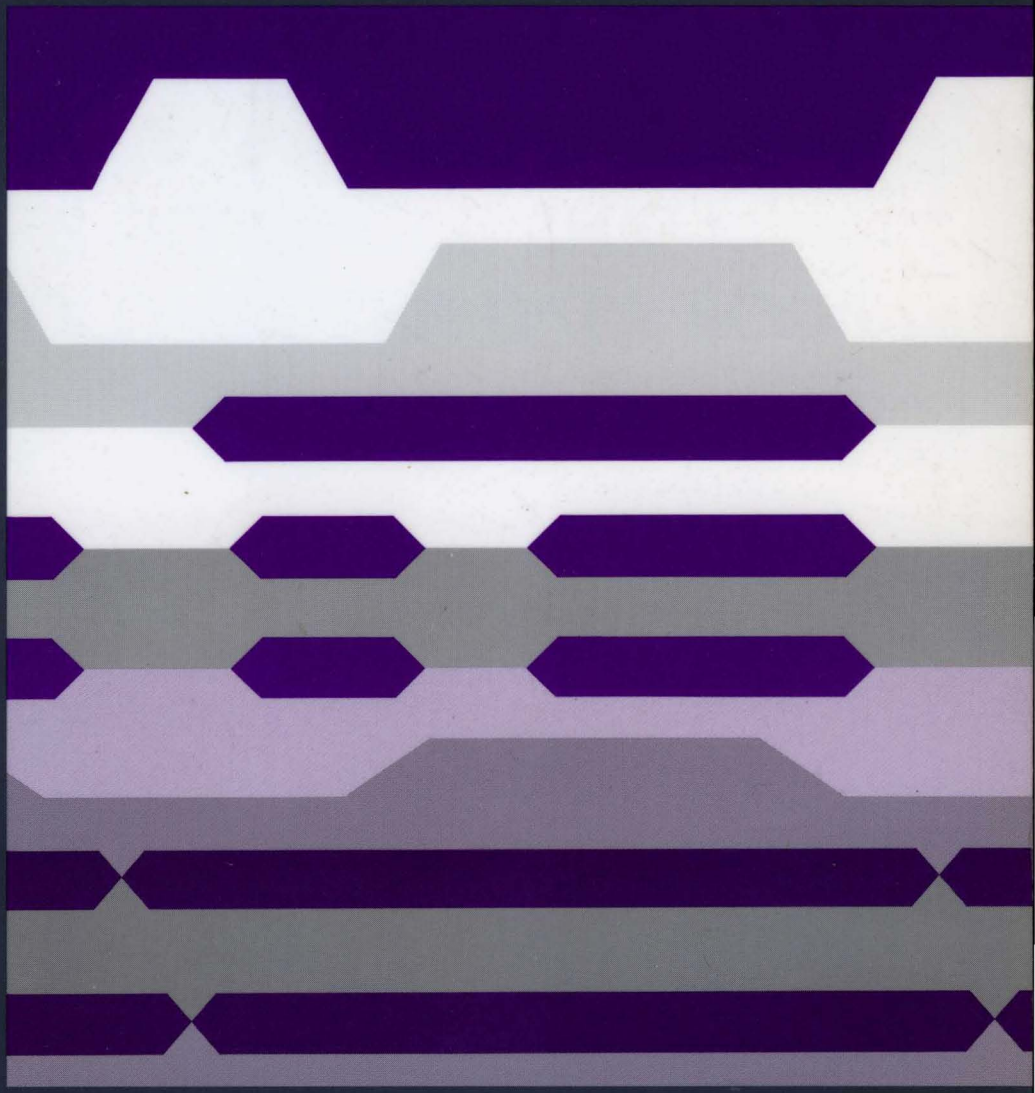
February, 1988

HITACHI®

8-BIT SINGLE-CHIP MICROCOMPUTER DATA BOOK



#U71

8-BIT SINGLE CHIP MICROCOMPUTER DATA BOOK

#U71

HITACHI®

MEDICAL APPLICATIONS

Hitachi's products are not authorized for use in MEDICAL APPLICATIONS, including, but not limited to, use in life support devices without the written consent of the appropriate officer of Hitachi's sales company. Buyers of Hitachi's products are requested to notify Hitachi's sales offices when planning to use the products in MEDICAL APPLICATIONS.

When using this manual, the reader should keep the following in mind:

1. This manual may, wholly or partially, be subject to change without notice.
2. All rights reserved: No one is permitted to reproduce or duplicate, in any form, the whole or part of this manual without Hitachi's permission.
3. Hitachi will not be responsible for any damage to the user that may result from accidents or any other reasons during operation of his unit according to this manual.
4. This manual neither ensures the enforcement of any industrial properties or other rights, nor sanctions the enforcement right thereof.
5. Circuitry and other examples described herein are meant merely to indicate characteristics and performance of Hitachi semiconductor-applied products. Hitachi assumes no responsibility for any patent infringements or other problems resulting from applications based on the examples described herein.
6. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi, Ltd.

February 1988

Printed in U.S.A.

CONTENTS

HITACHI

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

GENERAL INFORMATION

- Quick Reference Guide
- Introduction of Packages
- Reliability and Quality Assurance
- Reliability Test Data of Microcomputer
- Design Procedure and Support Tools for 8-bit Single-chip Microcomputer

QUICK REFERENCE GUIDE

■ NMOS 8-BIT SINGLE-CHIP MICROCOMPUTER HD6801 SERIES

Type No.			HD6801S0 HD6801S5	HD6801V0 HD6801V5	HD6803 HD6803-1
LSI Characteristics	Bus Timing (MHz)		1.0/1.25	1.0/1.25	1.0/1.25
	Supply Voltage (V)		5.0	5.0	5.0
	Operating Temperature * (°C)		0 ~ +70	0 ~ +70	0 ~ +70
	Package		DP-40	DP-40	DP-40
Functions	Memory	ROM (k byte)	2	4	–
		RAM (byte)	128	128	128
	I/O Port		29	29	13
	Interrupt	External	2	2	2
		Soft	1	1	1
		Timer	3	3	3
		Serial	1	1	1
	Timer		• Free running counter 16-bit x 1 • Output compare register 16-bit x 1 • Input capture register 16-bit x 1		
	SCI		Full double step-stop type		
	External Memory Expansion		• Address/data non-multiple mode (256 bytes) • Address/data multiple mode (65k bytes)		• Address/data multiple mode (65k bytes)
	Clock Pulse Generator		Built-in (External clock useable)		
	Built-in RAM Holding		Yes (64 bytes)		
EPROM on the Package Type**			HD68P01V07 HD68P01V07-1	HD68P01V07 HD68P01V07-1	–
Compatibility			MC6801 MC6801-1	–	MC6803 MC6803-1
Reference Page			43	77	111

* Wide Temperature Range (-40 ~ +85°C) version is available.
** HD68P01M0 and HD68P01M0-1 are useable.

■ NMOS 8-BIT SINGLE-CHIP MICROCOMPUTER HD6805 SERIES

Type No.			HD6805S1	HD6805S6	HD6805U1
LSI Characteristics	Clock Frequency (MHz)		1.0	1.0	1.0
	Supply Voltage (V)		5.25	5.25	5.25
	Operating Temperature** (°C)		0 ~ +70	0 ~ +70	0 ~ +70
	Package		DP-28	DP-28	DP-40
Functions	Memory	ROM (k byte)	1.1	1.8	2
		RAM (byte)	64	64	96
	I/O Port	I/O Port	20 (20)	20 (20)	32 (24)
		Input Port	20 (–)	20 (–)	32 (8)
	Interrupt	Nesting	6	6	6
		External	1	1	1
		Soft	1	1	1
		Timer	1	1	1
	Timer		• 8-bit timer with 7-bit prescaler • Event counter		
	Clock Pulse Generator		• Resistor • Crystal		
	Low-voltage Automatic Reset (LVI)		Yes	Yes	Yes
	Self-check Mode		Available	Available	Available
	Other Features				
EPROM on the Package Type			–	–	HD68P05V07
Compatibility			MC6805P2	MC6805P6	–
Reference Page			138	158	209

* Preliminary

** Wide Temperature Range (-40 ~ +85°C) version is available.

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

HD6805V1	HD6805T2*	HD6805W1
1.0	1.0	1.0
5.25	5.25	5.25
0 ~ +70	0 ~ +70	0 ~ +70
DP-40	DP-28	DP-40
4	2.5	4
96	64	96
32 / 24 / 8	19 / 19 / –	29 / 23 / 6
6	6	12
1	1	2
1	1	1
1	1	4
		• 8-bit timer with 7-bit prescaler • Event counter • 8-bit comparator
	• Crystal	
Yes	Yes	Yes
Available	Available	Available
	PLL logic for RF synthesizer	• 8-bit x 4-channel internal A/D converter • 8 bytes of standby RAM
HD68P05V07	–	HD68P05W0
–	MC6805T2	–
230	178	251

■ CMOS 8-BIT SINGLE-CHIP MICROCOMPUTER HD6301 SERIES

Type No.			HD6301V1 HD63A01V1 HD63B01V1		HD6301X0 HD63A01X0 HD63B01X0	
LSI Characteristics	Bus Timing (MHz)		1.0 (HD6301V1) 1.5 (HD63A01V1) 2.0 (HD63B01V1)		1.0 (HD6301X0) 1.5 (HD63A01X0) 2.0 (HD63B01X0)	
	Supply Voltage (V)		5.0		5.0	
	Operating Temperature*** (°C)		0 ~ +70		0 ~ +70	
	Package		DP-40, FP-54, CG-40		DP-64S, FP-80	
Functions	Memory	ROM (k byte)	4		4	
		RAM (byte)	128		192	
	I/O Port	I/O Port	29	29	53	24
		Input Port		–		8
		Output Port		–		21
	Interrupt	External	2		3	
		Soft	2		2	
		Timer	3		4	
		Serial	1		1	
	Timer		16-bit x 1 (Free running counter x 1 Output compare register x 1 Input capture register x1)		16-bit x 1 (Free running counter x 1 Output compare register x2 Input capture register x1) 8-bit x 1 (8-bit up counter x 1 Time constant register x 1)	
	SCI		Asynchronous		Asynchronous/Synchronous	
	External Memory Expansion		65k bytes		65k bytes	
	Other Features		•Error detection •Low power consumption modes (sleep and standby)		•Error detection •Low power consumption modes (sleep and standby) •Slow memory interface •Halt	
EPROM on the Package Type (EPROM On-Chip Type)			HD63P01M1 HD63PA01M1* HD63PB01M1* HD63701V0†** HD637A01V0†** HD637B01V0†**		HD63701X0†* HD637A01X0†* HD637B01X0†*	
Reference Page			279		319	

* Preliminary ** Under development *** Wide Temperature Range (-40 ~ +85°C) version is available.
†EPROM on-chip type

HD6301Y0 HD63A01Y0 HD63B01Y0	HD6303R HD63A03R HD63B03R	HD6303X HD63A03X HD63B03X	HD6303Y HD63A03Y HD63B03Y
1.0 (HD6301Y0) 1.5 (HD63A01Y0) 2.0 (HD63B01Y0)	1.0 (HD6303R) 1.5 (HD63A03R) 2.0 (HD63B03R)	1.0 (HD6303X) 1.5 (HD63A03X) 2.0 (HD63B03X)	1.0 (HD6303Y) 1.5 (HD63A03Y) 2.0 (HD63B03Y)
5.0	5.0	5.0	5.0
0 ~ +70	0 ~ +70	0 ~ +70	0 ~ +70
DP-64S	DP-40, FP-54, CG-40	DP-64S, FP-80	DP-64S
16	–	–	–
256	128	192	256
53: 48 / – / 5	13: 13 / – / –	24: 16 / 8 / –	24: 24 / – / –
3	2	3	3
2	2	2	2
4	3	4	4
1	1	1	1
16-bit x 1 (Free running counter x1, Output compare register x 2, Input capture register x 1) 8-bit x 1 (8-bit up counter x 1, Time constant register x 1)	16-bit x 1 (Free running counter x 1, Output compare register x 1, Input capture register x 1)	16-bit x 1 (Free running counter x1, Output compare register x 2, Input capture register x 1) 8-bit x 1 (8-bit up counter x 1, Time constant register x 1)	16-bit x 1 (Free running counter x 1, Output compare register x 2, Input capture register x 1) 8-bit x 1 (8-bit up counter x 1, Time constant register x 1)
Asynchronous/Synchronous	Asynchronous	Asynchronous/Synchronous	Asynchronous/Synchronous
65k bytes	65k bytes	65k bytes	65k bytes
•Error detection •Low power consumption modes (sleep and standby) •Slow memory interface •Halt	•Error detection •Low power consumption modes (sleep and standby)	•Error detection •Low power consumption modes (sleep and standby) •Slow memory interface •Halt	•Error detection •Low power consumption modes (sleep and standby) •Slow memory interface •Halt
358	406	440	477

■ CMOS 8-BIT SINGLE-CHIP MICROCOMPUTER HD6305 SERIES

Type No.			HD6305U0* HD63A05U0* HD63B05U0*	HD6305V0* HD63A05V0* HD63B05V0*	HD6305X0 HD63A05X0 HD63B05X0
LSI Characteristics	Clock Frequency (MHz)		1.0 (HD6305U0) 1.5 (HD63A05U0) 2.0 (HD63B05U0)	1.0 (HD6305V0) 1.5 (HD63A05V0) 2.0 (HD63B05V0)	1.0 (HD6305X0) 1.5 (HD63A05X0) 2.0 (HD63B05X0)
	Supply Voltage (V)		5.0	5.0	5.0
	Operating Temperature *** (°C)		0 ~ +70	0 ~ +70	0 ~ +70
	Package		DP-40	DP-40	DP-64S, FP-64
Functions	Memory	ROM (k byte)	2	4	4
		RAM (byte)	128	192	128
	I/O Port	I/O Port	31	31	32
		Input Port	31	31	55
		Output Port	–	–	7
			–	–	16
	Interrupt	External	2	2	2
		Soft	1	1	1
		Timer	2	2	2
		Serial	1	1	1
	Timer				
	SCI				
	External Memory Expansion		–	–	–
	Other Features				
EPROM on the Package Type (EPROM On-Chip Type)			HD63705V0†*** HD637A05V0†*** HD637B05V0†***		HD63P05Y0 HD63PA05Y0 HD63PB05Y0
Evaluation Chip			–	–	–
Reference Page			520	546	572

* Preliminary ** Under development *** Wide Temperature Range (-40 ~ +85°C) version is available.
†EPROM on-chip type

HD6305X1 HD63A05X1 HD63B05X1	HD6305X2 HD63A05X2 HD63B05X2	HD6305Y0 HD63A05Y0 HD63B05Y0	HD6305Y1 HD63A05Y1 HD63B05Y1	HD6305Y2 HD63A05Y2 HD63B05Y2	HD63L05F1
1.0 (HD6305X1) 1.5 (HD63A05X1) 2.0 (HD63B05X1)	1.0 (HD6305X2) 1.5 (HD63A05X2) 2.0 (HD63B05X2)	1.0 (HD6305Y0) 1.5 (HD63A05Y0) 2.0 (HD63B05Y0)	1.0 (HD6305Y1) 1.5 (HD63A05Y1) 2.0 (HD63B05Y1)	1.0 (HD6305Y2) 1.5 (HD63A05Y2) 2.0 (HD63B05Y2)	0.1
5.0	5.0	5.0	5.0	5.0	3.0
0 ~ +70	0 ~ +70	0 ~ +70	0 ~ +70	0 ~ +70	-20 ~ +75
DP-64S, FP-64	DP-64S, FP-64	DP-64S, FP-64	DP-64S, FP-64	DP-64S, FP-64	DP-64S, FP-80
4	–	8	8	–	4
128	128	256	256	256	96
31 / 24	31 / 24	55 / 32	31 / 24	31 / 24	20 / 20
7	7	7	7	7	–
–	–	16	–	–	(19)
2	2	2	2	2	1
1	1	1	1	1	1
2	2	2	2	2	1
1	1	1	1	1	–
• 8-bit x 1 (with 7-bit prescaler) • 15-bit x 1 (combined with SCI)					• 8-bit x 1 (with 7-bit prescaler)
Synchronous					–
12k bytes	16k bytes	–	8 k bytes	16k bytes	–
• Low power consumption modes (Wait, stop and standby)					• 8-bit A/D converter • LCD driver (6 x 7 segment) • Low power consumption modes (Standby and halt)
HD63P05Y1* HD63PA05Y1* HD63PB05Y1*	–	HD63P05Y0 HD63PA05Y0 HD63PB05Y0	HD63P05Y1* HD63PA05Y1* HD63PB05Y1*	–	–
–	–	–	–	–	HD63L05E0
599	599	628	655	655	684

■ NMOS 8-BIT SINGLE-CHIP MICROCOMPUTER EPROM ON THE PACKAGE TYPE

Type No.		HD68P01V07	HD68P01V07-1	HD68P01M0	HD68P01M0-1	HD68P05V07	HD68P05M0
LSI Characteristics	Supply Voltage (V)	5.0				5.0	
	Operating Temperature (°C)	0 ~ +70				0 ~ +70	
	Package	DC-40P				DC-40P	
Equivalent Device		HD6801S0 HD6801V0	HD6801S5 HD6801V5	–	–	HD6805U1 HD6805V1	–
Mountable EPROM		HN482732A-30	HN482732A-30	HN482764-3	HN482764-3	HN482732A-30	HN482764-3
Reference Page		717	717	717	717	756	756

■ CMOS 8-BIT SINGLE-CHIP MICROCOMPUTER EPROM ON THE PACKAGE TYPE

Type No.		HD63P01M1	HD63PA01M1*	HD63PB01M1*	HD63P05Y0	HD63PA05Y0	HD63PB05Y0
LSI Characteristics	Supply Voltage (V)	5.0			5.0		
	Operating Temperature (°C)	0 ~ +70			0 ~ +70		
	Package	DC-40P			DC-64SP		
Equivalent Device		HD6301V1	HD63A01V1	HD63B01V1	HD6305X0 HD6305Y0	HD63A05X0 HD63A05Y0	HD63B05X0 HD63B05Y0
Mountable EPROM		HN482732A-30 HN482764-3 HN27C64-30	HN482732A-30 HN482764-3 HN27C64-30	HN482732A-25 HN482764 HN27C64-25	HN482732A-30 HN482764-3 HN27C64-30	HN482732A-30 HN482764-3 HN27C64-30	HN482732A-25 HN482764 HN27C64-25
Reference Page		807	807	807	847	847	847

■ CMOS 8-BIT SINGLE-CHIP MICROCOMPUTER EPROM ON-CHIP TYPE

Type No.			HD6370IV0**	HD63701X0*	HD63705V0**
LSI Characteristics	Clock Frequency (MHz)		1.0, 1.5, 2.0	1.0, 1.5, 2.0	1.0, 1.5, 2.0
	Supply Voltage (V)		5.0	5.0	5.0
	Package		DC-40	DC-64S DP-64S	DC-40
Functions	Memory	EPROM (k byte)	4	4	4
		RAM (byte)	192	192	192
	I/O Port	I/O Port	29	53	31
		Input Port	–	8	–
		Output Port	–	21	–
	Interrupt	External	2	3	2
		Soft	2	2	1
		Timer	3	4	2
		Serial	1	1	1
	Timer		16 bit x 1	16 bit x 1 8 bit x 1	8 bit x 1
	Serial I/O		UART	UART synchronous	Synchronous
	External Memory Expansion		Possible (65k byte)	Possible (65k byte)	–
	Others		• Error detection • Low power consumption modes (Sleep and Standby)	• Error detection • Low power consumption modes (Sleep and Standby) • Slow memory interface	• Low power consumption modes (Wait, Stop and Standby)
EPROM Programming	Equivalent EPROM Type		27C256, 27256 (V_{PP} = 12.5V) High Performance Programming algorithm available	2732A (V_{PP} = 21V)	27C256, 27256 (V_{PP} = 12.5V) High Performance Programming algorithm available
	Socket Adapter	Hitachi	(under development)	H67PWA01A H67PWA01B	H35VSA00A H35VSA00B
		Data I/O	–	HD63701X0 (for 29A/29B)	HD63705V (for 29A/29B)
Reference Page			904	904	904

* Preliminary　　** Under development

HD68P05W0
5.0
0 ~ +70
DC-40P
HD6805W1
HN482732A-30 HN482764-3
778

HD63P05Y1*	HD63PA05Y1*	HD63PB05Y1*
5.0		
0 ~ +70		
DC-64SP		
HD6305X1 HD6305Y1	HD63A05X1 HD63A05Y1	HD63B05X1 HD63B05Y1
HN482732A-30 HN482764-3 HN27C64-30	HN482732A-30 HN482764-3 HN27C64-30	HN482732A-25 HN482764 HN27C64-25
874	874	874

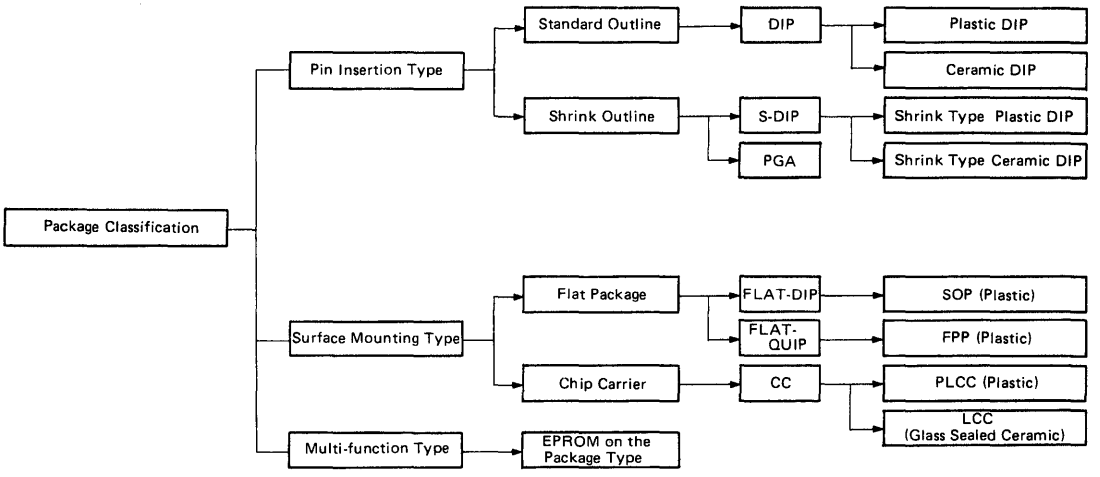
*Preliminary

INTRODUCTION OF PACKAGES

Hitachi microcomputer devices are offered in a variety of packages, to meet various user requirements.

1. Package Classification

When selecting suitable packaging, please refer to the Package Classifications given in Fig. 1 for pin insertion, surface mount, and multi-function types, in plastic and ceramic.



DIP; DUAL IN LINE PACKAGE
S-DIP; SHRINK DUAL IN LINE PACKAGE
PGA: PIN GRID ARRAY
FLAT-DIP; FLAT DUAL IN LINE PACKAGE
FLAT-QUIP; FLAT QUAD IN LINE PACKAGE
CC: CHIP CARRIER
SOP; SMALL OUTLINE PACKAGE
FPP; FLAT PLASTIC PACKAGE
PLCC; PLASTIC LEADED CHIP CARRIER
LCC ; LEADLESS CHIP CARRIER

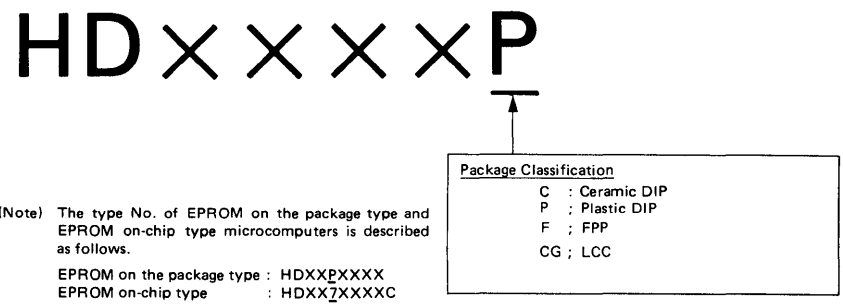
Fig. 1 Package Classification according to Material and Printed Circuit Board Mounting Type

2. Type No. and Package Code Indication

Type No. of Hitachi single-chip microcomputer device is followed by package material and outline specifications, as shown below. The package type used for each device is identified by code as follows, illustrated in the data sheet of each device.

When ordering, please write the package code beside the type number.

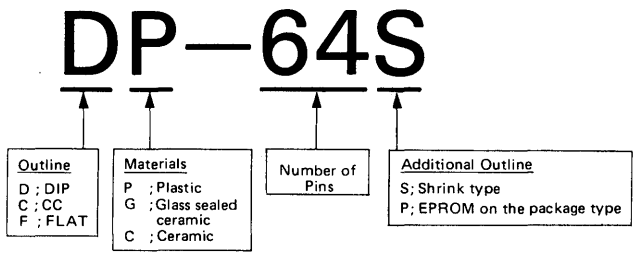
Type No. Indication



(Note) The type No. of EPROM on the package type and EPROM on-chip type microcomputers is described as follows.

EPROM on the package type : HDXX**P**XXXX
EPROM on-chip type : HDXX**7**XXXXC

Package Code Indication



3. Package Dimensional Outline

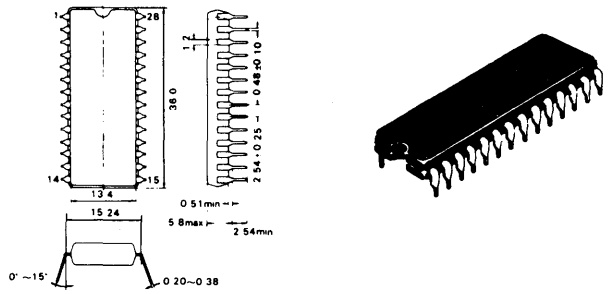
Hitachi single-chip microcomputer device employs the packages shown in Table 1 according to the mounting method on the PCB.

Table 1 Package List

Mounting method	Package classification	Package material	Package code
Pin insertion type	Standard outline (DIP)	Plastic	DP-28 DP-40
		Ceramic	DC-40
	Shrink outline (S-DIP)	Plastic	DP-64S
		Ceramic	DC-64S
Surface mounting type	Flat package (FPP)	Plastic	FP-54 FP-64 FP-80 FP-100
	Chip carrier (LCC)	Glass sealed ceramic	CG-40
Multi-function type	EPROM on the package type	Ceramic	DC-40P DC-64SP

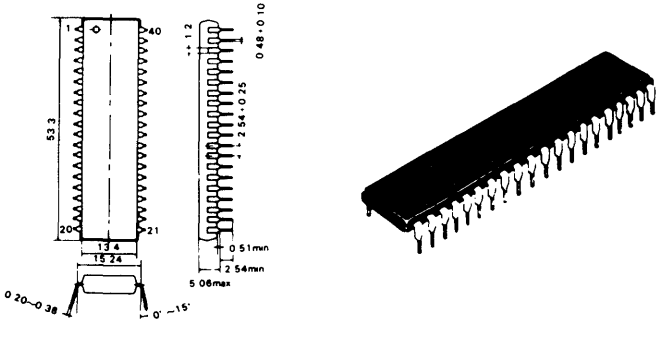
Plastic DIP

- **DP-28**



(Unit: mm)

- **DP-40**

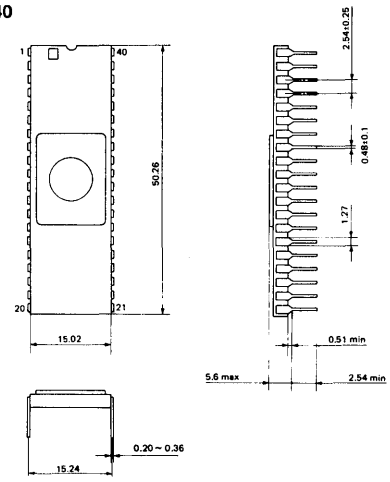


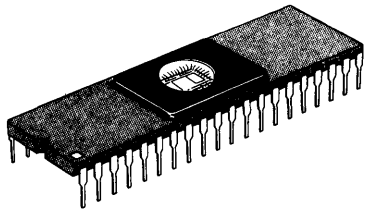
(Unit: mm)

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Ceramic DIP

• DC-40

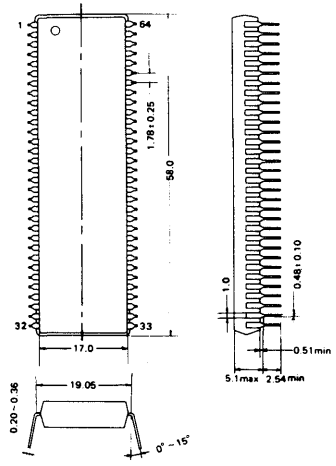


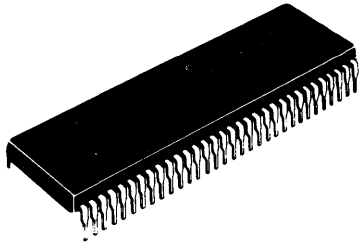


(Unit: mm)

Shrink Type Plastic DIP

• DP-64S



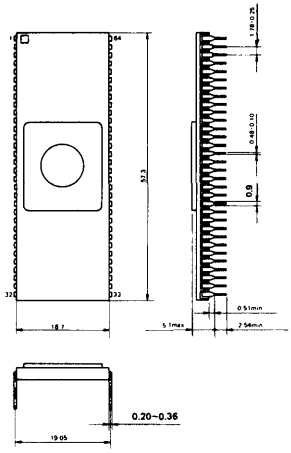


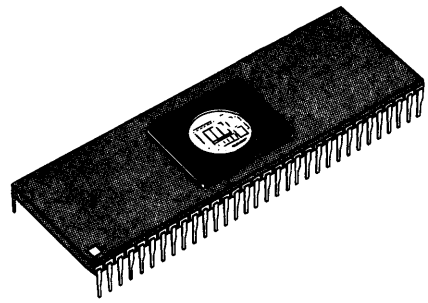
(Unit: mm)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Shrink Type Ceramic DIP

- **DC-64S**

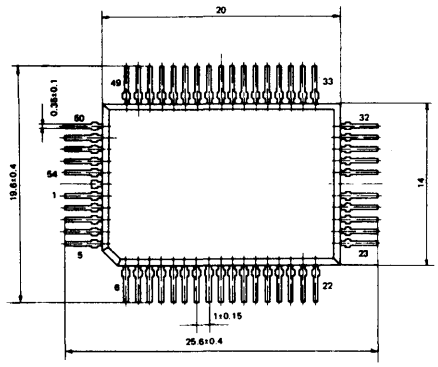


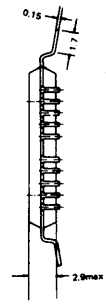


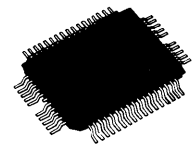
(Unit: mm)

Flat Package

- **FP-54**

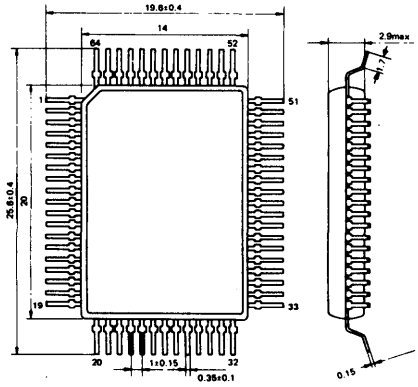


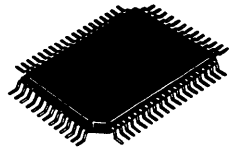




(Unit: mm)

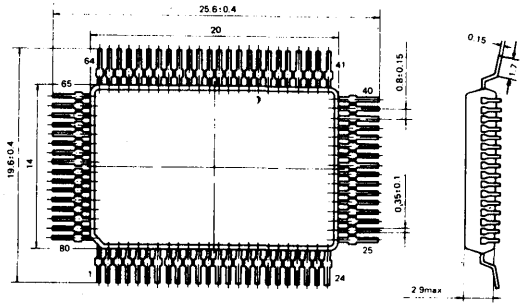
- **FP-64**

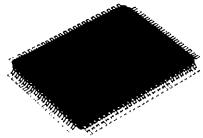




(Unit: mm)

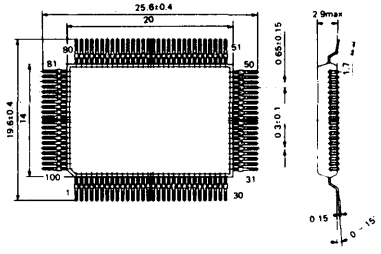
- **FP-80**

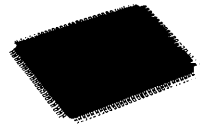




(Unit: mm)

- **FP-100**

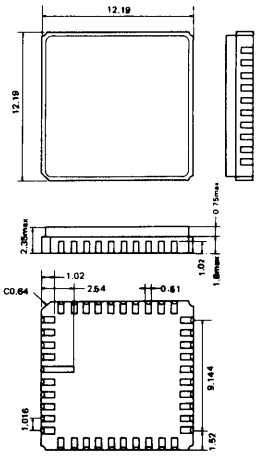




(Unit: mm)

Leadless Chip Carrier

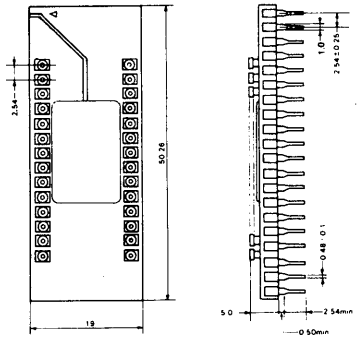
- **CG-40**

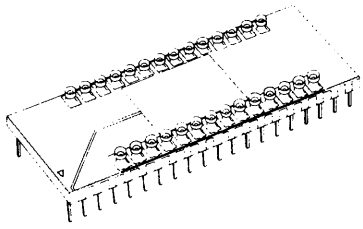


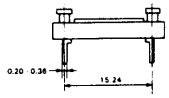
(Unit: mm)

EPROM on the Package Type

- **DC-40P**

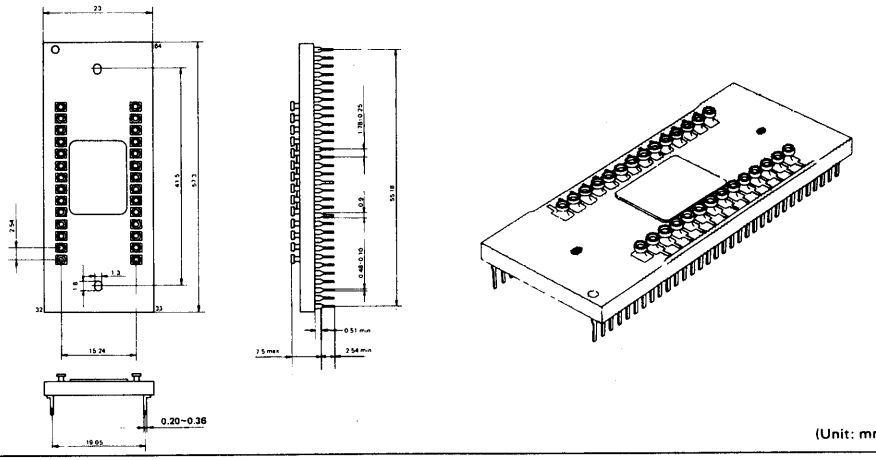






(Unit: mm)

• DC-64SP



4. Mounting Method

Package lead pins are surface treated with solder coating or plating to facilitate PCB mounting. The lead pins are connected to the package by eutectic solder. Common connecting method of leads and precautions are explained as follows:

4.1 Mounting Methods of Pin Insertion Type Package

Insert lead pins into the PCB through-holes (usually about ϕ0.8mm). Soak leads in a wave solder tub.

Lead pins held by the through-holes enable handling of the package through the soldering process, and facilitate automated soldering. When soldering leads in the wave solder tub, do not get solder on the package.

4.2 Mounting Method of Surface Mount Type Package

Apply the specified quantity of solder paste to the pattern on any printed board by the screen printing method, to temporarily fix the package to the board. The solder paste melts when heated in a reflowing furnace, and package leads and the pattern of the printed board are fixed by the surface tension of the melted solder and self alignment.

The size of the pattern where leads are attached should be 1.1 to 1.3 times the leads' width, depending on paste material or furnace adjustment.

The temperature of the reflowing furnace is dependent on packaging material and type. Fig. 2 lists the adjustment of the reflowing furnace for FPP. Pre-heat the furnace to 150° C. Surface temperature of the resin should be kept at 235° C maximum for 10 minutes or less.

(1) The temperature of the leads should be kept at 260°C for 10 minutes or less.
(2) The temperature of the resin should be kept at 235°C for 10 minutes or less.
(3) Below is shown the temperature profile when soldering a package by the reflowing method.

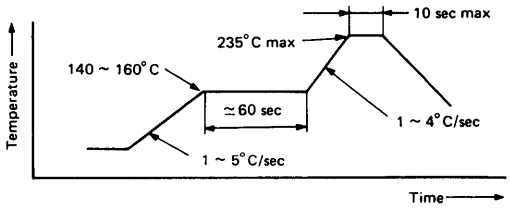


Fig. 2 Reflowing Furnace Adjustment for FPP

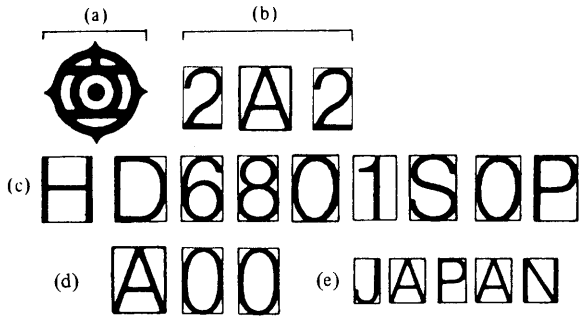
Employ adequate heating or temperature control equipment to prevent damage to the plastic package epoxy-resin material. When using an infrared heater, avoid long exposure at temperatures higher than the glass transition point of epoxy-resin (about 150° C), which may cause package damage and loss of reliability characteristics. Equalize the temperature inside and outside of packages by reducing the heat of the upper surface of the packages.

FPP leads may easily bend in shipment or during handling, and impact soldering onto the printed board. Heat the bent leads again with a soldering iron to reshape them.

Use a rosin flux when soldering. Do not use chloric flux because the chlorine in the flux has a tendency to remain on the leads and reduce reliability. Use alcohol, chlorothene or freon to wash away rosin flux from packages. These solvents should not remain on the packages for an excessive length of time, because the package markings may disappear.

5. Package Marking

The Hitachi trademark and product type No. are printed on packages, as shown in the following examples. Customer marking can be added to single-chip devices upon request.



Meaning of each mark

(a)	Hitachi Trademark
(b)	Lot Code
(c)	Type No.
(d)	ROM Code
(e)	Japan Mark

QUALITY ASSURANCE

1. VIEWS ON QUALITY AND RELIABILITY

Basic views on quality at Hitachi are to meet the individual uers' required quality level and maintain a general quality level equal to or above that of the general market. The quality required by the user may be specified by contract, or may be indefinite. In either case, efforts are made to assure reliable performance in actual operating circumstances. Quality control during the manufacturing process, and quality awareness from design through production lead to product quality and customer satisfaction. Our quality assurance technique consists basically of the following steps:

(1) Build in reliability at the design stage of new product development.
(2) Build in quality at all steps in the manufacturing process.
(3) Execute stringent inspection and reliability confirmation of final products.
(4) Enhance quality levels through field data feed back.
(5) Cooperate with research laboratories for higher quality and reliability.

With the views and methods mentioned above, utmost efforts are made to meet users' requirements.

2. RELIABILITY DESIGN OF SEMICONDUCTOR DEVICES

2.1 Reliability Targets

The reliability target is an important factor in sales, manufacturing, performance, and price. It is not adequate to set a reliability target based on a single set of common test conditions. The reliability target is set based on many factors:

(1) End use of semiconductor device.
(2) End use of equipment in which device is used.
(3) Device manufacturing process.
(4) End user manufacturing techniques.
(5) Quality control and screening test methods.
(6) Reliability target of system.

2.2 Reliability Design

The following steps are taken to meet the reliability targets:

(1) Design Standardization
As for design rules, critical items pertaining to quality and reliability are always studied at circuit design, device design, layout design, etc. Therefore, as long as standardized processing and materials are used the reliability risk is extremely small even in the case of new development devices, with the exception of special requirements imposed by functional needs.

(2) Device Design
It is important for the device design to consider total balance of process, structure, circuit, and layout design, especially in the case where new processes and/or new materials are employed. Rigorous technical studies are conducted prior to device development.

(3) Reliability Evaluation by Functional Test
Functional Testing is a useful method for design and process reliability evaluation of IC's and LSI devices which have complicated functions.

The objectives of Functional Test are:

- Determining the fundamental failure mode.
- Analysis of relation between failure mode and manufacturing process.
- Analysis of failure mechanism.
- Establishment of QC points in manufacturing process.

2.3 Design Review

Design Review is an organized method to confirm that a design satisfies the performance required and meets design specifications. In addition, design review helps to insure quality and reliability of the finished products. At Hitachi, design review is performed from the planning stage to production for new products, and also for design changes on existing products. Items discussed and considered at design review are:

(1) Description of the products based on design documents.
(2) From the standpoint of each participant, design documents are studied, and for points needing clarification, further investigation will be carried out.
(3) Specify quality control and test methods based on design documents and drawings.
(4) Check process and ability of manufacturing line to achieve design goal.
(5) Preparation for production.
(6) Planning and execution of sub-programs for design changes proposed by individual specialists,

for test, experiments, and calculations to confirm the design changes.

(7) Analysis of past failures with similar devices, discussion of methods to prevent them, and planning and execution of test programs to confirm success.

3. QUALITY ASSURANCE SYSTEM

3.1 Activity of Quality Assurance

General views of overall quality assurance in Hitachi are as follows:

(1) Problems in each individual process should be solved in the process. Therefore, at the finished product stage the potential failure factors have been removed.

(2) Feedback of information is used to insure a satisfactory level of ability process.

3.2 Quality Approval

To insure quality and reliability, quality approval is carried out at the preproduction stage of device design, as described in section 2. Our views on quality approval are:

(1) A third party executes approval objectively from the standpoint of the customer.

(2) Full consideration is given to past failures and information from the field.

(3) No design change or process change without QA approval.

(4) Parts, materials, and processes are closely monitored.

(5) Control points are established in mass production after studying the process abilities and variables.

3.3 Quality and Reliability Control at Mass Production

Quality control is accomplished through division of functions in manufacturing, quality assurance, and other related departments. The total function flow is shown in Fig. 2. The main points are described below.

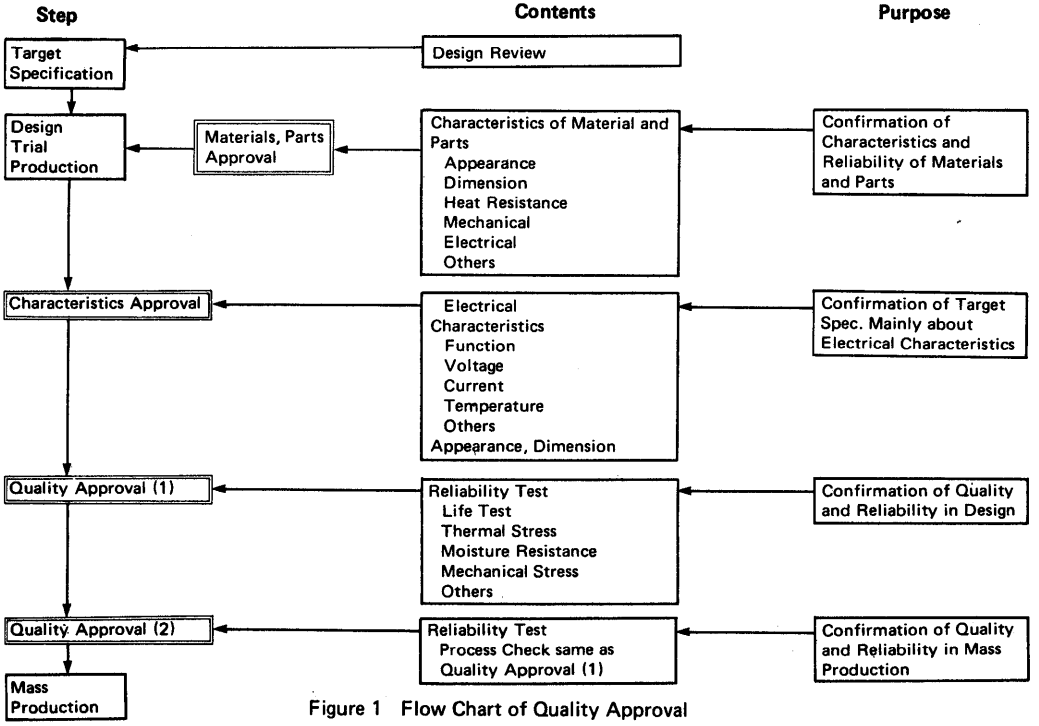


Figure 1 Flow Chart of Quality Approval

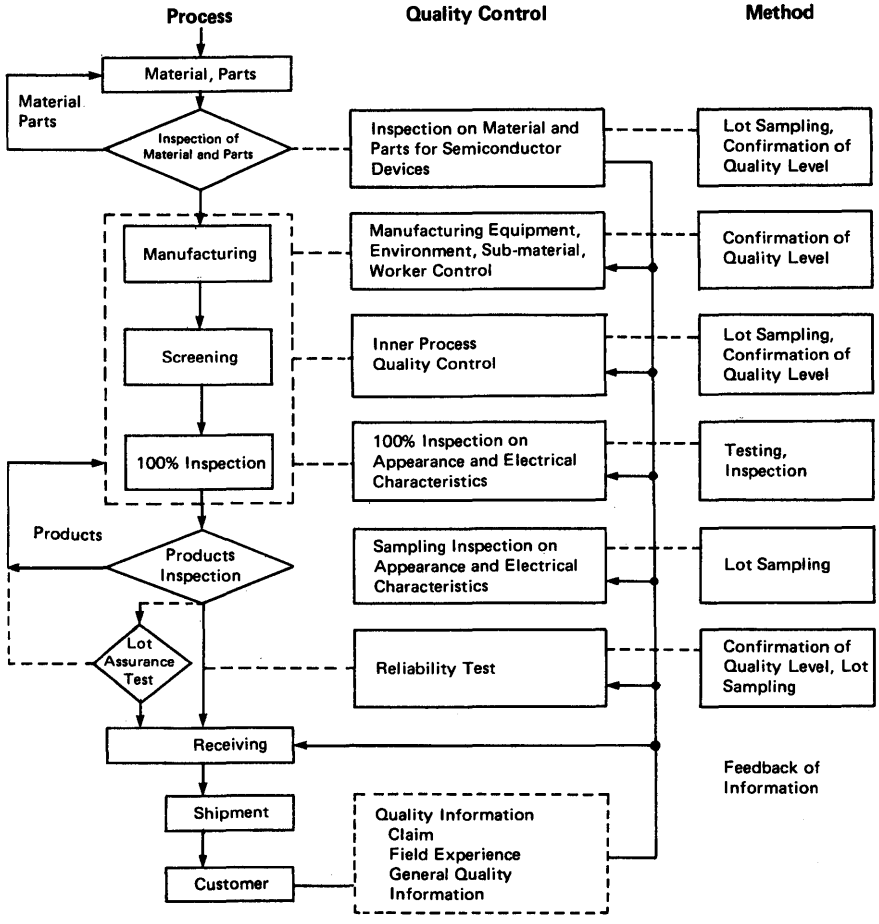


Figure 2 Flow Chart of Quality Control in Manufacturing Process

3.3.1 Quality Control of Parts and Materials

As semiconductor devices tend towards higher performance and higher reliability, the importance of quality control of parts and materials becomes paramount. Items such as crystals, lead frames, fine wire for wire bonding, packages, and materials needed in manufacturing processes such as masks and chemicals, are all subject to rigorous inspection and control. Incoming inspection is performed based on the purchase specification and drawing. The sampling is executed based mainly on MIL-STD-105D.

The other activities of quality assurance are as follows:

(1) Outside vendor technical information meeting.
(2) Approval and guidance of outside vendors.
(3) Chemical analysis and test.

The typical check points of parts and materials are shown in Table 1.

Table 1 Quality Control Check Points of Material and Parts (Example)

Material, Parts	Important Control Items	Point for Check
Wafer	Appearance	Damage and Contamination on Surface
	Dimension	Flatness
	Sheet Resistance	Resistance
	Defect Density	Defect Numbers
	Crystal Axis	
Mask	Appearance	Defect Numbers, Scratch
	Dimension	Dimension Level
	Resistoration	
	Gradation	Uniformity of Gradation
Fine Wire for Wire Bonding	Appearance	Contamination, Scratch, Bend, Twist
	Dimension	
	Purity	Purity Level
	Elongation Ratio	Mechanical Strength
Frame	Appearance	Contamination, Scratch
	Dimension	Dimension Level
	Processing Accuracy	
	Plating	Bondability, Solderability
	Mounting Characteristics	Heat Resistance
Ceramic Package	Appearance	Contamination, Scratch
	Dimension	Dimension Level
	Leak Resistance	Airtightness
	Plating	Bondability, Solderability
	Mounting Characteristics	Heat Resistance
	Electrical Characteristics	
	Mechanical Strength	Mechanical Strength
Plastic	Composition	Characteristics of Plastic Material
	Electrical Characteristics	
	Thermal Characteristics	
	Molding Performance	Molding Performance
	Mounting Characteristics	Mounting Characteristics

3.3.2 Inner Process Quality Control

Inner Process Quality Control performs very important functions in quality assurance of semiconductor devices. The manufacturing Inner Process Quality Control is shown in Fig. 3.

(1) Quality Control of Semi-final Products and Final Products

Potential failure factors of semiconductor devices are removed in the manufacturing process. To achieve this, check points are set-up in each process and products which have potential failure factors are not moved to the next process step. Manufacturing lines are rigidly selected and tight inner process quality controls are executed—rigid checks in each process and each lot, 100% inspection to remove failure factors caused by manufacturing variables and high temperature aging and temperature cycling. Elements of inner process quality control are as follows:

- Condition control of equipment and workers environment and random sampling of semi-final products.
- Suggestion system for improvement of work.
- Education of workers.
- Maintenance and improvement of yield.
- Determining quality problems, and implementing countermeasures.
- Transfer of quality information.

(2) Quality Control of Manufacturing Facilities and Measuring Equipment

Manufacturing equipment is improving as higher performance devices are needed. At Hitachi, the automation of manufacturing equipment is encouraged. Maintenance Systems maintain operation of high performance equipment. There are daily inspections which are performed based on related specifications. Inspection points are listed in the specification and are checked one by one to prevent any omission. As for adjustment and maintenance of measuring equipment, specifications are checked one by one to maintain and improve quality.

(3) Quality Control of Manufacturing Circumstances and Sub-Materials

The quality and reliability of semiconductor devices are highly affected by the manufacturing process. Therefore, controls of manufacturing circum-

stances such as temperature, humidity and dust, and the control of submaterials, like gas, and pure water used in a manufacturing process, are intensively executed.

Dust control is essential to realize higher integration and higher reliability of devices. At Hitachi, maintenance and improvement of cleanliness at manufacturing sites is accomplished through attention to buildings, facilities, air conditioning systems, delivered materials, clothes, work environment, and periodic inspection of floating dust concentration.

3.3.3 Final Product Inspection and Reliability Assurance

(1) Final Product Inspection

Lot inspection is done by the quality assurance

Process	Control Point		Purpose of Control
▽ Purchase of Material			
Wafer →	Wafer	Characteristics, Appearance	Scratch, Removal of Crystal Defect Wafer
○ Surface Oxidation	Oxidation		Assurance of Resistance
□ Inspection on Surface Oxidation		Appearance, Thickness of Oxide Film	Pinhole, Scratch
○ Photo Resist	Photo Resist		
□ Inspection on Photo Resist		Dimension, Appearance	Dimension Level
◇ PQC Level Check			Check of Photo Resist
○ Diffusion	Diffusion	Diffusion Depth, Sheet Resistance	Diffusion Status
□ Inspection on Diffusion		Gate Width	Control of Basic Parameters (V_{TH}, etc.) Cleanness of surface, Prior Check of V_{IH} Breakdown Voltage Check
◇ PQC Level Check		Characteristics of Oxide Film Breakdown Voltage	
○ Evaporation	Evaporation	Thickness of Vapor Film, Scratch, Contamination	Assurance of Standard Thickness
□ Inspection on Evaporation			
◇ PQC Level Check			
□ Wafer Inspection	Wafer	Thickness, V_{TH} Characteristics	Prevention of Crack, Quality Assurance of Scribe
□ Inspection on Chip Electrical Characteristics	Chip	Electrical Characteristics	
○ Chip Scribe		Appearance of Chip	
□ Inspection on Chip Appearance			
◇ PQC Lot Judgement			
Frame →			
○ Assembling	Assembling	Appearance after Chip Bonding	Quality Check of Chip Bonding
		Appearance after Wire Bonding	Quality Check of Wire Bonding
◇ PQC Level Check		Pull Strength, Compression Width, Shear Strength	Prevention of Open and Short
□ Inspection after Assembling		Appearance after Assembling	
◇ PQC Lot Judgement			
Package →			
○ Sealing	Sealing	Appearance after Sealing Outline, Dimension	Guarantee of Appearance and Dimension
◇ PQC Level Check	Marking	Marking Strength	
○ Final Electrical Inspection			
◇ Failure Analysis		Analysis of Failures, Failure Mode, Mechanism	Feedback of Analysis Information
□ Appearance Inspection			
△ Sampling Inspection on Products			
○ Receiving			
Shipment			

Figure 3 Example of Inner Process Quality Control

department for products which were judged good in 100% test . . . the final process in manufacturing. Though 100% yield is expected, sampling inspection is executed to prevent mixture of bad product by mistake. The inspection is executed not only to confirm that the products have met the users' requirements but also to consider potential quality factors. Lot inspection is executed based on MIL-STD-105D.

(2) Reliability Assurance Tests
To assure the reliability of semiconductor devices, reliability tests and tests on individual manufacturing lots that are required by the user, are periodically performed.

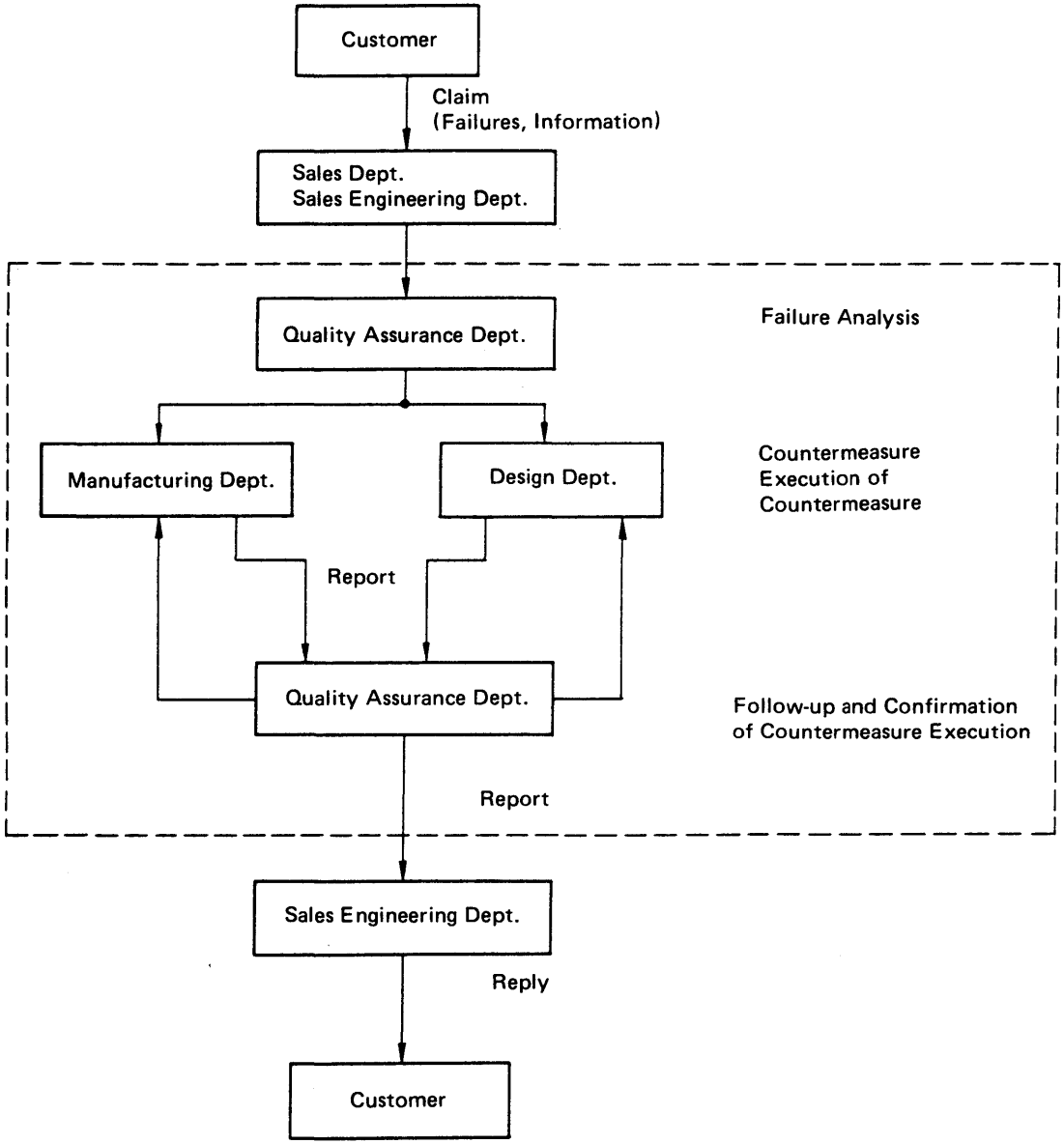


Figure 4 Process Flow Chart of Field Failure

RELIABILITY TEST DATA

1. INTRODUCTION

Microcomputers provide high reliability and quality to meet the demands of increased functions, enlarging scale, and widening application. Hitachi has improved the quality level of microcomputer products by evaluating reliability, building quality into the manufacturing process, strengthening inspection techniques, and analyzing field data.

The following reliability and quality assurance data for Hitachi 8-bit single-chip microcomputers indicates results from test and failure analysis.

2. PACKAGE AND CHIP STRUCTURE

2.1 Packaging

Production output and application of plastic packaging continues to increase, expanding to automobile measuring and control systems, and computer terminal equipment operating under severe conditions. To meet this demand, Hitachi has significantly improved moisture resistance and operational stability in the plastic manufacturing process.

Plastic and side-brazed ceramic package structures are shown in Figure 1 and Table 1.

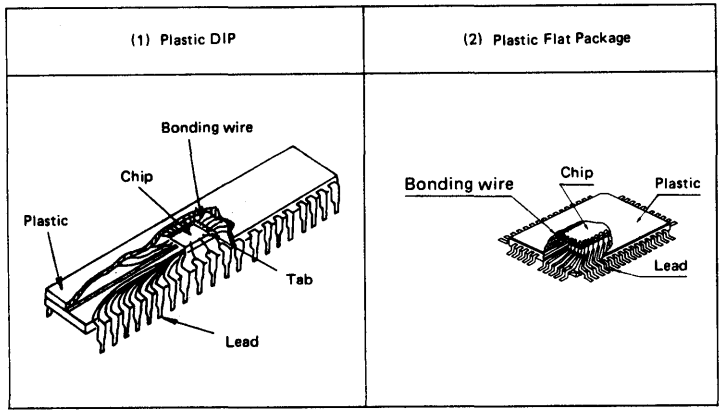


Figure 1 Package Structure

Table 1 Package Material and Properties

Item	Plastic DIP	Plastic Flat Package
Package	Epoxy	Epoxy
Lead	Solder dipping Alloy 42or Cu	Solder plating Alloy 42
Die bond	Au-Si or Ag paste	Au-Si or Ag paste
Wire bond	Thermo compression	Thermo compression
Wire	Au	Au

2.2 Chip Structure

The HMCS6800 family is produced in NMOS E/D technology or low power CMOS technology. Si-gate process is used in both types to achieve high reliability and density. Chip structure and basic circuitry are shown in Figure 2.

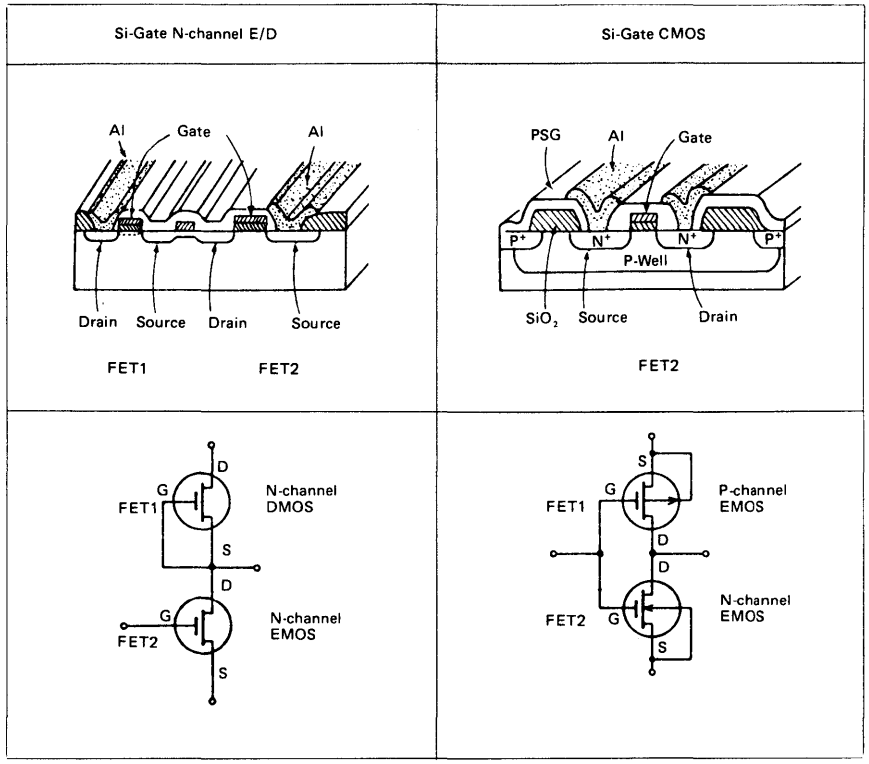


Figure 2 Chip Structure and Basic Circuit

3. QUALITY QUALIFICATION AND EVALUATION

3.1 Reliability Test Methods

Reliability test methods shown in Table 2 are used to qualify and evaluate new products and processes.

Table 2 Reliability Test Methods

Test Items	Test Condition	MIL-STD-883B Method No.
Operating Life Test	125°C, 1000hr	1005,2
High Temp, Storage	Tstg max, 1000hr	1008,1
Low Temp, Storage	Tstg min, 1000hr	
Steady State Humidity	65°C 95%RH, 1000hr	
Steady State Humidity Biased	85°C 85%RH, 1000hr	
Temperature Cycling	-55°C ~ 150°C, 10 cycles	1010,4
Temperature Cycling	-20°C ~ 125°C, 200 cycles	
Thermal Shock	0°C ~ 100°C, 100 cycles	1011,3
Soldering Heat	260°C, 10 sec	
Mechanical Shock	1500G 0.5 msec, 3 times/X, Y, Z	2002,2
Vibration Fatigue	60Hz 20G, 32hrs/X, Y, Z	2005,1
Variable Frequency	20~2000Hz 20G, 4 min/X, Y, Z	2007,1
Constant Acceleration	20000G, 1 min/X, Y, Z	2001,2
Lead Integrity	225gr, 90° 3 times	2004,3

3.2 Reliability Test Results

Reliability Test Results of 8-bit single-chip microcomputer devices are shown in Table 3 to Table 7.

Table 3 Dynamic Life Test

Device Type	Sample Size	Component Hours	Failures
HD6801P	191 pcs.	191000	0
HD6805P	114	114000	0
HD6301P	182	182000	0
HD6305P	77	77000	0
HD63L05P	40	40000	0
HD68P01	22	22000	0
HD63P01	32	32000	0
HD68P05	22	22000	0

Estimated Field Failure Rate
= 0.016%/1000 hrs at Ta = 75°C for NMOS (HD6801P, HD6805P)
= 0.037%/1000 hrs at Ta = 75°C for CMOS (HD6301P, HD63L05P)
(Activation Energy 0.7eV, Confidence Level 60%)

Table 4 High Temperature, High Humidity Test (Moisture Resistance Test)

(1) 85°C 85%RH Bias Test

Device Type	V_{CC} Bias	168 hrs	500 hrs	1000 hrs
HD6801P	5.5V	0/22	0/22	0/22
HD6805P	5.5V	0/22	0/22	0/22
HD6301P	5.5V	0/176	0/131	0/131
HD6305P	5.5V	0/22	0/22	0/22

(2) High Temperature-High Humidity Storage Life Test

Device Type	Condition	168 hrs	500 hrs	1000 hrs
HD6801P	65°C 95%RH	0/45	0/45	0/45
HD6805P	65°C 95%RH	0/45	0/45	0/45
HD6301P	65°C 95%RH	0/603	0/603	0/603
HD6301P	85°C 95%RH	0/234	1*/234	0/233
HD6305P	65°C 95%RH	0/112	0/112	0/112
HD63L05P	65°C 95%RH	0/160	0/160	0/160
HD63L05P	85°C 95%RH	0/160	1*/160	0/159

* Aluminum corrosion

(3) Pressure Cooker Test
(Condition: 2 atm 121°C)

Device Type	40 hrs	60 hrs	100 hrs	200 hrs
HD6801P	0/45	0/45	0/45	0/45
HD6805P	0/44	0/44	0/44	0/44
HD6301P	0/135	0/135	0/135	0/135
HD6305P	0/32	0/32	0/32	0/32
HD63L05P	0/80	0/80	1*/80	2**/79

*Leakage current
**Leakage current and Aluminum corrosion

(4) MIL-STD-883B Moisture Resistance Test
(Condition; 65°C~ -10°C, over 90%RH)

Device Type	10 cycles	20 cycles	40 cycles
HD6801P	0/50	0/50	0/50
HD6805P	0/32	0/32	0/32
HD6301P	0/75	0/75	0/75
HD63L05P	0/22	0/22	0/22

Table 5 Temperature Cycling Test

(Condition; -55°C ~ 25°C ~ 150°C)

Device Type	10 cycles	100 cycles	200 cycles
HD6801P	0/102	0/102	0/102
HD6805P	0/442	0/45	0/45
HD6301P	0/258	0/258	0/258
HD6305P	0/45	0/45	0/45
HD68P01	0/44	0/44	0/44
HD63P01	0/45	0/45	0/45
HD68P05	0/68	0/19	0/19

Table 6 High Temperature, Low Temperature Storage Life Test

Device Type	Ta	168 hrs	500 hrs	1000 hrs
HD6801P	150°C	0/22	0/22	0/22
	-55°C	0/22	0/22	0/22
HD6805P	150°C	0/44	0/44	0/44
	-55°C	0/22	0/22	0/22
HD6803P	150°C	0/22	0/22	0/22
	-55°C	0/22	0/22	0/22
HD6301P	150°C	0/45	0/45	0/45
	-55°C	0/22	0/22	0/22
HD6305P	150°C	0/22	0/22	0/22
	-55°C	0/22	0/22	0/22
HD63L05P	150°C	0/22	0/22	0/22
	-55°C	0/22	0/22	0/22

Table 7 Mechanical and Environmental Test

Test Item	Condition	Plastic DIP		Flat Plastic Package	
		Sample Size	Failure	Sample Size	Failure
Thermal Shock	0°C ~ 100°C 10 cycles	110	0	100	0
Soldering Heat	260°C, 10 sec.	164	0	22	0
Salt Water Spray	35°C, NaCl 5% 24 hrs	110	0	22	0
Solderability	230°C, 5 sec. Rosin flux	159	0	34	0
Drop Test	75cm, maple board 3 times	110	0	22	0
Mechanical Shock	1500G, 0.5ms 3 times/X, Y, Z	110	0	22	0
Vibration Fatigue	60 Hz, 20G 32hrs/X, Y, Z	110	0	22	0
Vibration Variable Freq.	100 ~ 2000Hz 20G, 4 times/X, Y, Z	110	0	22	0
Lead Integrity	225g, 90° Bonding 3 times	110	0	22	0

4. PRECAUTIONS

4.1 Storage

To prevent deterioration of electrical characteristics, solderability, appearance or structure, Hitachi recommends semiconductor devices be stored as follows:

(1) Store in ambient temperatures of 5 to 30° C, with a relative humidity of 40 to 60%.
(2) Store in a clean, dust- and active gas-free environment.
(3) Store in conductive containers to prevent static electricity.
(4) Store without any physical load.
(5) When storing devices for an extended period, store in an unfabricated form, to minimize corrosion of pre-formed lead wires.
(6) Unsealed chips should be stored in a cool, dry, dark and dust-free environment. Assembly should be performed within 5 days of unpacking. Devices can be stored for up to 20 days in dry nitrogen gas with a dew point at -30° C or less.
(7) Prevent condensation during storage due to rapid temperature changes.

4.2 Transportation

General precautions for electronic components are applicable in transporting semiconductors, units incorporating semiconductors, and other similar systems. In addition, Hitachi recommends the following:

(1) When transporting semiconductor devices or printed circuit boards, minimize mechanical vibration and shock. Use containers or jigs which will not induce static electricity as a result of vibration. Use of an electrically conductive container or aluminum foil is recommended.
(2) To prevent device deterioration from clothing-induced static electricity, workers should be properly grounded while handling devices. Use of a 1 M ohm resistor is recommended to prevent electric shock.
(3) When transporting printed curcuit boards containing semiconductor devices, suitable preventive measures against static electricity must be taken. Voltage build-up can be avoided by shorting the card-edge terminals. When a belt conveyor is used, apply some surface treatment to prevent build-up of electrical charge.
(4) Minimize mechanical vibration and shock when transporting semiconductor devices or printed circuit boards.

4.3 Handling for Measurement

Avoid static electricity, noise and voltage surge when measuring or mounting devices. Precaution should be taken against current leakage through terminals and housings of curve tracers, synchroscopes, pulse generators, and DC power sources.

When testing devices, prevent voltage surges from the tester, attached clamping circuit, and any excessive voltage possible through accidental contact.

In inspecting a printed circuit board, power should not be applied if any solder bridges or foreign matter is present.

4.4 Soldering

Semiconductor devices should not be exposed to high temperatures for excessive periods. Soldering must be performed consistent with temperature conditions of 260° C for 10 seconds, 350° C for 3 seconds, and at a distance of 1 to 1.5mm from the end of the device package.

A soldering iron with secondary voltage supplied through a grounded transformer is recommended to protect against leakage current. Use of alkali or acid flux, which may corrode the leads, is not recommended.

4.5 Removing Residual Flux

Detergent or ultrasonic removal of residual flux from circuit boards is necessary to ensure system reliability. Selection of detergent type and cleaning conditions are important factors.

When chloric detergent is used for plastic packaged devices, care must be taken against package corrosion. Extended cleaning periods and excessive temperature conditions can cause the chip coating to swell due to solvent permeation. Hitachi recommends use of Lotus and Dyfron solvents. Trichloroethylene solvent is not suitable.

The following conditions are advisable for ultrasonic cleaning:

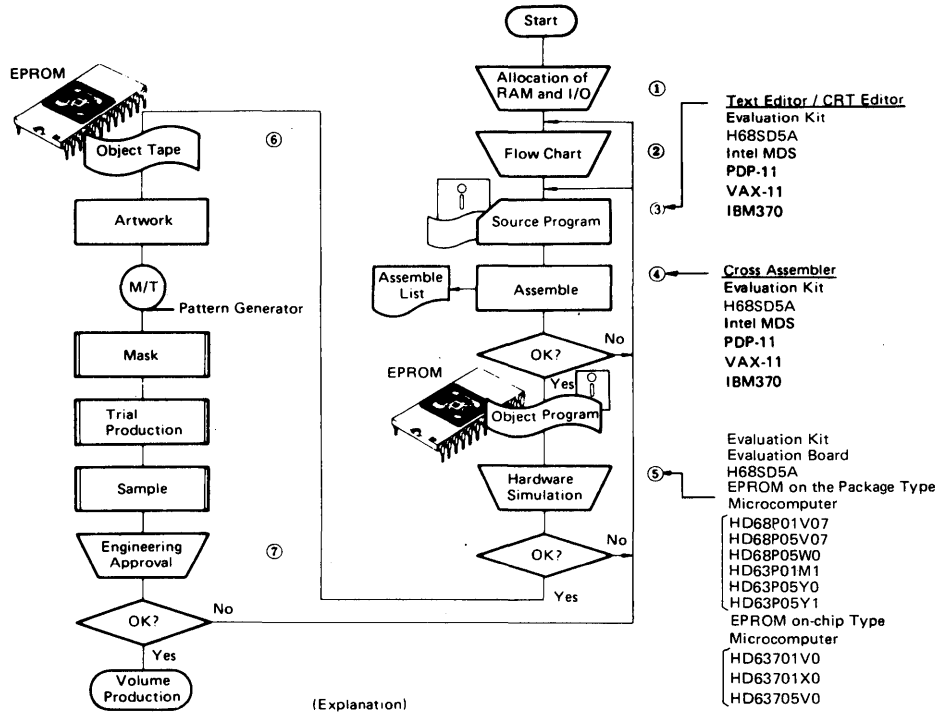
- Frequency: 28 to 29 k Hz (to avoid device resonation)
- Ultrasonic output: 15W/ℓ
- Keep devices from making direct contact with power generator
- Cleaning time: Less than 30 seconds.

DESIGN PROCEDURE AND SUPPORT TOOLS FOR 8-BIT SINGLE-CHIP MICROCOMPUTERS

The cross assmebler and hardware simulator using various types of computers are prepared by Hitachi as supporting systems to develop user's programs.

User's programs are mask programmed into the ROM and delivered as the LSI by the company.

Fig. 1 shows the typical program design procedure and Table 1 shows the system development support tools for the 8-bit single-chip microcomputer family used in these processes.



(Explanation)

① When the user programs the system, the predetermined functions are assigned to the I/O pin and the RAM before the programming.

② A flow chart is designed to achieve the predetermined functions and the flow chart is coded by using the prenumeric code.

③ The coded flow chart is punched into the card or the paper tape or written into the floppy disk, to generate a source program.

④ The source program is assembled by the resident system (evaluation kit) or the cross system, to generate the object program. In this case, errors during the assembling are also detected.

⑤ Hardware simulation is performed to confirm the program. The company provides four kinds of hardware, the H68SD5A, the evaluation kit, the evaluation board and the EPROM on the package type microcomputer. The consumers are able to choose the best suitable tool.

⑥ The completed program is sent to the company in the form of EPROM or the object tape.

⑦ Options such as ROM is masked by the company, LSI is testatively produced and the sample is handed in to the user. After the user has evaluated the sample and confirmed that the program is correct, mass production is started.

Figure 1 Program Design Procedure

Table 1 System Development Support Tools

Type No.	Resident System				Cross System		
	Evaluation Kit	Evaluation Board	EPROM on the Package	H68SD5A+Emulator Set (Hardware + Software)	IBM370	Intel MDS220/230	PDP-11/ VAX-11
HD6801S0 HD6801V0	H61EVT2 (Hardware) + S61MIX2-R (Software)	–	HD68P01V07	H68SD5A + H61MIX1	S31XSY1-T	S31MDS1-F (ISIS-II) S31MDS2-F (CP/M)	○
HD6805S1 HD6805S6	H65EVT2 (Hardware) + S65MIX1-R (Software)	–	–	H68SD5A + H65MIX1	–	S35MDS1-F (ISIS-II) S35MDS2-F (CP/M)	–
HD6805U1 HD6805V1		–	HD68P05V07		–		–
HD6805W1	H65EVT3 (Hardware) + S65MIX1-R (Software)	–	HD68P05W0	H68SD5A + H65MIX2	–		–
HD6301V1	H31EVT1 (Hardware) + S31MIX1-R (Software)	–	HD63P01M1 HD63701V0†**	H68SD5A+H31MIX1	S31XSY1-T	S31MDS1-F (ISIS-II) S31MDS2-F (CP/M)	○
HD6301X0	–	–	HD63701X0†*	H68SD5A + H31MIX2			○
HD6301Y0	–	**	–	H68SD5A + H31MIX3			○
HD6305U0*	–	–	HD63705V0†**	H68SD5A+H35MIX3**	–	S35MDS1-F (ISIS-II) S35MDS2-F (CP/M)	–
HD6305V0*	–	–	HD63705V0†**				–
HD6305X0	H35EVT1 (Hardware + Software)	–	HD63P05Y0	H68SD5A + H35MIX1	–		–
HD6305X1			HD63P05Y1*				
HD6305Y0		–	HD63P05Y0		–		–
HD6305Y1			HD63P05Y1*				
HD63L05F1	H3L5EVT1 (Hardware) + S3L5MIX1-R (Software)	H3L5EV00	–	H68SD5A + H3L5MIX1	–		–

* Preliminary ** Under development † EPROM On-chip Type ○ Available from Microtec.

■ SINGLE-CHIP MICROCOMPUTER DEVELOPMENT SYSTEM

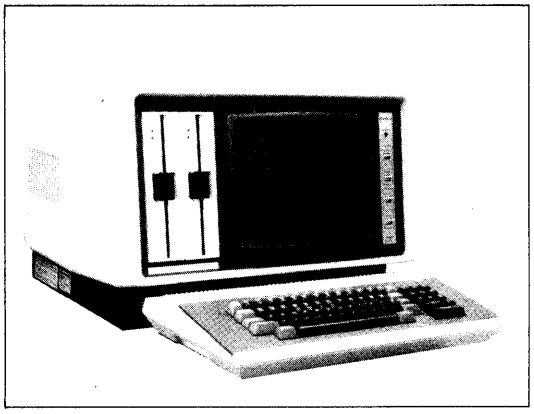
H68SD5A is a completely integrated hardware and software development system for Hitachi's 4-Bit and 8-Bit single-chip microcomputers. It offers high-level functions such as operation by CRT display, assembler (based on floppy disk), easy debugging (in-circuit emulator) and simulation.

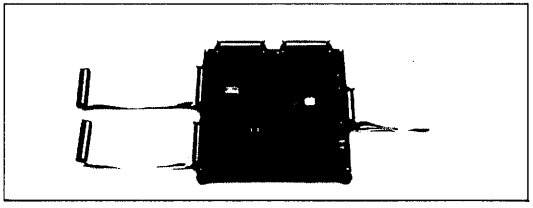
FEATURES

- Upward version of the H68SD5
- CRT Display, keyboard and two floppy disk drives
- Easy to debug user's prototype system by in-circuit emulators suited for each kind of MCU.
- The H68SD5A can perform system development by CRT editor, assembler, linkage editor, emulator and simulator.
- Parallel and/or serial interface ports are available for easy printer interface.
- EPROM programmer interface software is provided. DATA I/O's EPROM programmers (29A, 22A and 22B) are applicable.
- User's program developped by using the H68SD5 is usable.
- Emulators for the H68SD5 are usable.

System Configuration

H68SD5A





Emulator Module (Option) — 8-bit single-chip microcomputer family
- HMCS40 series
- HMCS400 series

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

DATA SHEETS

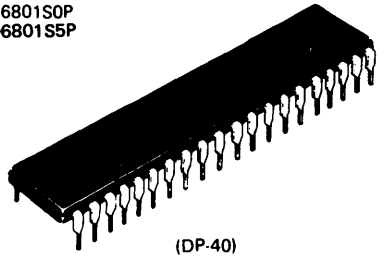
Preliminary data sheets herein contain information on new products. Specifications and information are subject to change without notice.

Advance Information data sheets herein contain information on a product under development. Hitachi reserves the right to change or discontinue these products without notice.

HD6801S0, HD6801S5
MCU (Microcomputer Unit)

The HD6801S MCU is an 8-bit microcomputer unit which is compatible with the HMCS6800 family of parts. The HD6801S MCU is object code compatible with the HD6800 with improved execution times of key instructions plus several new 16-bit and 8-bit instructions including an 8x8 unsigned multiply with 16-bit result. The HD6801S MCU can operate as a single – chip microcomputer or be expanded to 65k bytes. The HD6801S MCU is TTL compatible and requires one +5.0 volt power supply. The HD6801S MCU has 2k bytes of ROM and 128 bytes of RAM on chip. Serial Communication interface (S.C.I.), and parallel I/O as well as a three function 16-bit timer. Features and Block diagram of the HD6801S include the following:

HD6801S0P
HD6801S5P

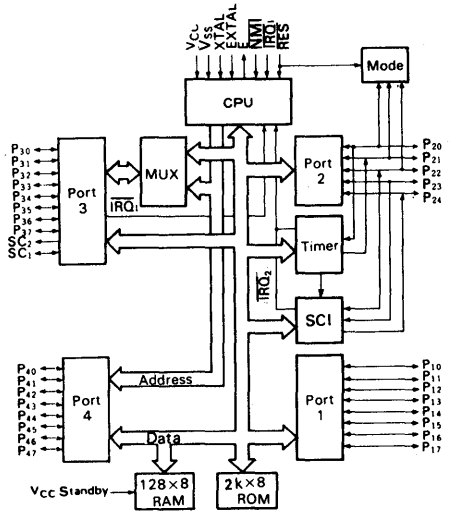


(DP-40)

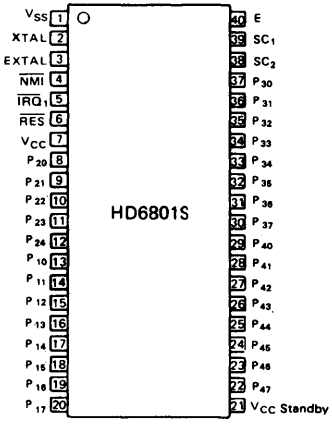
■ FEATURES

- Expanded HMCS6800 Instruction Set
- 8 x 8 Multiply
- On-Chip Serial Communication Interface (S.C.I.)
- Object Code Compatible With The HD6800 MPU
- 16-Bit Timer
- Single Chip Or Expandable To 65k Bytes
- 2k Bytes Of ROM
- 128 Bytes Of RAM (64 Bytes Retainable On Power Down)
- 29 Parallel I/O Lines And 2 Handshake Control Lines
- Internal Clock/Divided-By-Four Circuitry
- TTL Compatible Inputs And Outputs
- Interrupt Capability
- Compatible with MC6801 and MC6801-1

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ TYPE OF PRODUCTS

MCU	Bus Timing
HD6801S0	1 MHz
HD6801S5	1.25 MHz

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	0 ~ + 70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} =5.0V±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	Other Inputs*			2.0	–	V_{CC}	
Input "Low" Voltage	All Inputs*	V_{IL}		-0.3	–	0.8	V
Input Load Current	P_{40} ~ P_{47}	$\|I_{in}\|$	V_{in} = 0 ~ 2.4V	–	–	0.5	mA
	SC_1			–	–	0.8	
	EXTAL		V_{in} = 0 ~ V_{CC}	–	–	0.8	
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$	$\|I_{in}\|$	V_{in} = 0 ~ 5.25V	–	–	2.5	μA
Three State (Offset) Leakage Current	P_{10} ~ P_{17}, P_{30} ~ P_{37}	$\|I_{TSI}\|$	V_{in} = 0.5 ~ 2.4V	–	–	10	μA
	P_{20} ~ P_{24}			–	–	100	
Output "High" Voltage	P_{30} ~ P_{37}	V_{OH}	I_{LOAD} = -205 μA	2.4	–	–	V
	P_{40} ~ P_{47}, E, SC_1, SC_2		I_{LOAD} = -145 μA	2.4	–	–	
	Other Outputs		I_{LOAD} = -100 μA	2.4	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{LOAD} = 1.6 mA	–	–	0.5	V
Darlington Drive Current	P_{10} ~ P_{17}	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Power Dissipation		P_D		–	–	1200	mW
Input Capacitance	P_{30} ~ P_{37}, P_{40} ~ P_{47}, SC_1	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz	–	–	12.5	pF
	Other Inputs			–	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}		4.0	–	5.25	V
	Operating	V_{SB}		4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	V_{SBB} = 4.0V	–	–	8.0	mA

*Except Mode Programming Levels.

• AC CHARACTERISTICS

BUS TIMING (V_{CC} = 5.0V±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	HD6801S0			HD6801S5			Unit
				min	typ	max	min	typ	max	
Cycle Time		t_{cyc}		1	–	10	0.8	–	10	μs
Address Strobe Pulse Width "High"*		PW_{ASH}		200	–	–	150	–	–	ns
Address Strobe Rise Time		t_{ASr}		5	–	50	5	–	50	ns
Address Strobe Fall Time		t_{ASf}		5	–	50	5	–	50	ns
Address Strobe Delay Time*		t_{ASD}		60	–	–	30	–	–	ns
Enable Rise Time		t_{Er}		5	–	50	5	–	50	ns
Enable Fall Time		t_{Ef}		5	–	50	5	–	50	ns
Enable Pulse Width "High" Time*		PW_{EH}		450	–	–	340	–	–	ns
Enable Pulse Width "Low" Time*		PW_{EL}		450	–	–	350	–	–	ns
Address Strobe to Enable Delay Time*		t_{ASED}		60	–	–	30	–	–	ns
Address Delay Time		t_{AD}	Fig. 1	–	–	260	–	–	260	ns
Address Delay Time for Latch (f = 1.0MHz)*		t_{ADL}	Fig. 2	–	–	270	–	–	260	ns
Data Set-up Write Time		t_{DSW}		225	–	–	115	–	–	ns
Data Set-up Read Time		t_{DSR}		80	–	–	70	–	–	ns
Data Hold Time	Read	t_{HR}		10	–	–	10	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	
Address Set-up Time for Latch*		t_{ASL}		60	–	–	50	–	–	ns
Address Hold Time for Latch		t_{AHL}		20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	ns
Peripheral Read Access Time	Non-Multiplexed Bus*	(t_{ACCN})		–	–	(610)	–	–	(420)	ns
	Multiplexed Bus*	(t_{ACCM})		–	–	(600)	–	–	(420)	
Oscillator stabilization Time		t_{RC}	Fig. 10	100	–	–	100	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 11	200	–	–	200	–	–	ns

*These timings change in approximate proportion to tcyc. The figures in this characteristics represent those when tcyc is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Peripheral Data Setup Time	Port 1, 2, 3, 4	t_{PDSU}	Fig. 3	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		t_{OSD1}	Fig. 5	–	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		t_{OSD2}	Fig. 5	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4	t_{PWD}	Fig. 4	–	–	400	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	Port 2**, 4	t_{CMOS}	Fig. 4	–	–	2.0	μs
Input Strobe Pulse Width		t_{PWIS}	Fig. 6	200	–	–	ns
Input Data Hold Time	port 3	t_{IH}	Fig. 6	50	–	–	ns
Input Data Set-up Time	Port 3	t_{IS}	Fig. 6	20	–	–	ns

*Except P_{21} **10kΩ pull up register required for Port 2

TIMER, SCI TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Timer Input Pulse Width	t_{PWT}		$2t_{cyc}$+200	–	–	ns
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 7	–	–	600	ns
SCI Input Clock Cycle	t_{Scyc}		1	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Mode Programming Input "Low" Voltage		V_{MPL}	Fig. 8	–	–	1.7	V
Mode Programming Input "High" Voltage		V_{MPH}		4.0	–	–	V
$\overline{RES}$ "Low" Pulse Width		PW_{RSTL}		3.0	–	–	t_{cyc}
Mode Programming Set-up Time		t_{MPS}		2.0	–	–	t_{cyc}
Mode Programming Hold Time	$\overline{RES}$ Rise Time ≧ 1μs	t_{MPH}		0	–	–	ns
	$\overline{RES}$ Rise Time < 1μs			100	–	–	

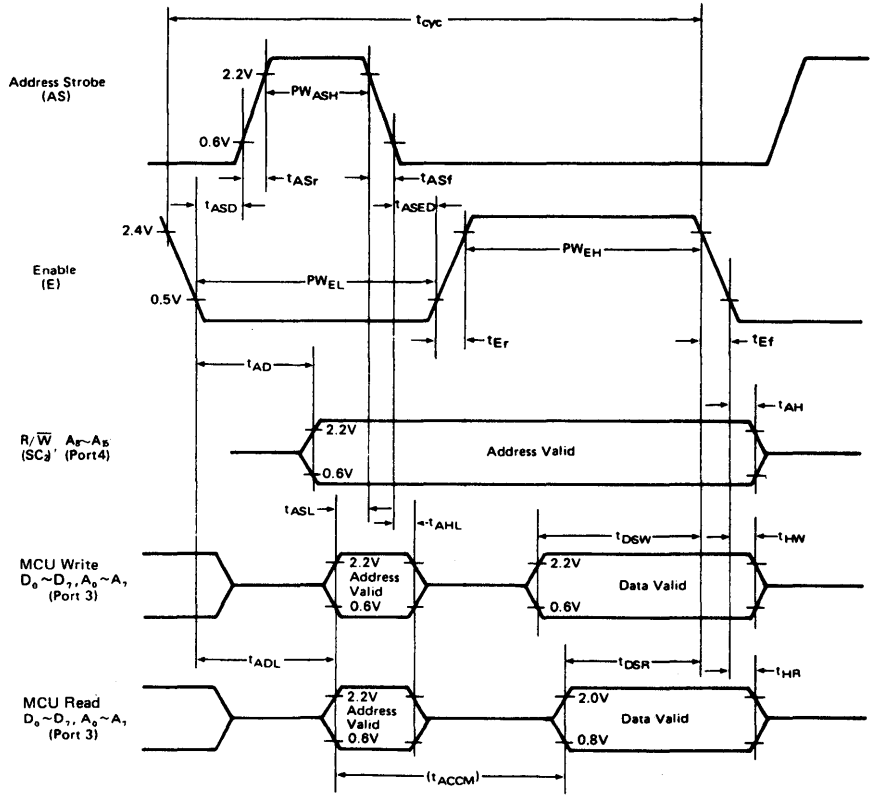


Figure 1 Expanded Multiplexed Bus Timing

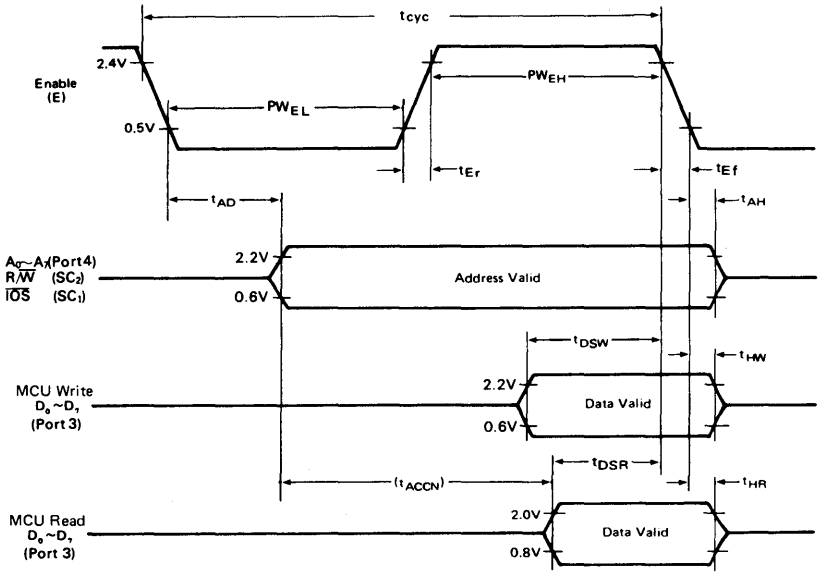
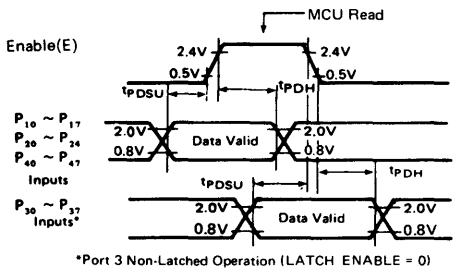
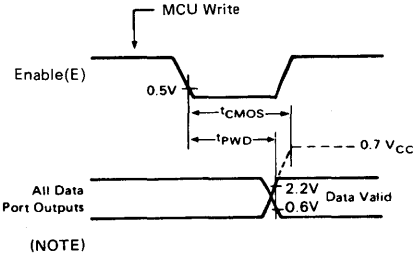


Figure 2 Expanded Non-Multiplexed Bus Timing



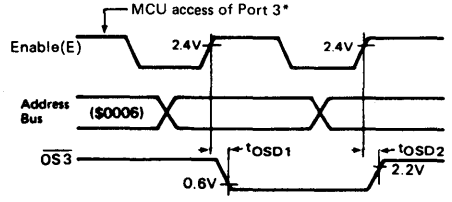
*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

Figure 3 Data Set-up and Hold Times (MCU Read)



(NOTE)
1. 10 kΩ Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P_{21}
3. Port 4 cannot be pulled above V_{CC}

Figure 4 Port Data Delay Timing (MCU Write)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

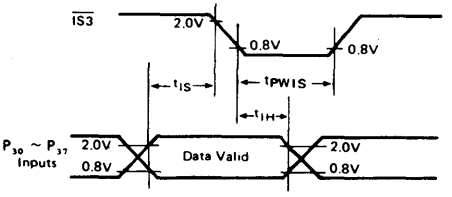


Figure 6 Port 3 Latch Timing (Single Chip Mode)

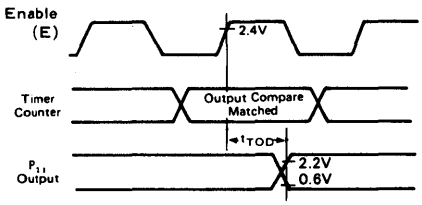


Figure 7 Timer Output Timing

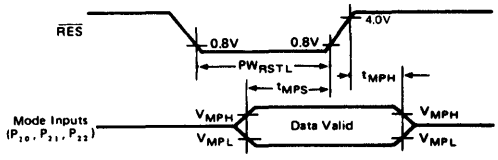
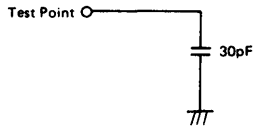
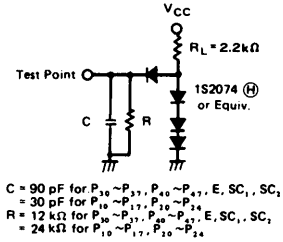


Figure 8 Mode Programming Timing



(a) CMOS Load



(b) TTL Load

Figure 9 Bus Timing Test Loads

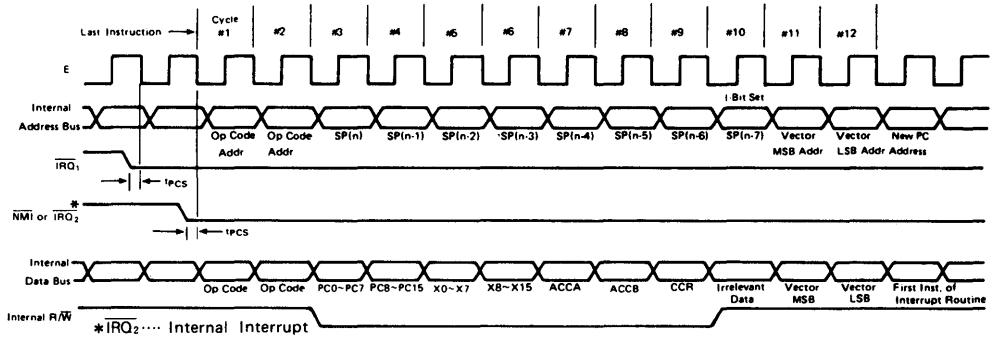


Figure 10 Interrupt Sequence

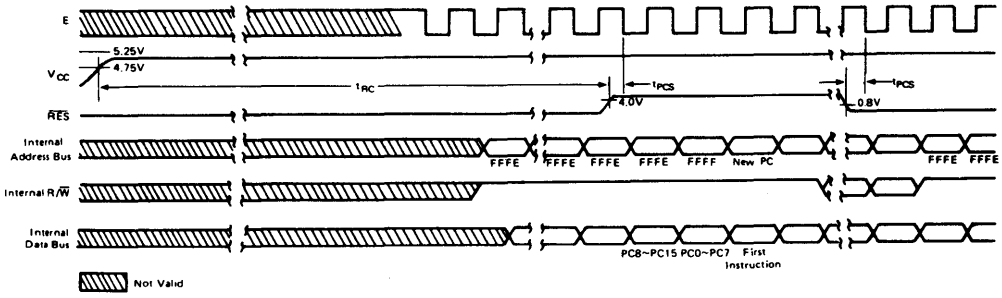


Figure 11 Reset Timing

■ SIGNAL DESCRIPTIONS

• Vcc and Vss

These two pins are used to supply power and ground to the chip. The voltage supplied will be +5 volts ±5%.

• XTAL and EXTAL

These connections are for a parallel resonant fundamental crystal, AT cut. Divide-by-4 circuitry is included with the internal clock, so a 4 MHz crystal may be used to run the system at 1 MHz. The divide-by-4 circuitry allows for use of the inexpensive 3.58 MHz Color TV crystal for non-time critical applications. Two 22pF capacitors are needed from the two crystal pins to ground to insure reliable operation. An example of the crystal interface is shown in Fig. 12. EXTAL may be driven by an external clock source at a 4 MHz rate to run at 1 MHz with 45% to 55% duty cycle. It is not restricted to 4 MHz, as it will divide by 4 any frequency less than or equal to 5 MHz. XTAL must be grounded if an external clock is used.

Nominal Crystal Parameter

Item \ Crystal	4 MHz	5 MHz
Co	7 pF max.	4.7 pF max.
R_S	60Ω max.	30Ω typ.

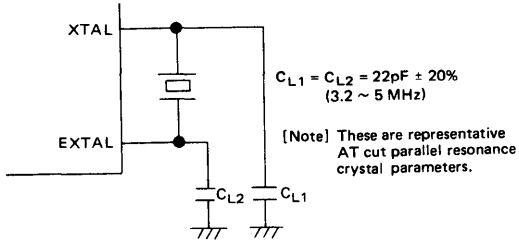


Figure 12 Crystal Interface

• Vcc Standby

This pin will supply +5 volts ±5% to the standby RAM on the chip. The first 64 bytes of RAM will be maintained in the power down mode with 8 mA current max. The circuit of figure 13 can be utilized to assure that V_{CC} Standby does not go below V_{SBB} during power down.

To retain information in the RAM during power down the following procedure is necessary:

1) Write "0" into the RAM enable bit, RAME. RAME is bit 6 of the RAM Control Register at location $0014. This disables the standby RAM, thereby protecting it at power down.
2) Keep V_{CC} Standby greater than V_{SBB}.

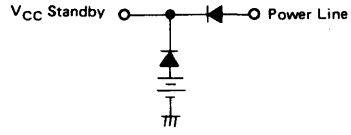


Figure 13 Battery Backup for V_{CC} Standby

• Reset ($\overline{RES}$)

This input is used to reset and start the MCU from a power down condition, resulting from a power failure or an initial startup of the processor. On power up, the reset must be held "Low" for at least 100 ms. When reset during operation, $\overline{RES}$ must be held "Low" at least 3 clock cycles.

When a "High" level is detected, the MCU does the following:

1) All the higher order address lines will be forced "High".
2) I/O Port 2 bits 2, 1, and 0 are latched into programmed control bits PC2, PC1 and PC0.
3) The last two ($FFFE, $FFFF) locations in memory will be used to load the program addressed by the program counter.
4) The interrupt mask bit is set. Clear before the CPU can recognize maskable interrupts.

• Enable (E)

This supplies the external clock for the rest of the system when the internal oscillator is used. It is a single phase, TTL compatible clock, and will be the divide-by-4 result of the crystal oscillator frequency. It will drive one TTL load and 90 pF capacitance.

• Non-Maskable Interrupt ($\overline{NMI}$)

A low-going edge on this input requests that a non-maskable-interrupt sequence be generated within the processor. As with interrupt Request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the Condition Code Register has no effect on $\overline{NMI}$.

In response to an $\overline{NMI}$ interrupt, the Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. At the end of the sequence, a 16-bit address will be loaded that points to a vectoring address located in memory locations $FFFC and $FFFD. An address loaded at these locations causes the CPU to branch to a non-maskable interrupt service routine in memory.

A 3.3 kΩ external resistor to Vcc should be used for wire-OR and optimum control of interrupts.

Inputs $\overline{IRQ_1}$ and $\overline{NMI}$ are hardware interrupt lines that are sampled during E and will start the interrupt routine on the $\overline{E}$ following the completion of an instruction.

• Interrupt Request ($\overline{IRQ_1}$)

This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will complete the current instruction before it recognizes the request. At that time, if the interrupt mask bit in the Condition Code Register is not set, the machine will begin an interrupt sequence. The Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. Next the CPU will respond to the interrupt request by setting the interrupt mask bit "High" so that no further maskable interrupts may occur. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations $FFF8 and $FFF9. An address loaded at these locations causes the CPU to branch to an interrupt routine in memory.

The $\overline{IRQ_1}$ requires a 3.3 kΩ external resistor to V_{CC} which should be used for wire-OR and optimum control of interrupts. Internal Interrupts will use an internal interrupt line ($\overline{IRQ_2}$). This interrupt will operate the same as $\overline{IRQ_1}$ except that it will use the vector address of $FFF0 through $FFF7. $\overline{IRQ_1}$ will have priority over $\overline{IRQ_2}$ if both occur at the same time. The Interrupt Mask Bit in the condition code register masks both interrupts (See Table 1).

Table 1 Interrupt Vector Location

	Vector MSB	Vector LSB	Interrupt
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$ (or $\overline{IS3}$)
	FFF6	FFF7	ICF (Input Capture)
	FFF4	FFF5	OCF (Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SC_1 (RDRF + ORFE + TDRE)

The following pins are available in the Single Chip Mode, and are associated with Port 3 only.

- **Input Strobe ($\overline{IS3}$) (SC_1)**

This sets an interrupt for the processor when the $\overline{IS3}$ Enable bit is set. As shown in Figure 6 Input Strobe Timing, $\overline{IS3}$ will fall t_{IS} minimum after data is valid on Port 3. If $\overline{IS3}$ Enable is set in the I/O Port 3 Control/Status Register, an interrupt will occur. If the latch enable bit in the I/O Port 3 Control/Status Register is set, this strobe will latch the input data from another device when that device has indicated that it has valid data.

- **Output Strobe ($\overline{OS3}$) (SC_2)**

This signal is used by the processor to strobe an external device, indicating valid data is on the I/O pins. The timing for the Output Strobe is shown in Figure 5. I/O Port 3 Control/Status Register is discussed in the following section.

The following pins are available in the Expanded Modes.

- **Read/Write (R/$\overline{W}$) (SC_2)**

This TTL compatible output signals the peripherals and memory devices whether the CPU is in a Read ("High") or a Write ("Low") state. The normal standby state of this signal is Read ("High"). This output can drive one TTL load and 90 pF capacitance.

- **I/O Strobe ($\overline{IOS}$) (SC_1)**

In the expanded non-multiplexed mode of operation, $\overline{IOS}$ internally decodes A_9 through A_{15} as "0"s and A_8 as a "1". This allows external access of the 256 locations from $0100 to $01FF. The timing diagrams are shown as figure 2.

- **Address Strobe (AS) (SC_1)**

In the expanded multiplexed mode of operation, address strobe is output on this pin. This signal is used to latch the 8 LSB's of address which are multiplexed with data on Port 3. An 8-bit latch is utilized in conjunction with Address Strobe, as shown in figure 19. So I/O Port 3 can become data bus during the E pulse. The timing for this signal is shown in Figure 1 of Bus Timing. This signal is also used to disable the address from the multiplexed bus allowing a deselect time, t_{ASD} before the data is enabled to the bus.

■ PORTS

There are four I/O ports on the HD6801S MCU; three 8-bit ports and one 5-bit port. There are two control lines associated with one of the 8-bit ports. Each port has an associated write only Data Direction Register which allows each I/O line to be programmed to act as an input or an output*. A "1" in the corresponding Data Direction Register bit will cause that I/O line to be an output. A "0" in the corresponding Data Direction Register bit will cause that I/O line to be an input. There are four ports: Port 1, Port 2, Port 3, and Port 4. Their addresses and the addresses of their Data Direction registers are given in Table 2.

* The only exception is bit 1 of Port 2, which can either be data input or Timer output.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

- **I/O Port 1**

This is an 8-bit port whose individual bits may be defined as inputs or outputs by the corresponding bit in its data direction register. The 8 output buffers have three-state capability, allowing them to enter a high impedance state when the peripheral data lines are used as inputs. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, these lines are TTL compatible and may also be used as a source of up to 1 mA at 1.5 V to directly drive a Darlington base. After Reset, the I/O lines are configured as inputs. In all three modes, Port 1 is always parallel I/O.

- **I/O Port 2**

This port has five lines that may be defined as inputs or outputs by its data direction register. The 5 output buffers have three-state capability, allowing them to enter a high impedance state when used as an input. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, this port has no internal pullup resistors but will drive TTL inputs directly. For driving CMOS inputs, external pullup resistors are required. After Reset, the I/O lines are configured as inputs. Three pins on Port 2 (pins 10, 9, and 8 of the chip) are used to program the mode of operation during reset. The values of these pins at reset are latched into the three MSB's (bits 7, 6, and 5) of Port 2 which are read-only. This is explained in the Mode Selection Section.

In all three modes, Port 2 can be configured as I/O and provides access to the Serial Communications Interface and the Timer. Bit 1 is the only pin restricted to data input or Timer output.

- **I/O Port 3**

This is an 8-bit port that can be configured as I/O, a data bus, or an address bus multiplexed with the data bus – depending on the mode of operation hardware programmed by the user at reset. As a data bus, Port 3 is bi-directional. As an input for peripherals, it must be supplied regular TTL levels, that is, greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0"

Its TTL compatible three-state output buffers can drive one TTL load and 90 pF capacitance. In the Expanded Modes, after reset, the data direction register is inhibited and data flow depends on the state of the R/$\overline{W}$ line. The input strobe ($\overline{IS3}$) and the output strobe ($\overline{OS3}$) used for handshaking are explained later.

In the three modes, Port 3 assumes the following characteristics:

Single Chip Mode: Parallel Inputs/Outputs as programmed by its associated Data Direction Register. There are two control lines associated with this port in this mode, an input strobe and an output strobe, that can be used for handshaking. They are controlled by the I/O Port 3 Control/Status Register explained at the end of this section. Three options of Port 3 operations are sumarized as follows: (1) Port 3 input data can be latched using $\overline{IS3}$ (SC_1) as a control signal, (2) $\overline{OS3}$ can be generated by either an CPU read or write to Port 3's Data Register, and (3) and $\overline{IRQ_1}$ interrupt can be enabled by an IS3 negative edge.
Port 3 latch and strobe timing is shown in Fig. 5 and Fig. 6.

Expanded Non-Multiplexed Mode: In this mode, Port 3 becomes the data bus (D_0~D_7).

Expanded Multiplexed Mode: In this mode, Port 3 becomes both the data bus (D_0~D_7) and lower bits of the address bus (A_0~A_7). An address strobe output is true when the address is on the port.

I/O PORT 3 CONTROL/STATUS REGISTER

	7	6	5	4	3	2	1	0
$000F	IS3 FLAG	$\overline{IS3}$ $\overline{IRQ_1}$ ENABLE	X	OSS	LATCH ENABLE	X	X	X

Bit 0; Not used.
Bit 1; Not used.
Bit 2; Not used.
Bit 3; LATCH ENABLE. This controls the input latch for I/O Port 3. If this bit is set "High" the input data will be latched with the falling edge of the Input Strobe, $\overline{IS3}$. This bit is cleared by reset, and the latch is "re-opened" with CPU read Port 3.
Bit 4; OSS. (Output Strobe Select) This bit will select if the Output Strobe should be generated at $\overline{OS3}$ (SC_2) by a write to I/O Port 3 or a read of I/O Port 3. When this bit is cleared the strobe is generated by a read Port 3. When this bit is set the strobe is generated by a write to Port 3.
Bit 5; Not used.
Bit 6; $\overline{IS3}$ $\overline{IRQ_1}$ ENABLE. When set, interrupt will be enabled whenever $\overline{IS3}$ FLAG is set; when clear, interrupt is inhibited. This bit is cleared by reset.
Bit 7; $\overline{IS3}$ FLAG. This is a read-only status bit that is set by the falling edge of the input strobe, $\overline{IS3}$ (SC_1). It is cleared by a read of the Control/Status Register followed by a read or write of I/O Port 3. Reset will clear this bit.

- **I/O Port 4**

This is an 8-bit port that can be configured as I/O or as address output lines depending on the mode of operation. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0"

As outputs, each line is TTL compatible and can drive 1 TTL load and 90 pF capacitance. After reset, the lines are configured as inputs. To use the pins as addresses, therefore, they should be programmed as outputs. In the three modes, Port 4 assumes the following characteristics:

Single Chip Mode: Parallel Inputs/Outputs as programmed by its associated Data Direction Register.

Expanded Non-Multiplexed Mode: Port 4 is configured as the lower order address lines (A_0~A_7) by writing "1"s to the data direction register. When all eight address lines are not needed, the remaining lines, starting with the most significant bit, may be used as I/O (inputs only).

Expanded Multiplexed Mode: Port 4 is configured as the higher order address lines (A_8~A_{15}) by writing "1"s to the data direction register. When all eight address lines are not needed, the remaining lines, starting with the most significant bit, may be used as I/O (Inputs only).

■ OPERATION MODES

The operation modes that HD6801S will operate after reset is determined by hardware that the user must wire on pins 10, 9, and 8 of the chip. These pins are the three LSB's (I/O 2, I/O 1, and I/O 0 respectively) of Port 2. They are latched into programmed control bits PC2, PC1, and PC0 when reset goes high. I/O Port 2 Register is shown below.

PORT 2 DATA REGISTER

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

An example of external hardware that could be used for Mode Selection is shown in Fig. 14. The HD14053B provides the isolation between the peripheral device and MCU during reset, which is necessary if data conflict can occur between peripheral device and Mode generation circuit.

As bits 5, 6 and 7 of Port 2 are read-only, the mode cannot be changed through software. The mode selections are shown in Table 3.

The HD6801S can operate in three basic modes;
(1) Single Chip Mode, (2) Expanded Multiplexed Mode (compatible with HMCS6800 peripheral family) (3) Expanded Non-Multiplexed Mode.

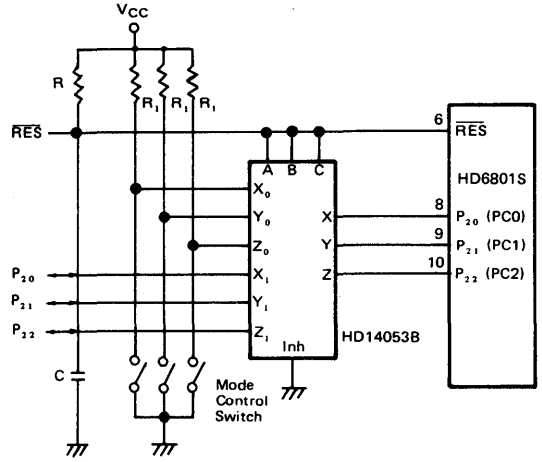
- **Single Chip Mode**

In the Single Chip Mode the Ports are configured as I/O.

This is shown in Figure 16 the single Chip Mode. In this mode, Port 3 will have two associated control lines, an input strobe and an output strobe for handshaking data.

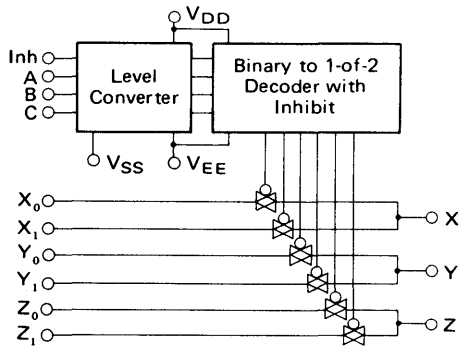
- **Expanded Non-Multiplexed Mode**

In this mode the HD6801S will directly address HMCS6800 peripherals with no external logic. In this mode Port 3 becomes the data bus. Port 4 becomes the A_0~A_7 address bus or partial address and I/O (inputs only). Port 2 can be parallel I/O, serial I/O, Timer, or any combination of them. Port 1 is parallel I/O only. In this mode the HD6801S is expandable to 256 locations. The eight address lines associated with Port 4 may be substituted for I/O (inputs only) if a fewer number of address lines will satisfy the application (See Figure 17).



[NOTES] 1) Mode 7 as shown
2) RC≈Reset time constant
3) R_1=10kΩ

Figure 14 Recommended Circuit for Mode Selection



Truth Table

Control Input				On Switch		
Inhibit	Select					
	C	B	A	HD14053B		
0	0	0	0	Z_0	Y_0	X_0
0	0	0	1	Z_0	Y_0	X_1
0	0	1	0	Z_0	Y_1	X_0
0	0	1	1	Z_0	Y_1	X_1
0	1	0	0	Z_1	Y_0	X_0
0	1	0	1	Z_1	Y_0	X_1
0	1	1	0	Z_1	Y_1	X_0
0	1	1	1	Z_1	Y_1	X_1
1	X	X	X	–		

Figure 15 HD14053B Multiplexers/Demultiplexers

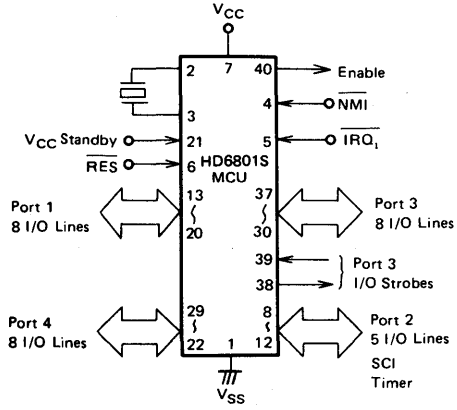


Figure 16 HD6801S MCU Single-Chip Mode

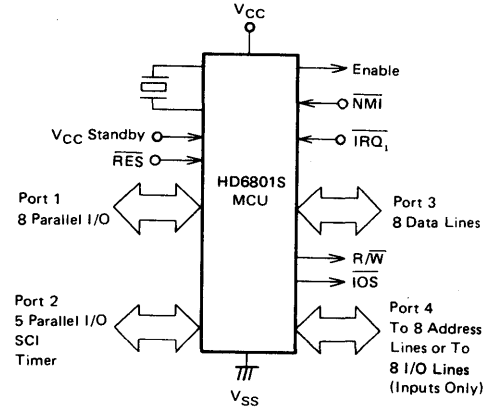


Figure 17 HD6801S MCU Expanded Non-Multiplexed Mode

- **Expanded Multiplexed Mode**

In this mode Port 4 becomes higher order address lines with an alternative of substituting some of the address lines for I/O (inputs only). Port 3 is the data bus multiplexed with the lower order address lines differentiated by an output called Address Strobe. Port 2 is 5 lines of Parallel I/O, SCI, Timer, or any combination of them. Port 1 is 8 Parallel I/O lines. In this mode it is expandable to 65k bytes. (See Figure 18).

- **Lower order Address Bus Latches**

Since the data bus is multiplexed with the lower order address bus in Port 3, latches are required to latch those address bits. The 74LS373 Transparent octal D-type latch can be used with the HD6801S to latch the least significant address byte. Figure 19 shows how to connect the latch to the HD6801S. The output control to the 74LS373 may be connected to ground.

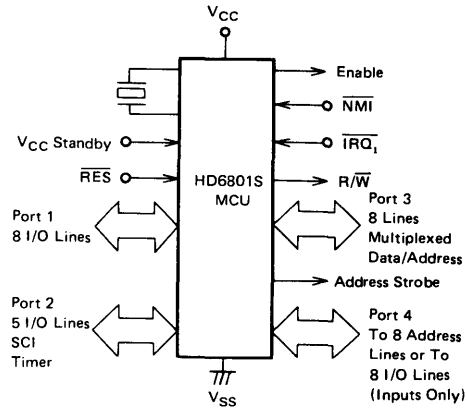
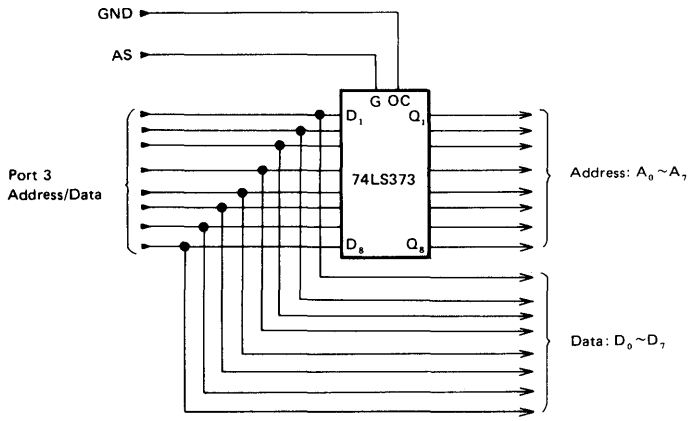


Figure 18 HD6801S MCU Expanded Multiplexed Mode



Function Table

Output Control	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 19 Latch Connection

- **Mode and Port Summary MCU Signal Description**

This section gives a description of the MCU signals for the various modes. SC_1 and SC_2 are signals which vary with the mode that the chip is in.

MODE	PORT 1 Eight Lines	PORT 2 Five Lines	PORT 3 Eight Lines	PORT 4 Eight Lines	SC_1	SC_2
SINGLE CHIP	I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)
EXPANDED MUX	I/O	I/O	ADDRESS BUS (A_0~A_7) DATA BUS (D_0~D_7)	ADDRESS BUS* (A_8~A_{15})	AS(O)	R/$\overline{W}$(O)
EXPANDED NON-MUX	I/O	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS* (A_0~A_7)	$\overline{IOS}$(O)	R/$\overline{W}$(O)

*These lines can be substituted for I/O (Input Only) starting with the most significant address line.
I = Input
O = Output
R/$\overline{W}$ = Read/$\overline{Write}$
$\overline{IS3}$ = Input Strobe
$\overline{OS3}$ = Output Strobe
$\overline{IOS}$ = I/O Select
SC = Strobe Control
AS = Address Strobe

Table 3 Mode Selection Summary

Mode	$P_{2.2}$ (PC2)	$P_{2.1}$ (PC1)	$P_{2.0}$ (PC0)	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX	Multiplexed/No RAM & ROM
2	L	H	L	E	I	E	MUX	Multiplexed/RAM
1	L	L	H	I	I	E	MUX	Multiplexed/RAM & ROM
0	L	L	L	I	I	I(3)	MUX	Multiplexed Test

LEGEND:
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

[NOTES]
1) Internal RAM is addressed at $XX80
2) Internal ROM is disabled
3) $\overline{RES}$ vector is external for 2 cycles after $\overline{RES}$ goes "High"
4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
5) Addresses associated with Port 3 are considered external in Modes 5 and 6
6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register

■ MEMORY MAPS

The MCU can provide up to 65k bytes address space depending on the operating mode. A memory map for each operating mode is shown in Figure 20. The first 32 locations of each map are reserved for the MCU's internal register area, as shown in Table 4. With exceptions as indicated.

Table 4 Internal Register Area

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External address in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No. $\overline{IOS}$)
** External addresses in Modes 0, 1, 2, 3
*** 1=Output, 0=Input

■ INTERRUPT FLOWCHART

The Interrupt flow chart is depicted in Figure 24 and is common to every interrupt excluding reset.

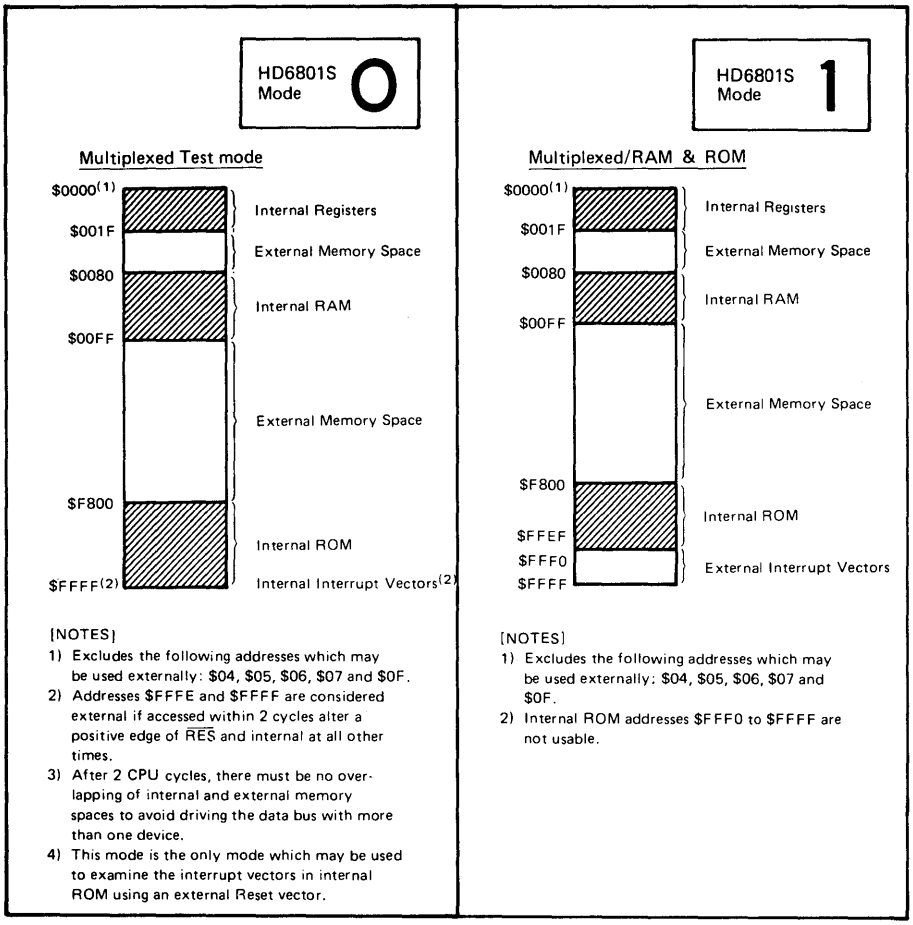


Figure 20 HD6801S Memory Maps

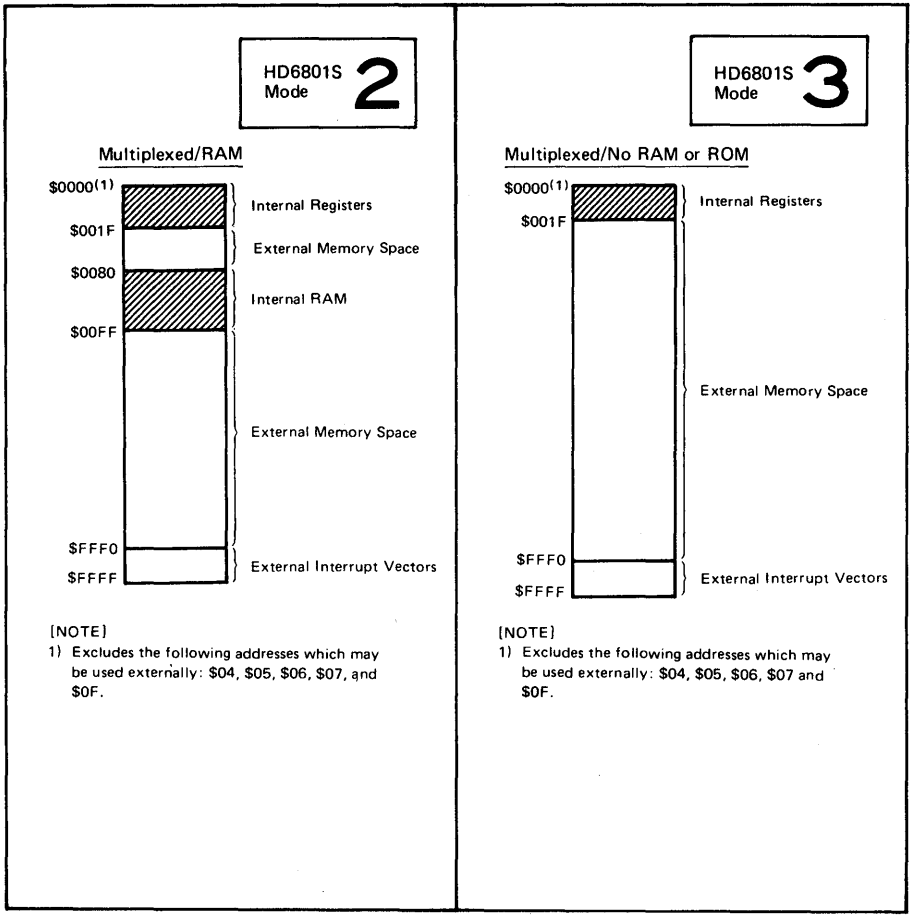


Figure 20 HD6801S Memory Maps (Continued)

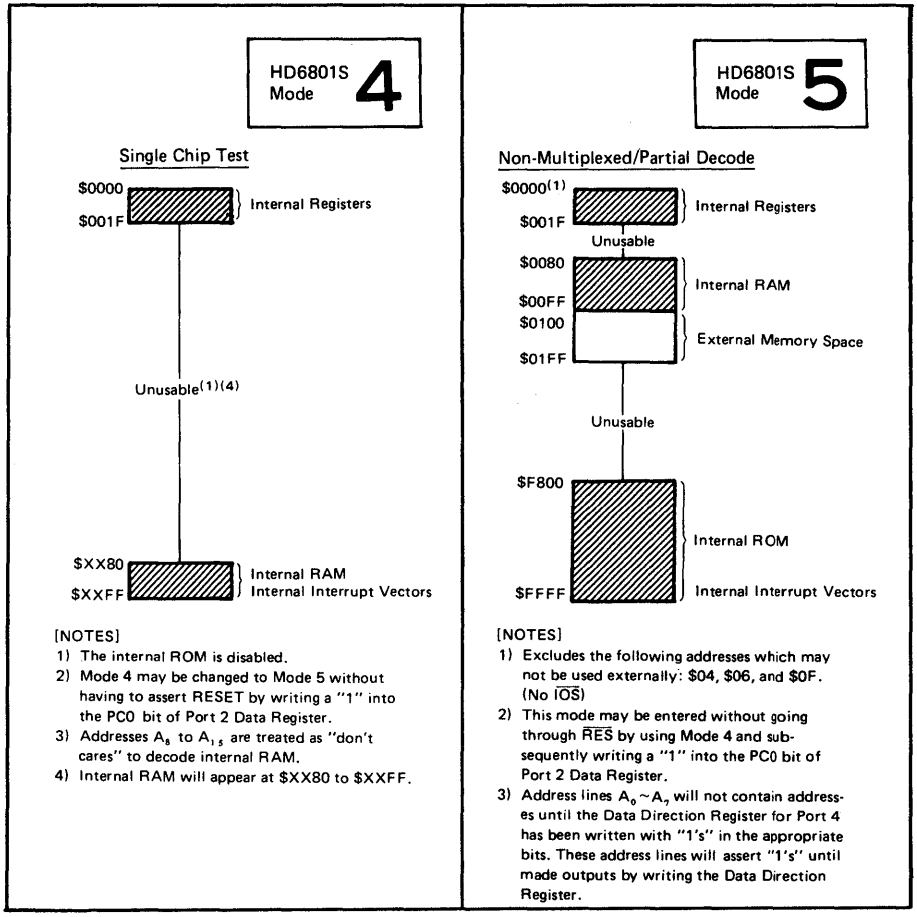
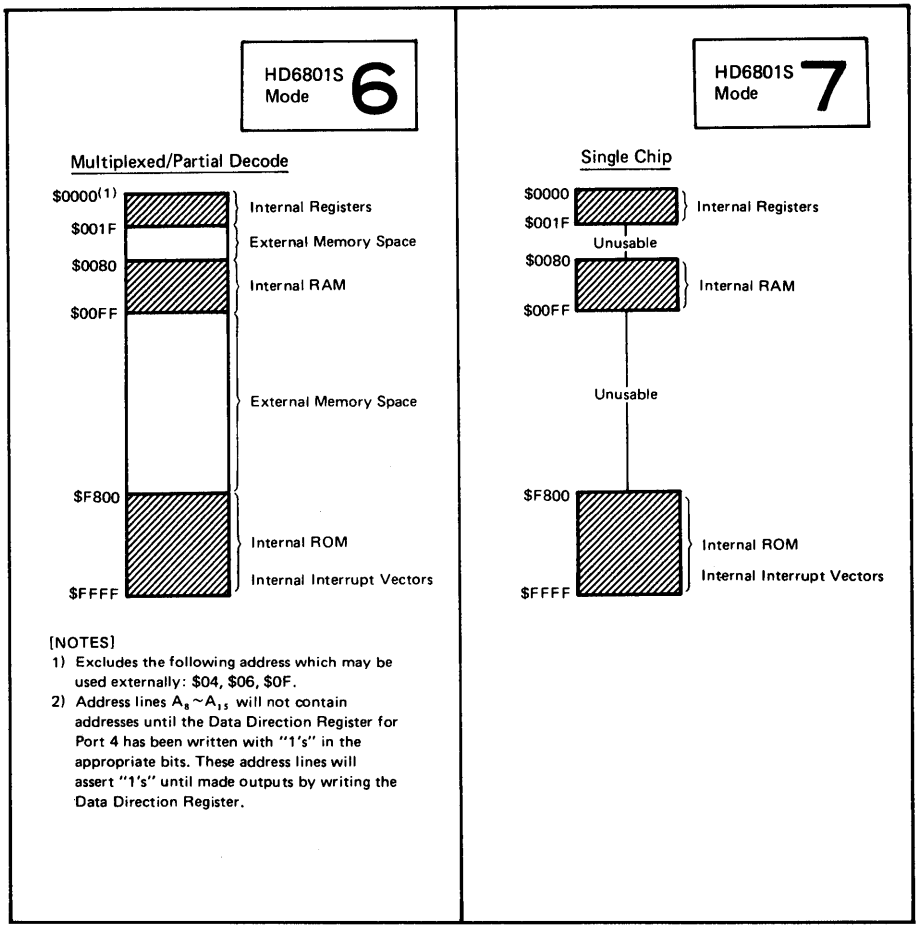


Figure 20 HD6801S Memory Maps (Continued)



[NOTES]
1) Excludes the following address which may be used externally: $04, $06, $0F.
2) Address lines $A_8 \sim A_{15}$ will not contain addresses until the Data Direction Register for Port 4 has been written with "1's" in the appropriate bits. These address lines will assert "1's" until made outputs by writing the Data Direction Register.

Figure 20 HD6801S Memory Maps (Continued)

■ PROGRAMMABLE TIMER

The HD6801S contains an on-chip 16-bit programmable timer which may be used to perform measurements on an input waveform while independently generating an output waveform. Pulse widths for both input and output signals may vary from a few microseconds to many seconds. The timer hardware consists of

- an 8-bit control and status register,
- a 16-bit free running counter,
- a 16-bit output compare register, and
- a 16-bit input capture register

A block diagram of the timer registers is shown in Figure 21.

- **Free Running Counter ($0009:$000A)**

The key element in the programmable timer is a 16-bit free running counter which is driven to increasing values by E (Enable). The counter value may be read by the CPU software at any time. The counter is cleared to zero by reset and may be considered a read-only register with one exception. Any CPU write to the counter's address ($09) will always result in preset value of $FFF8 being loaded into the counter regardless of the value involved in the write. This preset figure is intended for testing operation of the part, but may be of value in some applications.

- **Output Compare Register ($000B:$000C)**

The Output Compare Register is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly compared with the current value of the free running counter. When a match is found, a flag is set (OCF) in the Timer Control and Status Register (TCSR) and the current value of the Output Level bit (OLVL) in the TCSR is clocked to the Output Level Register. Providing the Data Direction Register for Port 2, Bit 1 contains a "1" (Output), the output level register value will appear on the pin for Port 2 Bit 1. The values in the Output Compare Register and Output level bit may then be changed to control the output level on the next compare value. The Output Compare Register is set to $FFFF during $\overline{RES}$. The Compare function is inhibited for one cycle following a write to the high byte of the Output Compare Register to insure a valid 16-bit value is in the register before a compare is made.

- **Input Capture Register ($000D:$000E)**

The Input Capture Register is a 16-bit read-only register used to store the current value of the free running counter when the proper transition of an external input signal occurs. The input transition change required to trigger the counter transfer is controlled by the input Edge bit (IEDG) in the TCSR. The Data Direction Register bit for Port 2 Bit 0, should* be clear (zero) in order to gate in the external input signal to the edge detect unit in the timer.

The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

* With Port 2 Bit 0 configured as an output and set to "1", the external input will still be seen by the edge detect unit.

- **Timer Control and Status Register (TCSR) ($0008)**

The Timer Control and Status Register consists of an 8-bit register of which all 8 bits are readable but only the low order 5 bits may be written. The upper three bits contain read-only timer status information and indicate the followings.

- a proper transition has taken place on the input pin with a subsequent transfer of the current counter value to the input capture register.
- a match has been found between the value in the free running counter and the output compare register, and
- when $0000 is in the free running counter.

Each of the flags may be enabled onto the HD6801 internal bus ($\overline{IRQ_2}$) with an individual Enable bit in the TCSR. If the

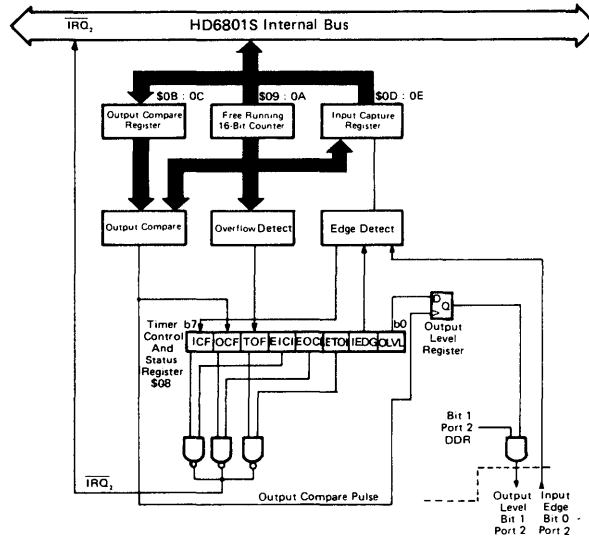


Figure 21 Block Diagram of Programmable Timer

Timer Control and Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

I-bit in the HD6801S Condition Code Register has been cleared, a priority vectored interrupt will occur corresponding to the flag bit(s) set. A description for each bit follows:

Bit 0 **OLVL** Output Level – This value is clocked to the output level register on a successful output compare. If the DDR for Port 2 bit 1 is set, the value will appear on the output pin.

Bit 1 **IEDG** Input Edge – This bit controls which transition of an input will trigger a transfer of the counter to the input capture register. The DDR for Port 2 Bit 0 must be clear for this function to operate. IEDG = 0 Transfer takes place on a negative edge ("High"-to-"Low" transition).
IEDG = 1 Transfer takes place on a positive edge ("Low"-to-"High" transition).

Bit 2 **ETOI** Enable Timer Overflow Interrupt – When set, this bit enables $\overline{IRQ_2}$ to occur on the internal bus for a TOF interrupt; when clear the interrupt is inhibited.

Bit 3 **EOCI** Enable Output Compare Interrupt – When set, this bit enables $\overline{IRQ_2}$ to appear on the internal bus for an output compare interrupt; when clear the interrupt is inhibited.

Bit 4 **EICI** Enable Input Capture Interrupt– When set, this bit enables $\overline{IRQ_2}$ to occur on the internal bus for an input capture interrupt; when clear the interrupt is inhibited.

Bit 5 **TOF** Timer Overflow Flag – This read-only bit is set when the counter contains $FFFF. It is cleared by a read of the TCSR (with TOF set) followed by an CPU read of the Counter ($09).

Bit 6 **OCF** Output Compare Flag – This read-only bit is set when a match is found between the output compare register and the free running counter. It is cleared by a read of the TCSR (with OCF set) followed by an CPU write to the output compare register ($0B or $0C).

Bit 7 **ICF** Input Capture Flag – This read-only status bit is set by a proper transition on the input; it is cleared by a read of the TCSR (with ICF set) followed by an CPU read of the Input Capture Register ($0D).

■ SERIAL COMMUNICATION INTERFACE

The HD6801S contains a full-duplex asynchronous serial communication interface (SCI) on chip. The controller comprises a transmitter and a receiver which operate independently or each other but in the same data format and at the same data rate. Both transmitter and receiver communicate with the CPU via the data bus and with the outside world via pins 2, 3, and 4 of Port 2. The hardware, software, and registers are explained in the following paragraphs.

• Wake-Up Feature

In a typical multi-processor application, the software protocol will usually contain a destination address in the initial byte(s) of the message. In order to permit non-selected MCU's to ignore the remainder of the message, a wake-up feature is included whereby all further interrupt processing may be optionally inhibited until the beginning of the next message. When the next message appears, the hardware re-enables (or "wakes-up") the for the next message. The "wake-up" is automatically triggered by a string of ten consecutive 1's which indicates an idle transmit line. The software protocol must provide for the short idle period between any two consecutive messages.

• Programmable Options

The following features of the HD6801S serial I/O section are programmable:

- format – standard mark/space (NRZ)
- Clock – external or internal
- baud rate – one of 4 per given CPU ϕ_2 clock frequency or external clock ×8 input
- wake-up feature – enabled or disabled
- Interrupt requests – enabled or masked individually for transmitter and receiver data registers
- clock output – internal clock enabled or disabled to Port 2 (Bit 2)
- Port 2 (bits 3 and 4) – dedicated or not dedicated to serial I/O individually for transmitter and receiver.

• Serial Communication Hardware

The serial communication hardware is controlled by 4 registers as shown in Figure 22. The registers include:

- an 8-bit control and status register
- a 4-bit rate and mode control register (write-only)
- an 8-bit read-only receive data register and
- an 8-bit write-only transmit data register.

In addition to the four registers, the serial I/O section utilizes bit 3 (serial input) and bit 4 (serial output) of Port 2. Bit 2 of Port 2 is utilized if the internal-clock-out or external-clock-in options are selected.

Transmit/Receive Control and Status (TRCS) Register

The TRCS register consists of an 8-bit register of which all 8 bits may be read while only bits 0~4 may be written. The register is initialized to $20 by reset. The bits in the TRCS register are defined as follows:

Transmit/Receive Control and Status Register

7	6	5	4	3	2	1	0	
RDRE	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR : $0011

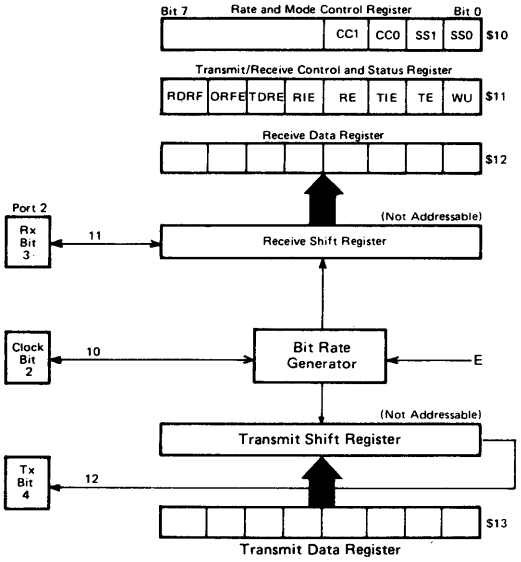


Figure 22 Serial I/O Registers

Bit 0 **WU** "Wake-up" on Next Message – set by HD6801S software and cleared by hardware on receipt of ten consecutive 1's or reset of RE flag. It should be noted that RE flag should be set in advance of CPU set of WU flag.

Bit 1 **TE** Transmit Enable – set by HD6801S to produce preamble of nine consecutive 1's and to enable gating of transmitter output to Port 2, bit 4 regardless of the DDR value corresponding to this bit; when clear, serial I/O has no effect on Port 2 bit 4.

TE set should be after at least one bit time of data transmit rate from the set-up of transmit data rate and mode.

Bit 2 **TIE** Transmit Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 5 (TDRE) is set; when clear, the TDRE value is masked from the bus.

Bit 3 **RE** Receiver Enable – when set, gates Port 2 bit 3 to input of receiver regardless of DDR value for this bit; when clear, serial I/O has no effect on Port 2 bit 3.

Bit 4 **RIE** Receiver Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 7 (RDRF) or bit 6 (ORFE) is set; when clear, the interrupt is masked.

Bit 5 **TDRE** Transmit Data Register Empty – set by hardware when a transfer is made from the transmit data register to the output shift register. The TDRE bit is cleared by reading the status register, then writing a new byte into the transmit data register, TDRE is initialized to 1 by reset.

Bit 6 **ORFE** Over-Run-Framing Error – set by hardware when an overrun or framing error occurs (receive only). An overrun is defined as a new byte received with last byte still in Data Register/Buffer. A framing error has occurred when the byte boundaries in bit stream are not synchronized to bit counter. The ORFE bit is cleared by reading the status register, then reading the Receive Data Register, or by reset.

Bit 7 **RDRF** Receiver Data Register Full – Set by hardware when a transfer from the input shift register to the receiver data register is made. The RDRF bit is cleared by reading the status register, then reading the Receive Data Register, or by reset.

Rate and Mode Control Register

The Rate and Mode Control register controls the following serial I/O variables:

- Baud rate
- format
- clocking source, and
- Port 2 bit 2 configuration

The register consists of 4 bits all of which are write-only and cleared by reset. The 4 bits in the register may be considered as a pair of 2-bit fields. The two low order bits control the bit rate for internal clocking and the remaining two bits control the format and clock select logic. The register definition is as follows:

Rate and Mode Control Register

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	ADDR : $0010

Bit 0 **SS0** } Speed Select – These bits select the Baud rate for
Bit 1 **SS1** } the internal clock. The four rates which may be selected are a function of the CPU ϕ_2 clock frequency. Table 5 lists the available Baud rates.

Bit 2 **CC0** } Clock Control and Format Select – this 2-bit field
Bit 3 **CC1** } controls the format and clock select logic. Table 6 defines the bit field.

Table 5 SCI Bit Times and Rates

SS1 : SS0	XTAL	2.4576 MHz	4.0 MHz	4.9152 MHz*
	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800 Baud
0 1	E ÷ 128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	E ÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	E ÷ 4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

* HD6801S5 Only

Table 6 SCI Format and Clock Source Control

CC1 : CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	**	**
0 1	NRZ	Internal	Not Used	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS; bit 4 is used for serial output if TE = "1" in TRCS.

Internally Generated Clock

If the user wishes for the serial I/O to furnish a clock, the following requirements are applicable:

- the values of RE and TE are immaterial.
- CC1, CC0 must be set to 10
- the maximum clock rate will be E ÷ 16.
- the clock will be at 1× the bit rate and will have a rising edge at mid-bit.

Externally Generated Clock

If the user wishes to provide an external clock for the serial I/O, the following requirements are applicable:

- the CC1, CC0, field in the Rate and Mode Control Register must be set to 11,
- the external clock must be set to 8 times (×8) the desired baud rate and
- the maximum external clock frequency is 1.0 MHz.

- **Serial Operations**

The serial I/O hardware should be initialized by the HD6801S software prior to operation. This sequence will normally consist of;

- writing the desired operation control bits to the Rate and Mode Control Register and
- writing the desired operational control bits in the Transmit/Receive Control and Status Register.

The Transmitter Enable (TE) and Receiver Enable (RE) bits may be left set for dedicated operations.

Transmit Operations

The transmit operation is enabled by the TE bit in the Transmit/Receive Control and Status Register. This bit when set, gates the output of the serial transmit shift register to Port 2 Bit 4 and takes unconditional control over the Data Direction Register value for Port 2, Bit 4.

Following a $\overline{\text{RES}}$ the user should configure both the Rate and Mode Control Register and the Transmit/Receive Control and Status Register for desired operation. Setting the TE bit during this procedure initiates the serial output by first transmitting a nine-bit preamble of 1's. Following the preamble, internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or,
2) if data has been loaded into the Transmit Data Register (TDRE = 0), the word is transferred to the output shift register and transmission of the data word will begin.

During the data transmit, the 0 start bit is first transmitted. Then the 8 data bits (beginning with bit 0) followed by the stop bit, are transmitted. When the Transmitter Data Register has been emptied, the hardware sets the TDRE flag bit.

If the HD6801S fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the parallel data register to the serial output register should occur) then a 1 will be sent (instead of a 0) at "Start" bit time, followed by more 1's until more data is supplied to the data register. No 0's will be sent while TDRE remains a 1.

Receive Operation

The receive operation is enabled by the RE bit which gates in the serial input through Port 2 Bit 3. The receiver section operation is conditioned by the contents of the Transmit/Receive Control and Status Register and the Rate and Mode Control Register.

The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the NRZ Mode, the received bit stream is synchronized by the first 0 (space) encountered.

The approximate center of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit) a framing error is assumed, and bit ORFE is set. If the tenth bit as a 1, the data is transferred to the Receive Data Register, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the HD6801S responds to either flag (RDRF or ORFE) by reading the status register followed by reading the Data Register, RDRF (or ORFE) will be cleared.

■ RAM CONTROL REGISTER

This register, which is addressed at $0014, gives status information about the standby RAM. A "0" in the RAM enable bit (RAME) will disable the standby RAM, thereby protecting it at power down if V_{CC} Standby is held greater than V_{SBB} volts, as explained previously in the signal description for V_{CC} Standby.

RAM Control Register

$0014	STBY PWR	RAME	X	X	X	X	X	X

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 **RAME** The RAM Enable control bit allows the user the ability to disable the standby RAM. This bit is set to a logic "1" by $\overline{RES}$ which enables the standby RAM and can be written to one or zero under program control. When the RAM is disabled, data is read from external memory.
Big 7 **STBY PWR** The Standby Power bit is cleared when the standby voltage is removed. This bit is a read/write status flag that the user can read which indicates that the standby RAM voltage has been applied, and the data in the standby RAM is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD6801S is upward object code compatible with the HD6800 as it implements the full HMCS6800 instruction set. The execution times of key instructions have been reduced to increase throughout. In addition, new instructions have been added; these include 16-bit operations and a hardware multiply.

Included in the instruction set section are the following:

- CPU Programming Model (Figure 23)
- Addressing modes
- Accumulator and memory instructions – Table 7
- New instructions
- Index register and stack manipulations instructions – Table 8
- Jump and branch instructions – Table 9
- Condition code register manipulation instructions – Table 10
- Instructions Execution times in machine cycles – Table 11
- Summary of cycle by cycle operation – Table 12
- Op codes Map – Table 13

• CPU Programming Model

The programming model for the HD6801S is shown in Figure 23. The double (D) accumulator is physically the same as the Accumulator A concatenated with the Accumulator B so that any operation using accumulator D will destroy information in A and B.

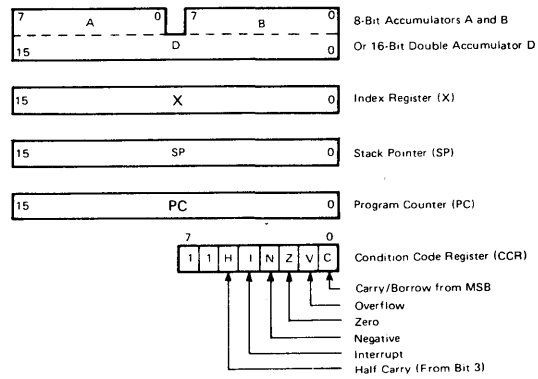


Figure 23 CPU Programming Model

• CPU Addressing Modes

The HD6801S eight-bit microcomputer unit has seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 11 along with the associated instruction execution time that is given in machine cycles. With a clock frequency of 4 MHz, these times would be microseconds.

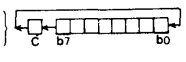
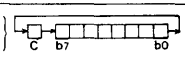
Accumulator (ACCX) Addressing

In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

Immediate Addressing

In immediate addressing, the operand is contained in the second byte of the instruction except LDS and LDX which have the operand in the second and third bytes of the instruction. The CPU addresses this location when it fetches the immediate instruction for execution. These are two or three-byte instructions.

Table 7 Accumulator & Memory Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	2	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	2	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	2	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	10	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	3	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	6	↕
	ROLA													49	2	1		•	•	↕	↕	6	↕
	ROLB													59	2	1		•	•	↕	↕	6	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	6	↕
	RORA													46	2	1		•	•	↕	↕	6	↕
	RORB													56	2	1		•	•	↕	↕	6	↕

The Condition Code Register notes are listed after Table 10.

(Continued)

Table 7 Accumulator & Memory Instructions (Continued)

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	3	1	ACC A/ ACC B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	3	1	0→ ACC A/ ACC B →C	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	2	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B - 00	•	•	↕	↕	R	R

The Condition Code Register notes are listed after Table 10.

Direct Addressing

In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine i.e., locations zero through 255. Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random access memory. These are two-byte instructions.

Extended Addressing

In extended addressing, the address contained in the second byte of the instruction is used as the higher eight-bits of the address of the operand. The third byte of the instruction is used as the lower eight-bits of the address for the operand. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest eight bits in the CPU. The carry is then added to the higher order eight bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

Implied Addressing

In the implied addressing mode the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

Relative Addressing

In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest eight bits plus two. The carry or borrow is then added to the high eight bits. This allows the user to address data within a range of -126 to +129 bytes of the present instruction. These are two-byte instructions.

- **New Instructions**

In addition to the existing 6800 Instruction Set, the following new instructions are incorporated in the HD6801S Microcomputer.

ABX Adds the 8-bit unsigned accumulator B to the 16-bit X-Register taking into account the possible carry out of the low order byte of the X-Register.

ADDD Adds the double precision ACCD* to the double precision value M:M+1 and places the results in ACCD.

ASLD Shifts all bits of ACCD one place to the left. Bit 0 is loaded with zero. The C bit is loaded from the most significant bit of ACCD.

LDD Loads the contents of double precision memory location into the double accumulator A:B. The condition codes are set according to the data.

LSRD Shifts all bits of ACCD one place to the right. Bit 15 is loaded with zero. The C bit is loaded from the least significant bit to ACCD.

MUL Multiplies the 8 bits in accumulator A with the 8 bits in accumulator B to obtain a 16-bit unsigned number in A:B, ACCA contains MSB of result.

PSHX The contents of the index register is pushed onto the stack at the address contained in the stack pointer. The stack pointer is decremented by 2.

PULX The index register is pulled from the stack beginning at the current address contained in the stack pointer +1. The stack pointer is incremented by 2 in total.

STD Stores the contents of double accumulator A:B in memory. The contents of ACCD remain unchanged.

SUBD Subtracts the contents of M:M + 1 from the contents of double accumulator AB and places the result in ACCD.

BRN Never branches. If effect, this instruction can be considered a two byte NOP (No operation) requiring three cycles for execution.

CPX Internal processing modified to permit its use with any conditional branch instruction.

*ACCD is the 16 bit register (A:B) formed by concatenating the A and B accumulators. The A-accumulator is the most significant byte.

Table 8 Index Register and Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X–M: M + 1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X – 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	3	1	SP – 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	3	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	3	1	X – 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{sp}, SP – 1 → SP X_H → M_{sp}, SP – 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•

The Condition Code Register notes are listed after Table 10.

Table 9 Jump and Branch Instructions

Operations	Mnemonic	Addressing Modes													Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	6	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		—— ⑧ ——					
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	⑨	•	•	•	•

Table10 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code Register					
		IMPLIED				5	4	3	2	1	0
		OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	—— ⑩ ——					
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

Condition Code Register Notes: (Bit set it test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to result of N⊕C after shift has occurred.
⑦ (Bit N) Test: Result less than zero? (Bit 15 = 1)
⑧ (All) Load Condition Code Register from Stack. (See Special Operations)
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All) Set according to the contents of Accumulator A.
⑪ (Bit C) Set equal to result of Bit 7 (ACCB)

Table 11 Instruction Execution Times in Machine Cycles

	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative		ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA	•	•	•	•	•	2	•	INX	•	•	•	•	•	3	•
ABX	•	•	•	•	•	3	•	JMP	•	•	•	3	3	•	•
ADC	•	2	3	4	4	•	•	JSR	•	•	5	6	6	•	•
ADD	•	2	3	4	4	•	•	LDA	•	2	3	4	4	•	•
ADDD	•	4	5	6	6	•	•	LDD	•	3	4	5	5	•	•
AND	•	2	3	4	4	•	•	LDS	•	3	4	5	5	•	•
ASL	2	•	•	6	6	•	•	LDX	•	3	4	5	5	•	•
ASLD	•	•	•	•	•	3	•	LSR	2	•	•	6	6	•	•
ASR	2	•	•	6	6	•	•	LSRD	•	•	•	•	•	3	•
BCC	•	•	•	•	•	•	3	MUL	•	•	•	•	•	10	•
BCS	•	•	•	•	•	•	3	NEG	2	•	•	6	6	•	•
BEQ	•	•	•	•	•	•	3	NOP	•	•	•	•	•	2	•
BGE	•	•	•	•	•	•	3	ORA	•	2	3	4	4	•	•
BGT	•	•	•	•	•	•	3	PSH	3	•	•	•	•	•	•
BHI	•	•	•	•	•	•	3	PSHX	•	•	•	•	•	4	•
BIT	•	2	3	4	4	•	•	PUL	4	•	•	•	•	•	•
BLE	•	•	•	•	•	•	3	PULX	•	•	•	•	•	5	•
BLS	•	•	•	•	•	•	3	ROL	2	•	•	6	6	•	•
BLT	•	•	•	•	•	•	3	ROR	2	•	•	6	6	•	•
BMI	•	•	•	•	•	•	3	RTI	•	•	•	•	•	10	•
BNE	•	•	•	•	•	•	3	RTS	•	•	•	•	•	5	•
BPL	•	•	•	•	•	•	3	SBA	•	•	•	•	•	2	•
BRA	•	•	•	•	•	•	3	SBC	•	2	3	4	4	•	•
BRN	•	•	•	•	•	•	3	SEC	•	•	•	•	•	2	•
BSR	•	•	•	•	•	•	6	SEI	•	•	•	•	•	2	•
BVC	•	•	•	•	•	•	3	SEV	•	•	•	•	•	2	•
BVS	•	•	•	•	•	•	3	STA	•	•	3	4	4	•	•
CBA	•	•	•	•	•	2	•	STD	•	•	4	5	5	•	•
CLC	•	•	•	•	•	2	•	STS	•	•	4	5	5	•	•
CLI	•	•	•	•	•	2	•	STX	•	•	4	5	5	•	•
CLR	2	•	•	6	6	•	•	SUB	•	2	3	4	4	•	•
CLV	•	•	•	•	•	2	•	SUBD	•	4	5	6	6	•	•
CMP	•	2	3	4	4	•	•	SWI	•	•	•	•	•	12	•
COM	2	•	•	6	6	•	•	TAB	•	•	•	•	•	2	•
CPX	•	4	5	6	6	•	•	TAP	•	•	•	•	•	2	•
DAA	•	•	•	•	•	2	•	TBA	•	•	•	•	•	2	•
DEC	2	•	•	6	6	•	•	TPA	•	•	•	•	•	2	•
DES	•	•	•	•	•	3	•	TST	2	•	•	6	6	•	•
DEX	•	•	•	•	•	3	•	TSX	•	•	•	•	•	3	•
EOR	•	2	3	4	4	•	•	TXS	•	•	•	•	•	3	•
INC	2	•	•	6	6	•	•	WAI	•	•	•	•	•	9	•
INS	•	•	•	•	•	3	•								

- **Summary of Cycle by Cycle Operation**

Table 12 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write line (R/$\overline{W}$) during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. (In general, instructions with the same addressing mode and number of cycles execute in the same manner; exceptions are indicated in the table).

Table 12 Cycle by Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Operand Data
LDS LDX LDD	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 2 Address Bus FFFF	1 1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	Op Code Address of Operand Operand Data
STA	3	1 2 3	Op Code Address Op Code Address + 1 Destination Address	1 1 0	Op Code Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Op Code Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Subroutine Address Stack Pointer Stack Pointer + 1	1 1 1 0 0	Op Code Irrelevant Data First Subroutine Op Code Return Address (Low Order Byte) Return Address (High Order Byte)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

*In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB = FFFF, DB = Low Byte of Reset Vector. (Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

*In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB=FFFF, DB=Low Byte of Reset Vector.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMPLIED					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address + 1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI**	9	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Op Code Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
WAI**		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)

**While the MCU is in the "Wait" state, its bus state will appear as a series of MCU reads of an address which is seven locations less than the original contents of the Stack Pointer. Contrary to the HD6800, none of the ports are driven to the high impedance state by a WAI instruction.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RELATIVE					
BCC BHT BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMT BVS BRN	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Branch Offset Low Byte of Restart Vector
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Address Bus FFFF Subroutine Starting Address Stack Pointer Stack Pointer – 1	1 1 1 1 0 0	Op Code Branch Offset Low Byte of Restart Vector Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

- **Summary of Undefined Instruction Operations**

The HD6801S has 36 undefined instructions. When these are carried out, the contents of Register and Memory in MCU change at random.

When the op codes (4E, 5E) are used to execute, the MCU continues to increase the program counter and it will not stop until the Reset signal enters. These op codes are used to test the LSI.

Table 13 Op codes Map

HD6801S MICROCOMPUTER INSTRUCTIONS

OP CODE HI / LO		0000	0001	0010	0011	ACC A 0100	ACC B 0101	IND 0110	EXT 0111	ACCA or SP 1000 IMM	ACCA or SP 1001 DIR	ACCA or SP 1010 IND	ACCA or SP 1011 EXT	ACCB or X 1100 IMM	ACCB or X 1101 DIR	ACCB or X 1110 IND	ACCB or X 1111 EXT	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								1
0010	2			BHI	PULA (+1)					SBC								2
0011	3			BLS	PULB (+1)	COM				*	SUBD (+2)			*	ADDD (+2)			3
0100	4	LSRD (+1)		BCC	DES	LSR				AND								4
0101	5	ASLD (+1)		BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX (+1)		BVC	PULX (+2)	ASL				EOR								8
1001	9	DEX (+1)	DAA	BVS	RTS (+2)	ROL				ADC								9
1010	A	CLV		BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI (+7)					ADD								B
1100	C	CLC		BGE	PSHX (+1)	INC				*	CPX (+2)			*	LDD (+1)			C
1101	D	SEC		BLT	MUL (+7)	TST				BSR (+4)	JSR (+2)			* (+1)	STD (+1)			D
1110	E	CLI		BGT	WAI (+6)	**		JMP (–3)		*	LDS (+1)			*	LDX (+1)			E
1111	F	SEI		BLE	SWI (+9)	CLR				* (+1)	STS (+1)			* (+1)	STX (+1)			F
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4	

[NOTES] 1) Undefined Op codes are marked with ▨.

2) ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

3) The instructions shown below are all 3 bytes and are marked with "*".
Immediate addressing mode of SUBD, CPX, LDS, ADDD, LDD and LDX instructions, and undefined op codes (8F, CD, CF).

4) The Op codes (4E, 5E) are 1 byte/∞ cycles instructions, and are marked with "**"

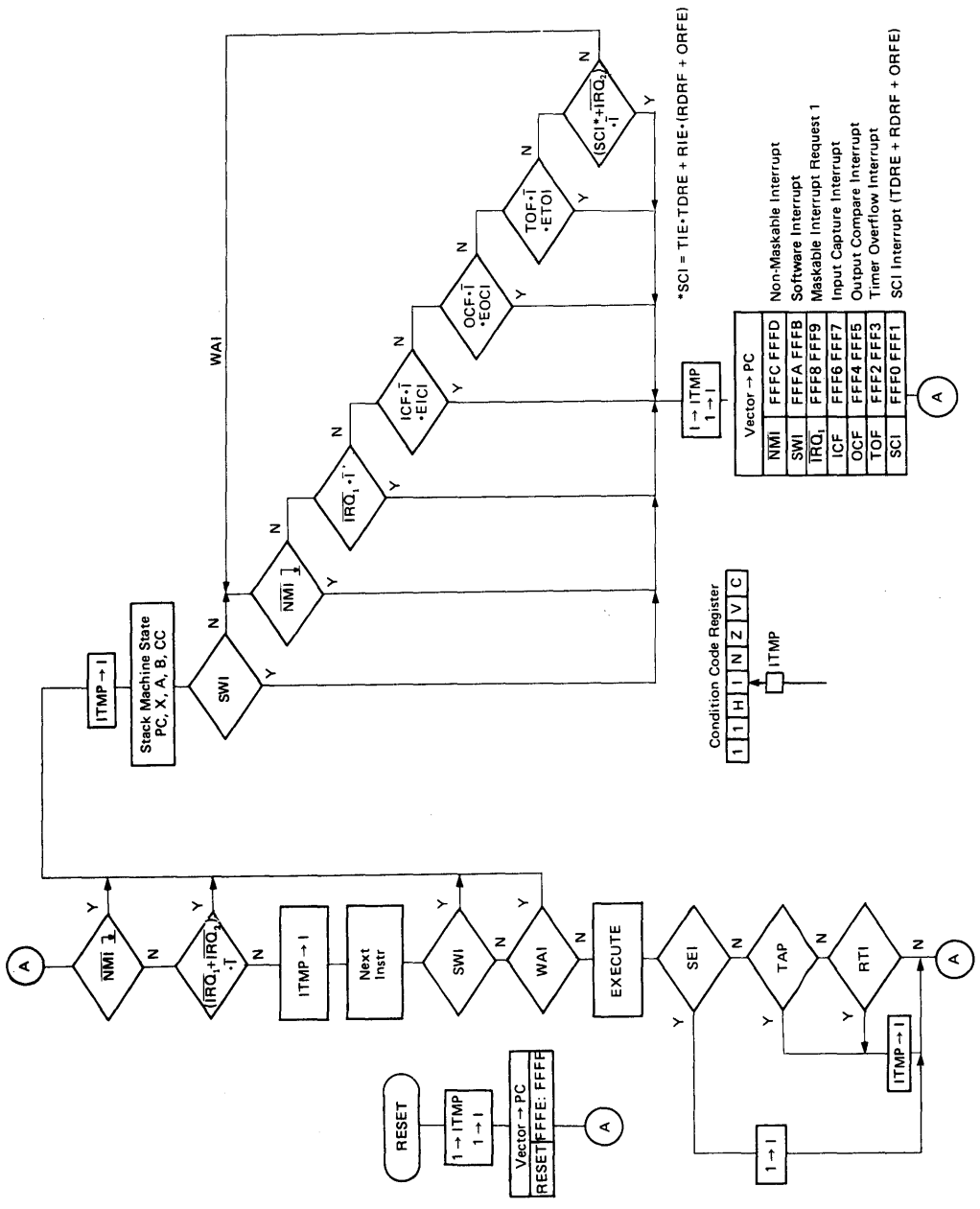


Figure 24 Interrupt Flowchart

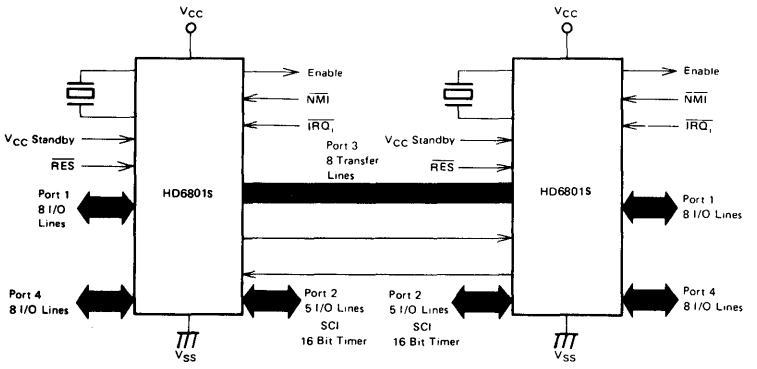


Figure 25 HD6801S MCU Single-Chip Dual Processor Configuration

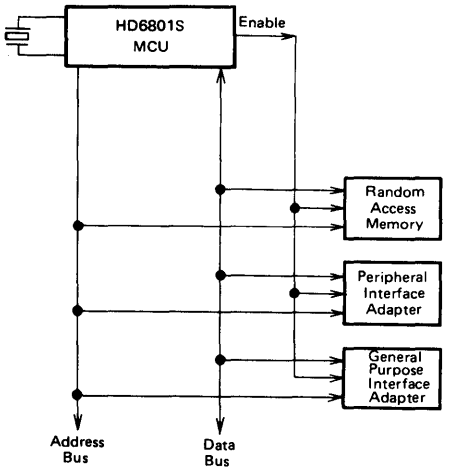


Figure 26 HD6801S MCU Expanded Non-Multiplexed Mode

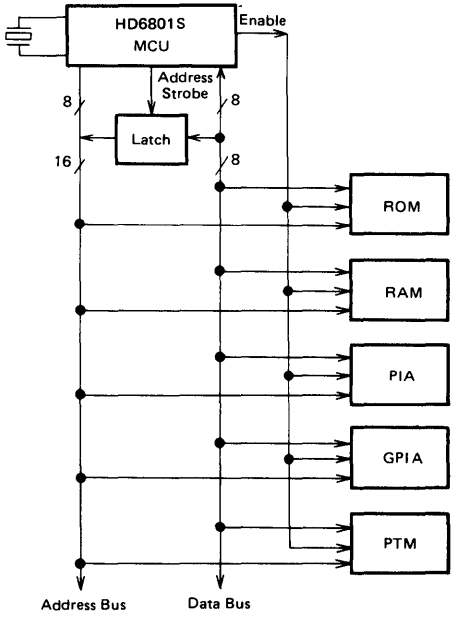


Figure 27 HD6801S MCU Expanded Multiplexed Mode

■ **Caution for the HD6801 Family SCI, TIMER Status Flag**

The flags shown in Table 14 are cleared by reading/writing (flag reset condition 2) the data register corresponding to each flag after reading the status register (flag reset condition 1).

To clear the flag correctly, take the following procedure:

1. Read the status register
2. Test the flag
3. Read the data register

Table 14 Status Flag Reset Conditions

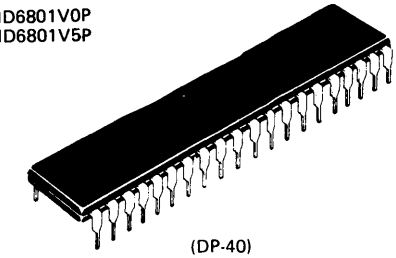
	Status Flag	Flag Reset Condition 1 (Status Register)	Flag Reset Condition 2 (Data Register)
TIMER	ICF	When each flag is "1", TRCSR/Read	ICR/Read
	OCF		OCR/Write
	TOF		TC/Read
SCI	RDRF	When each flag is "1", TRCSR/Read	RDR/Read
	ORFE		
	TDRE		TDR/Write

HD6801V0, HD6801V5

MCU (Microcomputer Unit)

The HD6801V MCU is an 8-bit microcomputer unit which is compatible with the HD6801S except the ROM size. The HD6801V MCU is object code compatible with the HD6800 with improved execution times of key instructions plus several new 16-bit and 8-bit instructions including an 8x8 unsigned multiply with 16-bit result. The HD6801V MCU can operate as a single chip microcomputer or be expanded to 65k bytes. The HD6801V MCU is TTL compatible and requires one +5.0 volt power supply. The HD6801V MCU has 4k bytes of ROM and 128 bytes of RAM on chip. Serial Communications Interface (SCI), and parallel I/O as well as a three function 16-bit timer. Features and Block diagram of the HD6801V include the following:

HD6801V0P
HD6801V5P

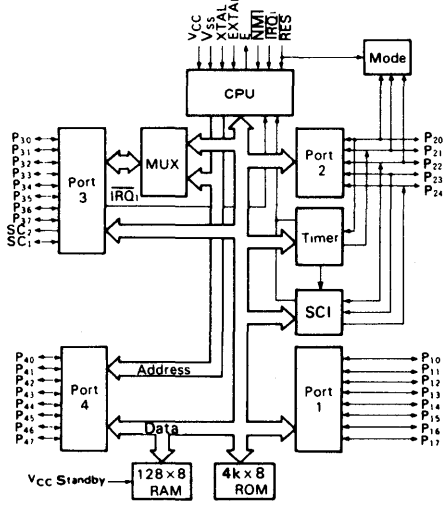


(DP-40)

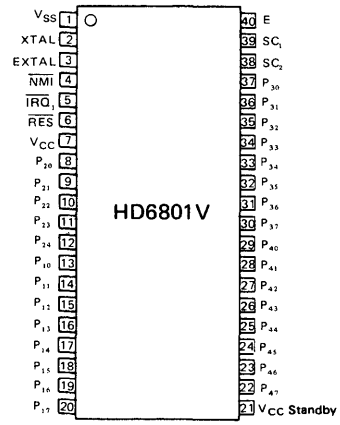
■ FEATURES

- Expanded HMCS6800 Instruction Set
- 8 x 8 Multiply
- On-Chip Serial Communication Interface (SCI)
- Object Code Compatible With The HD6800 MPU
- 16-Bit Timer
- Single Chip Or Expandable To 65k Bytes
- 4k Bytes Of ROM
- 128 Bytes Of RAM (64 Bytes Retainable On Power Down)
- 29 Parallel I/O Lines And 2 Handshake Control Lines
- Internal Clock/Divided-By-Four Circuitry
- TTL Compatible Inputs And Outputs
- Interrupt Capability
- Compatible with MC6801 (except ROM size)

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ TYPE OF PRODUCTS

MCU	Bus Timing
HD6801V0	1 MHz
HD6801V5	1.25 MHz

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} =5.0V±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	Other Inputs*			2.0	–	V_{CC}	
Input "Low" Voltage	EXTAL	V_{IL}		-0.3	–	0.6	V
	Other Inputs*			-0.3	–	0.8	
Input Load Current	P_{40} ~ P_{47}	$\|I_{in}\|$	V_{in} = 0 ~ 2.4V	–	–	0.5	mA
	SC_1			–	–	0.8	
	EXTAL		V_{in} = 0 ~ V_{CC}	–	–	1.2	
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$	$\|I_{in}\|$	V_{in} = 0 ~ 5.25V	–	–	2.5	μA
Three State (Offset) Leakage Current	P_{10} ~ P_{17}, P_{30} ~ P_{37}	$\|I_{TSI}\|$	V_{in} = 0.5 ~ 2.4V	–	–	10	μA
	P_{20} ~ P_{24}			–	–	100	
Output "High" Voltage	P_{30} ~ P_{37}	V_{OH}	I_{LOAD} = -205 μA	2.4	–	–	V
	P_{40} ~ P_{47}, E, SC_1, SC_2		I_{LOAD} = -145 μA	2.4	–	–	
	Other Outputs		I_{LOAD} = -100 μA	2.4	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{LOAD} = 1.6 mA	–	–	0.5	V
Darlington Drive Current	P_{10} ~ P_{17}	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Power Dissipation		P_D		–	–	1200	mW
Input Capacitance	P_{30} ~ P_{37}, P_{40} ~ P_{47}, SC_1	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz	–	–	12.5	pF
	Other Inputs			–	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}		4.0	–	5.25	V
	Operating	V_{SB}		4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	V_{SBB} = 4.0 V	–	–	8.0	mA

*Except Mode Programming Levels.

• AC CHARACTERISTICS

BUS TIMING (V_{CC} = 5.0V±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	HD6801V0			HD6801V5			Unit
				min	typ	max	min	typ	max	
Cycle Time		t_{cyc}		1	–	10	0.8	–	10	μs
Address Strobe Pulse Width "High" *		PW_{ASH}		200	–	–	150	–	–	ns
Address Strobe Rise Time		t_{ASr}		5	–	50	5	–	50	ns
Address Strobe Fall Time		t_{ASf}		5	–	50	5	–	50	ns
Address Strobe Delay Time *		t_{ASD}		60	–	–	30	–	–	ns
Enable Rise Time		t_{Er}		5	–	50	5	–	50	ns
Enable Fall Time		t_{Ef}		5	–	50	5	–	50	ns
Enable Pulse Width "High" Time *		PW_{EH}		450	–	–	340	–	–	ns
Enable Pulse Width "Low" Time *		PW_{EL}		450	–	–	350	–	–	ns
Address Strobe to Enable Delay Time *		t_{ASED}		60	–	–	30	–	–	ns
Address Delay Time		t_{AD}	Fig. 1	–	–	260	–	–	260	ns
Address Delay Time for Latch *		t_{ADL}	Fig. 2	–	–	270	–	–	260	ns
Data Set-up Write Time		t_{DSW}		225	–	–	115	–	–	ns
Data Set-up Read Time		t_{DSR}		80	–	–	80	–	–	ns
Data Hold Time	Read	t_{HR}		10	–	–	10	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	
Address Set-up Time for Latch *		t_{ASL}		60	–	–	50	–	–	ns
Address Hold Time for Latch		t_{AHL}		20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	ns
Peripheral Read Access Time	Non-Multiplexed Bus*	(t_{ACCN})		–	–	(610)	–	–	(410)	ns
	Multiplexed Bus*	(t_{ACCM})		–	–	(600)	–	–	(410)	
Oscillator stabilization Time		t_{RC}	Fig. 10	100	–	–	100	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 11	200	–	–	200	–	–	ns

*These timings change in approximate proportion to tcyc. The figures in this characteristics represent those when tcyc is minimum (=in the highest speed operation).

PERIPHERAL PORT TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Peripheral Data Setup Time	Port 1, 2, 3, 4	t_{PDSU}	Fig. 3	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		t_{OSD1}	Fig. 5	–	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		t_{OSD2}	Fig. 5	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4	t_{PWD}	Fig. 4	–	–	400	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	Port 2*, 4	t_{CMOS}	Fig. 4	–	–	2.0	μs
Input Strobe Pulse Width		t_{PWIS}	Fig. 6	200	–	–	ns
Input Data Hold Time	port 3	t_{IH}	Fig. 6	50	–	–	ns
Input Data Set-up Time	Port 3	t_{IS}	Fig. 6	20	–	–	ns

*Except P_{21} **10kΩ pull up register required for Port 2

TIMER, SCI TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Timer Input Pulse Width	t_{PWT}		$2t_{cyc}$+200	–	–	ns
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 7	–	–	600	ns
SCI Input Clock Cycle	t_{Scyc}		1	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Mode Programming Input "Low" Voltage		V_{MPL}	Fig. 8	–	–	1.7	V
Mode Programming Input "High" Voltage		V_{MPH}		4.0	–	–	V
$\overline{RES}$ "Low" Pulse Width		PW_{RSTL}		3.0	–	–	t_{cyc}
Mode Programming Set-up Time		t_{MPS}		2.0	–	–	t_{cyc}
Mode Programming Hold Time	$\overline{RES}$ Rise Time $\geqq 1\mu s$	t_{MPH}		0	–	–	ns
	$\overline{RES}$ Rise Time $< 1\mu s$			100	–	–	

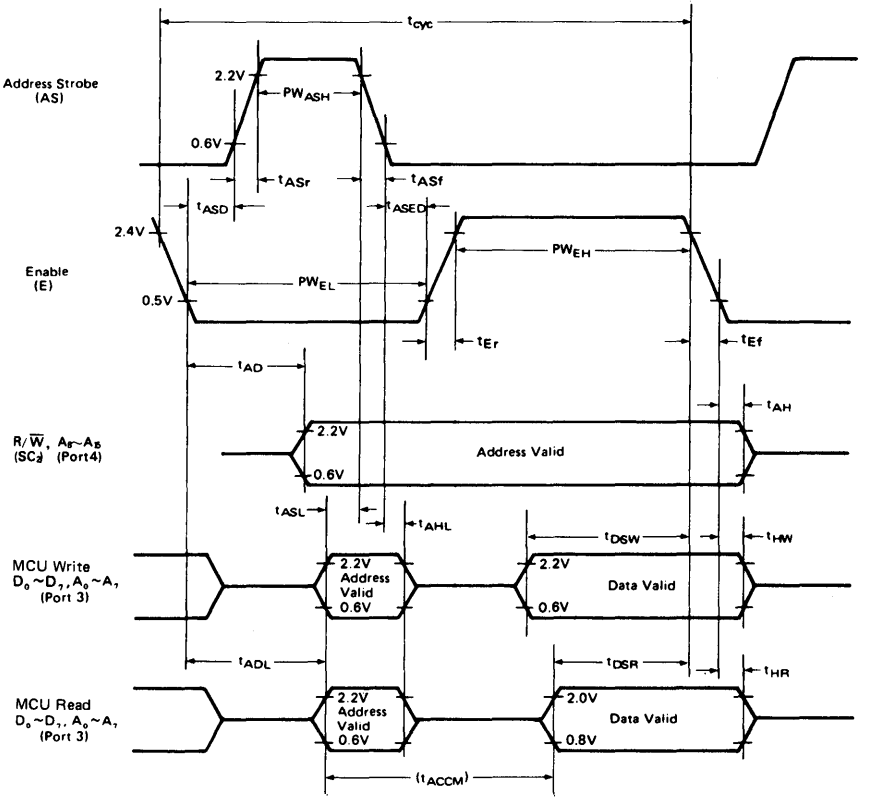


Figure 1 Expanded Multiplexed Bus Timing

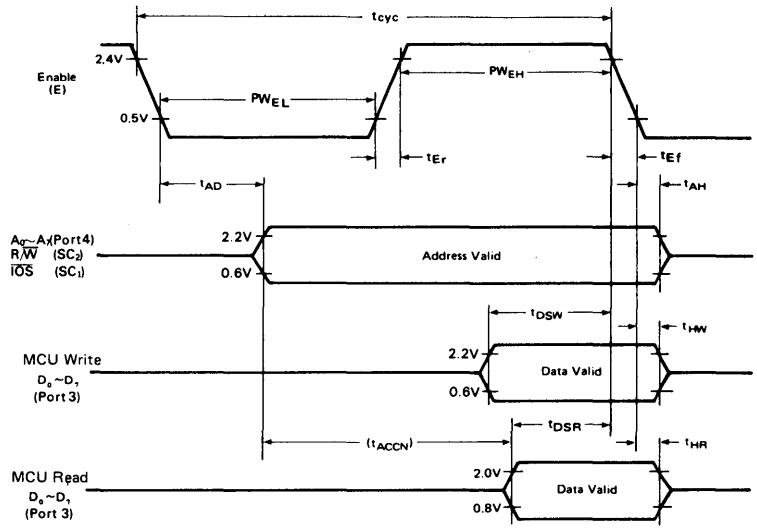
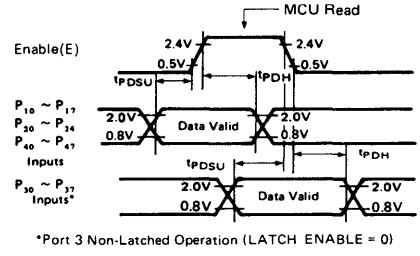
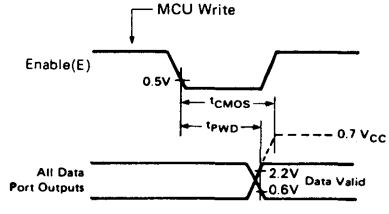


Figure 2 Expanded Non-Multiplexed Bus Timing



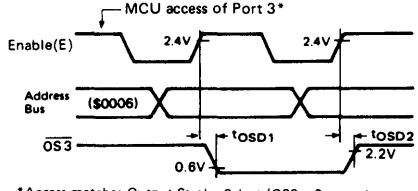
*Port 3 Non-Latched Operation (LATCH ENABLE = 0)

Figure 3 Data Set-up and Hold Times (MCU Read)



(Note)
1. 10 kΩ Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P_{21}
3. Port 4 cannot be pulled above V_{CC}

Figure 4 Port Data Delay Timing (MCU Write)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

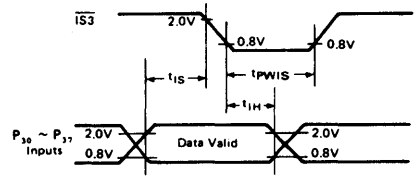


Figure 6 Port 3 Latch Timing (Single Chip Mode)

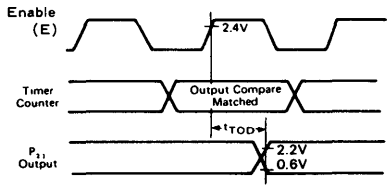


Figure 7 Timer Output Timing

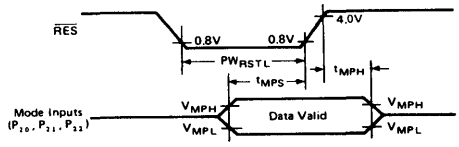
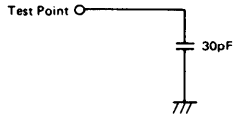
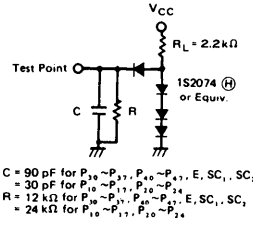


Figure 8 Mode Programming Timing



(a) CMOS Load



(b) TTL Load

Figure 9 Bus Timing Test Loads

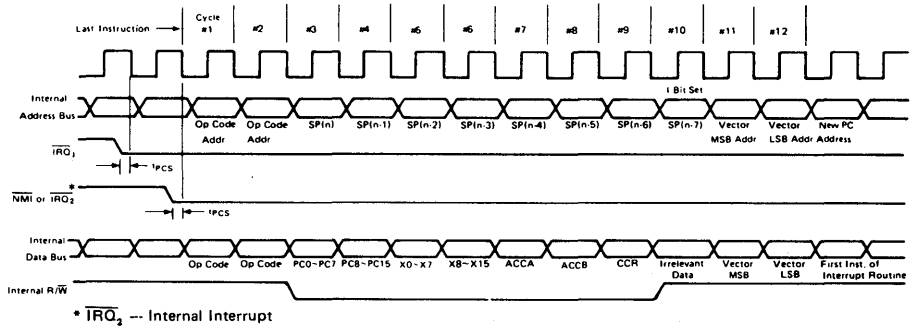


Figure 10 Interrupt Sequence

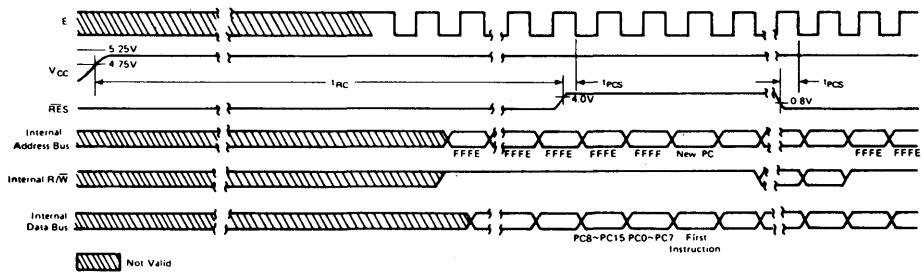


Figure 11 Reset Timing

■ SIGNAL DESCRIPTIONS

- **Vcc and Vss**

These two pins are used to supply power and ground to the chip. The voltage supplied will be +5 volts ±5%.

- **XTAL and EXTAL**

These connections are for a parallel resonant fundamental crystal, AT cut. Divide-by-4 circuitry is included with the internal clock, so a 4 MHz crystal may be used to run the system at 1 MHz. The divide-by-4 circuitry allows for use of the inexpensive 3.58 MHz Color TV crystal for non-time critical applications. Two 22pF capacitors are needed from the two crystal pins to ground to insure reliable operation. An example of the crystal interface is shown in Fig. 12. EXTAL may be driven by an external TTL compatible clock source with a 45% to 55% duty cycle. It will divide by 4 any frequency less than or equal to 5 MHz. XTAL must be grounded if an external clock is used.

Nominal Crystal Parameter

Item \ Crystal	4 MHz	5 MHz
Co	7 pF max.	4.7 pF max.
R_S	60Ω max.	30Ω typ.

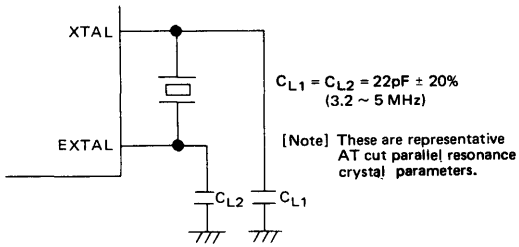


Figure 12 Crystal Interface

- **Vcc Standby**

This pin will supply +5 volts ±5% to the standby RAM on the chip. The first 64 bytes of RAM will be maintained in the power down mode with 8 mA current max. The circuit of figure 13 can be utilized to assure that V_{CC} Standby does not go below V_{SBB} during power down.

To retain information in the RAM during power down the following procedure is necessary:

1) Write "0" into the RAM enable bit, RAME. RAME is bit 6 of the RAM Control Register at location $0014. This disables the standby RAM, thereby protecting it at power down.
2) Keep V_{CC} Standby greater than V_{SBB}.

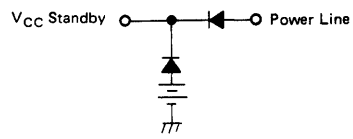


Figure 13 Battery Backup for V_{CC} Standby

- **Reset ($\overline{RES}$)**

This input is used to reset and start the CPU from a power down condition, resulting from a power failure or an initial startup of the processor. On power up, the reset must be held "Low" for at least 100 ms. When reset during operation, $\overline{RES}$ must be held "Low" at least 3 clock cycles.

When a "High" level is detected, the MCU does the following:

1) All the higher order address lines will be forced "High".
2) I/O Port 2 bits 2, 1, and 0 are latched into programmed control bits PC2, PC1 and PC0.
3) The last two ($FFFE, $FFFF) locations in memory will be used to load the program addressed by the program counter.
4) The interrupt mask bit is set. Clear before the CPU can recognize maskable interrupts.

- **Enable (E)**

This supplies the external clock for the rest of the system when the internal oscillator is used. It is a single phase, TTL compatible clock, and will be the divide-by-4 result of the crystal oscillator frequency. It will drive one TTL load and 90 pF capacitance.

- **Non-Maskable Interrupt ($\overline{NMI}$)**

A low-going edge on this input requests that a non-maskable-interrupt sequence be generated within the processor. As with interrupt Request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the Condition Code Register has no effect on $\overline{NMI}$.

In response to an $\overline{NMI}$ interrupt, the Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. At the end of the sequence, a 16-bit address will be loaded that points to a vectoring address located in memory locations $FFFC and $FFFD. An address loaded at these locations causes the CPU to branch to a non-maskable interrupt service routine in memory.

A 3.3 kΩ external resistor to VCC should be used for wire-OR and optimum control of interrupts.

Inputs $\overline{IRQ_1}$ and $\overline{NMI}$ are hardware interrupt lines that are sampled during E and will start the interrupt routine on the $\overline{E}$ following the completion of an instruction.

- **Interrupt Request ($\overline{IRQ_1}$)**

This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will complete the current instruction before it recognizes the request. At that time, if the interrupt mask bit in the Condition Code Register is not set, the machine will begin an interrupt sequence. The Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. Next the CPU will respond to the interrupt request by setting the interrupt mask bit "High" so that no further maskable interrupts may occur. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations $FFF8 and $FFF9. An address loaded at these locations causes the CPU to branch to an interrupt routine in memory.

The $\overline{IRQ_1}$ requires a 3.3 kΩ external resistor to V_{CC} which should be used for wire-OR and optimum control of interrupts. Internal Interrupts will use an internal interrupt line ($\overline{IRQ_2}$). This interrupt will operate the same as $\overline{IRQ_1}$ except that it will use the vector address of $FFF0 through $FFF7. $\overline{IRQ_1}$ will have priority over $\overline{IRQ_2}$ if both occur at the same time. The Interrupt Mask Bit in the condition code register masks both interrupts (See Table 1).

Table 1 Interrupt Vector Location

	Vector MSB	Vector LSB	Interrupt
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$ (or $\overline{IS3}$)
	FFF6	FFF7	ICF (Input Capture)
	FFF4	FFF5	OCF (Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SC_1 (RDRF + ORFE + TDRE)

The following pins are available in the Single Chip Mode, and are associated with Port 3 only.

- **Input Strobe ($\overline{IS3}$) (SC_1)**

The function of the $\overline{IS3}$ signal depends on the I/O Port 3 Control/Status Register. If $\overline{IS3}$ Enable bit is set, an interrupt will occur by the fall of the $\overline{IS3}$ signal. If the latch enable bit is set, the data in the I/O Port 3 will be latched at the I/O Port 3 Data Register. The timing condition of the $\overline{IS3}$ signal that is necessary to be latched the input data normally is shown in Figure 6.

- **Output Strobe ($\overline{OS3}$) (SC_2)**

This signal is used by the processor to strobe an external device, indicating valid data is on the I/O pins. The timing for the Output Strobe is shown in Figure 5. I/O Port 3 Control/Status Register is discussed in the following section.

The following pins are available in the Expanded Modes.

- **Read/Write (R/$\overline{W}$) (SC_2)**

This TTL compatible output signals the peripherals and memory devices whether the CPU is in a Read ("High") or a Write ("Low") state. The normal standby state of this signal is Read ("High"). This output can drive one TTL load and 90 pF capacitance.

- **I/O Strobe ($\overline{IOS}$) (SC_1)**

In the expanded non-multiplexed mode of operation, $\overline{IOS}$ internally decodes A_9 through A_{15} as zero's and A_8 as a one. This allows external access of the 256 locations from $0100 to $01FF. The timing diagrams are shown as figure 2.

- **Address Strobe (AS) (SC_1)**

In the expanded multiplexed mode of operation, address strobe is output on this pin. This signal is used to latch the 8 LSB's of address which are multiplexed with data on Port 3. An 8-bit latch is utilized in conjunction with Address Strobe, as shown in figure 19. So I/O port 3 can become data bus during the E pulse. The timing for this signal is shown in Figure 1 of Bus Timing. This signal is also used to disable the address from the multiplexed bus allowing a deselect time, t_{ASD} before the data is enabled to the bus.

- **PORTS**

There are four I/O ports on the HD6801V MCU; three 8-bit ports and one 5-bit port. There are two control lines associated with one of the 8-bit ports. Each port has an associated write only Data Direction Register which allows each I/O line to be programmed to act as an input or an output*. A "1" in the corresponding Data Direction Register bit will cause that I/O line to be an output. A "0" in the corresponding Data Direction Register bit will cause that I/O line to be an input. There are four ports: Port 1, Port 2, Port 3, and Port 4. Their addresses and the addresses of their Data Direction registers are given in Table 2.

* The only exception is bit 1 of Port 2, which can either be data input or Timer output.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

- **I/O Port 1**

This is an 8-bit port whose individual bits may be defined as inputs or outputs by the corresponding bit in its data direction register. The 8 output buffers have three-state capability, allowing them to enter a high impedance state when the peripheral data lines are used as inputs. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs these lines are TTL compatible and may also be used as a source of up to 1 mA at 1.5 V to directly drive a Darlington base. After Reset, the I/O lines are configured as inputs. In all three modes, Port 1 is always parallel I/O.

- **I/O Port 2**

This port has five lines that may be defined as inputs or outputs by its data direction register. The 5 output buffers have three-state capability, allowing them to enter a high impedance state when used as an input. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, this port has no internal pullup resistors but will drive TTL inputs directly. For driving CMOS inputs, external pullup resistors are required. After Reset, the I/O lines are configured as inputs. Three pins on Port 2 (pins 10, 9, and 8 of the chip) are used to program the mode of operation during reset. The values of these pins at reset are latched into the three MSB's (bits 7, 6, and 5) of Port 2 which are read only. This is explained in the Mode Selection Section.

In all three modes, Port 2 can be configured as I/O and provides access to the Serial Communications Interface and the Timer. Bit 1 is the only pin restricted to data input or Timer output.

- **I/O Port 3**

This is an 8-bit port that can be configured as I/O, a data bus, or an address bus multiplexed with the data bus – depending on the mode of operation hardware programmed by the user at reset. As a data bus, Port 3 is bi-directional. As an input for peripherals, it must be supplied regular TTL levels, that is, greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0"

Its TTL compatible three-state output buffers can drive one TTL load and 90 pF capacitance. In the Expanded Modes, after reset, the data direction register is inhibited and data flow depends on the state of the R/$\overline{W}$ line. The input strobe ($\overline{IS3}$) and the output strobe ($\overline{OS3}$) used for handshaking are explained later.

In the three modes, Port 3 assumes the following characteristics:

Single Chip Mode: Parallel Inputs/Outputs as programmed by its associated Data Direction Register. There are two control lines associated with this port in this mode, an input strobe and an output strobe, that can be used for handshaking. They are controlled by the I/O Port 3 Control/Status Register explained at the end of this section. Three options of Port 3 operations are sumarized as follows: (1) Port 3 input data can be latched using $\overline{IS3}$ (SC_1) as a control signal, (2) $\overline{OS3}$ can be generated by either an CPU read or write to Port 3's Data Register, and (3) and $\overline{IRQ_1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch and strobe timing is shown in Fig. 5 and Fig. 6.

Expanded Non-Multiplexed Mode: In this mode, Port 3 becomes the data bus (D_0~D_7).

Expanded Multiplexed Mode: In this mode, Port 3 becomes both the data bus (D_0~D_7) and lower bits of the address bus (A_0~A_7). An address strobe output is true when the address is on the port.

I/O PORT 3 CONTROL/STATUS REGISTER

	7	6	5	4	3	2	1	0
$000F	$\overline{IS3}$ FLAG	$\overline{IS3}$ $\overline{IRQ_1}$ ENABLE	X	OSS	LATCH ENABLE	X	X	X

Bit 0; Not used.
Bit 1; Not used.
Bit 2; Not used.
Bit 3; LATCH ENABLE. This controls the input latch for I/O Port 3. If this bit is set "High" the input data will be latched with the falling edge of the Input Strobe, $\overline{IS3}$. This bit is cleared by reset, and the latch is "re-opened" with CPU read Port 3.
Bit 4; OSS. (Output Strobe Select) This bit will select if the Output Strobe should be generated at $\overline{OS3}$ (SC_2) by a write to I/O Port 3 or a read of I/O Port 3. When this bit is cleared the strobe is generated by a read Port 3. When this bit is set the strobe is generated by a write to Port 3.
Bit 5; Not used.
Bit 6; $\overline{IS3}$ $\overline{IRQ_1}$ ENABLE. When set, interrupt will be enabled whenever $\overline{IS3}$ FLAG is set; when clear, interrupt is inhibited. This bit is cleared by reset.
Bit 7; $\overline{IS3}$ FLAG. This is a read-only status bit that is set by the falling edge of the input strobe, $\overline{IS3}$ (SC_1). It is cleared by a read of the Control/Status Register followed by a read or write of I/O Port 3. Reset will clear this bit.

- **I/O Port 4**

This is an 8-bit port that can be configured as I/O or as address output lines depending on the mode of operation. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0".

As outputs, each line is TTL compatible and can drive 1 TTL load and 90 pF capacitance. After reset, the lines are configured as inputs. To use the pins as addresses, therefore, they should be programmed as outputs. In the three modes, Port 4 assumes the following characteristics:

Single Chip Mode: Parallel Inputs/Outputs as programmed by its associated Data Direction Register.

Expanded Non-Multiplexed Mode: Port 4 is configured as the lower order address lines (A_0 ~ A_7) by writing "1"s to the data direction register. When all eight address lines are not needed, the remaining lines, starting with the most significant bit, may be used as I/O (inputs only).

Expanded Multiplexed Mode: Port 4 is configured as the higher order address lines (A_8 ~ A_{15}) by writing "1"s to the data direction register. When all eight address lines are not needed, the remaining lines, starting with the most significant bit, may be used as I/O (inputs only).

■ OPERATION MODES

The operation modes that HD6801V will operate after Reset is determined by hardware that the user must wire on pins 10, 9, and 8 of the chip. These pins are the three LSB's (I/O 2, I/O 1, and I/O 0 respectively) of Port 2. They are latched into programmed control bits PC2, PC1, and PC0 when reset goes high. I/O Port 2 Register is shown below.

PORT 2 DATA REGISTER

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

An example of external hardware that could be used for Mode Selection is shown in Fig 14. The HD14053B provides the isolation between the peripheral device and MCU during reset, which is necessary if data conflict can occur between peripheral device and Mode generation circuit.

As bits 5, 6 and 7 of Port 2 are read-only, the mode cannot be changed through software. The mode selections are shown in Table 3.

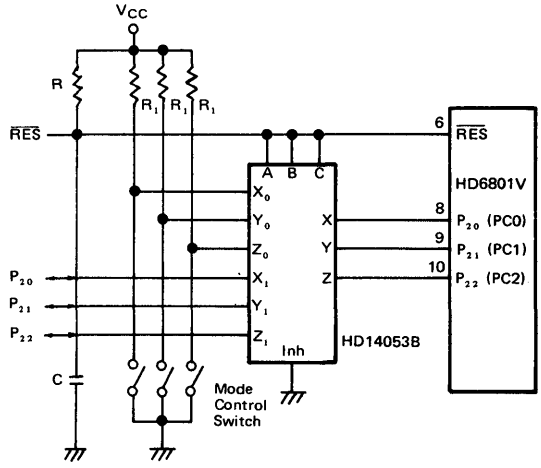
The HD6801V can operate three basic modes; (1) Single Chip Mode, (2) Expanded Multiplexed Mode (compatible with HMCS6800 peripheral family) (3) Expanded Non-Multiplexed Mode.

- **Single Chip Mode**

In the Single Chip Mode the Ports are configured as I/O. This is shown in Figure 16 the single Chip Mode. In this mode, Port 3 will have two associated control lines, an input strobe and an output strobe for handshaking data.

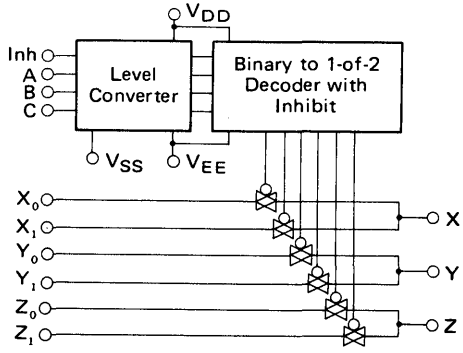
- **Expanded Non-Multiplexed Mode**

In this mode the HD6801V will directly address HMCS6800 peripherals with no external logic. In this mode Port 3 becomes the data bus. Port 4 becomes the A_0~A_7 address bus or partial address and I/O (inputs only). Port 2 can be parallel I/O, serial I/O, Timer, or any combination of them. Port 1 is parallel I/O only. In this mode the HD6801V is expandable to 256 locations. The eight address lines associated with Port 4 may be substituted for I/O (inputs only) if a fewer number of address lines will satisfy the application (See Figure 17).



[NOTES] 1) Mode 7 as shown
2) RC≈Reset time constant
3) R_1=10kΩ

Figure 14 Recommended Circuit for Mode Selection



Truth Table

Control Input				On Switch
Inhibit	Select			
	C	B	A	HD14053B
0	0	0	0	Z_0 Y_0 X_0
0	0	0	1	Z_0 Y_0 X_1
0	0	1	0	Z_0 Y_1 X_0
0	0	1	1	Z_0 Y_1 X_1
0	1	0	0	Z_1 Y_0 X_0
0	1	0	1	Z_1 Y_0 X_1
0	1	1	0	Z_1 Y_1 X_0
0	1	1	1	Z_1 Y_1 X_1
1	X	X	X	–

Figure 15 HD14053B Multiplexers/Demultiplexers

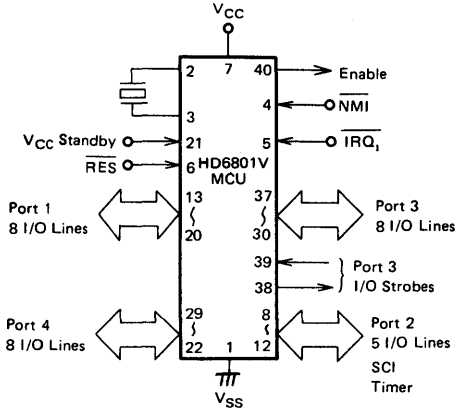


Figure 16 HD6801V MCU Single-Chip Mode

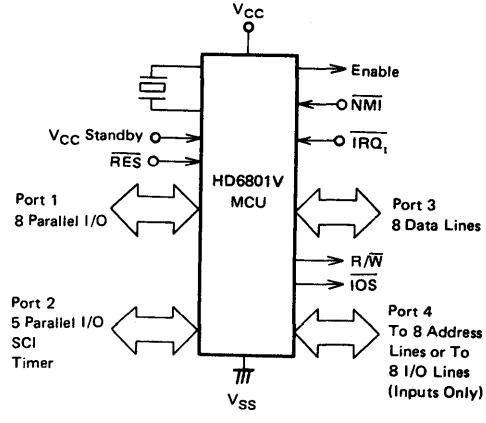


Figure 17 HD6801V MCU Expanded Non-Multiplexed Mode

- **Expanded Multiplexed Mode**

In this mode Port 4 becomes higher order address lines with an alternative of substituting some of the address lines for I/O (inputs only). Port 3 is the data bus multiplexed with the lower order address lines differentiated by an output called Address Strobe. Port 2 is 5 lines of Parallel I/O, SCI, Timer, or any combination of them. Port 1 is 8 Parallel I/O lines. In this mode it is expandable to 65k bytes. (See Figure 18).

- **Lower order Address Bus Latches**

Since the data bus is multiplexed with the lower order address bus in Port 3, latches are required to latch those address bits. The 74LS373 Transparent octal D-type latch can be used with the HD6801V to latch the least significant address byte. Figure 19 shows how to connect the latch to the HD6801V. The output control to the 74LS373 may be connected to ground.

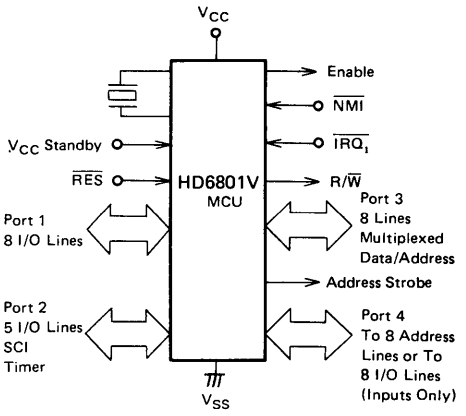
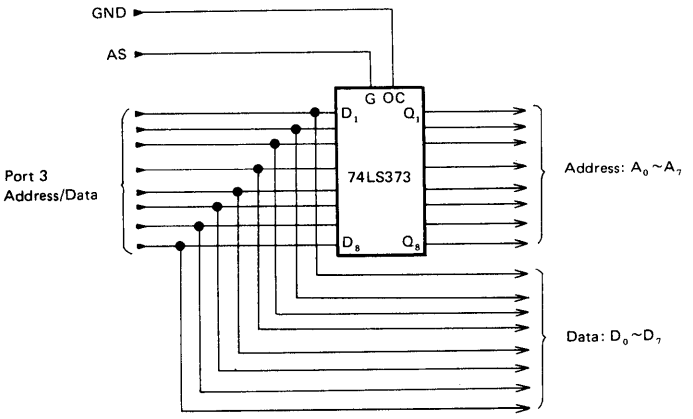


Figure 18 HD6801V MCU Expanded Multiplexed Mode



Function Table

Output Control	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 19 Latch Connection

- **Mode and Port Summary MCU Signal Description**

This section gives a description of the MCU signals for the various modes. SC_1 and SC_2 are signals which vary with the mode that the chip is in.

MODE	PORT 1 Eight Lines	PORT 2 Five Lines	PORT 3 Eight Lines	PORT 4 Eight Lines	SC_1	SC_2
SINGLE CHIP	I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)
EXPANDED MUX	I/O	I/O	ADDRESS BUS (A_0~A_7) DATA BUS (D_0~D_7)	ADDRESS BUS* (A_8~A_{15})	AS(O)	R/$\overline{W}$(O)
EXPANDED NON-MUX	I/O	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS* (A_0~A_7)	$\overline{IOS}$(O)	R/$\overline{W}$(O)

*These lines can be substituted for I/O (Input Only) starting with the most significant address line.

I = Input
O = Output
R/$\overline{W}$ = Read/Write
$\overline{IS3}$ = Input Strobe
$\overline{OS3}$ = Output Strobe
$\overline{IOS}$ = I/O Select
SC = Strobe Control
AS = Address Strobe

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 3 Mode Selection Summary

Mode	P_{22} ($PC2$)	P_{21} ($PC1$)	P_{20} ($PC0$)	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX	Multiplexed/No RAM & ROM
2	L	H	L	E	I	E	MUX	Multiplexed/RAM
1	L	L	H	I	I	E	MUX	Multiplexed/RAM & ROM
0	L	L	L	I	I	I(3)	MUX	Multiplexed Test

LEGEND:
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

[NOTES]
1) Internal RAM is addressed at $XX80
2) Internal ROM is disabled
3) $\overline{RES}$ vector is external for 2 cycles after $\overline{RES}$ goes "High"
4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
5) Addresses associated with Port 3 are considered external in Modes 5 and 6
6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register

■ MEMORY MAPS

The MCU can provide up to 65k bytes address space depending on the operating mode. A memory map for each operating mode is shown in Figure 20. The first 32 locations of each map are reserved for the MCU's internal register area, as shown in Table 4. With exceptions as indicated.

Table 4 Internal Register Area

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External address in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No. $\overline{IOS}$)
** External addresses in Modes 0, 1, 2, 3
*** 1=Output, 0=Input.

■ INTERRUPT FLOWCHART

The Interrupt flowchart is depicted in Figure 24 and is common to every interrupt excluding reset.

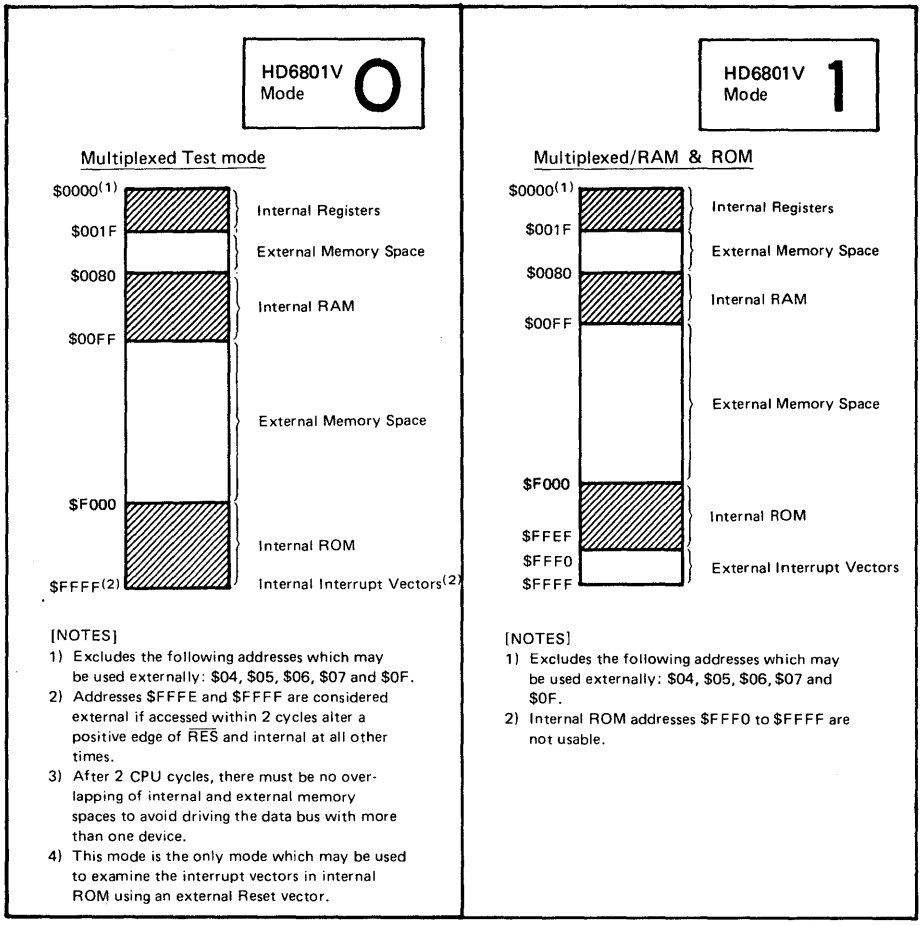


Figure 20 HD6801V Memory Maps

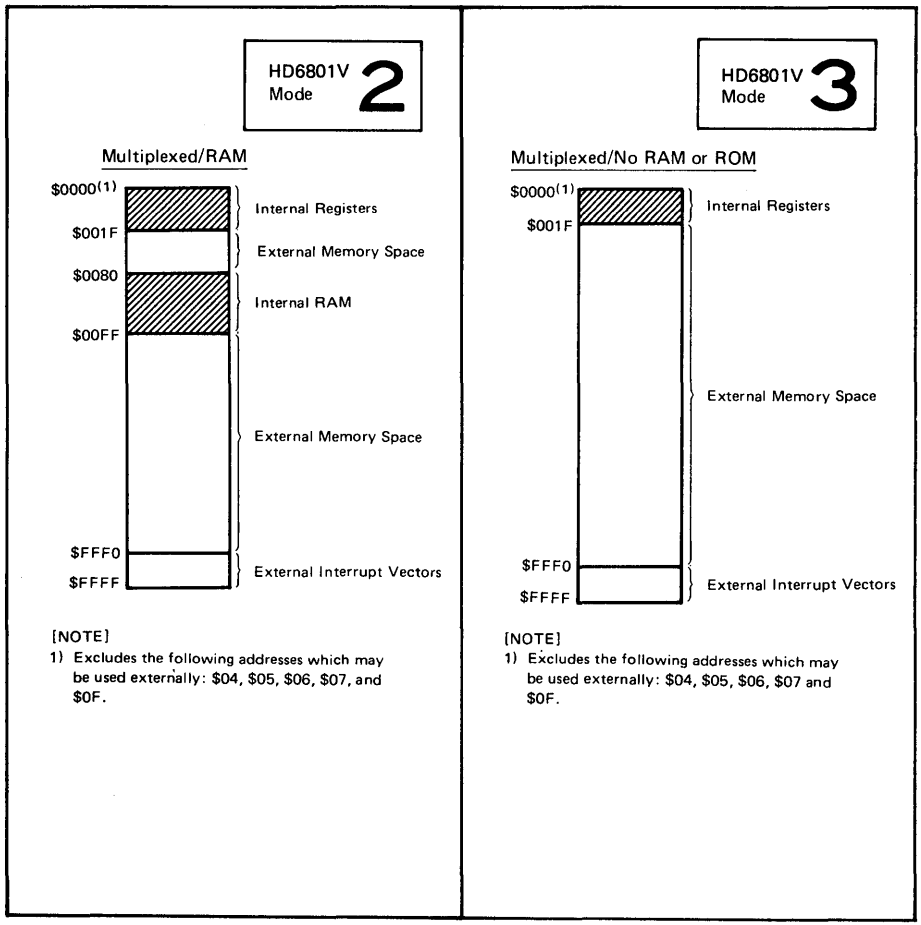


Figure 20 HD6801V Memory Maps (Continued)

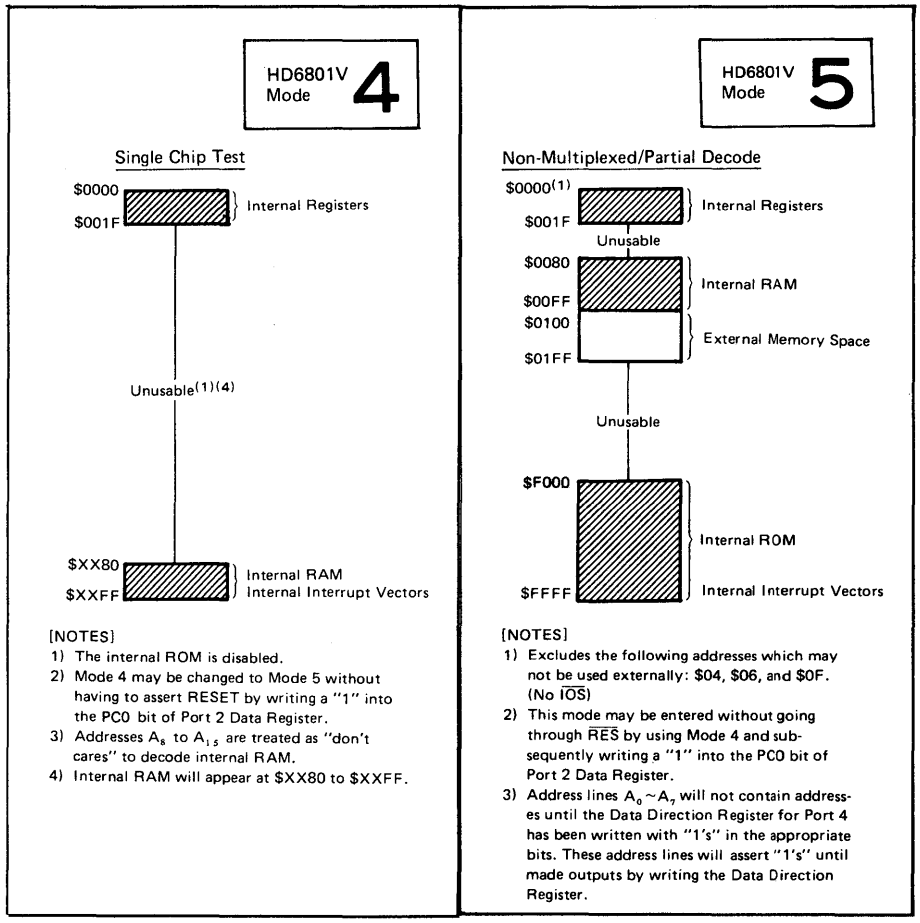
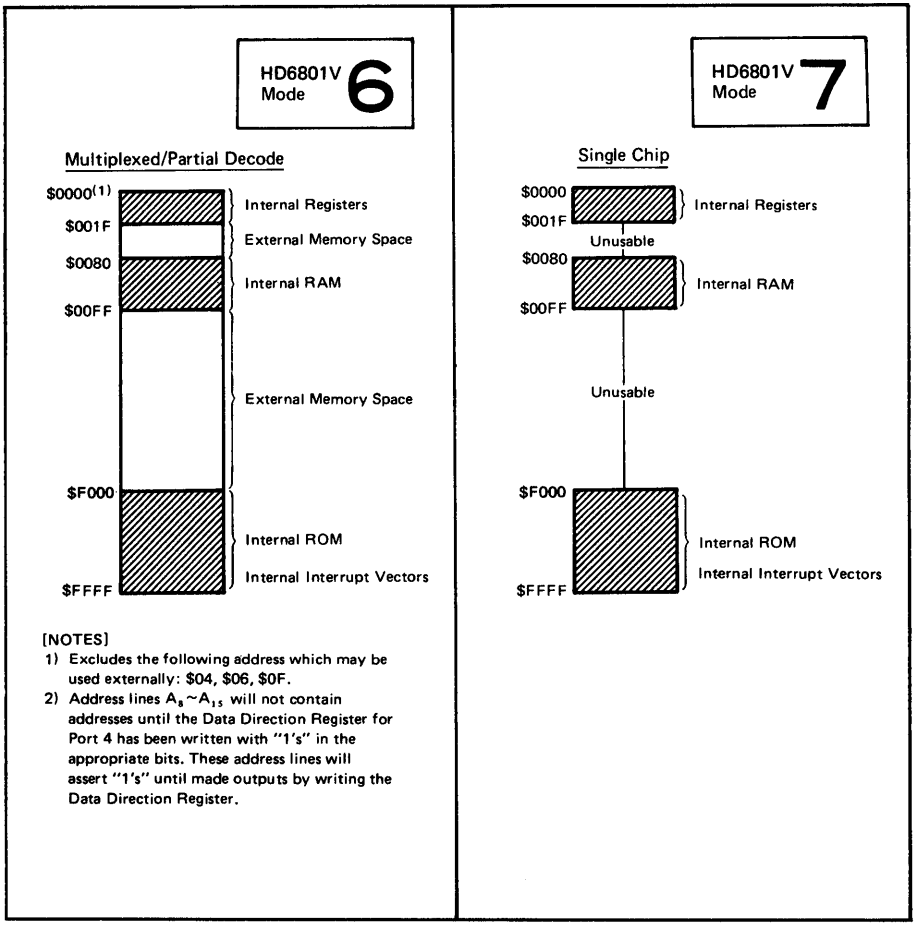


Figure 20 HD6801V Memory Maps (Continued)



[NOTES]
1) Excludes the following address which may be used externally: $04, $06, $0F.
2) Address lines $A_8 \sim A_{15}$ will not contain addresses until the Data Direction Register for Port 4 has been written with "1's" in the appropriate bits. These address lines will assert "1's" until made outputs by writing the Data Direction Register.

Figure 20 HD6801V Memory Maps (Continued)

■ PROGRAMMABLE TIMER

The HD6801V contains an on-chip 16-bit programmable timer which may be used to perform measurements on an input waveform while independently generating an output waveform. Pulse widths for both input and output signals may vary from a few microseconds to many seconds. The timer hardware consists of

- an 8-bit control and status register,
- a 16-bit free running counter,
- a 16-bit output compare register, and
- a 16-bit input capture register

A block diagram of the timer registers is shown in Figure 21.

● Free Running Counter ($0009:$000A)

The key element in the programmable timer is a 16-bit free running counter which is driven to increasing values by E (Enable). The counter value may be read by the CPU software at any time. The counter is cleared to zero by reset and may be considered a read-only register with one exception. Any CPU write to the counter's address ($09) will always result in preset value of $FFF8 being loaded into the counter regardless of the value involved in the write. This preset figure is intended for testing operation of the part, but may be of value in some applications.

● Output Compare Register ($000B:$000C)

The Output Compare Register is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly compared with the current value of the free running counter. When a match is found, a flag is set (OCF) in the Timer Control and Status Register (TCSR) and the current value of the Output Level bit (OLVL) in the TCSR is clocked to the Output Level Register. Providing the Data Direction Register for Port 2, Bit 1 contains a "1" (Output), the output level register value will appear on the pin for Port 2 Bit 1. The values in the Output Compare Register and Output level bit may then be changed to control the output level on the next compare value. The Output Compare Register is set to $FFFF during reset. The Compare function is inhibited for one cycle following a write to the high byte of the Output Compare Register to insure a valid 16-bit value is in the register before a compare is made.

● Input Capture Register ($000D:$000E)

The Input Capture Register is a 16-bit read-only register used to store the current value of the free running counter when the proper transition of an external input signal occurs. The input transition change required to trigger the counter transfer is controlled by the input Edge bit (IEDG) in the TCSR. The Data Direction Register bit for Port 2 Bit 0, should* be clear (zero) in order to gate in the external input signal to the edge detect unit in the timer.

The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

* With Port 2 Bit 0 configured as an output and set to "1", the external input will still be seen by the edge detect unit.

● Timer Control and Status Register (TCSR) ($0008)

The Timer Control and Status Register consists of an 8-bit register of which all 8 bits are readable but only the low order 5 bits may be written. The upper three bits contain read-only timer status information and indicate the followings:

- a proper transition has taken place on the input pin with a subsequent transfer of the current counter value to the input capture register.
- a match has been found between the value in the free running counter and the output compare register, and
- when $0000 is in the free running counter.

Each of the flags may be enabled onto the HD6801V internal bus ($\overline{IRQ_2}$) with an individual Enable bit in the TCSR. If the

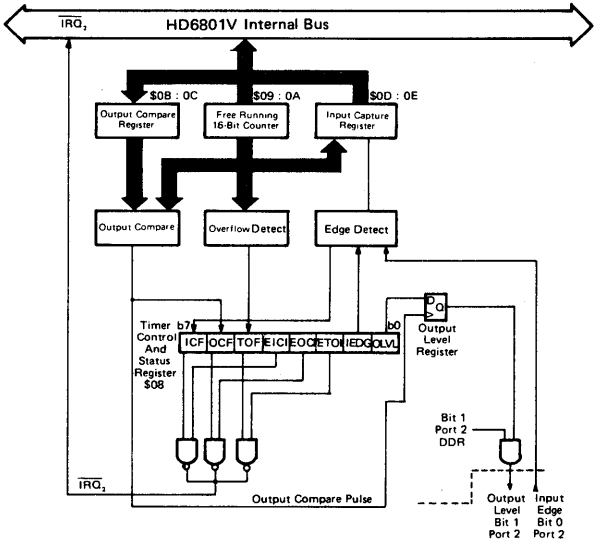


Figure 21 Block Diagram of Programmable Timer

Timer Control and Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

I-bit in the HD6801V Condition Code Register has been cleared, a priority vectored interrupt will occur corresponding to the flag bit(s) set. A description for each bit follows:

Bit 0 **OLVL** Output Level – This value is clocked to the output level register on a successful output compare. If the DDR for Port 2 bit 1 is set, the value will appear on the output pin.

Bit 1 **IEDG** Input Edge – This bit controls which transition of an input will trigger a transfer of the counter to the input capture register. The DDR for Port 2 Bit 0 must be clear for this function to operate. IEDG = 0 Transfer takes place on a negative edge ("High"-to-"Low" transition).
IEDG = 1 Transfer takes place on a positive edge ("Low"-to-"High" transition).

Bit 2 **ETOI** Enable Timer Overflow Interrupt – When set, this bit enables $\overline{IRQ_2}$ to occur on the internal bus for a TOF interrupt; when clear the interrupt is inhibited.

Bit 3 **EOCI** Enable Output Compare Interrupt – When set, this bit enables $\overline{IRQ_2}$ to appear on the internal bus for an output compare interrupt; when clear the interrupt is inhibited.

Bit 4 **EICI** Enable Input Capture Interrupt – When set, this bit enables $\overline{IRQ_2}$ to occur on the internal bus for an input capture interrupt; when clear the interrupt is inhibited.

Bit 5 **TOF** Timer Overflow Flag – This read-only bit is set when the counter contains $FFFF. It is cleared by a read of the TCSR (with TOF set) followed by an CPU read of the Counter ($09).

Bit 6 **OCF** Output Compare Flag – This read-only bit is set when a match is found between the output compare register and the free running counter. It is cleared by a read of the TCSR (with OCF set) followed by an CPU write to the output compare register ($0B or $0C).

Bit 7 **ICF** Input Capture Flag – This read-only status bit is set by a proper transition on the input; it is cleared by a read of the TCSR (with ICF set) followed by an CPU read of the Input Capture Register ($0D).

■ SERIAL COMMUNICATION INTERFACE

The HD6801V contains a full-duplex asynchronous serial communication interface (SCI) on chip. The controller comprises a transmitter and a receiver which operate independently or each other but in the same data format and at the same data rate. Both transmitter and receiver communicate with the CPU via the data bus and with the outside world via pins 2, 3, and 4 of Port 2. The hardware, software, and registers are explained in the following paragraphs.

• Wake-Up Feature

In a typical multi-processor application, the software protocol will usually contain a destination address in the initial byte(s) of the message. In order to permit non-selected MCU's to ignore the remainder of the message, a wake-up feature is included whereby all further interrupt processing may be optionally inhibited until the beginning of the next message. When the next message appears, the hardware re-enables (or "wakes-up") for the next message. The "wake-up" is automatically triggered by a string of ten consecutive 1's which indicates an idle transmit line. The software protocol must provide for the short idle period between any two consecutive messages.

• Programmable Options

The following features of the HD6801V serial I/O section are programmable:

- format – standard mark/space (NRZ)
- Clock – external or internal
- baud rate – one of 4 per given CPU ϕ_2 clock frequency or external clock ×8 input
- wake-up feature – enabled or disabled
- Interrupt requests – enabled or masked individually for transmitter and receiver data registers
- clock output – internal clock enabled or disabled to Port 2 (Bit 2)
- Port 2 (bits 3 and 4) – dedicated or not dedicated to serial I/O individually for transmitter and receiver.

• Serial Communication Hardware

The serial communication hardware is controlled by 4 registers as shown in Figure 22. The registers include:

- an 8-bit control and status register
- a 4-bit rate and mode control register (write only)
- an 8-bit read only receive data register and
- an 8-bit write only transmit data register.

In addition to the four registers, the serial I/O section utilizes bit 3 (serial input) and bit 4 (serial output) of Port 2. Bit 2 of Port 2 is utilized if the internal-clock-out or external-clock-in options are selected.

Transmit/Receive Control and Status (TRCS) Register

The TRCS register consists of an 8-bit register of which all 8 bits may be read while only bits 0~4 may be written. The register is initialized to $20 by reset. The bits in the TRCS register are defined as follows:

Transmit/Receive Control and Status Register

7	6	5	4	3	2	1	0	
RDRE	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR : $0011

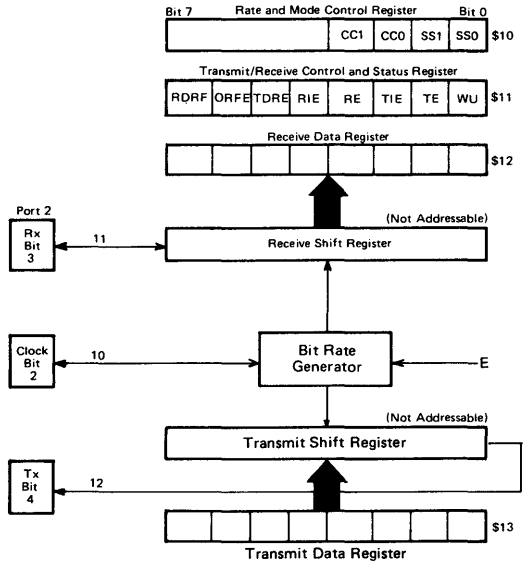


Figure 22 Serial I/O Registers

Bit 0 **WU** "Wake-up" on Next Message – set by HD6801V software and cleared by hardware on receipt of ten consecutive 1's or reset of RE flag. It should be noted that RE flag should be set in advance of CPU set of WU flag.

Bit 1 **TE** Transmit Enable – set by HD6801V to produce preamble of nine consecutive 1's and to enable gating of transmitter output to Port 2, bit 4 regardless of the DDR value corresponding to this bit; when clear, serial I/O has no effect on Port 2 bit 4.

TE set should be after at least one bit time of data transmit rate from the set-up of transmit data rate and mode.

Bit 2 **TIE** Transmit Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 5 (TDRE) is set; when clear, the TDRE value is masked from the bus.

Bit 3 **RE** Receiver Enable – when set, gates Port 2 bit 3 to input of receiver regardless of DDR value for this bit; when clear, serial I/O has no effect on Port 2 bit 3.

Bit 4 **RIE** Receiver Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 7 (RDRF) or bit 6 (ORFE) is set; when clear, the interrupt is masked.

Bit 5 **TDRE** Transmit Data Register Empty – set by hardware when a transfer is made from the transmit data register to the output shift register. The TDRE bit is cleared by reading the status register, then writing a new byte into the transmit data register, TDRE is initialized to 1 by reset.

Bit 6 **ORFE** Over-Run-Framing Error – set by hardware when an overrun or framing error occurs (receive only). An overrun is defined as a new byte received with last byte still in Data Register/Buffer. A framing error has occurred when the byte boundaries in bit stream are not synchronized to bit counter. If WU flag is set, the ORFE bit will not be set. The ORFE bit is cleared by reading the status register, then reading the Receive Data Register, or by reset.

Bit 7 **RDRF** Receiver Data Register Full – set by hardware when a transfer from the input shift register to the receiver data register is made. If WU flag is set, the RDRF bit will not be set. The RDRF bit is cleared by reading the status register, then reading the Receive Data Register, or by reset.

Rate and Mode Control Register

The Rate and Mode Control register controls the following serial I/O variables:

- Baud rate
- format
- clocking source, and
- Port 2 bit 2 configuration

The register consists of 4 bits all of which are write-only and cleared by reset. The 4 bits in the register may be considered as a pair of 2-bit fields. The two low order bits control the bit rate for internal clocking and the remaining two bits control the format and clock select logic. The register definition is as follows:

Rate and Mode Control Register

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	ADDR : $0010

Bit 0 **SS0** } Speed Select – These bits select the Baud rate for
Bit 1 **SS1** } the internal clock. The four rates which may be selected are a function of the CPU ϕ_2 clock frequency. Table 5 lists the available Baud rates.

Bit 2 **CC0** } Clock Control and Format Select – this 2-bit field
Bit 3 **CC1** } controls the format and clock select logic. Table 6 defines the bit field.

Table 5 SCI Bit Times and Rates

SS1 : SS0	XTAL	2.4576 MHz	4.0 MHz	4.9152 MHz*
	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800 Baud
0 1	E ÷ 128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	E ÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	E ÷ 4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

*HD6801V5 Only

Table 6 SCI Format and Clock Source Control

CC1: CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS; bit 4 is used for serial output if TE = "1" in TRCS.

Internally Generated Clock

If the user wishes for the serial I/O to furnish a clock, the following requirements are applicable:

- the values of RE and TE are immaterial.
- CC1, CC0 must be set to 10
- the maximum clock rate will be E ÷ 16.
- the clock will be at 1× the bit rate and will have a rising edge at mid-bit.

Externally Generated Clock

If the user wishes to provide an external clock for the serial I/O, the following requirements are applicable:

- the CC1, CC0, field in the Rate and Mode Control Register must be set to 11,
- the external clock must be set to 8 times (×8) the desired baud rate and
- the maximum external clock frequency is 1.0 MHz.

• Serial Operations

The serial I/O hardware should be initialized by the HD6801V software prior to operation. This sequence will normally consist of;

- writing the desired operation control bits to the Rate and Mode Control Register and
- writing the desired operational control bits in the Transmit/Receive Control and Status Register.

The Transmitter Enable (TE) and Receiver Enable (RE) bits may be left set for dedicated operations.

Transmit Operations

The transmit operation is enabled by the TE bit in the Transmit/Receive Control and Status Register. This bit when set, gates the output of the serial transmit shift register to Port 2 Bit 4 and takes unconditional control over the Data Direction Register value for Port 2, Bit 4.

Following a $\overline{RES}$ the user should configure both the Rate and Mode Control Register and the Transmit/Receive Control and Status Register for desired operation. Setting the TE bit during this procedure initiates the serial output by first transmitting a nine-bit preamble of 1's. Following the preamble, internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or,
2) if data has been loaded into the Transmit Data Register (TDRE = 0), the word is transferred to the output shift register and transmission of the data word will begin.

During the data transmit, the 0 start bit is first transmitted. Then the 8 data bits (beginning with bit 0) followed by the stop bit, are transmitted. When the Transmitter Data Register has been emptied, the hardware sets the TDRE flag bit.

If the HD6801V fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the parallel data register to the serial output register should occur) then a 1 will be sent (instead of a 0) at "Start" bit time, followed by more 1's until more data is supplied to the data register. No 0's will be sent while TDRE remains a 1.

Receive Operation

The receive operation is enabled by the RE bit which gates in the serial input through Port 2 Bit 3. The receiver section operation is conditioned by the contents of the Transmit/Receive Control and Status Register and the Rate and Mode Control Register.

The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the NRZ Mode, the received bit stream is synchronized by the first 0 (space) encountered.

The approximate center of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit) a framing error is assumed, and bit ORFE is set. If the tenth bit as a 1, the data is transferred to the Receive Data Register, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the HD6801V responds to either flag (RDRF or ORFE) by reading the status register followed by reading the Data Register, RDRF (or ORFE) will be cleared.

■ RAM CONTROL REGISTER

This register, which is addressed at $0014, gives status information about the standby RAM. A 0 in the RAM enable bit (RAME) will disable the standby RAM, thereby protecting it at power down if V_{CC} Standby is held greater than V_{SBB} volts, as explained previously in the signal description for V_{CC} Standby.

RAM Control Register

$0014	STBY PWR	RAME	X	X	X	X	X	X

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 **RAME** The RAM Enable control bit allows the user the ability to disable the standby RAM. This bit is set to a logic "1" by $\overline{RES}$ which enables the standby RAM and can be written to one or zero under program control. When the RAM is disabled, data is read from external memory.
Big 7 **STBY PWR** The Standby Power bit is cleared when the standby voltage is removed. This bit is a read/write status flag that the user can read which indicates that the standby RAM voltage has been applied, and the data in the standby RAM is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD6801V is upward object code compatible with the HD6800 as it implements the full HMCS6800 instruction set. The execution times of key instructions have been reduced to increase throughout. In addition, new instructions have been added; these include 16-bit operations and a hardware multiply.

Included in the instruction set section are the following:

- CPU Programming Model (Figure 23)
- Addressing modes
- Accumulator and memory instructions – Table 7
- New instructions
- Index register and stack manipulations instructions – Table 8
- Jump and branch instructions – Table 9
- Condition code register manipulation instructions – Table 10
- Instructions Execution times in machine cycles – Table 11
- Summary of cycle by cycle operation – Table 12
- Summary of undefined instructions operation
- Op codes Map – Table 13

● CPU Programming Model

The programming model for the HD6801V is shown in Figure 23. The double (D) accumulator is physically the same as the Accumulator A concatenated with the Accumulator B so that any operation using accumulator D will destroy information in A and B.

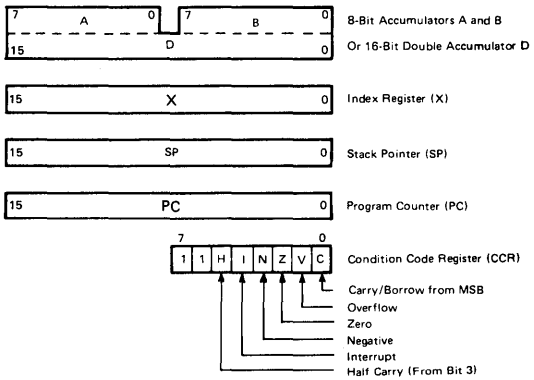


Figure 23 CPU Programming Model

● CPU Addressing Modes

The HD6801V eight-bit microcomputer unit has seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 11 along with the associated instruction execution time that is given in machine cycles. With a clock frequency of 4 MHz, these times would be microseconds.

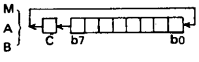
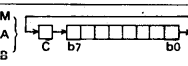
Accumulator (ACCX) Addressing

In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

Immediate Addressing

In immediate addressing, the operand is contained in the second byte of the instruction except LDS and LDX which have the operand in the second and third bytes of the instruction. The CPU addresses this location when it fetches the immediate instruction for execution. These are two or three-byte instructions.

Table 7 Accumulator & Memory Instructions

Operations	Mnemonic	Addressing Modes														Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	2	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	2	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	2	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	10	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	3	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	⑥	↕
	ROLA													49	2	1		•	•	↕	↕	⑥	↕
	ROLB													59	2	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	⑥	↕
	RORA													46	2	1		•	•	↕	↕	⑥	↕
	RORB													56	2	1		•	•	↕	↕	⑥	↕

The Condition Code Register notes are listed after Table 10.

(Continued)

Table 7 Accumulator & Memory Instructions (Continued)

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	‡	‡	⑥	‡
	ASLA													48	2	1	A	•	•	‡	‡	⑥	‡
	ASLB													58	2	1	B	•	•	‡	‡	⑥	‡
Double Shift Left, Arithmetic	ASLD													05	3	1	C ACC A/ ACC B ← 0 (A7 A0 B7 B0)	•	•	‡	‡	⑥	‡
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	‡	‡	⑥	‡
	ASRA													47	2	1	A	•	•	‡	‡	⑥	‡
	ASRB													57	2	1	B	•	•	‡	‡	⑥	‡
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	‡	⑥	‡
	LSRA													44	2	1	A 0→ (b7 b0 C)	•	•	R	‡	⑥	‡
	LSRB													54	2	1	B	•	•	R	‡	⑥	‡
Double Shift Right Logical	LSRD													04	3	1	0→ ACC A/ ACC B → C (A7 A0 B7 B0)	•	•	R	‡	⑥	‡
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	‡	‡	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	‡	‡	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	‡	‡	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A – M → A	•	•	‡	‡	‡	‡
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B – M → B	•	•	‡	‡	‡	‡
Double Subtract	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				A : B – M : M + 1 → A : B	•	•	‡	‡	‡	‡
Subtract Accumulators	SBA													10	2	1	A – B → A	•	•	‡	‡	‡	‡
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A – M – C → A	•	•	‡	‡	‡	‡
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B – M – C → B	•	•	‡	‡	‡	‡
Transfer Accumulators	TAB													16	2	1	A → B	•	•	‡	‡	R	•
	TBA													17	2	1	B → A	•	•	‡	‡	R	•
Test Zero or Minus	TST							6D	6	2	7D	6	3				M – 00	•	•	‡	‡	R	R
	TSTA													4D	2	1	A – 00	•	•	‡	‡	R	R
	TSTB													5D	2	1	B – 00	•	•	‡	‡	R	R

The Condition Code Register notes are listed after Table 10.

Direct Addressing

In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine i.e., locations zero through 255. Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random access memory. These are two-byte instructions.

Extended Addressing

In extended addressing, the address contained in the second byte of the instruction is used as the higher eight-bits of the address of the operand. The third byte of the instruction is used as the lower eight-bits of the address for the operand. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest eight bits in the CPU. The carry is then added to the higher order eight bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

Implied Addressing

In the implied addressing mode the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

Relative Addressing

In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest eight bits plus two. The carry or borrow is then added to the high eight bits. This allows the user to address data within a range of -126 to +129 bytes of the present instruction. These are two-byte instructions.

- **New Instructions**

In addition to the existing 6800 Instruction Set, the following new instructions are incorporated in the HD6801V Microcomputer.

ABX Adds the 8-bit unsigned accumulator B to the 16-bit X-Register taking into account the possible carry out of the low order byte of the X-Register.

ADDD Adds the double precision ACCD* to the double precision value M:M+1 and places the results in ACCD.

ASLD Shifts all bits of ACCD one place to the left. Bit 0 is loaded with zero. The C bit is loaded from the most significant bit of ACCD.

LDD Loads the contents of double precision memory location into the double accumulator A:B. The condition codes are set according to the data.

LSRD Shifts all bits of ACCD one place to the right. Bit 15 is loaded with zero. The C bit is loaded from the least significant bit to ACCD.

MUL Multiplies the 8 bits in accumulator A with the 8 bits in accumulator B to obtain a 16-bit unsigned number in A:B, ACCA contains MSB of result.

PSHX The contents of the index register is pushed onto the stack at the address contained in the stack pointer. The stack pointer is decremented by 2.

PULX The index register is pulled from the stack beginning at the current address contained in the stack pointer +1. The stack pointer is incremented by 2 in total.

STD Stores the contents of double accumulator A:B in memory. The contents of ACCD remain unchanged.

SUBD Subtracts the contents of M:M + 1 from the contents of double accumulator AB and places the result in ACCD.

BRN Never branches. If effect, this instruction can be considered a two byte NOP (No operation) requiring three cycles for execution.

CPX Internal processing modified to permit its use with any conditional branch instruction.

*ACCD is the 16 bit register (A:B) formed by concatenating the A and B accumulators. The A-accumulator is the most significant byte.

Table 8 Index Register and Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M : M + 1	•	•	‡	‡	‡	‡
Decrement Index Reg	DEX													09	3	1	X − 1 → X	•	•	•	‡	•	•
Decrement Stack Pntr	DES													34	3	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	3	1	X + 1 → X	•	•	•	‡	•	•
Increment Stack Pntr	INS													31	3	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	‡	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	‡	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	‡	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	‡	R	•
Index Reg → Stack Pntr	TXS													35	3	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•

The Condition Code Register notes are listed after Table 10.

Table 9 Jump and Branch Instructions

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	6	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——			⑧	——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	⑨	•	•	•	•

Table10 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code Register					
		IMPLIED				5	4	3	2	1	0
		OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	——			⑩	——	
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

Condition Code Register Notes: (Bit set it test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to result to N ⊕ C after shift has occurred.
⑦ (Bit N) Test: Result less than zero? (Bit 15 = 1)
⑧ (All) Load Condition Code Register from Stack. (See Special Operations)
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All) Set according to the contents of Accumulator A.
⑪ (Bit C) Set equal to result of Bit 7 (ACCB)

Table 11 Instruction Execution Times in Machine Cycle

	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative		ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA	•	•	•	•	•	2	•	INX	•	•	•	•	•	3	•
ABX	•	•	•	•	•	3	•	JMP	•	•	•	3	3	•	•
ADC	•	2	3	4	4	•	•	JSR	•	•	5	6	6	•	•
ADD	•	2	3	4	4	•	•	LDA	•	2	3	4	4	•	•
ADDD	•	4	5	6	6	•	•	LDD	•	3	4	5	5	•	•
AND	•	2	3	4	4	•	•	LDS	•	3	4	5	5	•	•
ASL	2	•	•	6	6	•	•	LDX	•	3	4	5	5	•	•
ASLD	•	•	•	•	•	3	•	LSR	2	•	•	6	6	•	•
ASR	2	•	•	6	6	•	•	LSRD	•	•	•	•	•	3	•
BCC	•	•	•	•	•	•	3	MUL	•	•	•	•	•	10	•
BCS	•	•	•	•	•	•	3	NEG	2	•	•	6	6	•	•
BEQ	•	•	•	•	•	•	3	NOP	•	•	•	•	•	2	•
BGE	•	•	•	•	•	•	3	ORA	•	2	3	4	4	•	•
BGT	•	•	•	•	•	•	3	PSH	3	•	•	•	•	•	•
BHI	•	•	•	•	•	•	3	PSHX	•	•	•	•	•	4	•
BIT	•	2	3	4	4	•	•	PUL	4	•	•	•	•	•	•
BLE	•	•	•	•	•	•	3	PULX	•	•	•	•	•	5	•
BLS	•	•	•	•	•	•	3	ROL	2	•	•	6	6	•	•
BLT	•	•	•	•	•	•	3	ROR	2	•	•	6	6	•	•
BMI	•	•	•	•	•	•	3	RTI	•	•	•	•	•	10	•
BNE	•	•	•	•	•	•	3	RTS	•	•	•	•	•	5	•
BPL	•	•	•	•	•	•	3	SBA	•	•	•	•	•	2	•
BRA	•	•	•	•	•	•	3	SBC	•	2	3	4	4	•	•
BRN	•	•	•	•	•	•	3	SEC	•	•	•	•	•	2	•
BSR	•	•	•	•	•	•	6	SEI	•	•	•	•	•	2	•
BVC	•	•	•	•	•	•	3	SEV	•	•	•	•	•	2	•
BVS	•	•	•	•	•	•	3	STA	•	•	3	4	4	•	•
CBA	•	•	•	•	•	2	•	STD	•	•	4	5	5	•	•
CLC	•	•	•	•	•	2	•	STS	•	•	4	5	5	•	•
CLI	•	•	•	•	•	2	•	STX	•	•	4	5	5	•	•
CLR	2	•	•	6	6	•	•	SUB	•	2	3	4	4	•	•
CLV	•	•	•	•	•	2	•	SUBD	•	4	5	6	6	•	•
CMP	•	2	3	4	4	•	•	SWI	•	•	•	•	•	12	•
COM	2	•	•	6	6	•	•	TAB	•	•	•	•	•	2	•
CPX	•	4	5	6	6	•	•	TAP	•	•	•	•	•	2	•
DAA	•	•	•	•	•	2	•	TBA	•	•	•	•	•	2	•
DEC	2	•	•	6	6	•	•	TPA	•	•	•	•	•	2	•
DES	•	•	•	•	•	3	•	TST	2	•	•	6	6	•	•
DEX	•	•	•	•	•	3	•	TSX	•	•	•	•	•	3	•
EOR	•	2	3	4	4	•	•	TXS	•	•	•	•	•	3	•
INC	2	•	•	6	6	•	•	WAI	•	•	•	•	•	9	•
INS	•	•	•	•	•	3	•								

- **Summary of Cycle by Cycle Operation**

Table 12 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write line (R/$\overline{W}$) during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. (In general, instructions with the same addressing mode and number of cycles execute in the same manner; exceptions are indicated in the table).

Table 12 Cycle by Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR	2	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Operand Data
AND ORA					
BIT SBC					
CMP SUB					
LDS	3	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Operand Data (High Order Byte)
LDD		3	Op Code Address + 2	1	Operand Data (Low Order Byte)
CPX	4	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Data (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Data (Low Order Byte)
		4	Address Bus FFFF	1	Low Byte of Restart Vector
DIRECT					
ADC EOR	3	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand
AND ORA		3	Address of Operand	1	Operand Data
BIT SBC					
CMP SUB					
STA	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address
		3	Destination Address	0	Data from Accumulator
LDS	4	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand
LDD		3	Address of Operand	1	Operand Data (High Order Byte)
		4	Operand Address + 1	1	Operand Data (Low Order Byte)
STS	4	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand
STD		3	Address of Operand	0	Register Data (High Order Byte)
		4	Address of Operand + 1	0	Register Data (Low Order Byte)
CPX	5	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Address of Operand
ADDD		3	Operand Address	1	Operand Data (High Order Byte)
		4	Operand Address + 1	1	Operand Data (Low Order Byte)
		5	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	5	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Subroutine Address	1	First Subroutine Op Code
		4	Stack Pointer	0	Return Address (Low Order Byte)
		5	Stack Pointer + 1	0	Return Address (High Order Byte)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1*	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

* In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB=FFFF, DB=Low Byte of Reset Vector. (Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

* In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMPLIED					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address + 1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI**	9	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Op Code Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
WAI**		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)

**While the MCU is in the "Wait" state, its bus state will appear as a series of MCU reads of an address which is seven locations less than the original contents of the Stack Pointer. Contrary to the HD6800, none of the ports are driven to the high impedance state by a WAI instruction.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instruction	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RELATIVE					
BCC BHT BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMT BVS BRN	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Branch Offset Low Byte of Restart Vector
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Address Bus FFFF Subroutine Starting Address Stack Pointer Stack Pointer – 1	1 1 1 1 0 0	Op Code Branch Offset Low Byte of Restart Vector Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

• Summary of Undefined Instruction Operations

The HD6801V has 36 undefined instructions. When these are carried out, the contents of Register and Memory in MCU change at random.

When the op codes (4E, 5E) are used to execute, the MCU continues to increase the program counter and it will not stop until the Reset signal enters. These op codes are used to test the LSI.

Table 13 Op codes Map

HD6801V MICROCOMPUTER INSTRUCTIONS																		
OP CODE						ACC A	ACC B	IND	EXT	ACCA or SP				ACCB or X				
										IMM	DIR	IND	EXT	IMM	DIR	IND	EXT	
LO \ HI		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								
0010	2			BHI	PULA (+1)					SBC								
0011	3			BLS	PULB (+1)	COM				*	SUBD (+2)			*	ADDD (+2)			3
0100	4	LSRD (+1)		BCC	DES	LSR				AND								4
0101	5	ASLD (+1)		BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX (+1)		BVC	PULX (+2)	ASL				EOR								8
1001	9	DEX (+1)	DAA	BVS	RTS (+2)	ROL				ADC								9
1010	A	CLV		BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI (+7)					ADD								B
1100	C	CLC		BGE	PSHX (+1)	INC				*	CPX (+2)			*	LDD (+1)			C
1101	D	SEC		BLT	MUL (+7)	TST				BSR (+4)	JSR (+2)			* (+1)	STD (+1)			D
1110	E	CLI		BGT	WAI (+6)	**		JMP (-3)		*	LDS (+1)			*	LDX (+1)			E
1111	F	SEI		BLE	SWI (+9)	CLR				* (+1)	STS (+1)			* (+1)	STX (+1)			F
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4	

[NOTES] 1) Undefined Op codes are marked with ▨.

2) ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

3) The instructions shown below are all 3 bytes and are marked with "*".
Immediate addressing mode of SUBD, CPX, LDS, ADDD, LDD and LDX instructions, and undefined op codes (8F, CD, CF).

4) The Op codes (4E, 5E) are 1 byte/∞ cycles instructions, and are marked with "**"

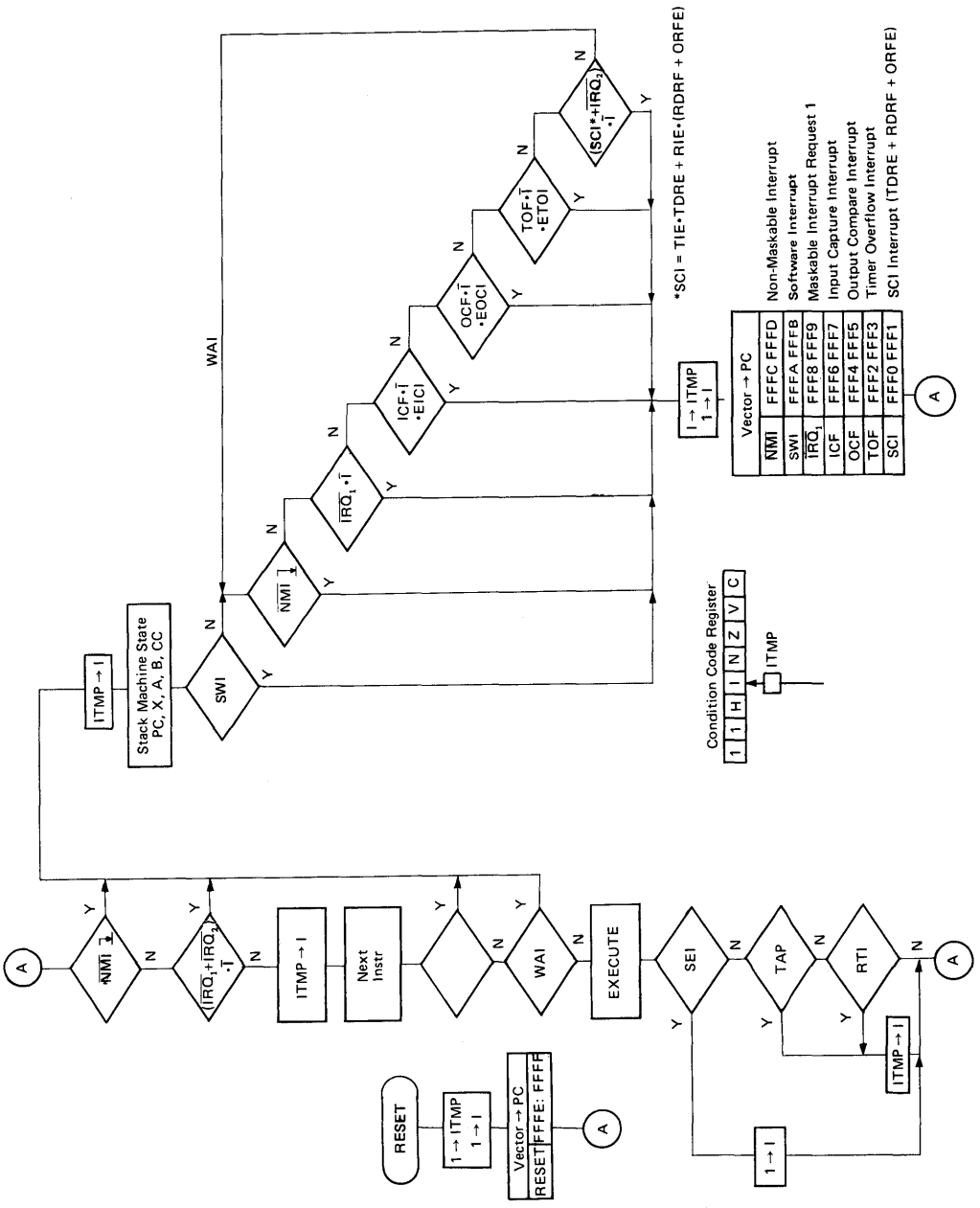


Figure 24 Interrupt Flowchart

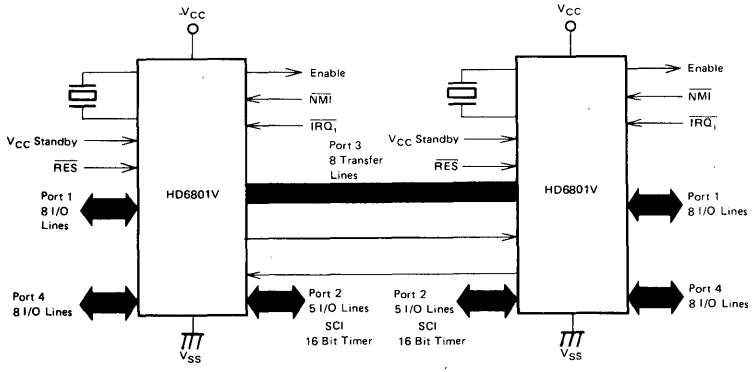


Figure 25 HD6801V MCU Single-Chip Dual Processor Configuration

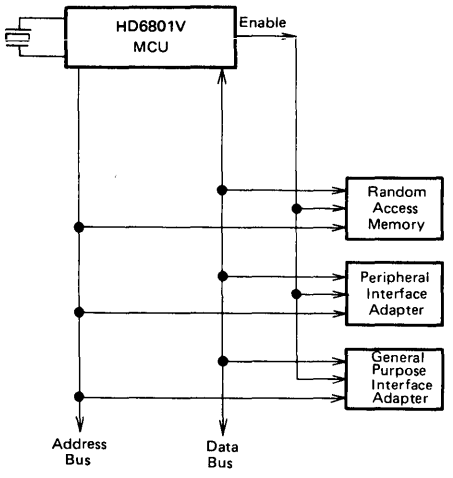


Figure 26 HD6801V MCU Expanded Non-Multiplexed Mode

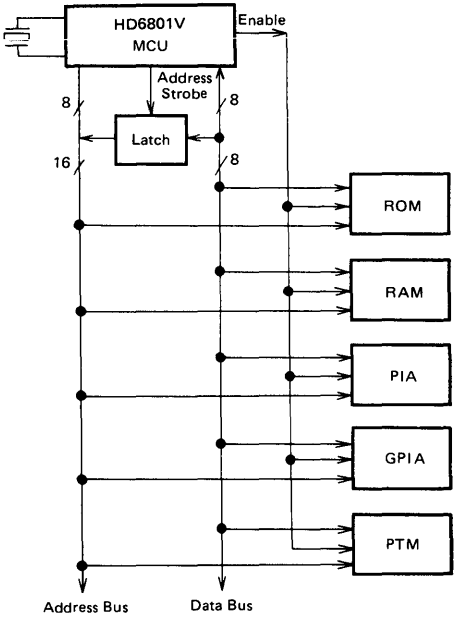


Figure 27 HD6801V MCU Expanded Multiplexed Mode

■ **Caution for the HD6801 Family SCI, TIMER Status Flag**

The flags shown in Table 14 are cleared by reading/writing (flag reset condition 2) the data register corresponding to each flag after reading the status register (flag reset condition 1).

To clear the flag correctly, take the following procedure:

1. Read the status register
2. Test the flag
3. Read the data register

Table 14 Status Flag Reset Conditions

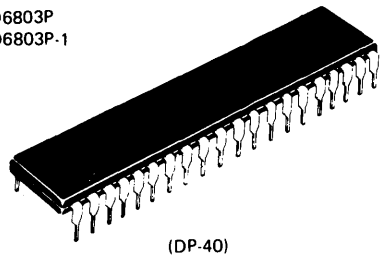
	Status Flag	Flag Reset Condition 1 (Status Register)	Flag Reset Condition 2 (Data Register)
TIMER	ICF	When each flag is	ICR/Read
	OCF	"1",	OCR/Write
	TOF	TRCSR/Read	TC/Read
SCI	RDRF	When each flag is	RDR/
	ORFE	"1",	Read
	TDRE	TRCSR/Read	TDR/Write

HD6803, HD6803-1

MPU (Micro Processing Unit)

The HD6803 MPU is an 8-bit micro processing unit which is compatible with the HMCS6800 family of parts. The HD6803 MPU is object code compatible with the HD6800 with improved execution times of key instructions plus several new 16-bit and 8-bit instruction including an 8 × 8 unsigned multiply with 16-bit result. The HD6803 MPU can be expanded to 65k bytes. The HD6803 MPU is TTL compatible and requires one +0.5 volt power supply. The HD6803 MPU has 128 bytes of RAM, Serial Communications Interface (S.C.I.), and parallel I/O as well as a three function 16-bit timer. Features and Block Diagram of the HD6803 include the following:

HD6803P
HD6803P-1

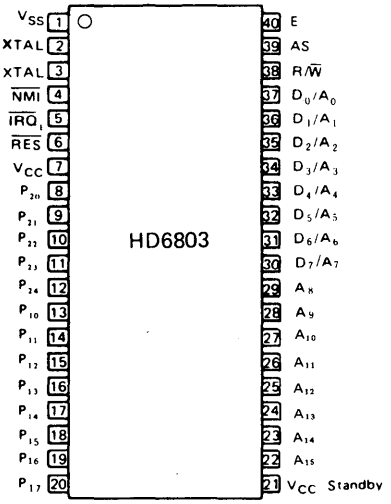


(DP-40)

■ FEATURES

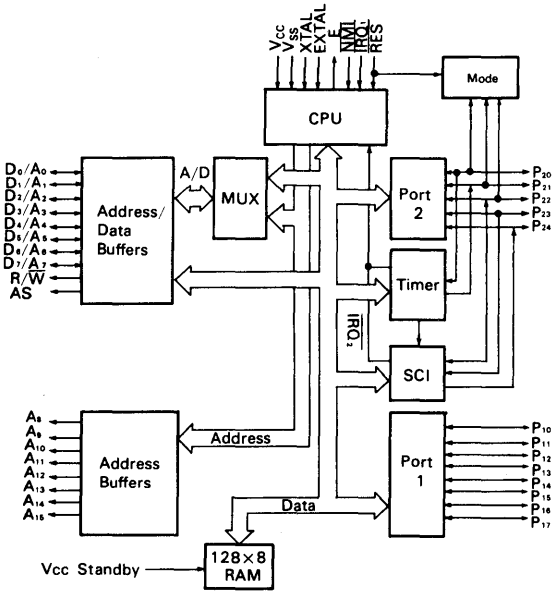
- Expanded HMCS6800 Instruction Set
- 8 x 8 Multiply
- On-Chip Serial Communications Interface (S.C.I.)
- Object Code Compatible with The HD6800 MPU
- 16-Bit Timer
- Expandable to 65k Bytes
- Multiplexed Address and Data
- 128 Bytes of RAM (64 Bytes Retainable On Power Down)
- 13 Parallel I/O Lines
- Internal Clock/Divided-By-Four
- TTL Compatible Inputs and Outputs
- Interrupt Capability
- Compatible with MC6803 and MC6803-1

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



■ TYPE OF PRODUCTS

Type No.	Bus Timing
HD6803	1.0MHz
HD6803-1	1.25MHz

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} =5.0V±5%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	Other Inputs*			2.0	–	V_{CC}	
Input "Low" Voltage	All Inputs*	V_{IL}		-0.3	–	0.8	V
Input Load Current	EXTAL	$\|I_{in}\|$	V_{in} = 0 ~ V_{CC}	–	–	0.8	mA
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$	$\|I_{in}\|$	V_{in} = 0 ~ 5.25V	–	–	2.5	μA
Three State (Offset) Leakage Current	P_{10}~P_{17}, D_0/A_0~D_7/A_7	$\|I_{TSI}\|$	V_{in} = 0.5 ~ 2.4V	–	–	10	μA
	P_{20} ~ P_{24}			–	–	100	
Output "High" Voltage	D_0/A_0 ~ D_7/A_7	V_{OH}	I_{LOAD} = -205 μA	2.4	–	–	V
	A_8 ~ A_{15}, E, R/$\overline{W}$, AS		I_{LOAD} = -145 μA	2.4	–	–	
	Other Outputs		I_{LOAD} = -100 μA	2.4	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{LOAD} = 1.6 mA	–	–	0.5	V
Darlington Drive Current	P_{10} ~ P_{17}	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Power Dissipation		P_D		–	–	1200	mW
Input Capacitance	D_0/A_0 ~ D_7/A_7	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz	–	–	12.5	pF
	Other Inputs			–	–	10.0	
V_{CC} Standby	Powerdown	V_{SBB}		4.0	–	5.25	V
	Operating	V_{SB}		4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	V_{SBB} = 4.0V	–	–	8.0	mA

*Except Mode Programming Levels.

• AC CHARACTERISTICS

BUS TIMING (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	HD6803			HD6803-1			Unit
				min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.8	–	10	μs
Address Strobe Pulse Width "High" *		PW_{ASH}		200	–	–	150	–	–	ns
Address Strobe Rise Time		t_{ASr}		5	–	50	5	–	50	ns
Address Strobe Fall Time		t_{ASf}		5	–	50	5	–	50	ns
Address Strobe Delay Time *		t_{ASD}		60	–	–	30	–	–	ns
Enable Rise Time		t_{Er}		5	–	50	5	–	50	ns
Enable Fall Time		t_{Ef}		5	–	50	5	–	50	ns
Enable Pulse Width "High" Time *		PW_{EH}		450	–	–	340	–	–	ns
Enable Pulse Width "Low" Time *		PW_{EL}		450	–	–	350	–	–	ns
Address Strobe to Enable Delay Time *		t_{ASED}		60	–	–	30	–	–	ns
Address Delay Time		t_{AD}		–	–	260	–	–	260	ns
Address Delay Time for Latch *		t_{ADL}		–	–	270	–	–	260	ns
Data Set-up Write Time		t_{DSW}		225	–	–	115	–	–	ns
Data Set-up Read Time		t_{DSR}		80	–	–	70	–	–	ns
Data Hold Time	Read	t_{HR}		10	–	–	10	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	
Address Set-up Time for Latch *		t_{ASL}		60	–	–	50	–	–	ns
Address Hold Time for Latch		t_{AHL}		20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	ns
Peripheral Read Access Time (Multiplexed Bus)*		(t_{ACCM})		–	–	(600)	–	–	(420)	ns
Oscillator stabilization Time		t_{RC}	Fig. 8	100	–	–	100	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 7,8	200	–	–	200	–	–	ns

*These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING (V_{CC} = 5.0V ± 5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Peripheral Data Setup Time	Port 1, 2	t_{PDSU}	Fig. 2	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2	t_{PDH}	Fig. 2	200	–	–	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*	t_{PWD}	Fig. 3	–	–	400	ns

* Except P_{21}

TIMER, SCI TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Timer Input Pulse Width	t_{PWT}		$2t_{cyc}$+200	–	–	ns
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 4	–	–	600	ns
SCI Input Clock Cycle	t_{Scyc}		1	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Mode Programming Input "Low" Voltage		V_{MPL}	Fig. 5	–	–	1.7	V
Mode Programming Input "High" Voltage		V_{MPH}		4.0	–	–	V
$\overline{RES}$ "Low" Pulse Width		PW_{RSTL}		3.0	–	–	t_{cyc}
Mode Programming Set-up Time		t_{MPS}		2.0	–	–	t_{cyc}
Mode Programming Hold Time	$\overline{RES}$ Rise Time $\geqq 1\mu s$	t_{MPH}		0	–	–	ns
	$\overline{RES}$ Rise Time $< 1\mu s$			100	–	–	

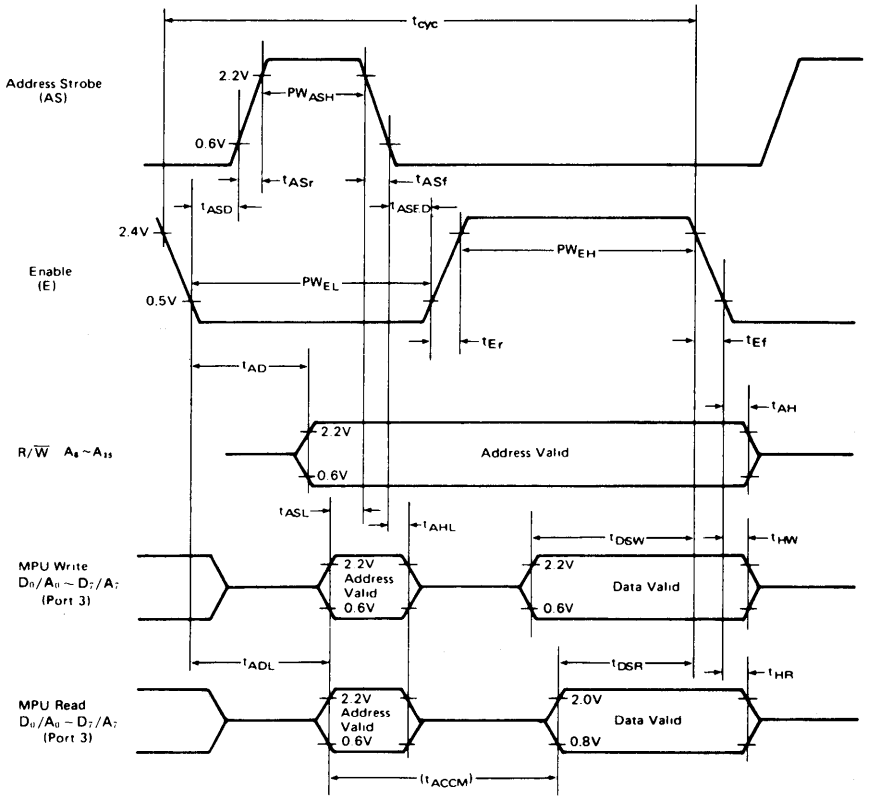


Figure 1 Expanded Multiplexed Bus Timing

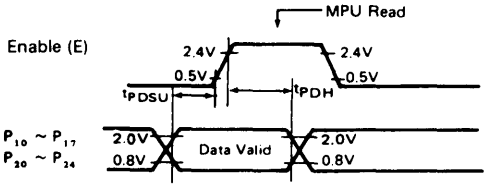


Figure 2 Data Set-up and Hold Times (MPU Read)

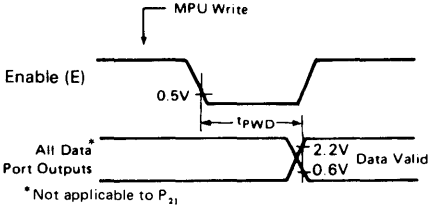


Figure 3 Port Data Delay Timing (MPU Write)

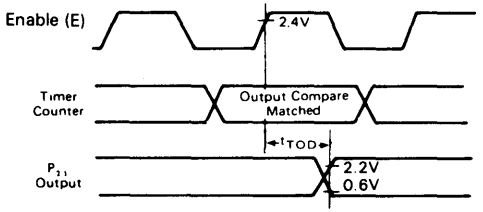


Figure 4 Timer Output Timing

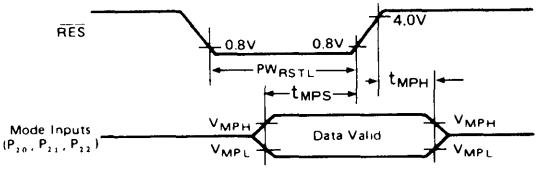


Figure 5 Mode Programming Timing

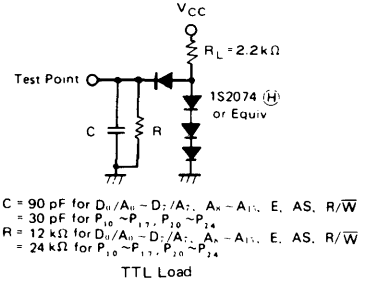


Figure 6 Bus Timing Test Load

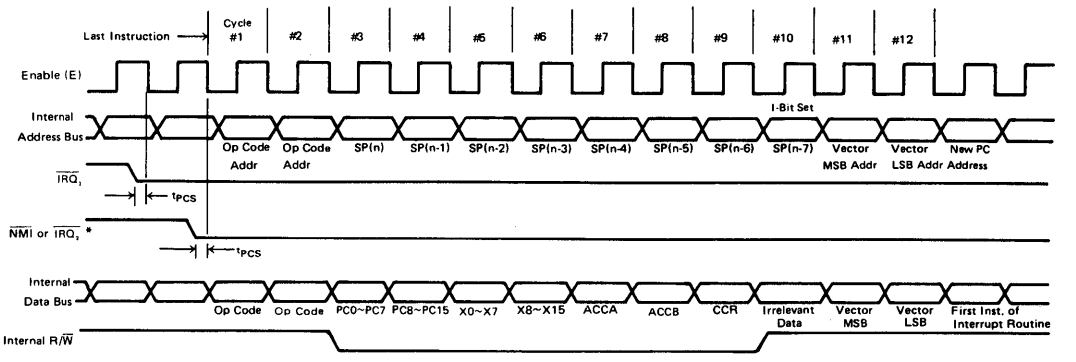


Figure 7 Interrupt Sequence

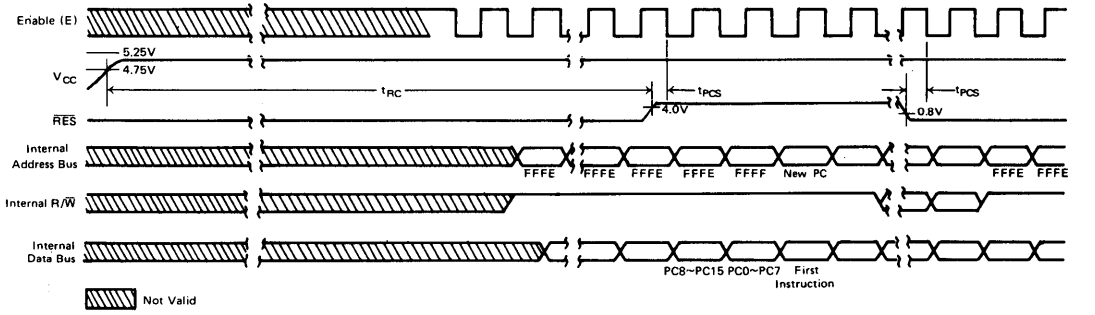


Figure 8 Reset Timing

■ SIGNAL DESCRIPTIONS

• VCC and VSS

These two pins are used to supply power and ground to the chip. The voltage supplied will be +5 volts ±5%.

• XTAL and EXTAL

These connections are for a parallel resonant fundamental crystal, AT cut. Devide-by-4 circuitry is included with the internal clock, so a 4 MHz crystal may be used to run the system at 1 MHz. The devide-by-4 circuitry allows for use of the inexpensive 3.58 MHz Color TV crystal for non-time critical applications. Two 22pF capacitors are needed from the two crystal pins to ground to insure reliable operation. An example of the crystal interface is shown in Fig. 9. EXTAL may be driven by an external TTL compatible source with a 45% to 55% duty cycle. It will devided by 4 any frequency less than or equal to 5 MHz. XTAL must be grounded if an external clock is used.

Nominal Crystal Parameter

Item \ Crystal	4 MHz	5 MHz
C_O	7pF max.	4.7pF max.
R_S	60Ω max.	30Ω typ.

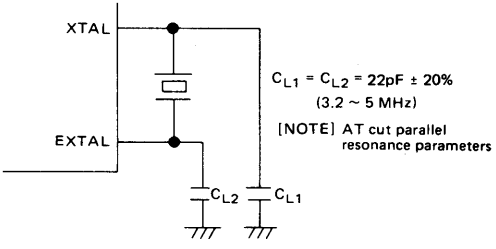


Figure 9 Crystal Interface

- **V_{CC} Standby**

This pin will supply +5 volts ±5% to the standby RAM on the chip. The first 64 bytes of RAM will be maintained in the power down mode with 8 mA current max. The circuit of figure 13 can be utilized to assure that V_{CC} Standby does not go below V_{SBB} during power down.

To retain information in the RAM during power down the following procedure is necessary:

1) Write "0" into the RAM enable bit, RAME. RAME is bit 6 of the RAM Control Register at location $0014. This disables the standby RAM, thereby protecting it at power down.
2) Keep V_{CC} Standby greater than V_{SBB}.

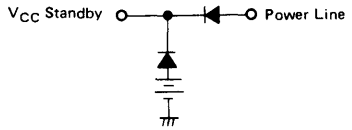


Figure 10 Battery Backup for V_{CC} Standby

- **Reset ($\overline{RES}$)**

This input is used to reset and start the MPU from a power down condition, resulting from a power failure or an initial startup of the processor. On power up, the reset must be held "Low" for at least 100 ms. When reset during operation, $\overline{RES}$ must be held "Low" at least 3 clock cycles.

When a "High" level is detected, the CPU does the following;

1) All the higher order address lines will be forced "High".
2) I/O Port 2 bits, 2, 1, and 0 are latched into programmed control bits PC2, PC1 and PC0.
3) The last two ($FFFE, $FFFF) locations in memory will be used to load the program addressed by the program counter.
4) The interrupt mask bit is set. Clear before the CPU can recognize maskable interrupts.

- **Enable (E)**

This supplies the external clock for the rest of the system when the internal oscillator is used. It is a single phase, TTL compatible clock, and will be the divide-by-4 result of the crystal oscillator frequency. It will drive one TTL load and 90 pF capacitance.

- **Non-Maskable Interrupt ($\overline{NMI}$)**

When the falling edge of the input signal is detected at this pin, the CPU begins non-maskable interrupt sequence internally. As with interrupt Request signal, the processor will complete the current instruction that is being executed before it recognizes the $\overline{NMI}$ signal. The interrupt mask bit in the Condition Code Register has no effect on $\overline{NMI}$.

In response to an $\overline{NMI}$ interrupt, the Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. At the end of the sequence, a 16-bit address will be loaded that points to a vectoring address located in memory locations $FFFC and $FFFD. An address loaded at these locations causes the CPU to branch to a non-maskable interrupt service routine in memory.

A 3.3 kΩ external resistor to V_{CC} should be used for wire-OR and optimum control of interrupts.

Inputs $\overline{IRQ_1}$ and $\overline{NMI}$ are hardware interrupt lines that are sampled during E and will start the interrupt routine on the $\overline{E}$ following the completion of an instruction.

- **Interrupt Request ($\overline{IRQ_1}$)**

This level sensitive input requests that an interrupt sequence be generated within the machine. The processor will complete the current instruction before it recognizes the request. At that time, if the interrupt mask bit in the Condition Code Register is not set, the machine will begin an interrupt sequence. The Index Register, Program Counter, Accumulators, and Condition Code Register are stored on the stack. Next the CPU will respond to the interrupt request by setting the interrupt mask bit "High" so that no further maskable interrupts may occur. At the end of the cycle, a 16-bit address will be loaded that points to a vectoring address which is located in memory locations $FFF8 and $FFF9. An address loaded at these locations causes the CPU to branch to an interrupt routine in memory.

The $\overline{IRQ_1}$ requires a 3.3 kΩ external resistor to V_{CC} which should be used for wire-OR and optimum control of interrupts. Internal Interrupts will use an internal interrupt line ($\overline{IRQ_2}$). This interrupt will operate the same as $\overline{IRQ_1}$ except that it will use the vector address of $FFF0 through $FFF7. $\overline{IRQ_1}$ will have priority to $\overline{IRQ_2}$ if both occur at the same time. The Interrupt Mask Bit in the condition code register masks both interrupts (See Table 1).

Table 1 Interrupt Vector Location

	Vector		Interrupt
	MSB	LSB	
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$
	FFF6	FFF7	ICF (Input Capture)
	FFF4	FFF5	OCF (Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

- **Read/Write (R/$\overline{W}$)**

This TTL compatible output signals the peripherals and memory devices whether the CPU is in a Read ("High") or a Write ("Low") state. The normal standby state of this signal is Read ("High"). This output can drive one TTL load and 90pF capacitance.

- **Address Strobe (AS)**

In the expanded multiplexed mode of operation, address strobe is output on this pin. This signal is used to latch the 8 LSB's of address which are multiplexed with data on D_0/A_0 to D_7/A_7. An 8-bit latch is utilized in conjunction with Address Strobe, as shown in figure 11. So D_0/A_0 to D_7/A_7 can become data bus during the E pulse. The timing for this signal is shown in Figure 1 of Bus Timing. This signal is also used to disable the address from the multiplexed bus allowing a deselect time, t_{ASD} before the data is enabled to the bus.

■ PORTS

There are two I/O ports on the HD6803 MPU; one 8-bit port and one 5-bit port. Each port has an associated write

only Data Direction Register which allows each I/O line to be programmed to act as an input or an output*. A "1" in the corresponding Data Direction Register bit will cause that I/O line to be an output. A "0" in the corresponding Data Direction Register bit will cause that I/O line to be an input. There are two ports: Port 1, Port 2. Their addresses and the addresses of their Data Direction registers are given in Table 2.

* The only exception is bit 1 of Port 2, which can either be data input or Timer output.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001

- **I/O Port 1**

This is an 8-bit port whose individual bits may be defined as inputs or outputs by the corresponding bit in its data direction register. The 8 output buffers have three-state capability, allowing them to enter a high impedance state when the peripheral data lines are used as inputs. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, these lines are TTL compatible and may also be used as a source of up to 1 mA at 1.5 V to directly drive a Darlington base. After reset, the I/O lines are configured as inputs.

- **I/O Port 2**

This port has five lines that may be defined as inputs or outputs by its data direction register. The 5 output buffers have three-state capability, allowing them to enter a high impedance state when used as an input. In order to be read properly, the voltage on the input lines must be greater than 2.0 V for a logic "1" and less than 0.8 V for a logic "0". As outputs, this port has no internal pullup resistors but will drive TTL inputs directly. For driving CMOS inputs, external pullup resistors are required. After reset, the I/O lines are configured as inputs. Three pins on Port 2 (pin 8, 9 and 10 of the chip) are requested to set following values (Table 3) during reset. The values of above three pins during reset are latched into the three MSBs (Bit 5, 6 and 7) of Port 2 which are read only.

Port 2 can be configured as I/O and provides access to the Serial Communications Interface and the Timer. Bit 1 is the only pin restricted to data input or Timer output.

Table 3 The Values of three pins

Pin Number	Value
8	L
9	H
10	L

[NOTES] L; Logical "0"
H; Logical "1"

■ BUS

- **Data/Address Lines ($D_0/A_0 \sim D_7/A_7$)**

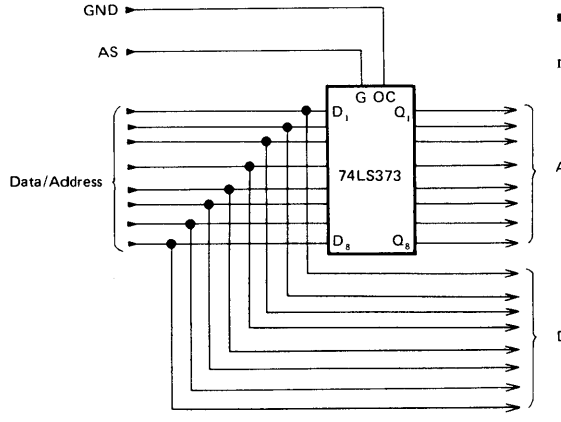
Since the data bus is multiplexed with the lower order address bus in Data/Address, latches are required to latch those address bits. The 74LS373 Transparent Octal D-type latch can be used with the HD6803 to latch the least significant address byte. Figure 11 shows how to connect the latch to the HD6803. The output control to the 74LS373 may be connected to ground.

- **Address Lines ($A_8 \sim A_{15}$)**

Each line is TTL compatible and can drive one TTL load and 90 pF. After reset, these pins become output for upper order address lines (A_8 to A_{15}).

■ INTERRUPT FLOWCHART

The Interrupt flowchart is depicted in Figure 16 and is common to every interrupt excluding reset.



Function Table

Output Control	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 11 Latch Connection

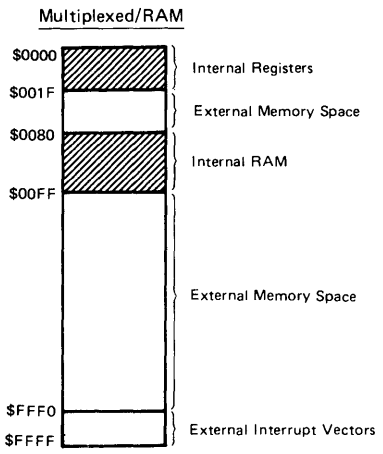
■ MEMORY MAP

The MPU can provide up to 65k byte address space. A memory map is shown in Figure 12. The first 32 locations are reserved for the MPU's internal register area, as shown in Table 4 with exceptions as indicated.

Table 4 Internal Register Area

Register	Address
Port 1 Data Direction Register**	00
Port 2 Data Direction Register**	01
Port 1 Data Register	02
Port 2 Data Register	03
Not Used	04*
Not Used	05*
Not Used	06*
Not Used	07*
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Not Used	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External Address
** 1; Output, 0; Input



[NOTE]
Excludes the following addresses which may be used externally: $04, $05, $06, $07, and $0F.

Figure 12 HD6803 Memory Map

■ PROGRAMMABLE TIMER

The HD6803 contains an on-chip 16-bit programmable timer which may be used to measure an input waveform while independently generating an output waveform. Pulse widths for both input and output signals may vary from a few microseconds to many seconds. The timer hardware consists of

- an 8-bit control and status register,
- a 16-bit free running counter,
- a 16-bit output compare register,
- a 16-bit input capture register

A block diagram of the timer registers is shown in Figure 13.

● Free Running Counter ($0009:$000A)

The key element in the programmable timer is a 16-bit free running counter which is driven to increasing values by E (Enable). The counter value may be read by the CPU software at any time. The counter is cleared to zero by reset and may be considered a read-only register with one exception. Any CPU write to the counter's address ($09) will always result in preset value of $FFF8 being loaded into the counter regardless of the value involved in the write. This preset figure is intended for testing operation of the part, but may be of value in some applications.

● Output Compare Register ($000B:$000C)

The Output Compare Register is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly compared with the current value of the free running counter. When a match is found, a flag is set (OCF) in the Timer Control and Status Register (TCSR) and the current value of the Output Level bit (OLVL) in the TCSR is clocked to the Output Level Register. Providing the Data Direction Register for Port 2, Bit 1 contains a "1" (Output), the output level register value will appear on the pin for Port 2 Bit 1. The values in the Output Compare Register and Output Level bit may then be changed to control the output level on the next compare value. The Output Compare Register is set to $FFFF during reset. The Compare function is inhibited for one cycle following a write to the high byte of the Output Compare Register to insure a valid 16-bit value is in the register before a compare is made.

● Input Capture Register ($000D:$000E)

The Input Capture Register is a 16-bit read-only register used to store the current value of the free running counter when the proper transition of an external input signal occurs. The input transition change required to trigger the counter transfer is controlled by the input Edge bit (IEDG) in the TCSR. The Data Direction Register bit for Port 2 Bit 0, should* be clear (zero) in order to gate in the external input signal to the edge detect unit in the timer.

The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

* With Port 2 Bit 0 configured as an output and set to "1", the external input will still be seen by the edge detect unit.

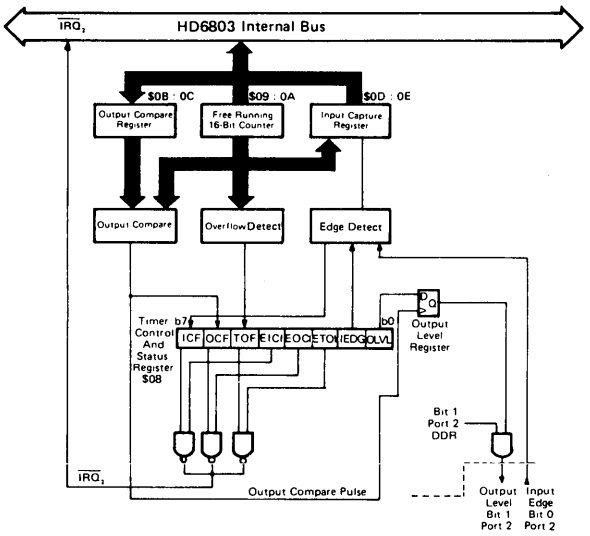


Figure 13 Block Diagram of Programmable Timer

Timer Control and Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

- **Timer Control and Status Register (TCSR) ($0008)**

The Timer Control and Status Register consists of an 8-bit register of which all 8 bits are readable but only the low order 5 bits may be written. The upper three bits contain read-only timer status information and indicate the followings:

- a proper transition has taken place on the input pin with a subsequent transfer of the current counter value to the input capture register.
- a match has been found between the value in the free running counter and the output compare register, and when $0000 is in the free running counter.

Each of the flags may be enabled onto the HD6803 internal bus ($\overline{IRQ_2}$) with an individual Enable bit in the TCSR. If the I-bit in the HD6803 Condition Code register has been cleared, a prior vectored interrupt will occur corresponding to the flag bit(s) set. A description for each bit follows:

Bit 0 **OLVL** Output Level – This value is clocked to the output level register on a successful output compare. If the DDR for Port 2 bit 1 is set, the value will appear on the output pin.

Bit 1 **IEDG** Input Edge – This bit controls which transition of an input will trigger a transfer of the counter to the input capture register. The DDR for Port 2 Bit 0 must be clear for this function to operate. IEDG = 0 Transfer takes place on a negative edge ("High"-to-"Low" transition).
IEDG = 1 Transfer takes place on a positive edge ("Low"-to-"High" transition).

Bit 2 **ETOI** Enable Timer Overflow Interrupt – When set, this bit enables $\overline{IRQ_2}$ to occur on the internal bus for a TOF interrupt; when clear the interrupt is inhibited.

Bit 3 **EOCI** Enable Output Compare Interrupt – When set, this bit enables $\overline{IRQ_2}$ to appear on the internal bus for an output compare interrupt; when clear the interrupt is inhibited.

Bit 4 **EICI** Enable input Capture Interrupt – When set, this bit enables $\overline{IRQ_2}$ to occur on the internal bus for an input capture interrupt; when clear the interrupt is inhibited.

Bit 5 **TOF** Timer Overflow Flag – This read-only bit is set when the counter contains $FFFF. It is cleared by a read of the TCSR (with TOF set) followed by an CPU read of the Counter ($09).

Bit 6 **OCF** Output Compare Flag – This read-only bit is set when a match is found between the output compare register and the free running counter. It is cleared by a read of the TCSR (with OCF set) followed by an CPU write to the output compare register ($0B or $0C).

Bit 7 **ICF** Input Capture Flag – This read-only status bit is set by a proper transition on the input; it is cleared by a read of the TCSR (with ICF set) followed by an CPU read of the Input Capture Register ($0D).

■ SERIAL COMMUNICATIONS INTERFACE

The HD6803 contains a full-duplex asynchronous serial communications interface (SCI) on chip. The controller comprises a transmitter and a receiver which operate independently or each other but in the same data format and at the same data rate. Both transmitter and receiver communicate with the CPU via the data bus and with the outside world via pins 2, 3, and 4 of Port 2. The hardware, software, and registers are explained in the following paragraphs.

• Wake-Up Feature

In a typical multi-processor application, the software protocol will usually contain a destination address in the initial byte(s) of the message. In order to permit non-selected MPU's to ignore the remainder of the message, a wake-up feature is included whereby all further interrupt processing may be optionally inhibited until the beginning of the next message. When the next message appears, the hardware re-enables (or "wakes-up") for the next message. The "wake-up" is automatically triggered by a string of ten consecutive 1's which indicates an idle transmit line. The software protocol must provide for the short idle period between any two consecutive messages.

• Programmable Options

The following features of the HD6803 serial I/O section are programmable:

- format – standard mark/space (NRZ)
- Clock – external or internal
- baud rate – one of 4 per given CPU ϕ_2 clock frequency or external clock ×8 input
- wake-up feature – enabled or disabled
- Interrupt requests – enabled or masked individually for transmitter and receiver data registers
- clock output – internal clock enabled or disabled to Port 2 (Bit 2)
- Port 2 (bits 3 and 4) – dedicated or not dedicated to serial I/O individually for transmitter and receiver.

• Serial Communications Hardware

The serial communications hardware is controlled by 4 registers as shown in Figure 14. The registers include:

- an 8-bit control and status register
- a 4-bit rate and mode control register (write only)
- an 8-bit read only receive data register and
- an 8-bit write only transmit data register.

In addition to the four registers, the serial I/O section utilizes bit 3 (serial input) and bit 4 (serial output) of Port 2. Bit 2 of Port 2 is utilized if the internal-clock-out or external-clock-in options are selected.

Transmit/Receive Control and Status (TRCS) Register

The TRCS register consists of an 8-bit register of which all 8 bits may be read while only bits 0~4 may be written. The register is initialized to $20 by reset. The bits in the TRCS register are defined as follows:

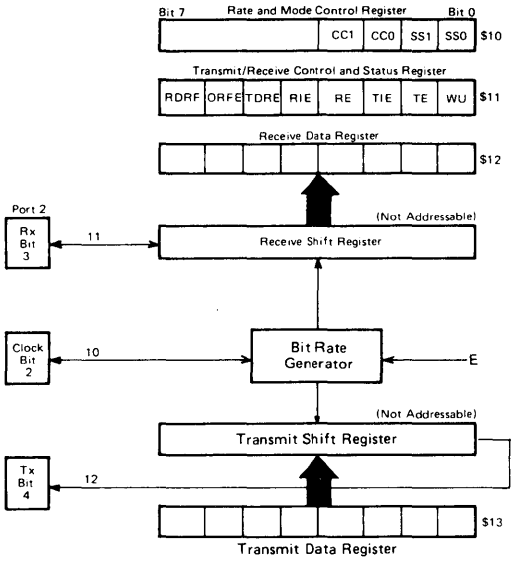


Figure 14 Serial I/O Registers

Bit 0 **WU** "Wake-up" on Next Message – set by HD6803 software and cleared by hardware on receipt of ten consecutive 1's or reset of RE flag. It should be noted that RE flag should be set in advance of CPU set of WU flag.

Bit 1 **TE** Transmit Enable – set by HD6803 to produce preamble of nine consecutive 1's and to enable gating of transmitter output to Port 2, bit 4 regardless of the DDR value corresponding to this bit; when clear, serial I/O has no effect on Port 2 bit 4.

TE set should be after at least one bit time of data transmit rate from the set-up of transmit data rate and mode.

Bit 2 **TIE** Transmit Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 5 (TDRE) is set; when clear, the TDRE value is masked from the bus.

Bit 3 **RE** Receiver Enable – when set, gates Port 2 bit 3 to input of receiver regardless of DDR value for this bit; when clear, serial I/O has no effect on Port 2 bit 3.

Bit 4 **RIE** Receiver Interrupt Enable – when set, will permit an $\overline{IRQ_2}$ interrupt to occur when bit 7 (RDRF) or bit 6 (ORFE) is set; when clear, the interrupt is masked.

Transmit/Receive Control and Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR : $0011

Bit 5 **TDRE** Transmit Data Register Empty – set by hardware when a transfer is made from the transmit data register to the output shift register. The TDRE bit is cleared by reading the status register, then writing a new byte into the transmit data register, TDRE is initialized to 1 by reset.

Bit 6 **ORFE** Over-Run-Framing Error – set by hardware when an overrun or framing error occurs (receive only).

Rate and Mode Control Register

7	6	5	4	3	2	1	0
X	X	X	X	CC1	CC0	SS1	SS0

ADDR : $0010

An overrun is defined as a new byte received with last byte still in Data Register/Buffer. A framing error has occured when the byte boundaries in bit stream are not synchronized to bit counter. If WU-flag is set, the ORFE bit will not be set. The ORFE bit is cleard by reading the status register, then reading the Receive Data Register, or by reset.

Bit 7 **RDRF** Receiver Data Register Full-set by hardware when a transfer from the input shift register to the receiver data register is made. If WU-flag is set, the RDRF bit will not be set. The RDRF bit is cleared by reading the status register, then reading the Receive Data Register, or by reset.

Rate and Mode Control Register (RMCR)

The Rate and Mode Control register controls the following serial I/O variables:

- Baud rate
- format
- clocking source,
- Port 2 bit 2 configuration

The register consists of 4 bits all of which are write-only and cleared by reset. The 4 bits in the register may be considered as a pair of 2-bit fields. The two low order bits control the bit rate for internal clocking and the remaining two bits control the format and clock select logic. The register definition is as follows:

Bit 0 **SS0** } Speed Select – These bits select the Baud rate for
Bit 1 **SS1** } the internal clock. The four rates which may be selected are a function of the CPU ϕ_2 clock frequency. Table 5 lists the available Baud rates.

Bit 2 **CC0** } Clock Control and Format Select – this 2-bit field
Bit 3 **CC1** } controls the format and clock select logic. Table 6 defines the bit field.

Table 5 SCI Bit Times and Rates

	XTAL	2.4576 MHz	4.0 MHz	4.9152 MHz*
SS1 : SS0	E	614.4 kHz	1.0 MHz	1.2288 MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13.0 μs/76,800 Baud
0 1	E ÷ 128	208 μs/4,800 Baud	128 μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	E ÷ 1024	1.67 ms/600 Baud	1.024 ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	E ÷ 4096	6.67 ms/150 Baud	4.096 ms/244.1 Baud	3.33 ms/300 Baud

* HD6803-1 Only

Table 6 SCI Format and Clock Source Control

CC1: CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS; bit 4 is used for serial output if TE = "1" in TRCS.

Internally Generated Clock

If the user wishes for the serial I/O to furnish a clock, the following requirements are applicable:

- the values of RE and TE are immaterial.
- CC1, CC0 must be set to 10
- the maximum clock rate will be E ÷ 16.
- the clock will be at 1x the bit rate and will have a rising edge at mid-bit.

Externally Generated Clock

If the user wishes to provide an external clock for the serial I/O, the following requirements are applicable:

- the CC1, CC0, field in the Rate and Mode Control Register must be set to 11,
- the external clock must be set to 8 times (×8) the desired baud rate and
- the maximum external clock frequency is 1.0 MHz.

- **Serial Operations**

The serial I/O hardware should be initialized by the HD6803 software prior to operation. This sequence will normally consist of;

- writing the desired operation control bits to the Rate and Mode Control Register and
- writing the desired operational control bits in the Transmit/Receive Control and Status Register.

The Transmitter Enable (TE) and Receiver Enable (RE) bits may be left set for dedicated operations.

Transmit Operations

The transmit operation is enabled by the TE bit in the Transmit/Receive Control and Status Register. This bit when set, gates the output of the serial transmit shift register to Port 2 Bit 4 and takes unconditional control over the Data Direction Register value for Port 2, Bit 4.

Following a $\overline{RES}$ the user should configure both the Rate and Mode Control Register and the Transmit/Receive Control and Status Register for desired operation. Setting the TE bit during this procedure initiates the serial output by first transmitting a nine-bit preamble of 1's. Following the preamble, internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or,
2) if data has been loaded into the Transmit Data Register (TDRE = 0), the word is transferred to the output shift register and transmission of the data word will begin.

During the data transmit, the 0 start bit is first transmitted. Then the 8 data bits (beginning with bit 0) followed by the stop bit, are transmitted. When the Transmitter Data Register has been emptied, the hardware sets the TDRE flag bit.

If the HD6803 fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the parallel data register to the serial output register should occur) then a 1 will be sent (instead of a 0) at "Start" bit time, followed by more 1's until more data is supplied to the data register. No 0's will be sent while TDRE remains a 1.

Receive Operation

The receive operation is enabled by the RE bit which gates in the serial input through Port 2, Bit 3. The receiver section operation is conditioned by the contents of the Transmit/Receive Control and Status Register and the Rate and Mode Control Register.

The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the NRZ Mode, the received bit stream is synchronized by the first 0 (space) encountered.

The approximate center of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit) a framing error is assumed, and bit ORFE is set. If the tenth bit as a 1, the data is transferred to the Receive Data Register, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an overrun has occurred. When the HD6803 responds to either flag (RDRF or ORFE) by reading the status register followed by reading the Data Register, RDRF (or ORFE) will be cleared.

■ RAM CONTROL REGISTER

This register, which is addressed at $0014, gives status information about the standby RAM. A 0 in the RAM enable bit (RAME) will disable the standby RAM, thereby protecting it at power down if V_{CC} Standby is held greater than V_{SBB} volts, as explained previously in the signal description for V_{CC} Standby.

RAM Control Register

$0014	STBY PWR	RAME	X	X	X	X	X	X

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 **RAME** The RAM Enable control bit allows the user the ability to disable the standby RAM. This bit is set to a logic "1" by $\overline{RES}$ which enables the standby RAM and can be written to one or zero under program control. When the RAM is disabled, data is read from external memory.
Bit 7 **STBY PWR** The Standby Power bit is cleared when the standby voltage is removed. This bit is a read/write status flag that the user can read which indicates that the standby RAM voltage has been applied, and the data in the standby RAM is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD6803 is upward object code compatible with the HD6800 as it implements the full HMCS6800 instruction set. The execution times of key instructions have been reduced to increase throughout. In addition, new instructions have been added; these include 16-bit operations and a hardware multiply.

Included in the instruction set section are the following:

- CPU Programming Model–Figure 15.
- Addressing modes
- Accumulator and memory instructions – Table 7
- New instructions
- Index register and stack manipulations instructions – Table 8
- Jump and branch instructions – Table 9
- Condition code register manipulation instructions – Table 10
- Instructions Execution times in machine cycles – Table 11
- Summary of cycle by cycle operation – Table 12
- Summary of undefined instructions – Table 13

- **CPU Programming Model**

The programming model for the HD6803 is shown in Figure 15. The double (D) accumulator is physically the same as the Accumulator A concatenated with the Accumulator B so that any operation using accumulator D will destroy information in A and B.

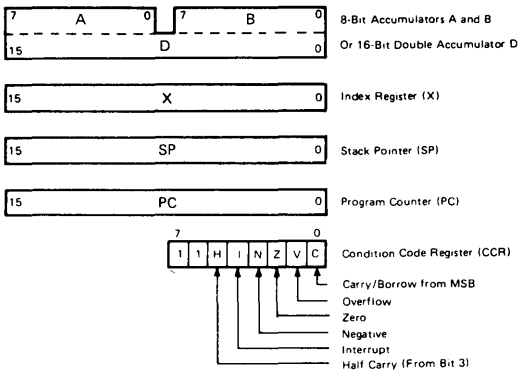


Figure 15 CPU Programming Model

- **CPU Addressing Modes**

The HD6803 8-bit micro processing unit has seven address modes that can be used by a programmer, with the addressing mode a function of both the type of instruction and the coding within the instruction. A summary of the addressing modes for a particular instruction can be found in Table 11 along with the associated instruction execution time that is given in machine cycles. With a clock frequency of 4 MHz, these times would be microseconds.

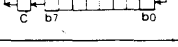
Accumulator (ACCX) Addressing

In accumulator only addressing, either accumulator A or accumulator B is specified. These are one-byte instructions.

Immediate Addressing

In immediate addressing, the operand is contained in the second byte of the instruction except LDS and LDX which have the operand in the second and third bytes of the instruction. The CPU addresses this location when it fetches the immediate instruction for execution. These are two or three-byte instructions.

Table 7 Accumulator & Memory Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A − M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B − M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	2	1	A − B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 − M → M	•	•	↕	↕	①	②
	NEGA													40	2	1	00 − A → A	•	•	↕	↕	①	②
	NEGB													50	2	1	00 − B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M − 1 → M	•	•	↕	↕	④	•
	DECA													4A	2	1	A − 1 → A	•	•	↕	↕	④	•
	DECB													5A	2	1	B − 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	10	1	A × B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Msp, SP − 1 → SP	•	•	•	•	•	•
	PSHB													37	3	1	B → Msp, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	4	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	4	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	ROLA													49	2	1		•	•	↕	↕	⑥	↕
	ROLB													59	2	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	RORA													46	2	1		•	•	↕	↕	⑥	↕
	RORB													56	2	1		•	•	↕	↕	⑥	↕

The Condition Code Register notes are listed after Table 10.

(Continued)

Table 7 Accumulator & Memory Instructions (Continued)

Operations	Mnemonic	Addressing Modes: IMMED. OP	IMMED. ~	IMMED. #	DIRECT OP	DIRECT ~	DIRECT #	INDEX OP	INDEX ~	INDEX #	EXTEND OP	EXTEND ~	EXTEND #	IMPLIED OP	IMPLIED ~	IMPLIED #	Boolean/ Arithmetic Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	2	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	2	1	B	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	3	1	ACC A/ ACC B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	2	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	2	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	2	1	A	•	•	R	↕	⑥	↕
	LSRB													54	2	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	3	1	ACC A/ ACC B	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A – M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B – M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				A : B – M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	2	1	A – B → A	•	•	↕	↕	↕	↕
Subtract, With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A – M – C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B – M – C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	6	2	7D	6	3				M – 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A – 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B – 00	•	•	↕	↕	R	R

The Condition Code Register notes are listed after Table 10.

Direct Addressing

In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in the machine i.e., locations zero through 255. Enhanced execution times are achieved by storing data in these locations. In most configurations, it should be a random access memory. These are two-byte instructions.

Extended Addressing

In extended addressing, the address contained in the second byte of the instruction is used as the higher 8-bits of the address of the operand. The third byte of the instruction is used as the lower 8-bits of the address for the operand. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In indexed addressing, the address contained in the second byte of the instruction is added to the index register's lowest 8-bits in the CPU. The carry is then added to the higher order 8-bits of the index register. This result is then used to address memory. The modified address is held in a temporary address register so there is no change to the index register. These are two-byte instructions.

Implied Addressing

In the implied addressing mode the instruction gives the address (i.e., stack pointer, index register, etc.). These are one-byte instructions.

Relative Addressing

In relative addressing, the address contained in the second byte of the instruction is added to the program counter's lowest 8-bits plus two. The carry or borrow is then added to the high 8-bits. This allows the user to address data within a range of -126 to +129 bytes of the present instruction. These are two-byte instructions.

- **New Instructions**

In addition to the existing 6800 Instruction Set, the following new instructions are incorporated in the HD6803 Microcomputer.

ABX Adds the 8-bit unsigned accumulator B to the 16-bit X-Register taking into account the possible carry out of the low order byte of the X-Register.

ADDD Adds the double precision ACCD* to the double precision value M:M+1 and places the results in ACCD.

ASLD Shifts all bits of ACCD one place to the left. Bit 0 is loaded with zero. The C bit is loaded from the most significant bit of ACCD.

LDD Loads the contents of double precision memory location into the double accumulator A:B. The condition codes are set according to the data.

LSRD Shifts all bits of ACCD one place to the right. Bit 15 is loaded with zero. The C bit is loaded from the least significant bit to ACCD.

MUL Multiplies the 8 bits in accumulator A with the 8 bits in accumulator B to obtain a 16-bit unsigned number in A:B, ACCA contains MSB of result.

PSHX The contents of the index register is pushed onto the stack at the address contained in the stack pointer. The stack pointer is decremented by 2.

PULX The index register is pulled from the stack beginning at the current address contained in the stack pointer +1. The stack pointer is incremented by 2 in total.

STD Stores the contents of double accumulator A:B in memory. The contents of ACCD remain unchanged.

SUBD Subtracts the contents of M:M + 1 from the contents of double accumulator AB and places the result in ACCD.

BRN Never branches. If effect, this instruction can be considered a two byte NOP (No operation) requiring three cycles for execution.

CPX Internal processing modified to permit its use with any conditional branch instruction.

*ACCD is the 16 bit register (A:B) formed by concatenating the A and B accumulators. The A-accumulator is the most significant byte.

Table 8 Index Register and Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M : M + 1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	3	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	3	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				$M \rightarrow X_H$, $(M+1) \rightarrow X_L$	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				$M \rightarrow SP_H$, $(M+1) \rightarrow SP_L$	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				$X_H \rightarrow M$, $X_L \rightarrow (M+1)$	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				$SP_H \rightarrow M$, $SP_L \rightarrow (M+1)$	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	3	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	$X_L \rightarrow M_{sp}$, SP − 1 → SP $X_H \rightarrow M_{sp}$, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, $M_{sp} \rightarrow X_H$ SP + 1 → SP, $M_{sp} \rightarrow X_L$	•	•	•	•	•	•

The Condition Code Register notes are listed after Table 10.

Table 9 Jump and Branch Instructions

Operations	Mnemonic	Addressing Modes														Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	6	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		—— ⑧ ——					
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	⑨	•	•	•	•

Table10 Condition Code Register Manipulation Instructions

Operations	Mnemonic	AddressingModes			Boolean Operation	Condition Code Register					
		IMPLIED				5	4	3	2	1	0
		OP	~	#		H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	—— ⑩ ——					
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

Condition Code Register Notes: (Bit set it test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result = 00000000?
③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to result of N ⊕ C after shift has occurred.
⑦ (Bit N) Test: Result less than zero? (Bit 15 = 1)
⑧ (All) Load Condition Code Register from Stack. (See Special Operations)
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All) Set according to the contents of Accumulator A.
⑪ (Bit C) Set equal to result of Bit 7 (ACCB)

Table 11 Instruction Execution Times in Machine Cycle

	ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative		ACCX	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA	•	•	•	•	•	2	•	INX	•	•	•	•	•	3	•
ABX	•	•	•	•	•	3	•	JMP	•	•	•	3	3	•	•
ADC	•	2	3	4	4	•	•	JSR	•	•	5	6	6	•	•
ADD	•	2	3	4	4	•	•	LDA	•	2	3	4	4	•	•
ADDD	•	4	5	6	6	•	•	LDD	•	3	4	5	5	•	•
AND	•	2	3	4	4	•	•	LDS	•	3	4	5	5	•	•
ASL	2	•	•	6	6	•	•	LDX	•	3	4	5	5	•	•
ASLD	•	•	•	•	•	3	•	LSR	2	•	•	6	6	•	•
ASR	2	•	•	6	6	•	•	LSRD	•	•	•	•	•	3	•
BCC	•	•	•	•	•	•	3	MUL	•	•	•	•	•	10	•
BCS	•	•	•	•	•	•	3	NEG	2	•	•	6	6	•	•
BEQ	•	•	•	•	•	•	3	NOP	•	•	•	•	•	2	•
BGE	•	•	•	•	•	•	3	ORA	•	2	3	4	4	•	•
BGT	•	•	•	•	•	•	3	PSH	3	•	•	•	•	•	•
BHI	•	•	•	•	•	•	3	PSHX	•	•	•	•	•	4	•
BIT	•	2	3	4	4	•	•	PUL	4	•	•	•	•	•	•
BLE	•	•	•	•	•	•	3	PULX	•	•	•	•	•	5	•
BLS	•	•	•	•	•	•	3	ROL	2	•	•	6	6	•	•
BLT	•	•	•	•	•	•	3	ROR	2	•	•	6	6	•	•
BMI	•	•	•	•	•	•	3	RTI	•	•	•	•	•	10	•
BNE	•	•	•	•	•	•	3	RTS	•	•	•	•	•	5	•
BPL	•	•	•	•	•	•	3	SBA	•	•	•	•	•	2	•
BRA	•	•	•	•	•	•	3	SBC	•	2	3	4	4	•	•
BRN	•	•	•	•	•	•	3	SEC	•	•	•	•	•	2	•
BSR	•	•	•	•	•	•	6	SEI	•	•	•	•	•	2	•
BVC	•	•	•	•	•	•	3	SEV	•	•	•	•	•	2	•
BVS	•	•	•	•	•	•	3	STA	•	•	3	4	4	•	•
CBA	•	•	•	•	•	2	•	STD	•	•	4	5	5	•	•
CLC	•	•	•	•	•	2	•	STS	•	•	4	5	5	•	•
CLI	•	•	•	•	•	2	•	STX	•	•	4	5	5	•	•
CLR	2	•	•	6	6	•	•	SUB	•	2	3	4	4	•	•
CLV	•	•	•	•	•	2	•	SUBD	•	4	5	6	6	•	•
CMP	•	2	3	4	4	•	•	SWI	•	•	•	•	•	12	•
COM	2	•	•	6	6	•	•	TAB	•	•	•	•	•	2	•
CPX	•	4	5	6	6	•	•	TAP	•	•	•	•	•	2	•
DAA	•	•	•	•	•	2	•	TBA	•	•	•	•	•	2	•
DEC	2	•	•	6	6	•	•	TPA	•	•	•	•	•	2	•
DES	•	•	•	•	•	3	•	TST	2	•	•	6	6	•	•
DEX	•	•	•	•	•	3	•	TSX	•	•	•	•	•	3	•
EOR	•	2	3	4	4	•	•	TXS	•	•	•	•	•	3	•
INC	2	•	•	6	6	•	•	WAI	•	•	•	•	•	9	•
INS	•	•	•	•	•	3	•								

- **Summary of Cycle by Cycle Operation**

Table 12 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write line (R/$\overline{W}$) during each cycle for each instruction.

This information is useful in comparing actual with expected results during debug of both software and hardware as the control program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. (In general, instructions with the same addressing mode and number of cycles execute in the same manner; exceptions are indicated in the table).

Table 12 Cycle by Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Operand Data
LDS LDX LDD	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 2 Address Bus FFFF	1 1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	Op Code Address of Operand Operand Data
STA	3	1 2 3	Op Code Address Op Code Address + 1 Destination Address	1 1 0	Op Code Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Op Code Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Subroutine Address Stack Pointer Stack Pointer + 1	1 1 1 0 0	Op Code Irrelevant Data First Subroutine Op Code Return Address (Low Order Byte) Return Address (High Order Byte)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
LDD		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

* In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

* In the TST instruction, R/$\overline{W}$ line of the sixth cycle is "1" level, and AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMPLIED					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address + 1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)
WAI**	9	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer – 1	1 1 0 0	Op Code Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

(Continued)

Table 12 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
WAI**		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)

(Continued)

** While the MPU is in the "Wait" state, its bus state will appear as a series of MPU reads of an address which is seven locations less than the original contents of the Stack Pointer. Contrary to the HD6800, none of the ports are driven to the high impedance state by a WAI instruction.

Table 12 Cycle by Cycle Operation (Continued)

RELATIVE

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
BCC BHT BNE BCS BLE BPL BEQ BLS BRA BGE BLT BVC BGT BMT BVS BRN	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Branch Offset Low Byte of Restart Vector
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Address Bus FFFF Subroutine Starting Address Stack Pointer Stack Pointer – 1	1 1 1 1 0 0	Op Code Branch Offset Low Byte of Restart Vector Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

- **Summary of Undefined Instruction Operations**

The HD6803 has 36 underfined instructions. When these are carried out, the contents of Register and Memory in MPU change at random.

When the op codes (4E, 5E) are used to execute. the MPU continues to increase the program counter and it will not stop until the Reset signal enters. These op codes are used to test the LSI.

Table 13 Op codes Map

HD6803 MICROPROCESSOR INSTRUCTIONS

OP CODE HI / LO		0000	0001	0010	0011	ACC A 0100	ACC B 0101	IND 0110	EXT 0111	ACCA or SP IMM 1000	ACCA or SP DIR 1001	ACCA or SP IND 1010	ACCA or SP EXT 1011	ACCB or X IMM 1100	ACCB or X DIR 1101	ACCB or X IND 1110	ACCB or X EXT 1111	
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS					CMP								1
0010	2			BHI	PULA (+1)					SBC								2
0011	3			BLS	PULB (+1)	COM				•	SUBD (+2)			•	ADDD (+2)			3
0100	4	LSRD (+1)		BCC	DES	LSR				AND								4
0101	5	ASLD (+1)		BCS	TXS					BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX (+1)		BVC	PULX (+2)	ASL				EOR								8
1001	9	DEX (+1)	DAA	BVS	RTS (+2)	ROL				ADC								9
1010	A	CLV		BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI (+7)					ADD								B
1100	C	CLC		BGE	PSHX (+1)	INC				•	CPX (+2)			•	LDD (+1)			C
1101	D	SEC		BLT	MUL (+7)	TST				BSR (+4)	JSR (+2)			• (+1)	STD (+1)			D
1110	E	CLI		BGT	WAI (+6)	••		JMP (-3)		•	LDS (+1)			•	LDX (+1)			E
1111	F	SEI		BLE	SWI (+9)	CLR				• (+1)	STS (+1)			• (+1)	STX (+1)			F
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4	

[NOTES] 1) Undefined Op codes are marked with ▨.

2) ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

3) The instructions shown below are all 3 bytes and are marked with "•".
Immediate addressing mode of SUBD, CPX, LDS, ADDD, LDD and LDX instructions, and undefined op codes (8F, CD, CF).

4) The Op codes (4E, 5E) are 1 byte/∞ cycles instructions, and are marked with "••"

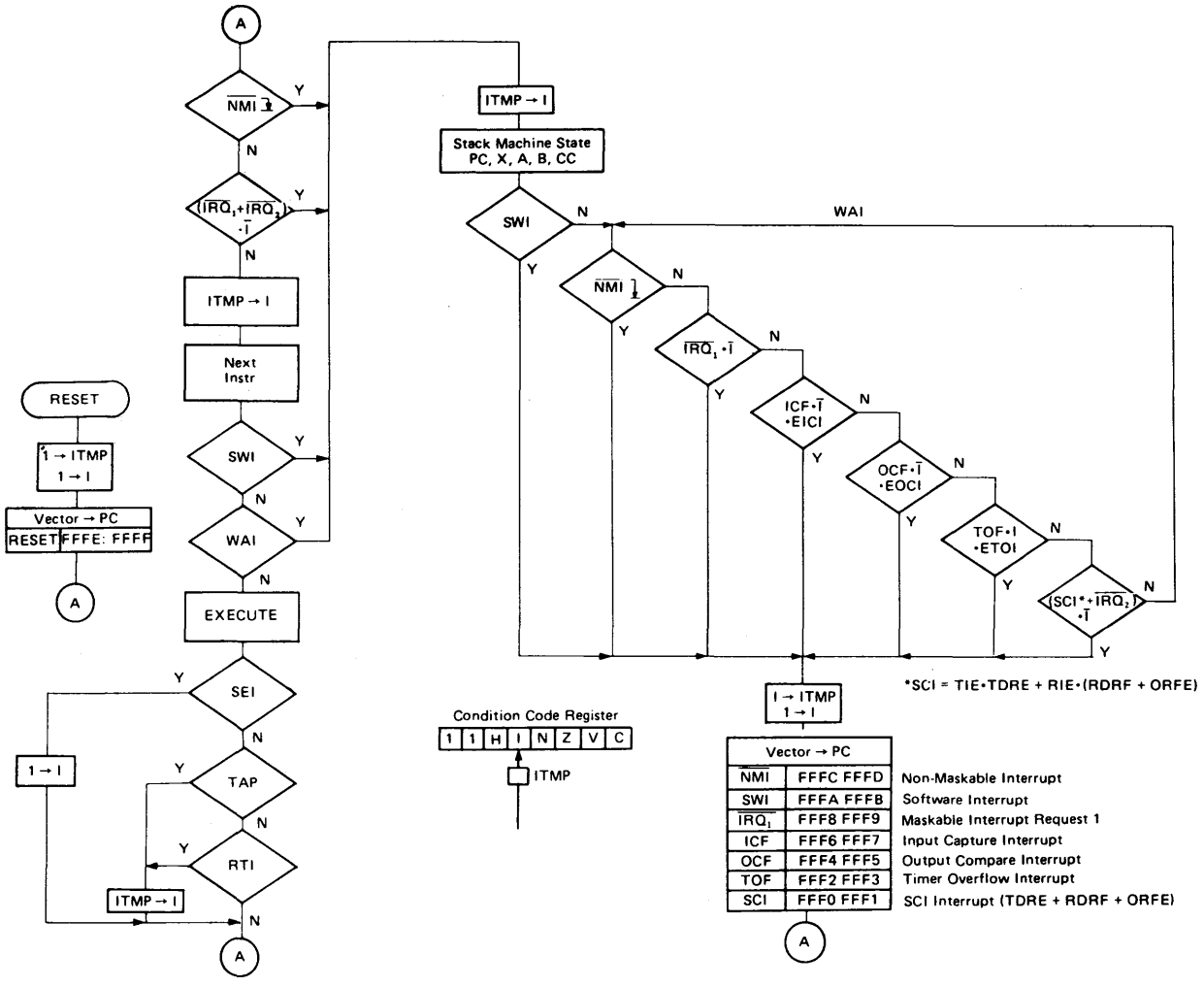


Figure 16 Interrupt Flowchart

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

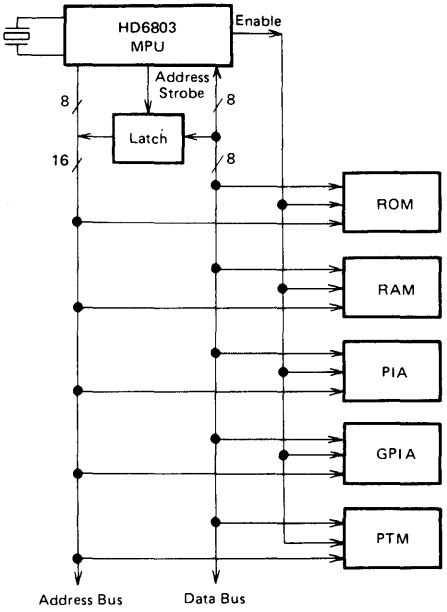


Figure 17 HD6803 MPU Expanded Multiplexed Bus

■ **Caution for the HD6803 Family SCI, TIMER Status Flag**

The flags shown in Table 14 are cleared by reading/writing (flag reset condition 2) the data register corresponding to each flag after reading the status register (flag reset condition 1).

To clear the flag correctly, take the following procedure:

1. Read the status register.
2. Test the flag.
3. Read the data register.

Table 14 Status Flag Reset Conditions

	Status Flag	Flag Reset Condition 1 (Status Register)	Flag Reset Condition 2 (Data Register)
TIMER	ICF	When each flag is "1", TRCSR/Read	ICR/Read
	OCF		OCR/Write
	TOF		TC/Read
SCI	RDRF	When each flag is "1", TRCSR/Read	RDR/Read
	ORFE		
	TDRE		TDR/Write

HD6805S1

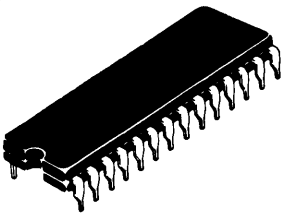
MCU (Microcomputer Unit)

The HD6805S1 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, I/O and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD 6800-based instruction set.

The following are some of the hardware and software highlights of the MCU.

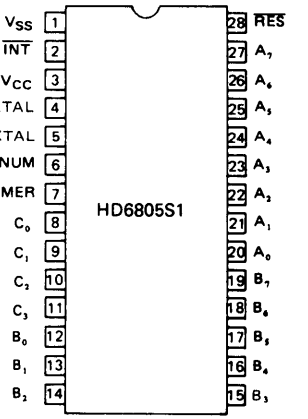
- **HARDWARE FEATURES**
- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 1100 Bytes of User ROM
- Internal 8-Bit Timer with 7-Bit Prescaler
- Vectored Interrupts – External and Timer
- 20 TTL/CMOS Compatible I/O Lines; 8 Lines LED Compatible
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- Easy for System Development and Debugging
- 5 Vdc Single Supply
- Compatible with MC6805P2
- **SOFTWARE FEATURES**
- Similar to HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible with MC6805P2

HD6805S1P



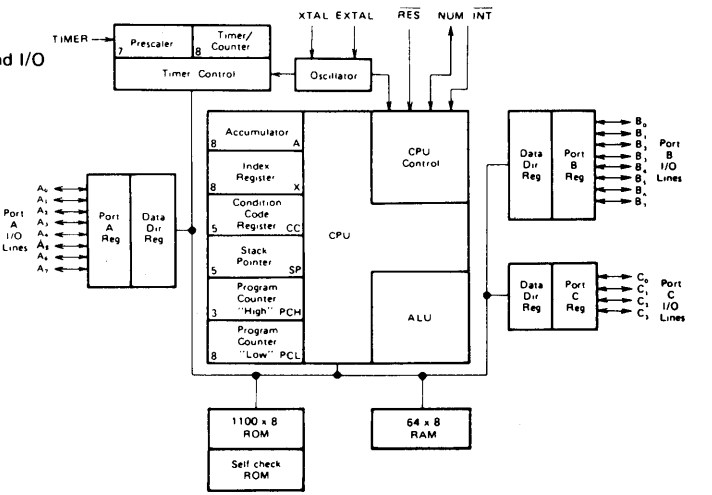
(DP-28)

■ **PIN ARRANGEMENT**



(Top View)

■ **BLOCK DIAGRAM**



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in} *	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +12.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0~+70°C, unless otherwise noted.)

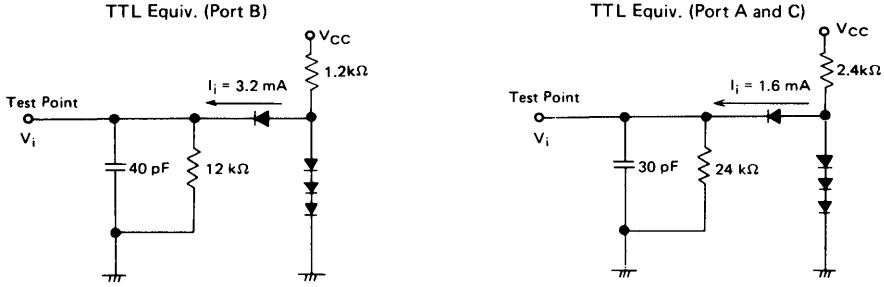
Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT}$			3.0	–	V_{CC}	V
	All Other			2.0	–	V_{CC}	V
Input "High" Voltage(Timer)	Timer Mode			2.0	–	V_{CC}	V
	Self-Check Mode			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT}$			-0.3	–	0.8	V
	EXTAL(Crystal Mode)			-0.3	–	0.6	V
	All Other			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	700	mW
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT}$			-50	–	50	μA
	EXTAL(Crystal Mode)			-1200	–	0	μA

• AC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
Oscillation Frequency (External Resistor Mode)		f_{EXT}	R_{CP}=15.0kΩ±1%	–	3.4	–	MHz
$\overline{INT}$ Pulve Width		t_{IWL}		$\frac{t_{cyc}}{250}$+	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		$\frac{t_{cyc}}{250}$+	–	–	ns
TIMER Pulse Width		t_{TWL}		$\frac{t_{cyc}}{250}$+	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20%, R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2 μF	100	–	–	ms
Input Capacitance	XTAL	C_{in}	V_{in}=0V	–	–	30	pF
	All Other			–	–	10	pF

● PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = −10 μA	3.5	–	–	V
			I_{OH} = −100 μA	2.4	–	–	V
	Port B		I_{OH} = −200 μA	2.4	–	–	V
			I_{OH} = −1 mA	1.5	–	–	V
	Port C		I_{OH} = −100 μA	2.4	–	–	V
Output "Low" Voltage	Port A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.4	V
	Port B		I_{OL} = 3.2 mA	–	–	0.4	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Port A, B, C	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	−500	–	–	μA
			V_{in} = 2V	−300	–	–	μA
	Port B, C		V_{in} = 0.4V ~ V_{CC}	− 20	–	20	μA



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074 (H) or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

● V_{CC} and V_{SS}

Power is supplied to the MCU using these two pins. V_{CC} is +5.25 V ±0.5 V. V_{SS} is the ground connection.

● $\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

● XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4 MHz maximum), a resistor or an external signal can be connected to these pins to provide a system clock with various stability/cost tradeoffs. Refer to INTERNAL OSCILLATOR OPTIONS for recommendations about these inputs.

● TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER for additional information about the timer circuitry.

● $\overline{RES}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to RESETS for additional information.

● NUM

This pin is not for user application and should be connected to V_{SS}.

● Input/Output Lines (A_0 ~ A_7, B_0 ~ B_7, C_0 ~ C_3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the Data Direction Registers (DDR). Refer to INPUT/OUTPUT for additional information.

■ MEMORY

The MCU memory is configured as shown in Figure 2. During the processing of an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Figure 3. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

Caution: – Self Test ROM Address Area
Self test ROM locations can not be used for a user program. If the user's program is in this location, it will be removed when manufacturing mask for production.

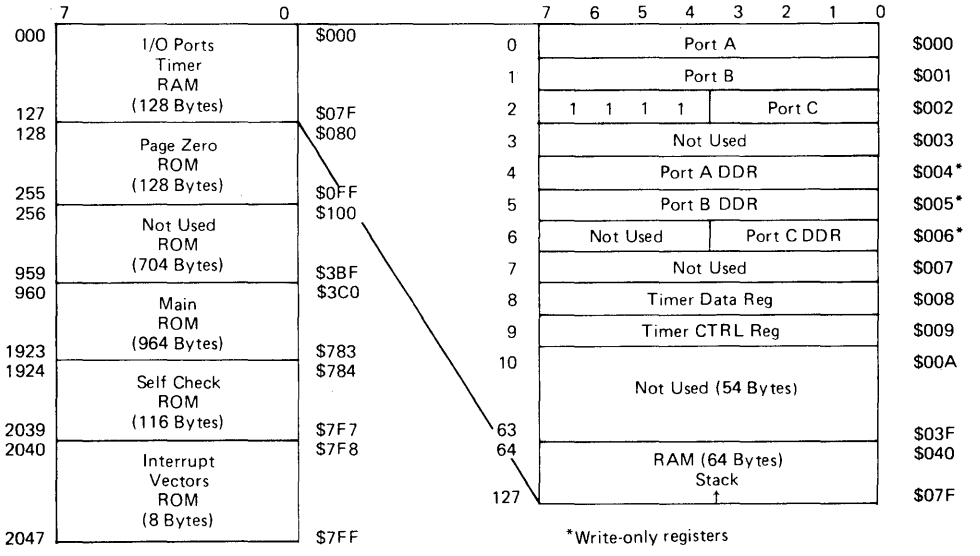


Figure 2 MCU Memory Configuration

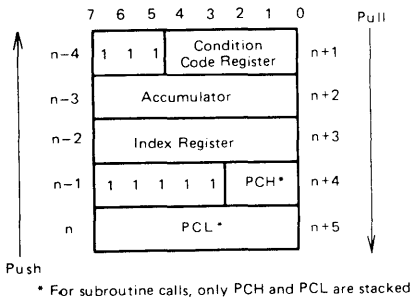


Figure 3 Interrupt Stacking Order

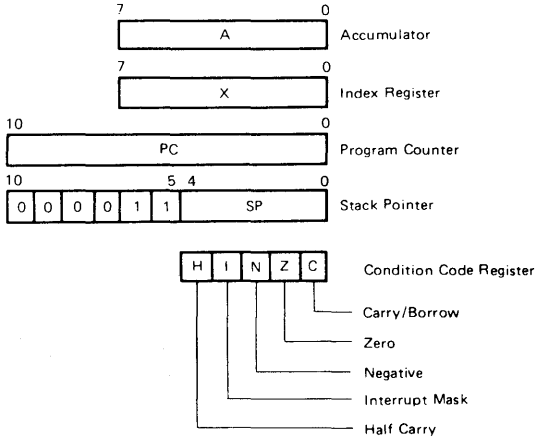


Figure 4 Programming Model

■ REGISTERS

The CPU has five registers available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

- **Accumulator (A)**

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

- **Index Register (X)**

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

- **Program Counter (PC)**

The program counter is an 11-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The six most significant bits of the stack pointer are permanently set to 000011. During a MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 which allows the programmer to use up to 15 levels of subroutine calls.

- **Condition Code Register (CC)**

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

Used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in result equal to a logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

■ TIMER

The MCU timer circuitry is shown in Figure 5. The 8-bit counter, the Timer Data Register (TDR), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR) is set. the CPU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the Condition Code Register also prevents a time interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When the internal ϕ_2 signal is selected as the input source, the node a is connected to b (see Fig. 5).

In case of the external source, the node b connects with c.

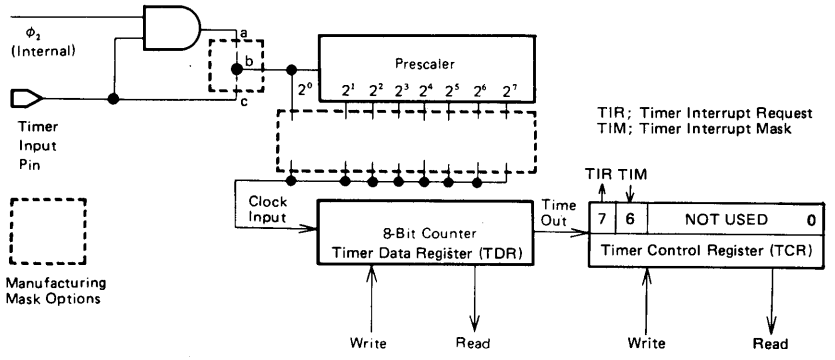


Figure 5 Timer Block Diagram

When the ϕ_2 signal is used as the source, the clock signal is input to the prescaler while the TIMER input is "High".

The source of the clock input is one of the options that has to be specified before manufacture of the MCU. A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter (TDR). The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter (TDR) can be read at any time by monitoring the TDR. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones; the timer interrupt request bit (bit 7) is cleared, and the timer interrupt mask bit (bit 6) is set.

(NOTE) If the MCU Timer is not used, the TIMER input pin must be grounded.

■ SELF CHECK

The self-check capability of the MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 6 and monitor the output of port C bit 3 for an oscillation of approximately 3Hz. ROM, RAM, TIMER, Interrupts, I/O of Port A, B and C are checked by this capability.

■ RESETS

The MCU can be reset three ways; by initial power-up, by the external reset input ($\overline{RES}$) and by an optional internal low voltage inhibit circuit, see Figure 7. All the I/O port are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum of 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High".

This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 8, typically provides sufficient delay.

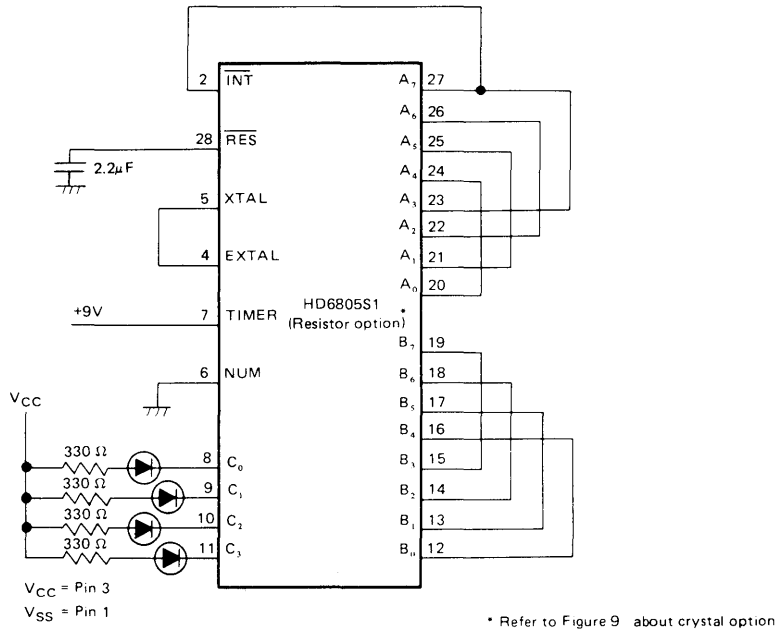


Figure 6 Self Check Connections

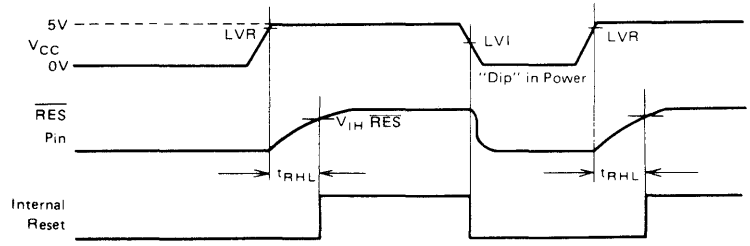


Figure 7 Power Up and $\overline{RES}$ Timing

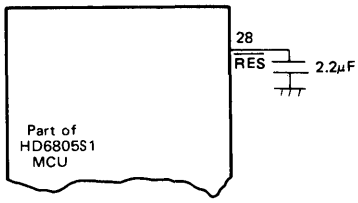


Figure 8 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR OPTIONS

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoff. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different connection methods are shown in Figure 9. Crystal specifications are given in Figure 10. A resistor selection graph is given in Figure 11.

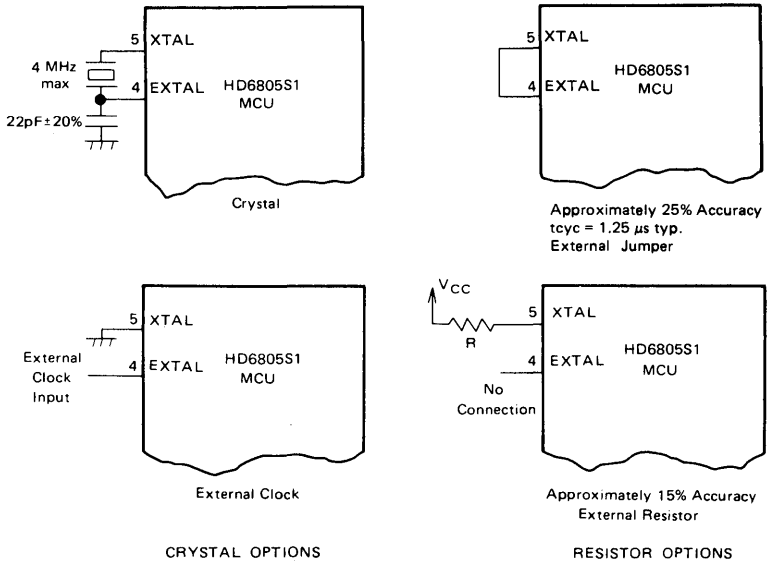
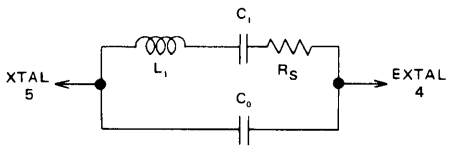


Figure 9 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz (C_1 =22pF±20%)
R_S = 60 Ω max.

Figure 10 Crystal Parameters

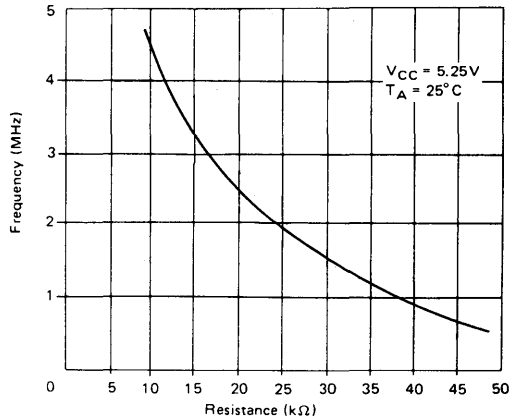


Figure 11 Typical Resistor Selection Graph

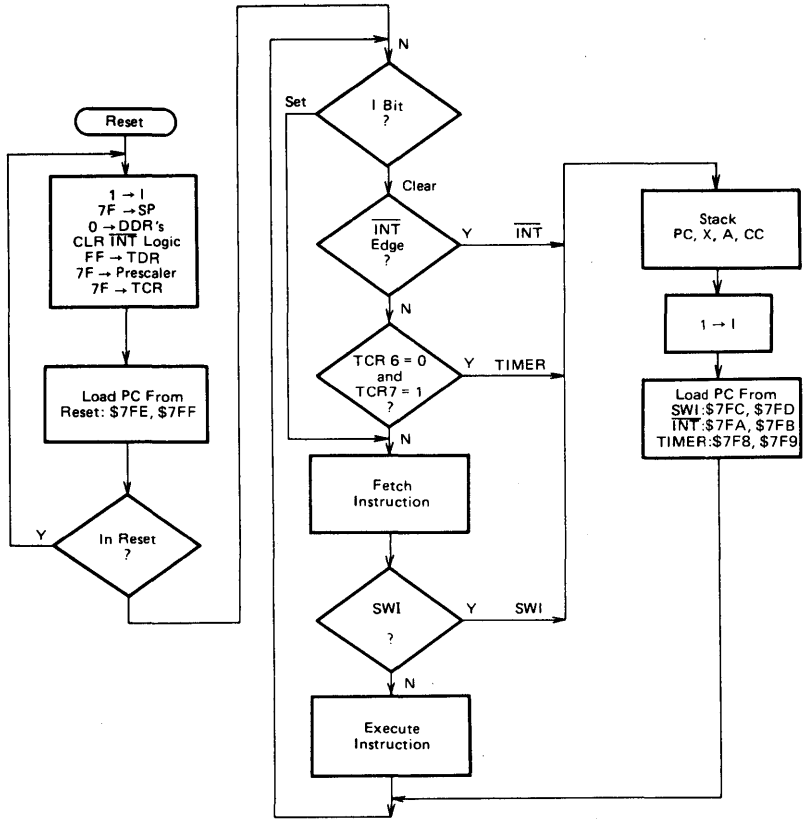
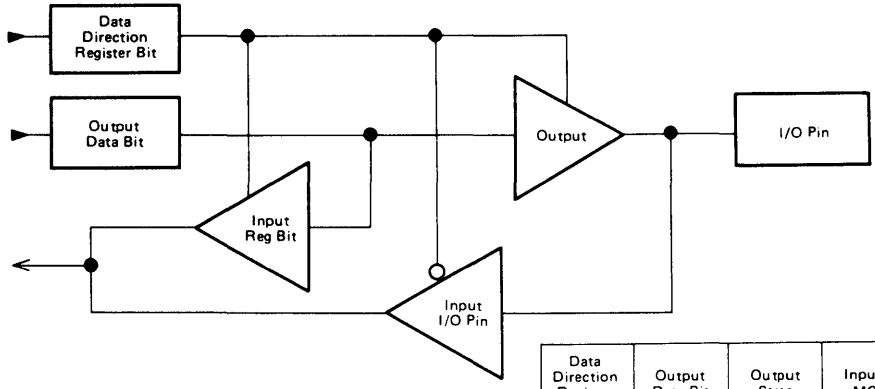


Figure 12 Interrupt Processing Flowchart



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

Figure 13 Typical Port I/O Circuitry

■ INTERRUPTS

The CPU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires 11 cycles. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. Table 1 provides a listing of the interrupts, their priority, and the vector address that contain-the starting address of the appropriate interrupt routine.

A flowchart of the interrupt processing sequence is given in Figure 12.

Table 1 Interrupt Priorities

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$7FE and $7FF
SWI	2	$7FC and $7FD
$\overline{INT}$	3	$7FA and $7FB
TIMER	4	$7F8 and $7F9

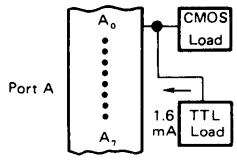
■ INPUT/OUTPUT

There are 20 input/output pins. All pins are programmable as either inputs or outputs under software control of the corresponding Data Direction Register (DDR). When programmed as outputs, all I/O pins the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (see Figure 13). When port B is programmed for outputs, it is capable of sinking 10mA on each pin (V_{OL} = 1V max). All input/output lines are TTL compatible as both inputs and outputs. Port A is CMOS compatible as outputs, and Port B and C are CMOS compatible as inputs. Figure 14 provides some examples of port connections.

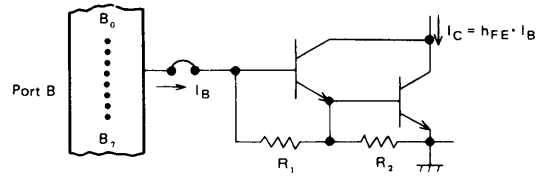
■ BIT MANIPULATION

The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 15 illustrates the usefulness of the bit manipulation and test instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which powers the controlled hardware.

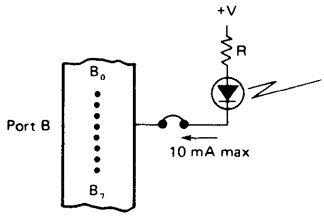
This program, which uses only seven ROM locations, provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer could also be incorporated to provide turn-on at some later time which would permit pulse-width modulation of the controlled power.



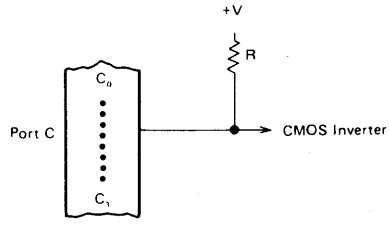
Port A Programmed as output(s), driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s), driving Darlington-base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)



Port C Programmed as output(s), driving CMOS loads, using external pull-up resistors. (d)

Figure 14 Typical Port Connections

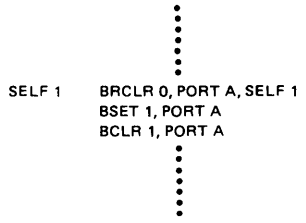


Figure 15 Bit Manipulation Example

■ ADDRESSING MODES

The CPU has ten addressing modes available for use by the programmer. They are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 16. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 17. In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 18. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 19. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA=(PC)+2+Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken Rel=0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 20. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 21. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 22. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 23. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 24. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($00-$FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 25. The implied mode of addressing has no EA. All the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are one byte long.

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

- **Register/Memory Instructions**

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 2.

- **Read/Modity/Write Instructions**

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 3.

- **Branch Instructions**

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

- **Bit Manipulation Instructions**

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 5.

- **Control Instructions**

The control instructions control the MCU operations during program execution. Refer to Table 6.

- **Alphabetical Listing**

The complete instruction set is given in alphabetical order in Table 7.

- **Opcode Map**

Table 8 is an opcode map for the instructions used on the MCU.

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

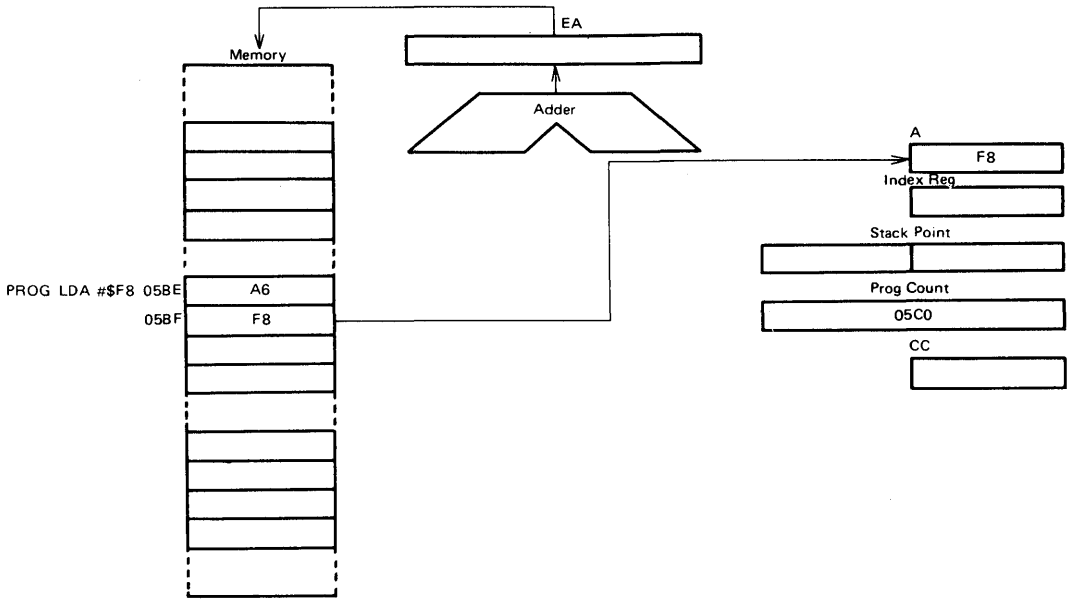


Figure 16 Immediate Addressing Example

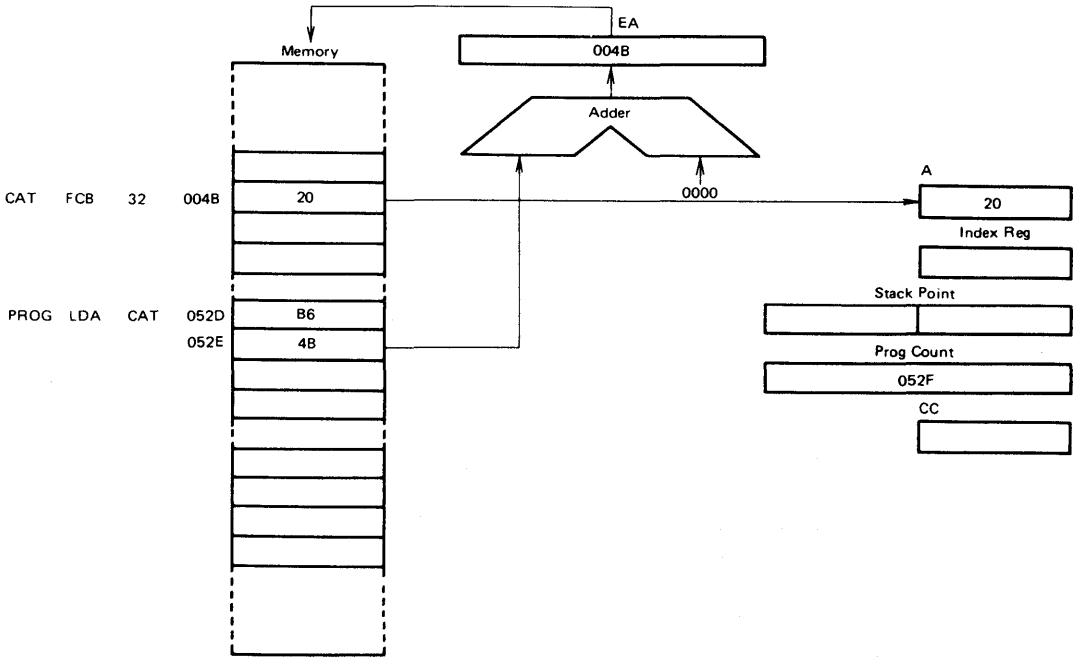


Figure 17 Direct Addressing Example

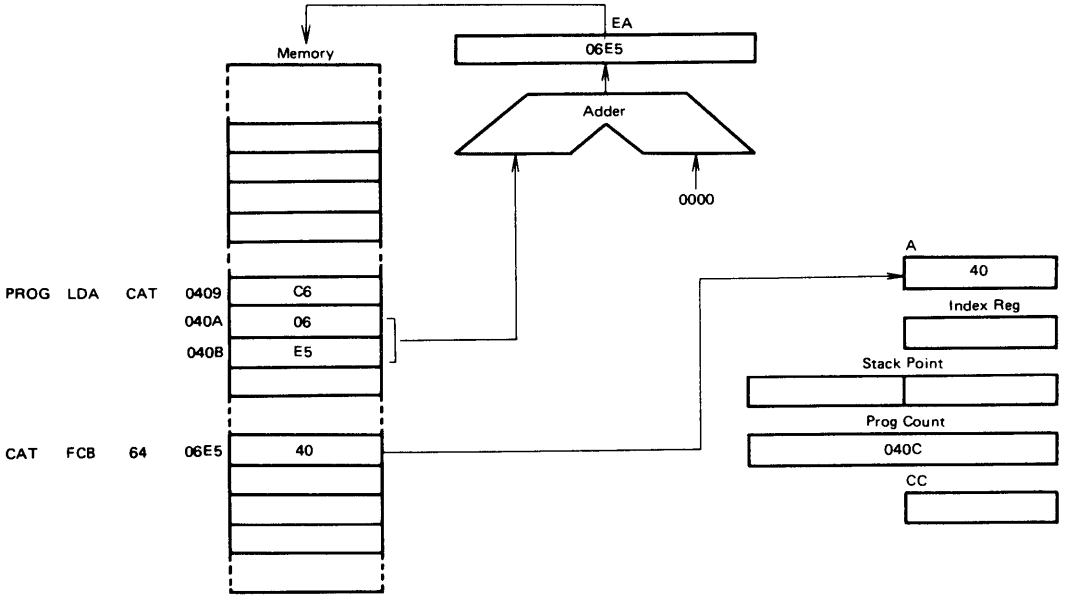


Figure 18 Extended Addressing Example

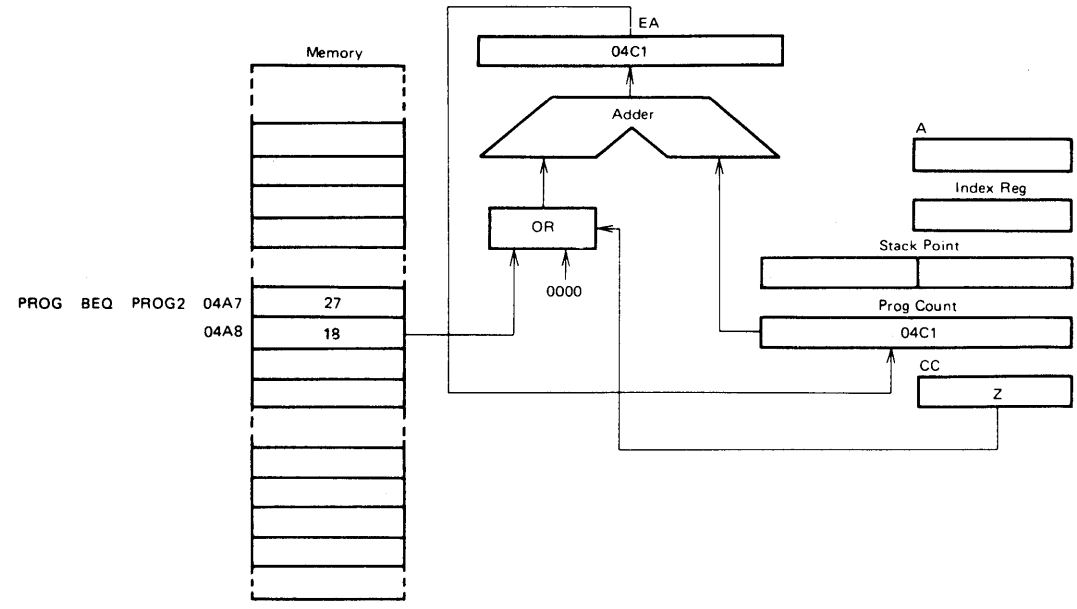


Figure 19 Relative Addressing Example

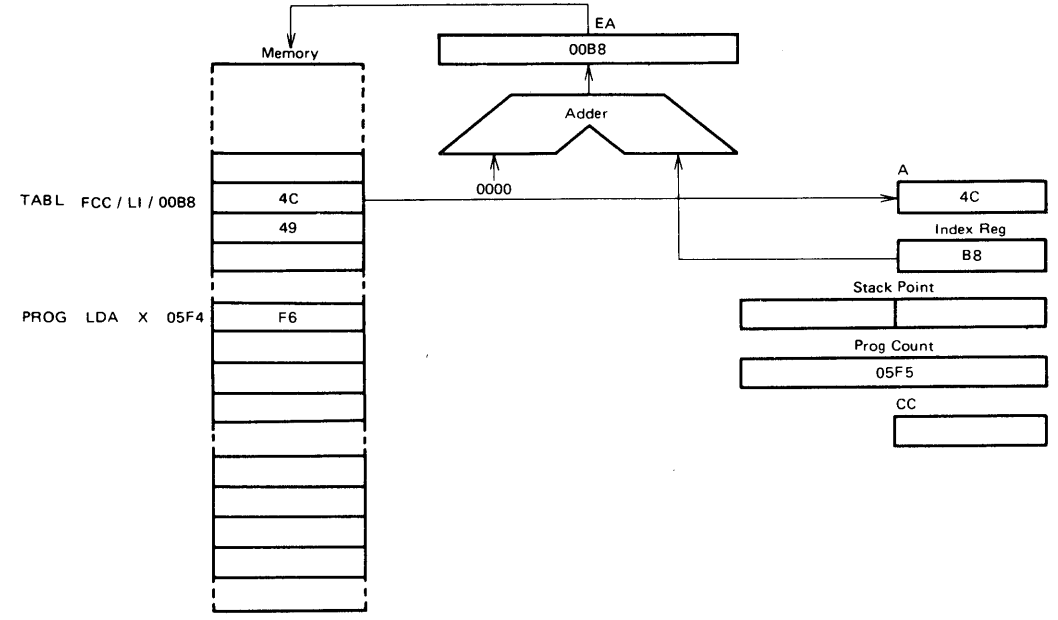


Figure 20 Indexed (No Offset) Addressing Example

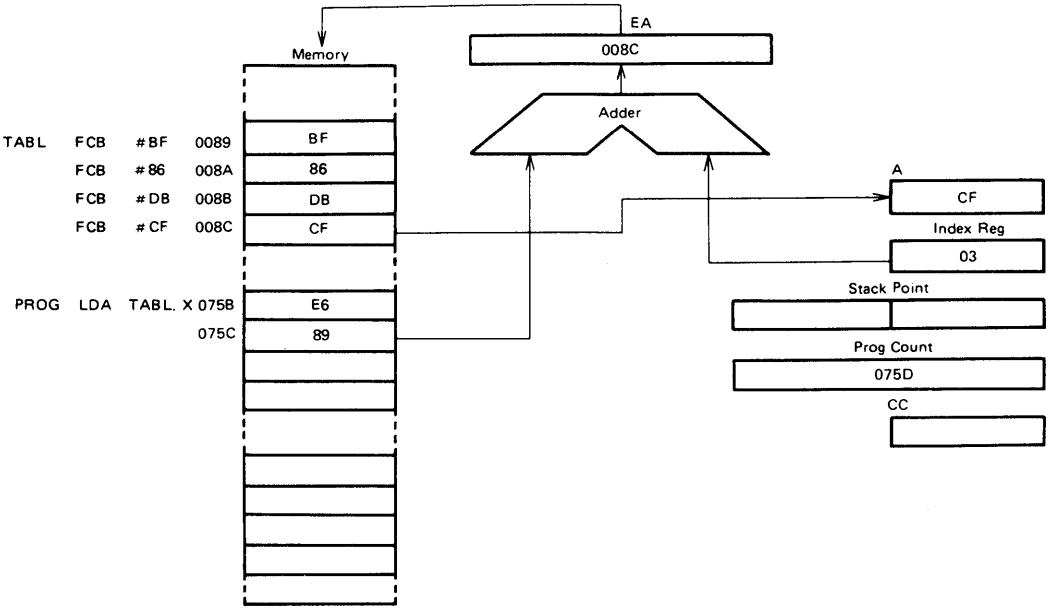


Figure 21 Indexed (8-Bit Offset) Addressing Example

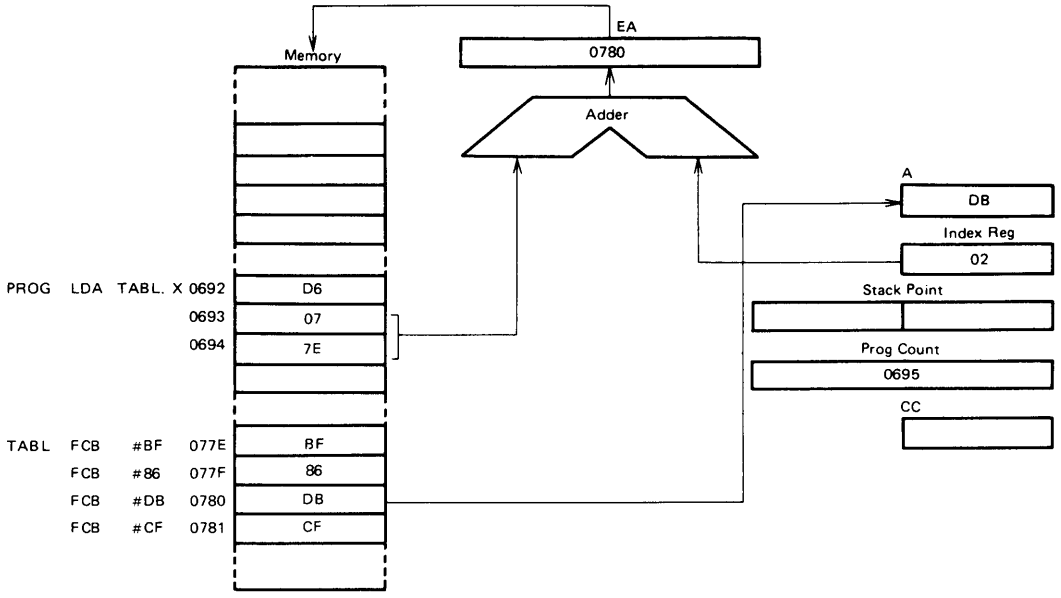


Figure 22 Indexed (16-Bit Offset) Addressing Example

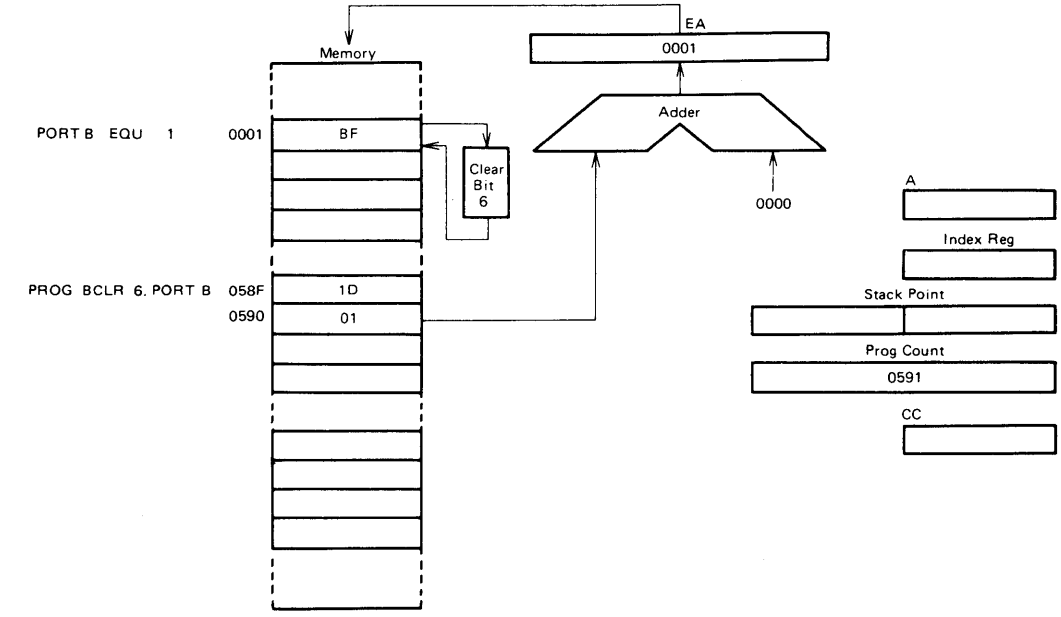


Figure 23 Bit Set/Clear Addressing Example

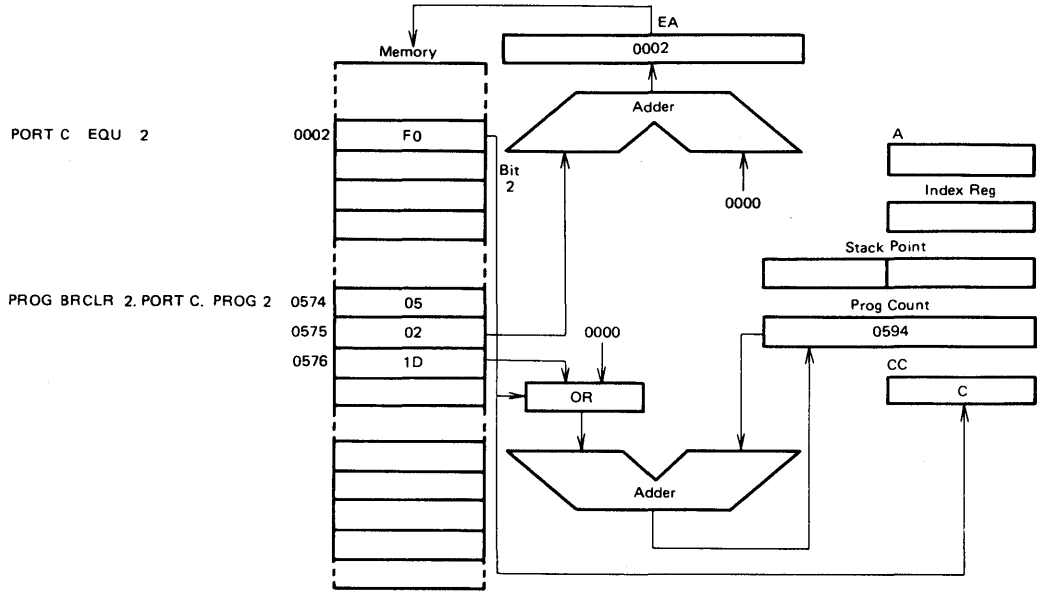


Figure 24 Bit Test and Branch Addressing Example

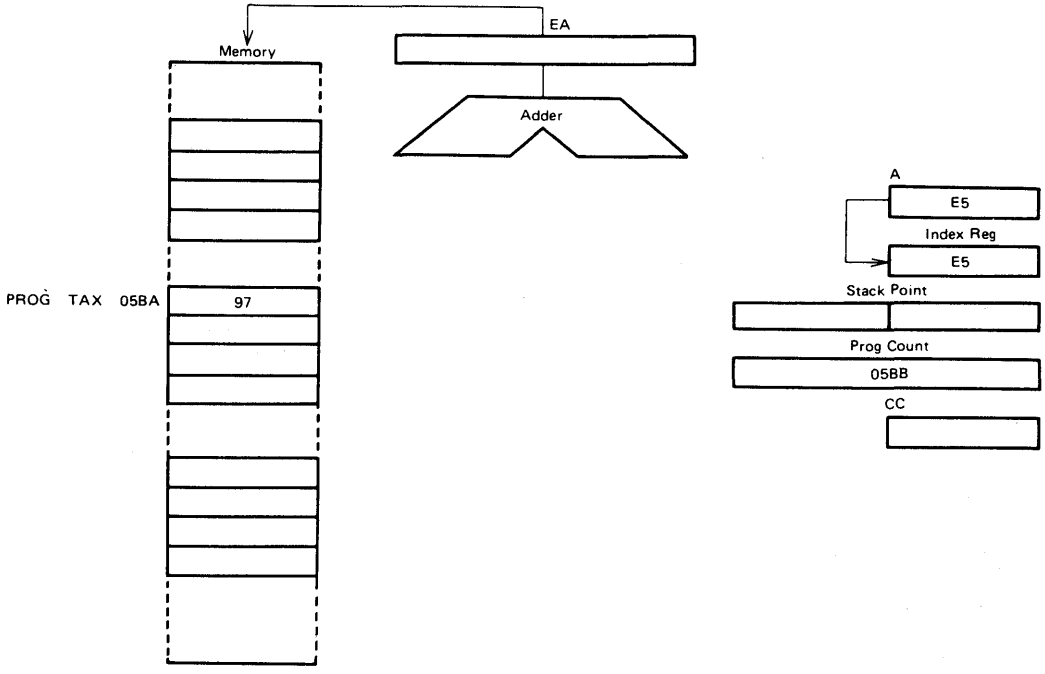


Figure 25 Implied Addressing Example

Table 2 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 3 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 4 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 5 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 6 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•

(to be continued)

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Condition Code Symbols:

- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

Table 8 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and the number of cycles required (Bytes/Cycles).
Mnemonics followed by a "*" require a different number of cycles as follows:

RTI	9
RTS	6
SWI	11
BSR	8

3. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

HD6805S6

MCU (Microcomputer Unit)

The HD6805S6 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, I/O and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD 6800-based instruction set.

The following are some of the hardware and software highlights of the MCU.

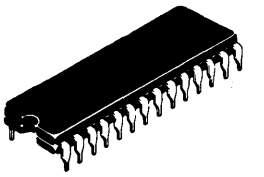
■ HARDWARE FEATURES

- 8-Bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 1804 Bytes of User ROM
- Internal 8-Bit Timer with 7-Bit Prescaler
- 20 TTL/CMOS Compatible I/O Lines; 8 Lines LED Compatible
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- Easy for System Development and Debugging
- 5 Vdc Single Supply
- Compatible with MC6805P6

■ SOFTWARE FEATURES

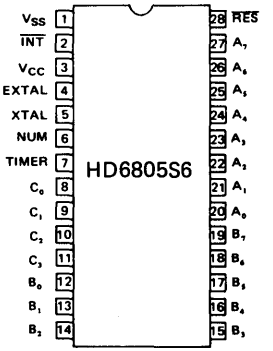
- Similar to HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible with MC6805P6

HD6805S6P



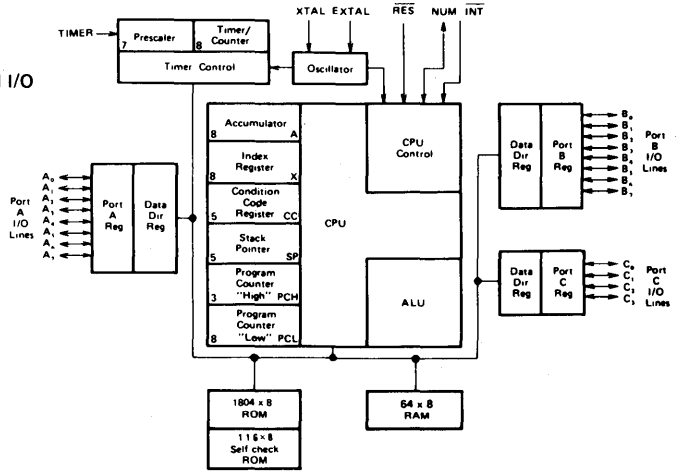
(DP-28)

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in} *	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +12.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0~+70°C, unless otherwise noted.)

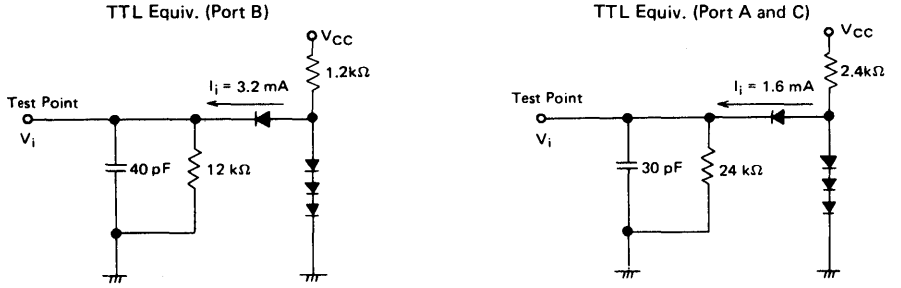
Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT}$			3.0	–	V_{CC}	V
	All Other			2.0	–	V_{CC}	V
Input "High" Voltage (Timer)	Timer Mode			2.0	–	V_{CC}	V
	Self-Check Mode			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT}$			-0.3	–	0.8	V
	EXTAL(Crystal Mode)			-0.3	–	0.6	V
	All Other			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	700	mW
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT}$			-50	–	50	μA
	EXTAL(Crystal Mode)			-1200	–	0	μA

• AC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
Oscillation Frequency (External Resistor Mode)		f_{EXT}	R_{CP}=15.0kΩ±1%	–	3.4	–	MHz
$\overline{INT}$ Pulse Width		t_{IWL}		t_{cyc}+250	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		t_{cyc}+250	–	–	ns
TIMER Pulse Width		t_{TWL}		t_{cyc}+250	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20%, R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2 μF	100	–	–	ms
Input Capacitance	XTAL	C_{in}	V_{in}=0V	–	–	35	pF
	All Other			–	–	10	pF

● PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 ~ +70°C , unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = −10 μA	3.5	–	–	V
			I_{OH} = −100 μA	2.4	–	–	V
	Port B		I_{OH} = −200 μA	2.4	–	–	V
			I_{OH} = −1 mA	1.5	–	–	V
	Port C		I_{OH} = −100 μA	2.4	–	–	V
Output "Low" Voltage	Port A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.4	V
	Port B		I_{OL} = 3.2 mA	–	–	0.4	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Port A, B, C	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		-0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	−500	–	–	μA
			V_{in} = 2V	−300	–	–	μA
	Port B, C		V_{in} = 0.4V ~ V_{CC}	− 20	–	20	μA



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074 Ⓗ or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

● V_{CC} and V_{SS}

Power is supplied to the MCU using these two pins. V_{CC} is +5.25 V ± 0.5 V. V_{SS} is the ground connection.

● $\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

● XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4 MHz maximum), a resistor or an external signal can be connected to these pins to provide a system clock with various stability/cost tradeoffs. Refer to INTERNAL OSCILLATOR OPTIONS for recommendations about these inputs.

● TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER for additional information about the timer circuitry.

● $\overline{RES}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to RESETS for additional information.

● NUM

This pin is not for user application and should be connected to V_{SS}.

● Input/Output Lines (A_0 ~ A_7, B_0 ~ B_7, C_0 ~ C_3)

These 20 lines are arranged into two 8-bit ports (A and B) and one 4-bit port (C). All lines are programmable as either inputs or outputs under software control of the data direction registers. Refer to INPUT/OUTPUT for additional information.

■ MEMORY

The MCU memory is configured as shown in Figure 2. During the processing of an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Figure 3. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

Caution: – Self Test ROM Address Area
Self test ROM locations can not be used for a user program. If the user's program is in this location, it will be removed when manufacturing mask for production.

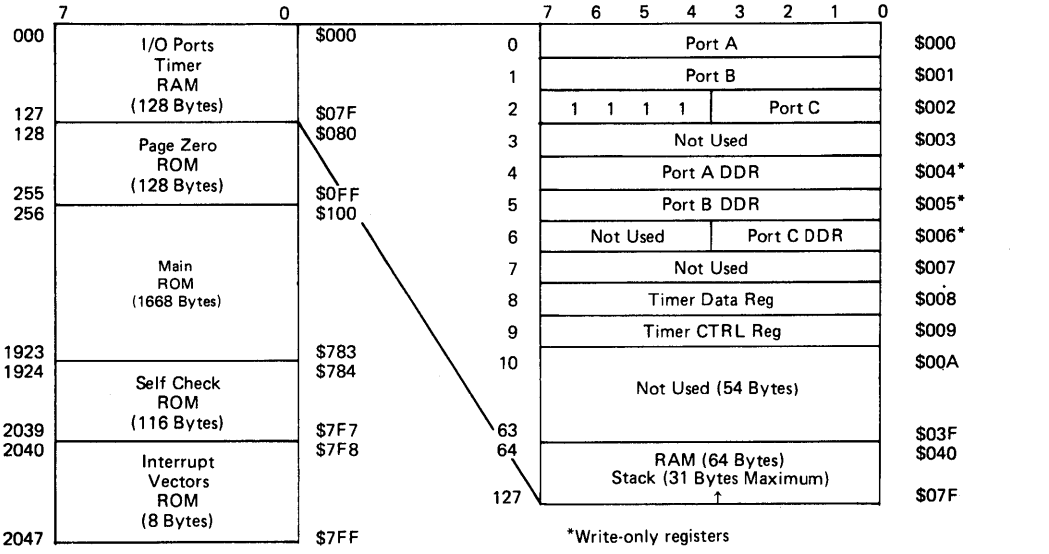


Figure 2 MCU Memory Configuration

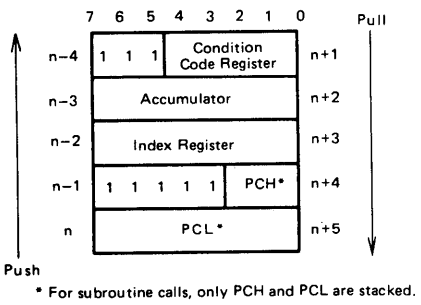


Figure 3 Interrupt Stacking Order

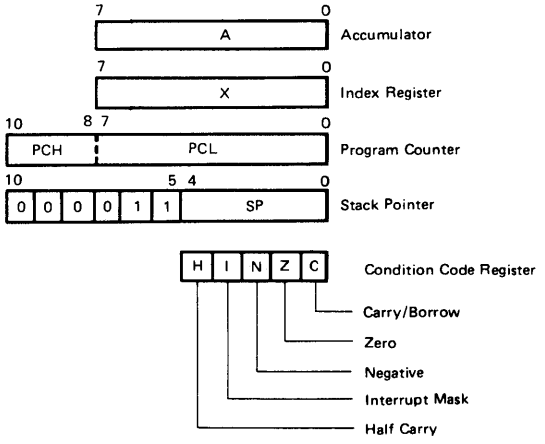


Figure 4 Programming Model

■ REGISTERS

The CPU has five registers available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

- **Accumulator (A)**

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

- **Index Register (X)**

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

- **Program Counter (PC)**

The program counter is an 11-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is an 11-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The six most significant bits of the stack pointer are permanently set to 000011. During a MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 which allows the programmer to use up to 15 levels of subroutine calls.

- **Condition Code Register (CC)**

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

Used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in result equal to a logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

■ TIMER

The MCU timer circuitry is shown in Figure 5. The 8-bit counter, the Timer Data Register (TDR), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR), is set. The CPU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $7F8 and $7F9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the Condition Code Register also prevents a timer interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When the internal ϕ_2 signal is selected as the input source, the node a is connected to b (see Fig. 5).

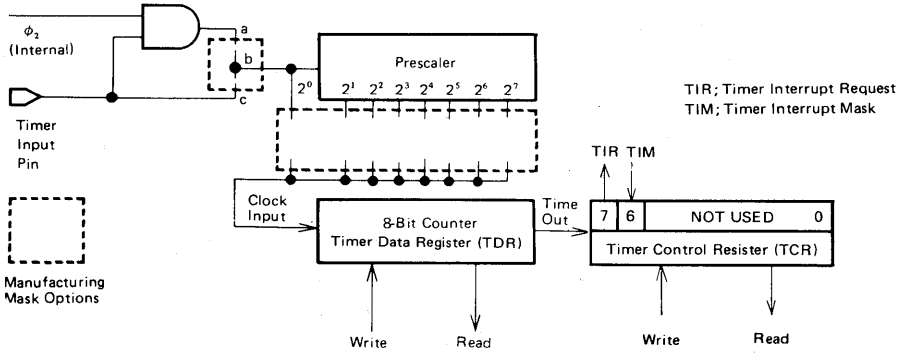


Figure 5 Timer Block Diagram

In case of the external source, the node b connects with c. When the ϕ_2 signal is used as the source, the clock signal is input to the prescaler while the TIMER input is "High".

The source of the clock input is one of the options that has to be specified before manufacture of the MCU. A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter (TDR). The timer continues to count past zero, falling through to $FF from zero, and then continuing the count. Thus, the counter can be read at any time by reading the TDR. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones; the timer interrupt request bit (bit 7) is cleared, and the timer interrupt mask bit (bit 6) is set.

(NOTE) If the MCU Timer is not used, the TIMER input pin must be grounded.

■ SELF CHECK

The self-check capability of the MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 6 and monitor the output of port C bit 3 for an oscillation of approximately 3Hz. ROM, RAM, TIMER, Interrupts, I/O of Port A, B and C are checked by this capability.

■ RESETS

The MCU can be reset three ways: by initial power-up, by the external reset input ($\overline{RES}$) and by an optional internal low voltage detect circuit, see Figure 7. All the I/O port are initialized to Input mode (DDR's are cleared) during RESET.

During power-up, a minimum of 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High".

This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 8, typically provides sufficient delay.

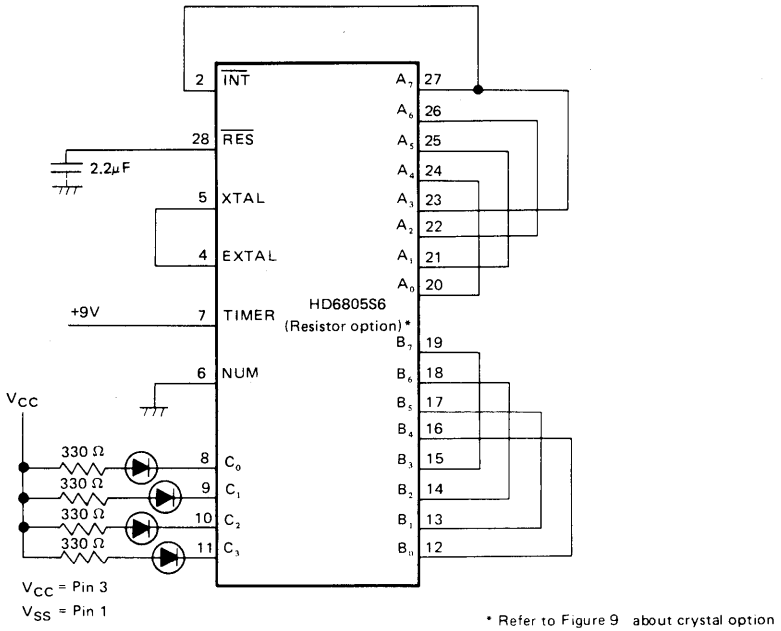


Figure 6 Self Check Connections

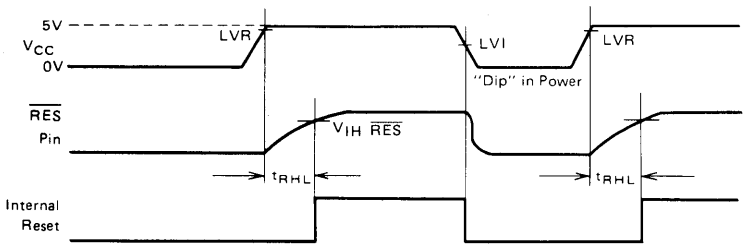


Figure 7 Power Up and $\overline{RES}$ Timing

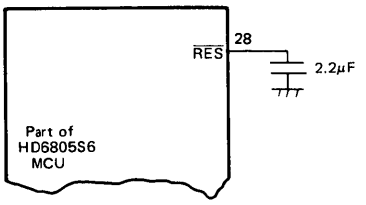


Figure 8 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR OPTIONS

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoff. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different connection methods are shown in Figure 9. Crystal specifications are given in Figure 10. A resistor selection graph is given in Figure 11.

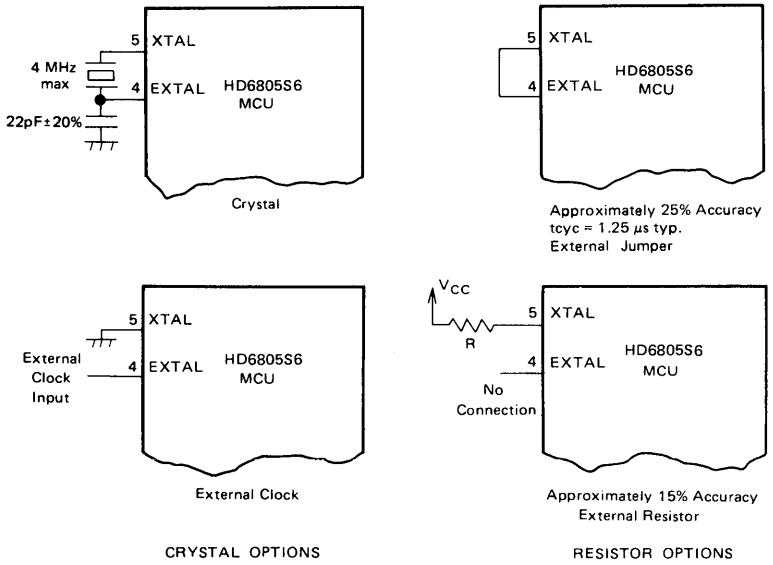
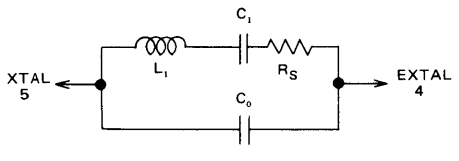


Figure 9 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz (C_1 =22pF±20%)
R_S = 60 Ω max.

Figure 10 Crystal Parameters

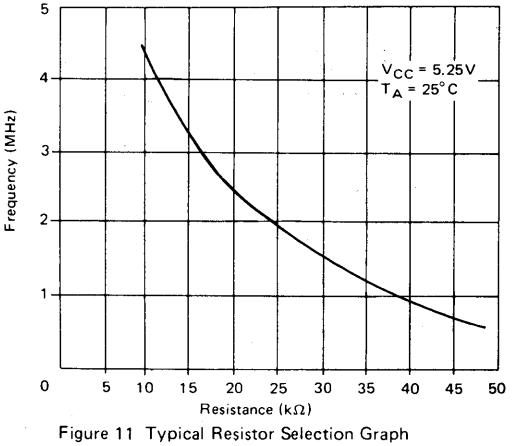


Figure 11 Typical Resistor Selection Graph

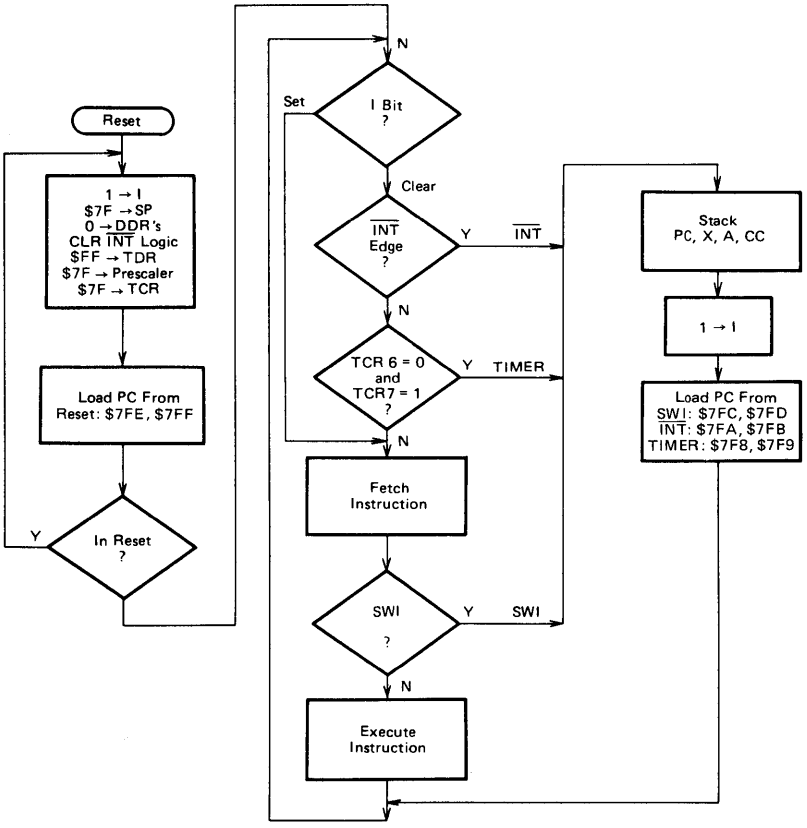
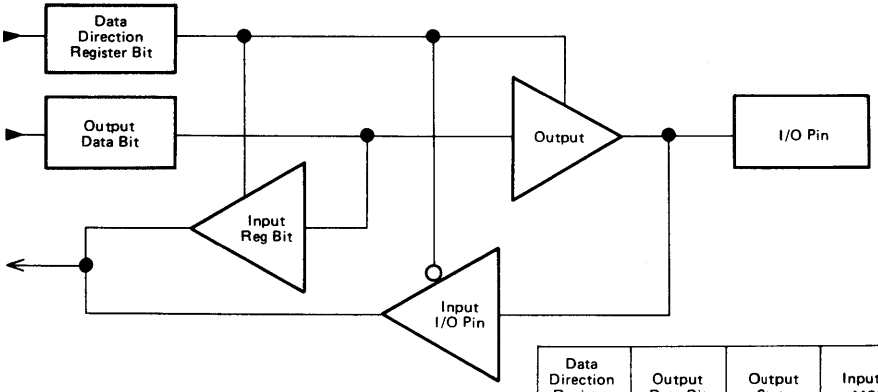


Figure 12 Interrupt Processing Flowchart



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

Figure 13 Typical Port I/O Circuitry

■ INTERRUPTS

The CPU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires 11 cycles. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. Table 1 provides a listing of the interrupts, their priority, and the vector address that contain the starting address of the appropriate interrupt routine.

A flowchart of the interrupt processing sequence is given in Figure 12.

Table 1 Interrupt Priorities

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$7FE and $7FF
SWI	2	$7FC and $7FD
$\overline{INT}$	3	$7FA and $7FB
TIMER	4	$7F8 and $7F9

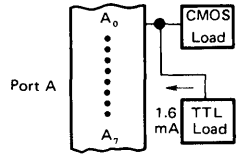
■ INPUT/OUTPUT

There are 20 input/output pins. All pins are programmable as either inputs or outputs under software control of the corresponding data direction register (DDR). When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (see Figure 13). When port B is programmed for outputs, it is capable of sinking 10 mA on each pin (V_{OL} = 1V max). All input/output lines are TTL compatible as both inputs and outputs. Port A are CMOS compatible as outputs, and Port B and C are CMOS compatible as inputs. Figure 14 provides some examples of port connections.

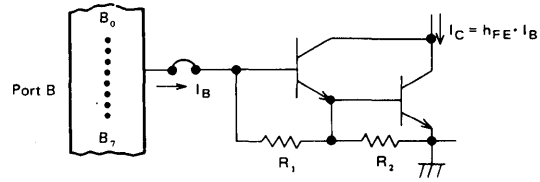
■ BIT MANIPULATION

The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 15 illustrates the usefulness of the bit manipulation and test instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which powers the controlled hardware.

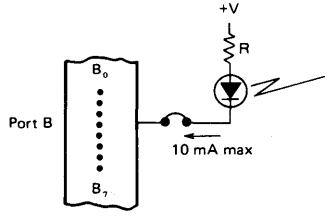
This program, which uses only seven ROM locations, provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer could also be incorporated to provide turn-on at some later time which would permit pulse-width modulation of the controlled power.



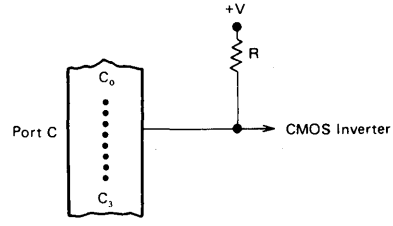
Port A Programmed as output(s), driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s), driving Darlington base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)



Port C Programmed as output(s), driving CMOS loads, using external pull-up resistors. (d)

Figure 14 Typical Port Connections

```
SELF 1    BRCLR 0, PORT A, SELF 1
          BSET 1, PORT A
          BCLR 1, PORT A
```

Figure 15 Bit Manipulation Example

■ ADDRESSING MODES

The CPU has ten addressing modes available for use by the programmer. They are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 16. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 17. In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 18. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 19. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA=(PC)+2+Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken Rel=0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 20. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 21. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 22. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 23. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 24. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($00-$FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 25. The implied mode of addressing has no EA. All the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are one byte long.

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

- **Register/Memory Instructions**

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 2.

- **Read/Modify/Write Instructions**

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 3.

- **Branch Instructions**

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

- **Bit Manipulation Instructions**

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 5.

- **Control Instructions**

The control instructions control the MCU operations during program execution. Refer to Table 6.

- **Alphabetical Listing**

The complete instruction set is given in alphabetical order in Table 7.

- **Opcode Map**

Table 8 is an opcode map for the instructions used on the MCU.

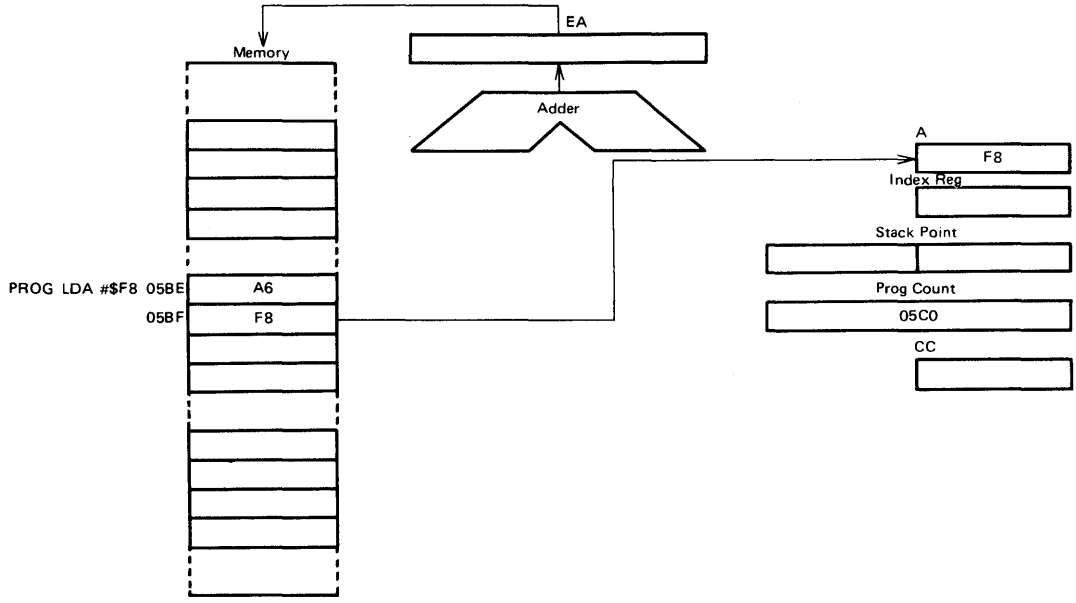


Figure 16 Immediate Addressing Example

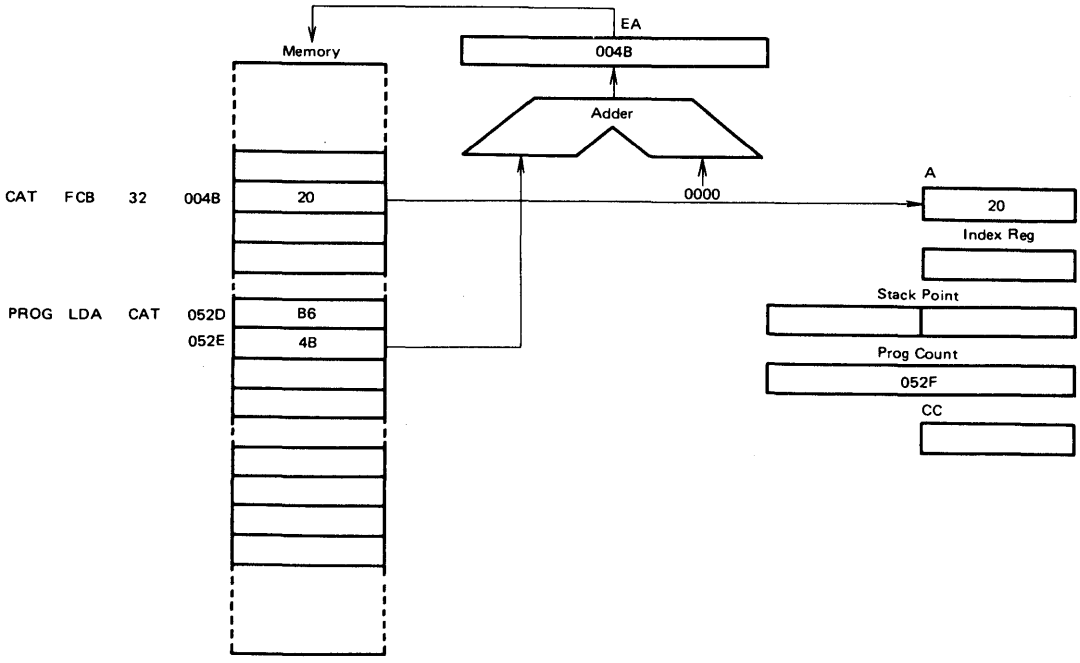


Figure 17 Direct Addressing Example

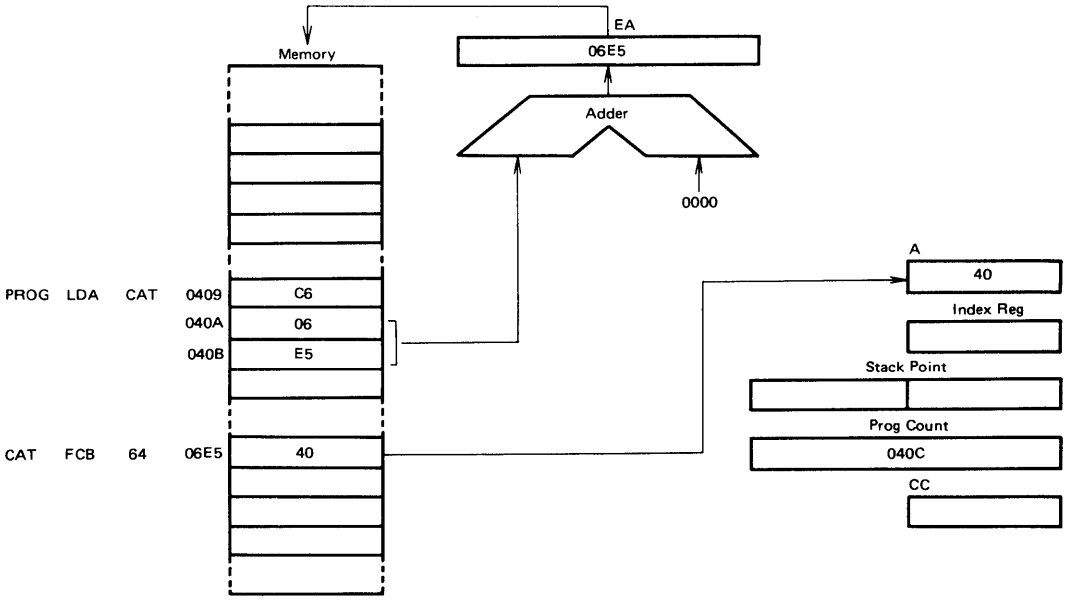


Figure 18 Extended Addressing Example

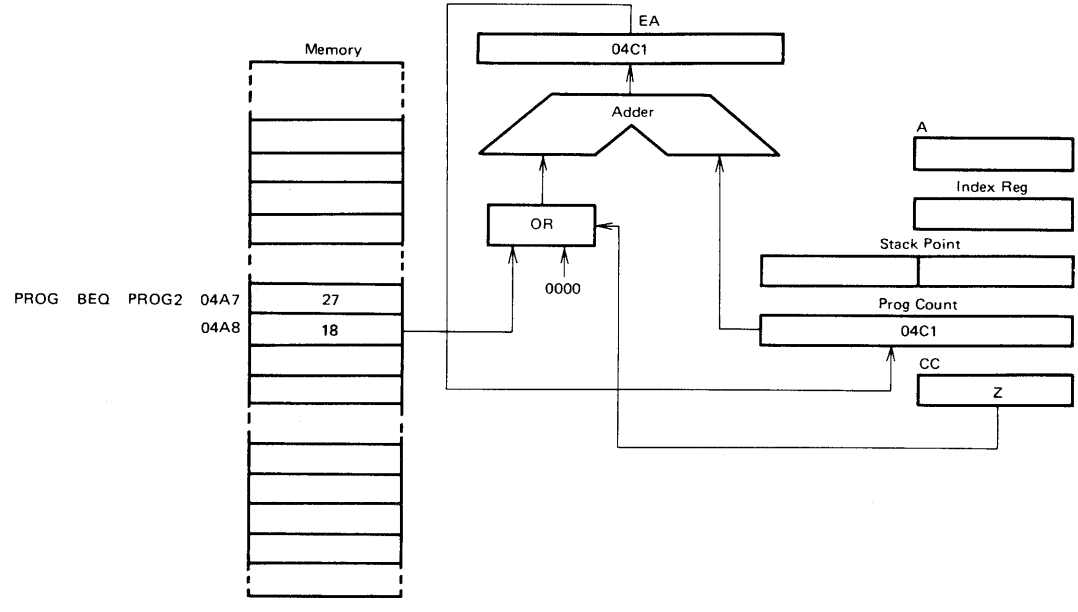


Figure 19 Relative Addressing Example

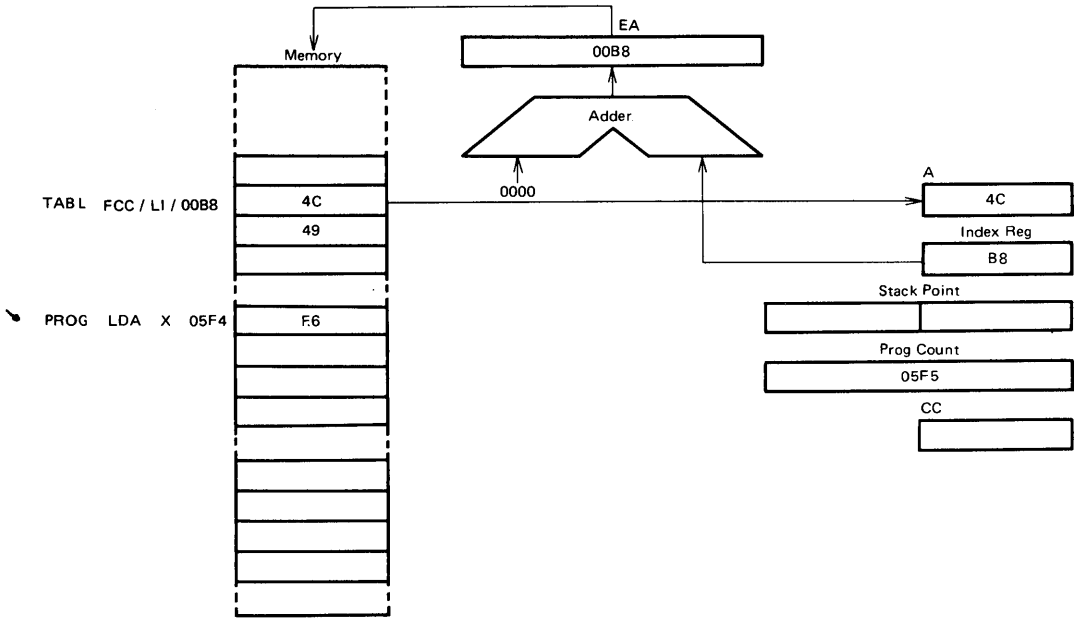


Figure 20 Indexed (No Offset) Addressing Example

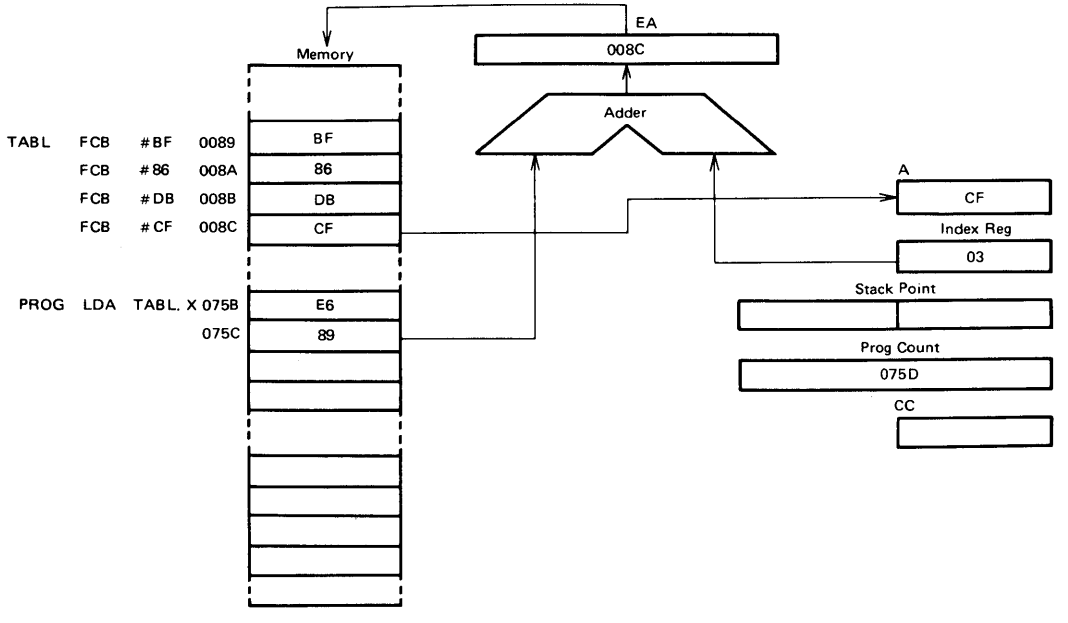


Figure 21 Indexed (8-Bit Offset) Addressing Example

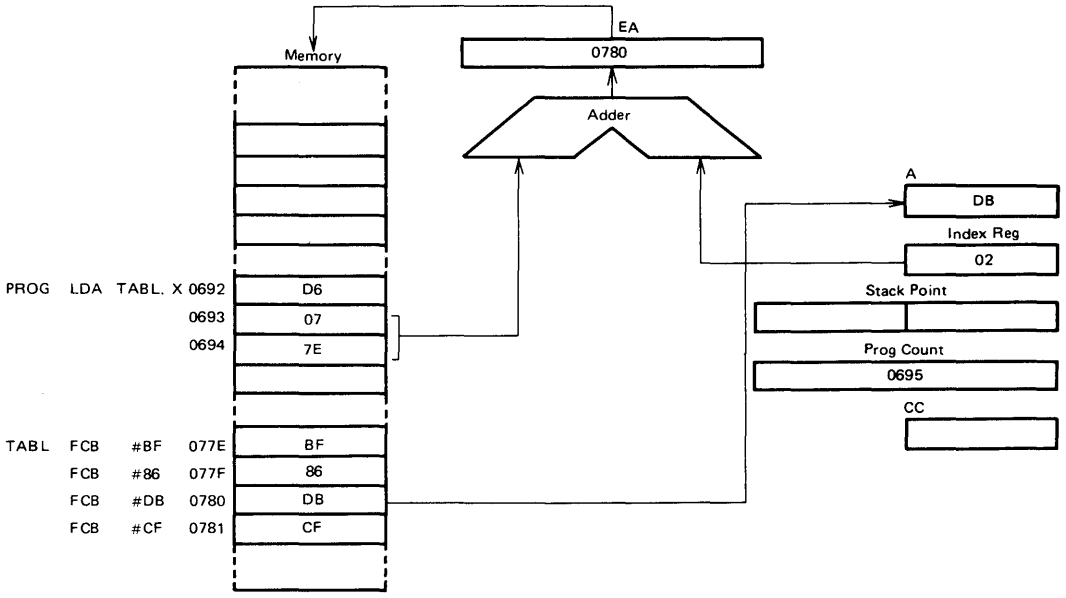


Figure 22 Indexed (16-Bit Offset) Addressing Example

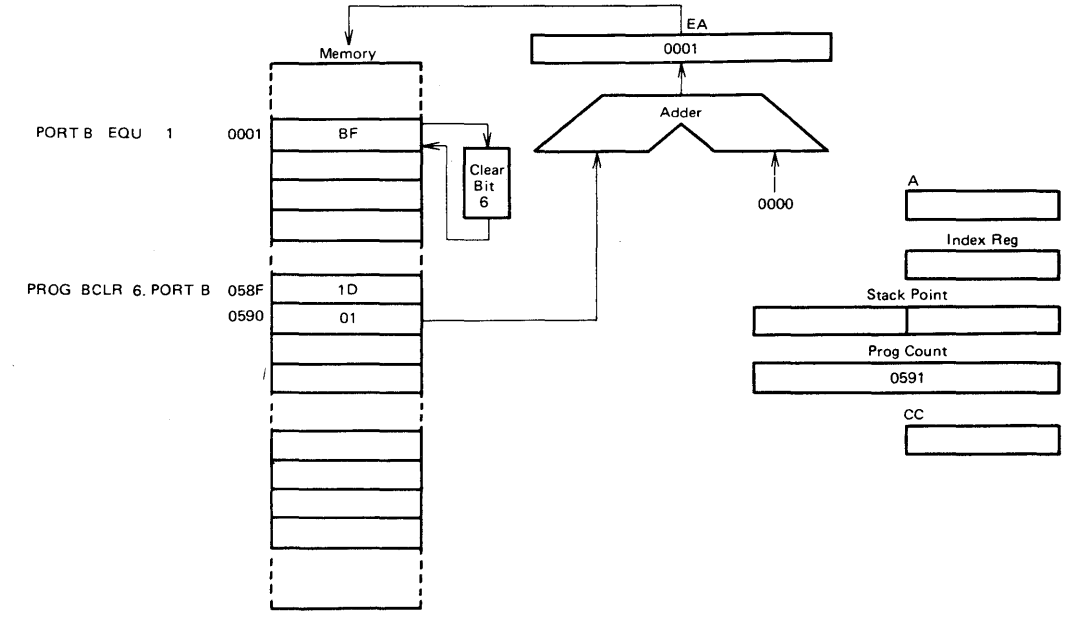


Figure 23 Bit Set/Clear Addressing Example

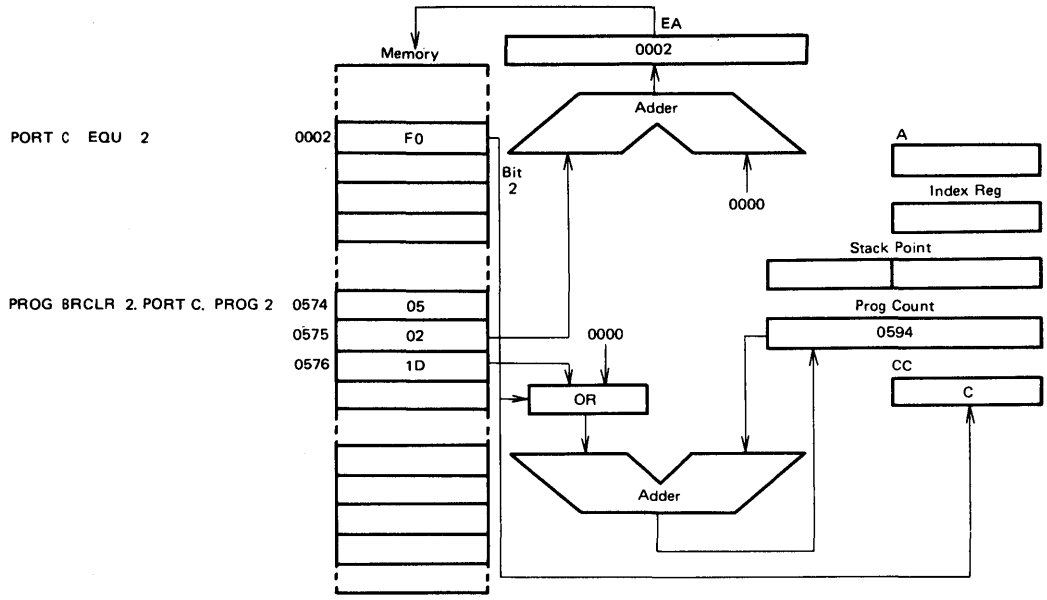


Figure 24 Bit Test and Branch Addressing Example

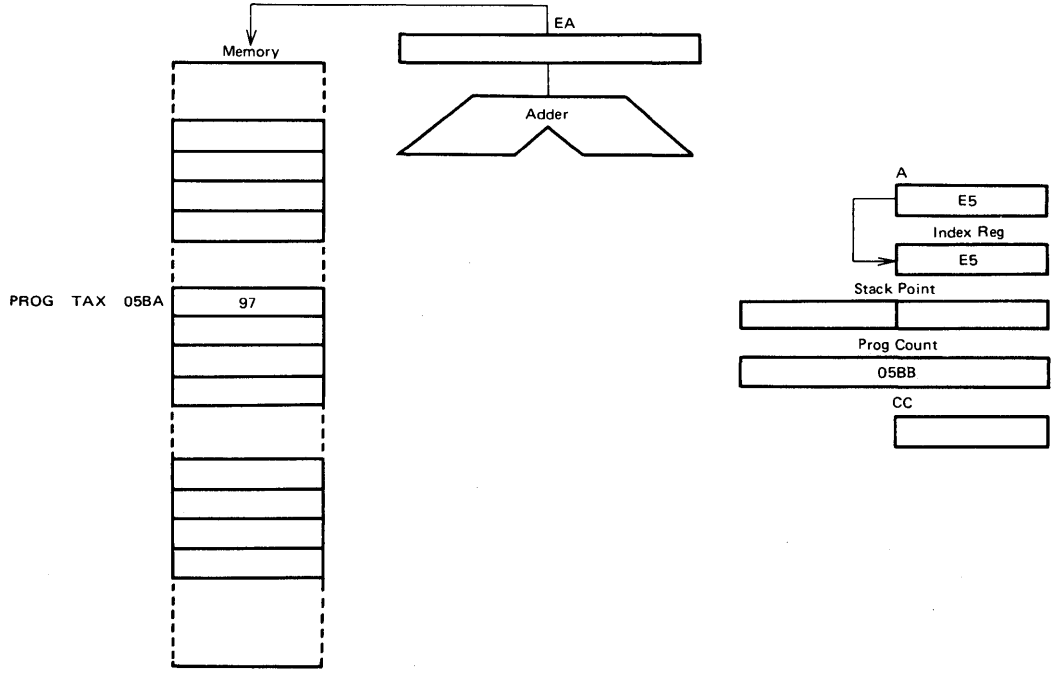


Figure 25 Implied Addressing Example

Table 2 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 3 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 4 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 5 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 6 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•

(to be continued)

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Condition Code Symbols:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
∧ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register From Stack

Table 8 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and the number of cycles required (Bytes/Cycles).
Mnemonics followed by a "*" require a different number of cycles as follows:

RTI	9
RTS	6
SWI	11
BSR	8

3. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

HD6805T2

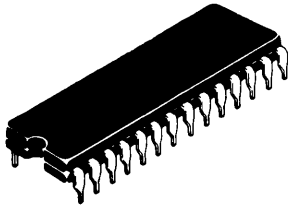
MCU (Microcomputer Unit with PLL Logic)

—PRELIMINARY—

The HD6805T2 is an 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, I/O, Timer, and the PLL Logic for an RF synthesizer. It meets the needs of users who needs economical single chip microcomputer with the proven abilities of MPU instruction set of HD6800.

The Principal characteristics of the hardware and software of the MCU are listed below.

HD6805T2P



(DP-28)

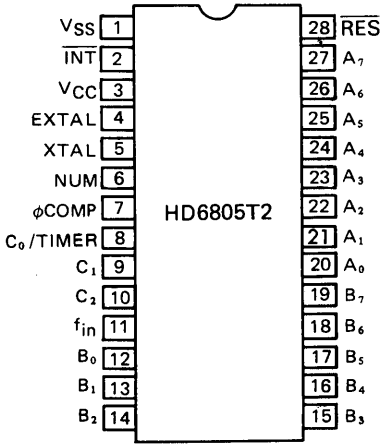
■ HARDWARE FEATURES

- 8-bit Architecture
- 64 Bytes of RAM
- Memory Mapped I/O
- 2508 Bytes of User ROM
- Internal 8-bit Timer with 7-bit Prescaler
- Timer Start/Stop and Source Select
- Vectored Interrupts — External and Timer
- 19 TTL/CMOS Compatible I/O Lines; 8 Lines are LED Compatible
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- 14-Bit Binary Variable Divider
- 10-Stage Mask-Programmable Reference Divider
- Three-State Phase and Frequency Comparator
- Suitable for TV Frequency Synthesizers
- $5V_{dc}$ Single Supply

■ SOFTWARE FEATURES

- Resembles HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Handing
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM, and I/O
- Compatible with MC6805T2

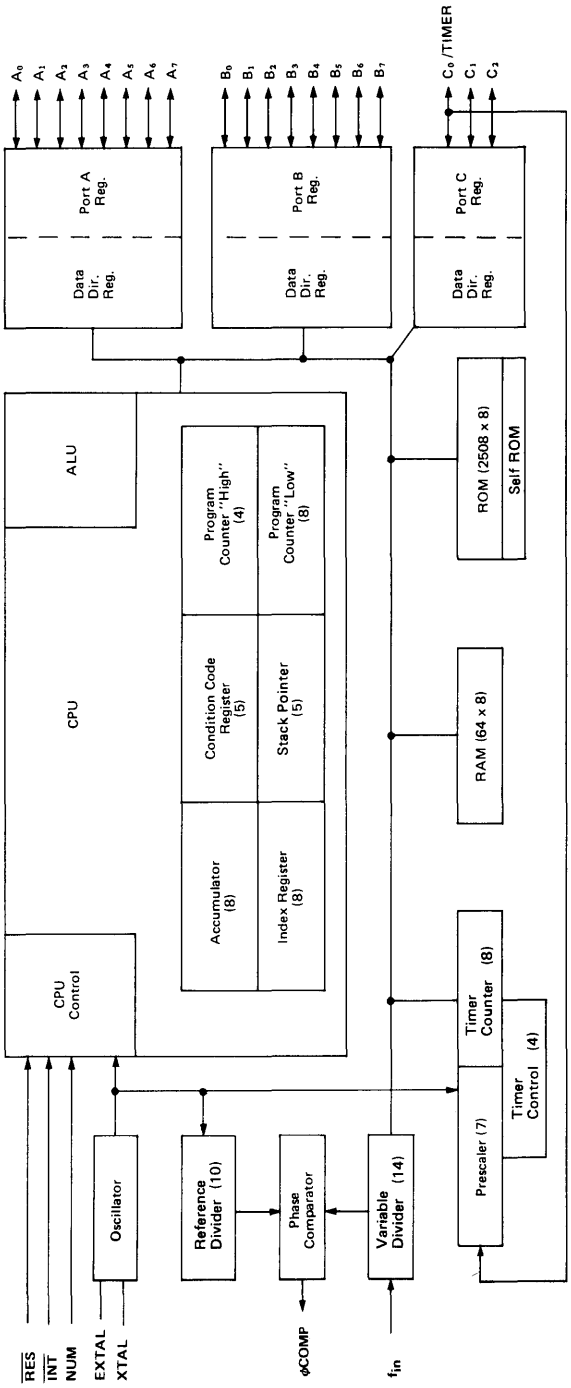
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BLOCK DIAGRAM**



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 to +7.0	V
Input Voltage (Except ϕCOMP)	V_{in}*	−0.3 to +7.0	V
Input Voltage (ϕCOMP)		−0.3 to +12.0	V
Operating Temperature	T_{opr}	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +150	°C

* With respect to V_{SS} (SYSTEM GND).
(Note) When the maximum ratings are exceeded, the LSI may be irreparably damaged. Recommended operating conditions should be adhered to.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min.	typ.	max.	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT}$*			3.0	–	V_{CC}	V
	All Other Except f_{in}			2.0	–	V_{CC}	V
Input "High" Voltage ϕCOMP	Normal Mode			2.0	–	V_{CC}	V
	Self-Check Mode**			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		−0.3	–	0.8	V
	$\overline{INT}$*			−0.3	–	0.8	V
	EXTAL			−0.3	–	0.6	V
	All Other Except f_{in}			−0.3	–	0.8	V
Power Dissipation		P_D	Not Port Loading	–	–	850	mW
AC Coupled Input Voltage Swing	f_{in}	V_{FIP}		0.5	–	2.4	V_{ACP-P}
Input Leak Current	f_{in}	$\lvert I_{IL} \rvert$		–	–	40	μA
Output "Low" Current	ϕCOMP	I_{CML}	V_{OL} = 1.0V	–	–	300	μA
Output "High" Current	ϕCOMP	I_{CMH}	$V_{OH} = V_{CC} - 1V$	–	–	200	μA
Input Leak Current	ϕCOMP	$\lvert I_{IL} \rvert$		–	–	1.0	μA
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	$\overline{INT}$	$\lvert I_{IL} \rvert$	V_{in} = 0.4V	–	–	50	μA
	EXTAL			–	–	1600	μA
	$\overline{RES}$		V_{in} = 0.8V	4.0	–	50	μA

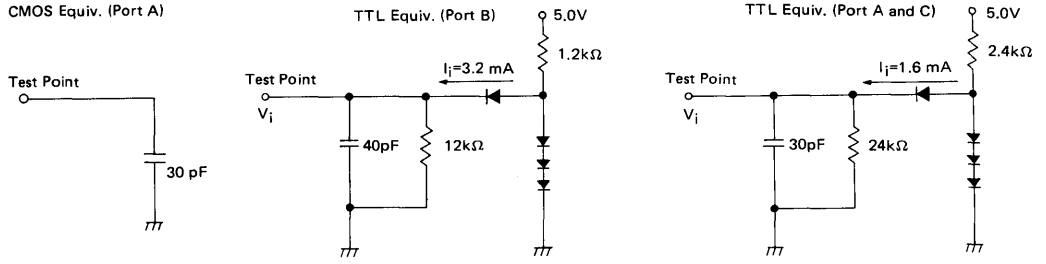
* Internal biasing makes the input float to approximately 2.0V when unused.
** In self-check mode, ϕCOMP may be connected to V_{IH} through 10 kΩ register.

- **AC CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	min.	typ.	max.	Unit
Clock Frequency	f_{cl}		0.4	–	4.2	MHz
Cycle Time	t_{cyc}		0.95	–	10	μs
$\overline{INT}$ Pulse Width	t_{IWL}		t_{cyc}+ 250	–	–	ns
$\overline{RES}$ Pulse Width	t_{RWL}		t_{cyc}+ 250	–	–	ns
TIMER Pulse Width	t_{TWL}		t_{cyc}+ 250	–	–	ns
Delay Time Reset	t_{RHL}	External Cap. = 1.0μF	–	100	–	ms
Input Frequency	f_{in}		1	–	16	MHz
Input Frequency Rise Time at f_{in}	t_{inr}		–	–	20	ns
Input Frequency Fall Time at f_{in}	t_{inf}		–	–	20	ns
Duty Cycle of f_{in} and External Input on EXTAL	–		40	–	60	%
Injection Pulse Active Time	t_{err}		–	70	–	ns
Input Capacitance (XTAL)	C_{in}	V_{in} = 0V	–	–	35	pF
Input Capacitance (All Other)	C_{in}	V_{in} = 0V	–	–	10	pF

- **PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 to +70°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min.	typ.	max.	Unit
Output "High" Voltage	Port A with CMOS Drive Enable	V_{OH}	I_{OH} = −100μA	2.4	–	–	V
			I_{OH} = −10μA	3.5	–	–	V
	Port A with CMOS Drive Disable		I_{OH} = −100μA	2.4	–	–	V
	Port B		I_{OH} = −200μA	2.4	–	–	V
	Port C		I_{OH} = −100μA	2.4	–	–	V
Output "Low" Voltage	Port A	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
	Port B		I_{OL} = 3.2mA	–	–	0.4	V
			I_{OL} = 10mA	–	–	1.0	V
	Port C		I_{OL} = 1.6mA	–	–	0.4	V
Input "High" Voltage	Port A, B, C	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Port A with CMOS Drive Enable	$\lvert I_{IL} \rvert$	V_{in} = 2.0V	–	–	300	μA
			V_{in} = 0.8V	–	–	500	μA
	Port A with CMOS Drive Disable			–	–	20	μA
	Port B			–	–	20	μA
	Port C			–	–	20	μA



(Note) 1. Load capacitance contains the floating capacitance of the probe, the jig, etc.
2. All diodes are 1S2074 Ⓗ or the similar.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The following describes the input and output signals of HD6805T2.

• V_{CC}, V_{SS}

The MCU is supplied power through these pins. V_{CC} is 5.25V ± 0.5V. V_{SS} is grounded.

• $\overline{INT}$

This pin provides an external interrupt to the MCU. For more details, see INTERRUPTS.

• XTAL, EXTAL

These are control input pins for the internal clock circuit. Connection of these pins to a crystal (AT cut, 4.2 MHz maximum) or to an external input provides the internal oscillator with varying degrees of stability. For suggestions pertaining to these pins, see INTERNAL OSCILLATOR OPTIONS.

• f_{in}

This f_{in} pin is a high frequency digital input to the variable divider portion of the on-chip frequency synthesizer. The reference frequency for the phase lock loop is divided down from internal clock ϕ_2. The frequency synthesizer features are explained in PHASE LOCK LOOP.

• ϕCOMP

This is a three-state output signal. The state varies according to the result of the comparison between the internal reference frequency and the variable divided signal. See PLL for details. ϕCOMP is raised to 9V through 10 kΩ in self-check mode.

• $\overline{RES}$

Besides the resetting capability which the MCU already has, this pin also makes resetting of the MCU possible. See RESETS for more details.

• NUM

This pin should be grounded as it is not applicable to users.

• INPUT/OUTPUT Lines (A_0 to A_7, B_0 to B_7, C_0 to C_2)

These 19 lines form two 8-bit ports (Port A, Port B) and one 3-bit port (Port C). By software control of the data direction registers, these lines can be programmed to be either inputs or outputs. See INPUTS/OUTPUTS for more information. The C_0/TIMER pin also can be programmed as an external input to the internal timer. For information on the timer modes, see TIMER.

■ MEMORY

The MCU memory diagram is indicated in Figure 2. The MCU, with its program counter, can address 4096 bytes of memory and I/O registers. The MCU has implemented 2698 of the 4096 memory locations, 2508 bytes user ROM, 116 bytes self-check ROM, 64 bytes of user RAM, 6 bytes of port I/O, 2 timer registers, and 2 PLL registers. The user ROM is divided into four areas. The first area (begins at $080) provides users with access to ROM locations by using the direct and table look-up indexed addressing mode. The second and third user ROM areas begin at memory location $100 and $D40 respectively. The last eight-byte user ROM which begins at $FF8 is for the interrupt vectors.

The first 16 memory locations of the MCU are reserved for I/O features and 10 of them have been implemented. These locations are used for the ports, the port DDRs, the timer, and the PLL registers.

The MCU has 64 bytes of user RAM. 31 bytes out of the 64 bytes are shared with the stack area. Careful utilization of the stack is necessary when data shares the stack area.

While interrupt and subroutine calls are processed to save the processor state, the shared stack area is occupied.

The register contents are saved in the stack as indicated in Figure 3. The low order byte (PCL) of the program counter is stacked first as the stack pointer decrements during saving. Next, the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly after pulls from the stack, as the stack pointer increments when it fetches data from the stack. Only when a subroutine call is made, the program counter (PCH, PCL) contents are saved onto the stack, and the remaining CPU registers are not saved.

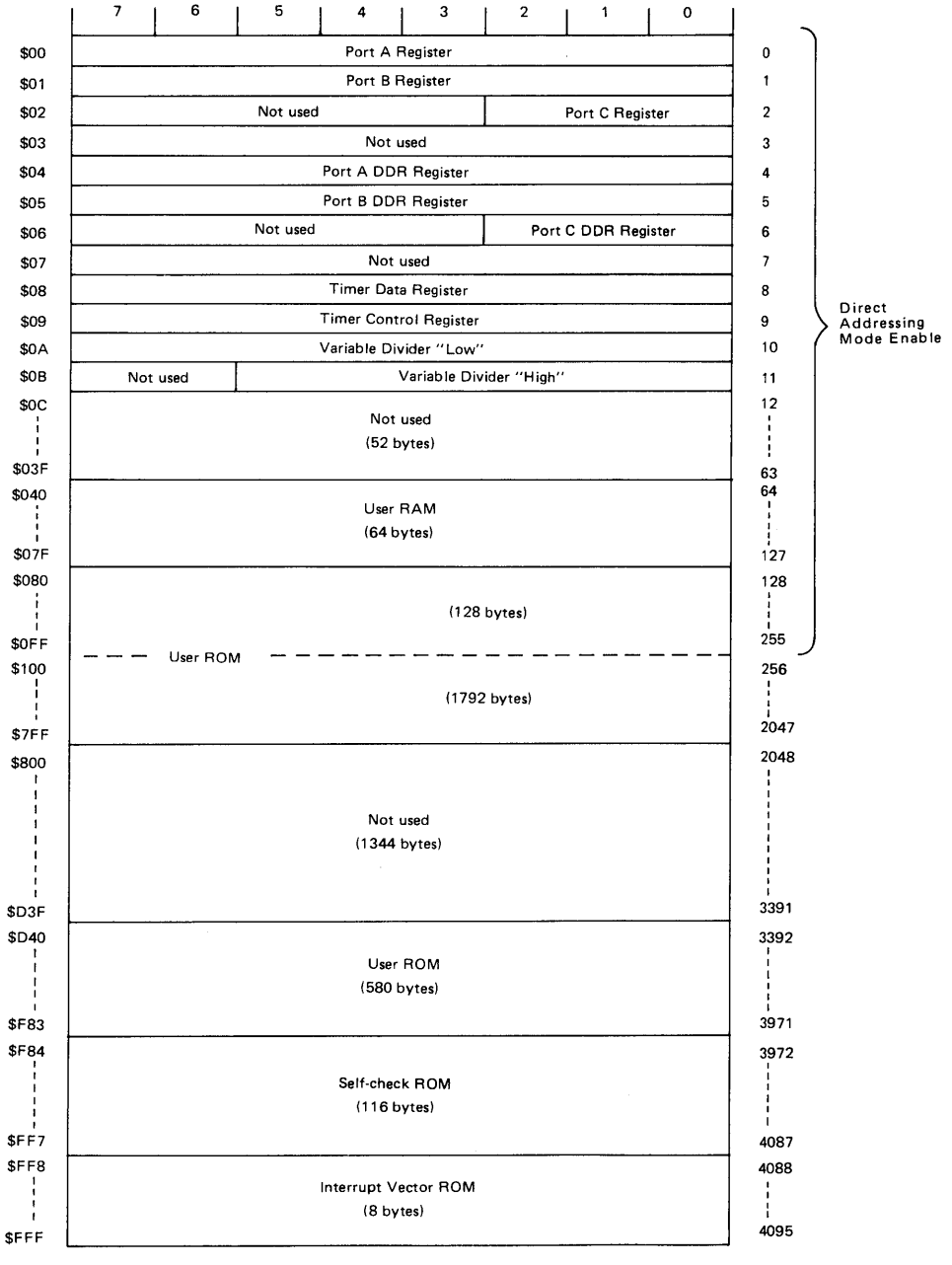
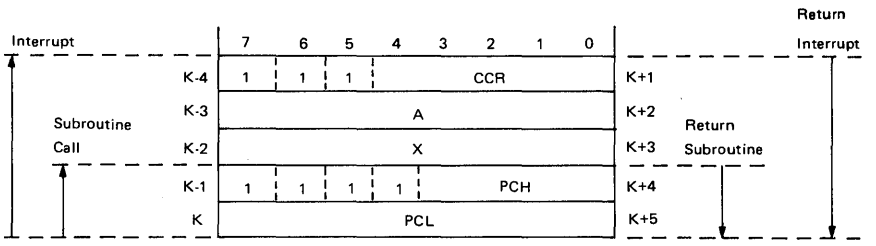


Figure 2 MCU Memory Configuration

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300



Only PCH and PCL are saved for subroutine calls.

Figure 3 Interrupt Stacking Order

■ REGISTERS

The MCU provides five registers for the programmer which are indicated in Figure 4. These registers are explained in the following.

• Accumulator (A)

The accumulator is an 8-bit general purpose register. It holds operands and results of arithmetic calculations or data manipulations.

• Index Register (X)

The index register is an 8-bit register for the indexed addressing mode. It has an 8-bit address which can be added to an offset value to make an effective address. When using read/modify/write instructions, it may also be used for data manipulation and limited calculations. When not required by the code sequence being executed, it can be a temporary storage area.

• Program Counter (PC)

The program counter is an 12-bit register. It contains the address that decides which instruction is to be executed next.

• Stack Pointer (SP)

The stack pointer is a 12-bit register. It contains the address of the next free location in the stack. Firstly, the stack pointer is set to location $07F. Then it is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The seven most crucial bits of the stack pointer are set to 0000011 permanently. The stack pointer is set to location $07F during and MCU reset or the reset stack pointer (RSP) instruction. Subroutines and interrupts can be nested down to location $061 which enables the programmer to use a maximum of 15 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register. In the register, the results of the instruction just executed is indicated or flagged by each bit. These bits can be individually program tested specific action taken as a result of their state. The following paragraphs explain each individual code register bit.

Half Carry (H)

Indicates that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD and ADC).

Interrupt (I)

This bit is set to mask external interrupt ($\overline{INT}$) and the timer. If an interrupt takes place while this bit is set, it is latched and will not be processed until the interrupt bit is reset.

Negative (N)

Indicates that the result of the last data, arithmetic, or logical manipulation was negative (bit 7 in result equal to a logic "one").

Zero (Z)

It indicates that the result of the last data, arithmetic, or logical manipulation was zero.

Carry/Borrow (C)

During the last arithmetic operation, it indicates that a carry or borrow out of the arithmetic unit (ALU) occurred. During branch instructions, rotates, bit test, and shifts, this bit is also affected.

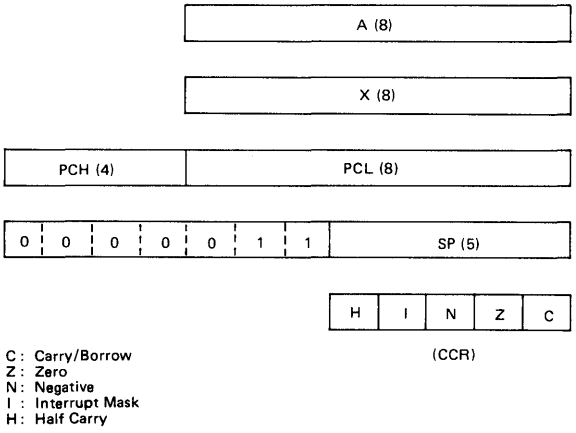


Figure 4 Programming Model

■ TIMER

The MCU timer circuitry is indicated in Figure 5. Program control enables the 8-bit counter to be loaded and the clock input (prescaler output) decrements the counter toward zero. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR) is set. The timer interrupt mask bit (bit 6) in the TCR enables the timer interrupt to be masked (disabled). The timer also becomes inhibited by the interrupt bit (I-bit) in the Condition Code Register. When the MCU responds to the interrupt requirement, it maintains the present CPU state in the stack, fetches the timer interrupt vectors from locations $FF8 and $FF9, and executes the interrupt routine. See INTERRUPT for more details.

The timer clock input is established by way of bit 5 (TCR 5) in the Timer Control Register. When this bit is set to a logic "one" (external mode), the C_0/TIMER pin is the time clock source. In this mode, a mask option selects either the gated ϕ_2 with C_0 or the positive transition on C_0/TIMER as timer clock source. This makes pulse widths or pulse counts easily measured. The timer clock source is the internal ϕ_2 when TCR 5 is set to a logic "zero". When bit 4 in the Timer Control Register is set to a logic "one", the time clock source is disabled.

The timer continues to count past zero, falling through to $FF from zero, and then continuing to count. The counter can be monitored by reading the Timer Data Register (TDR). This allows a program to determine the length of time since a timer interrupt has occurred without disturbing the counting process.

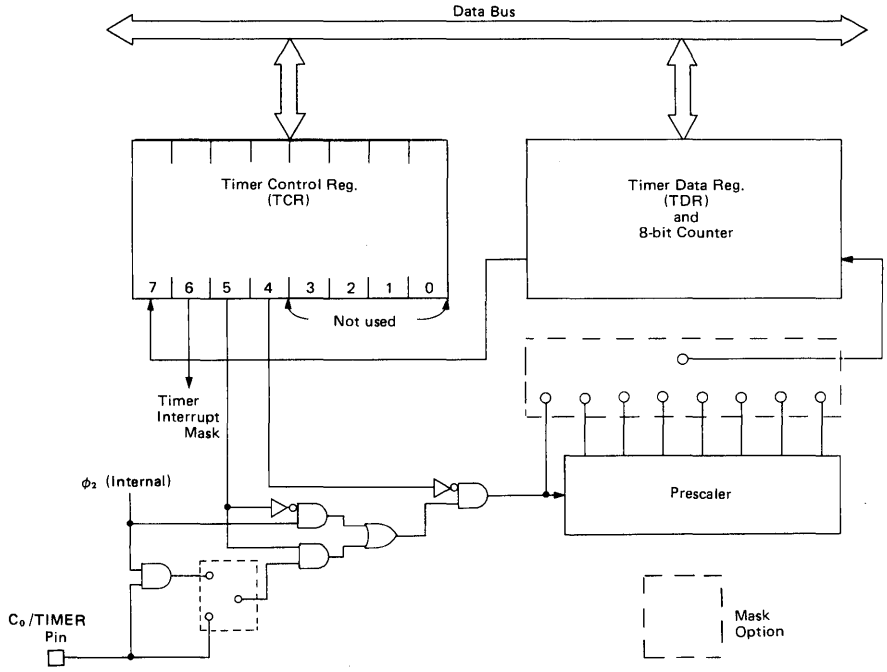


Figure 5 Timer Block Diagram

7	6	5	4	3	2	1	0
TCR7	TCR6	TCR5	TCR4	TCR3	TCR2	TCR1	TCR0

TCR7 – – Timer Interrupt Request Status Bit;
Set when TDR "0"; cleared to "0" during reset.

TCR6 – – Timer Interrupt Mask Bit;
When "1", timer interrupt is masked (disabled). Set to "1" during reset.

TCR5 – – External Timer Source;
External when set to "1" and internal when set to "0". Cleared to "0" during reset.

TCR4 – – Disable Timer;
Timer source is disconnected when set to "1" and timer input enabled when set to "0". Cleared to "0" during reset.

TCR bits 3, 2, 1 and 0; set to all "1"'s (not used).

■ SELF-CHECK

The MCU has a self-check capability which allows an internal check to see whether a port is functional or not. Connect the MCU as indicated in Figure 6 and monitor the output of Port C bit 2 for an oscillation of approximately 7 Hz. Pin 7, a 9 volt level on the ϕCOMP input, energizes the ROM-based self-check feature. The self-check program exercises the timer, interrupts, I/O ports, RAM, and ROM.

■ RESETS

There are three ways to reset the MCU; initial power up, the external reset input $\overline{RES}$), and by an internal low voltage detect circuit. See Figure 7 for details. All the I/O ports are initialized to Input Mode (DDR's are cleared) during RESET.

A delay of t_{RHL} is essential upon power up before allowing the reset input to go "High".

This time allows the internal crystal oscillator to stabilize. Sufficient delay can be provided by connecting a capacitor to the $\overline{RES}$ input as shown in Figure 8.

■ INTERNAL OSCILLATOR

The design of the internal oscillator circuit aims at the requirement of minimum of external components. A crystal or an external signal can be used to drive the internal oscillator. Figures 9 and 10 show different connection methods. Figure 11 indicates the crystal specifications.

The crystal oscillator startup time changes according to many variables: crystal parameters (especially R_S), oscillator load capacitances, IC parameters, ambient temperature, and supply capacitances. To ensure rapid oscillator startup, neither the load capacitance nor the crystal characteristics should exceed the recommended value.

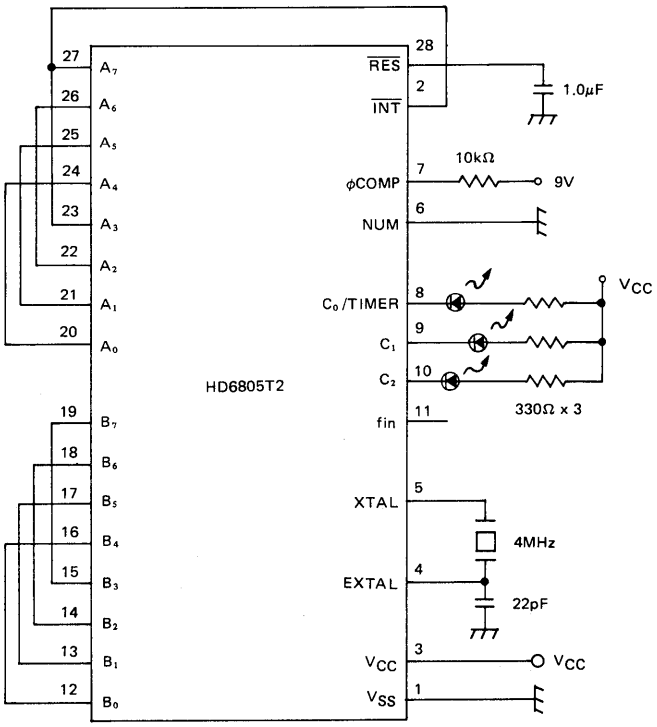


Figure 6 Self Check Connections

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

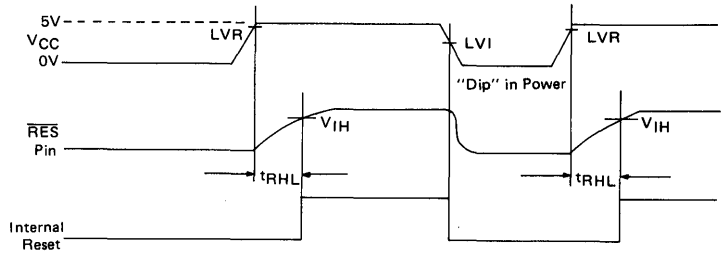


Figure 7 Power Up and $\overline{RES}$ Timing

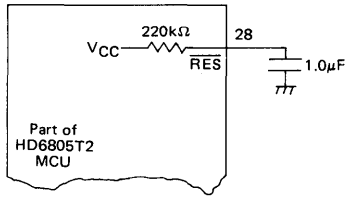


Figure 8 Power Up Reset Delay Circuit

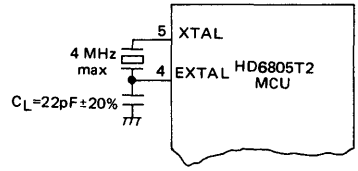


Figure 9 Crystal

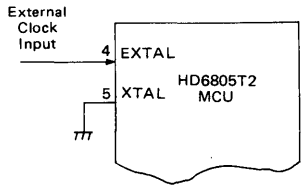


Figure 10 External Clock

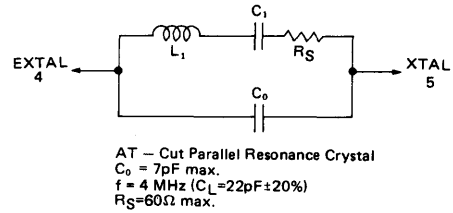


Figure 11 Crystal Parameters

■ INTERRUPTS

The MCU can be interrupted three ways as follows: ① through the external interrupt ($\overline{INT}$) input pin, ② the internal timer interrupt request, ③ the software interrupt instruction (SWI). If any interrupt occurs, processing is in pending, the current CPU state is saved in the stack, the interrupt bit (I) in the condition Code Register is set, the address of the interrupt routine is received from the proper interrupt vector address, and finally the interrupt routine is carried out. The complete stacking the CPU registers, setting the I-bit, and vector fetching, a total of 11 t_{cyc} periods are required.

Figure 12 illustrates a flowchart of the interrupt sequence. When a return from interrupt (RTI) instruction is issued, the interrupt service routine must be terminated to cause the MCU to resume processing of the program held by the interrupt (by popping off the previous CPU state). Table 1 lists the interrupts, their priority, and the address of the vector containing the start address of the proper interrupt service routine. The interrupt priority applies to those suspended when the CPU is ready for a new interrupt. Though $\overline{RES}$ is listed in Table 1 in that it is occasionally regarded as an interrupt, it is not normally used as such. When the interrupt mask bit in the Condition Code Register is set, the interrupt is latched for executing a later interrupt.

The external interrupt is internally synchronized and then latched onto the negative edge of $\overline{INT}$, which can be driven by a digital signal at a maximum period of t_{IWL}.

Table 1

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$FFE and $FFF
SWI	2*	$FFC and $FFD
$\overline{INT}$	3	$FFA and $FFB
TIMER	4	$FF8 and $FF9

* Priority 2 applies when the I-bit in the condition Code Register is set. When I = 0, SWI has priority 4, like any other instruction. Therefore, $\overline{INT}$ has priority 2 and the timer has priority 3.

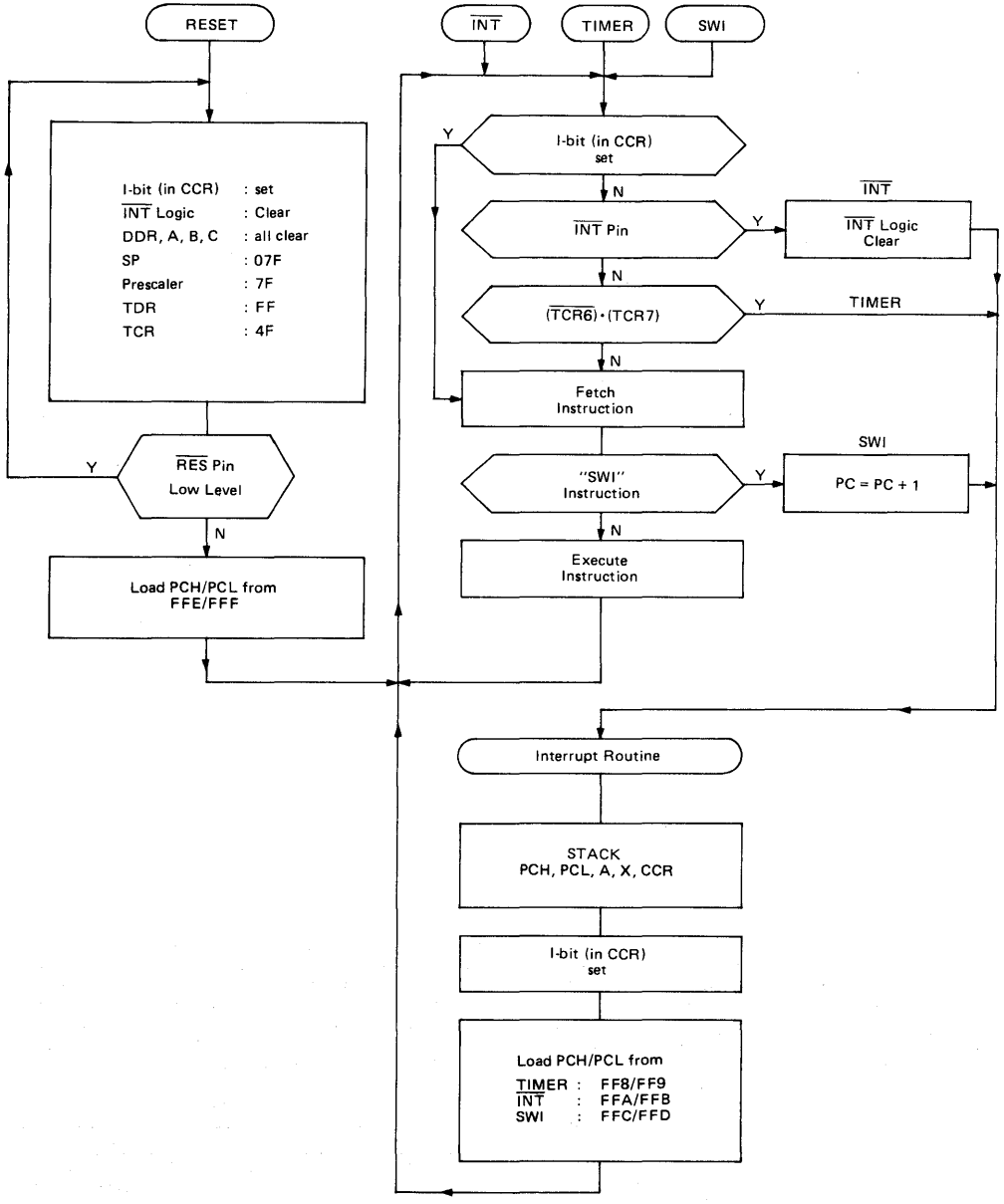
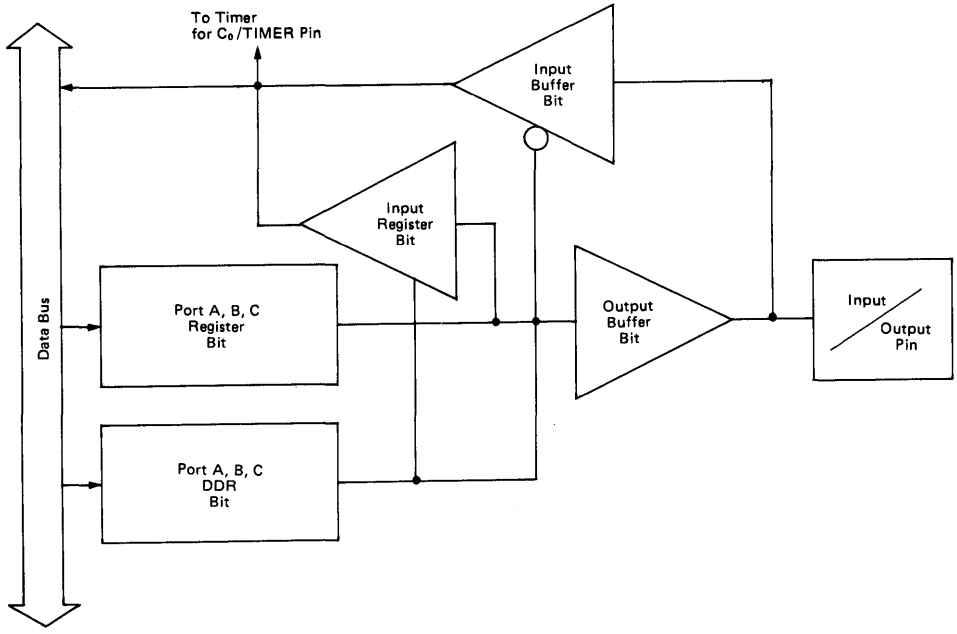


Figure 12 Interrupt Processing Flowchart

■ INPUT/OUTPUT

The HD6805T2 is provided with 19 I/O pins ($\overline{INT}$ is also an input pin but is used only as an interrupt). The Data Direction Register (DDR) of each pin defines whether it is an input or an output ("1" is an output; "0" is an input).

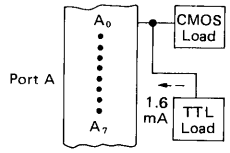


Port A, B, C DDR Bit	1	1	0
Port A, B, C Register Bit	0	1	0/1
Output State	0	1	3-State*
Input to MCU	0	1	Pin

* Except Port A (CMOS drive enable option)

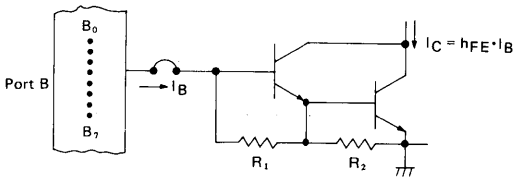
Figure 13 Port A, B, C Block Diagram

The DDR is initialized to all "0"'s by resetting and all pins become input. The output registers, not initialized by resetting, can be set before programming the DDR to prevent undefined level. Fig. 13 shows the latched output data may be read as input data in the case programmed as an output regardless of the logic levels at the output pin due to output loading. If programmed as an output, port B can sink 10 mA (V_{OL} max = 1V) and source 1 mA on each pin.



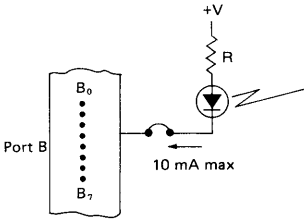
(a)

Port A: Drives CMOS and TTL loads directly when programmed as an output. Internal pull-up devices are included for CMOS drivability. For more details, refer to ELECTRICAL CHARACTERISTICS.



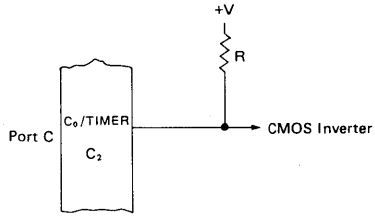
(b)

Port B: Drives Darlington base directly when programmed as an output.



(c)

Port B: Drives LED(s) directly when programmed as an output.



(d)

Port C: Drives CMOS by external pull-up resistors when programmed as an output.

Figure 14 Typical Port Connections

All I/O lines, either for inputs or outputs, are TTL compatible. Both ports B and C as inputs are CMOS compatible; port A as outputs is CMOS compatible by mask option. Some of the port connections are shown in Fig. 14. Fig. 2 shows the data register and the DDR's address.

Caution

The DDR's corresponding to ports A, B and C (at $004, $005 and $006) are provided for write operation only; the read operation is not defined. BSET and BCLR for read/modify/write operations don't set or clear DDRs, but unaffected bits can be set, so it is recommended to use a single-store instruction to write into DDR's.

The latched output data can be written as shown in Fig. 13. When writing to a port, any data can be written even in its DDR "input" mode. This initializes the data registers to prevent any uncertain output. But read/modify/write instructions should be handled carefully because read data depends on the I/O level of the pin with the DDR in the input (0) mode and on the latched output data with the DDR in the output (1) mode.

■ PHASE LOCK LOOP (PLL)

The HD6805T2 has a Phase Lock Loop (PLL) which consists of a 14-bit binary variable divider, a 10-phase reference divider, a frequency and phase comparator which has a three-state output, and the circuits which prevent "backlash" in phase lock states.

Fig. 15 gives an easily established frequency synthesizer system driven by a voltage-controlled oscillator when connecting an adequate high-frequency prescaler and an active integrator. The equations controlling the PLL is shown in Fig. 16.

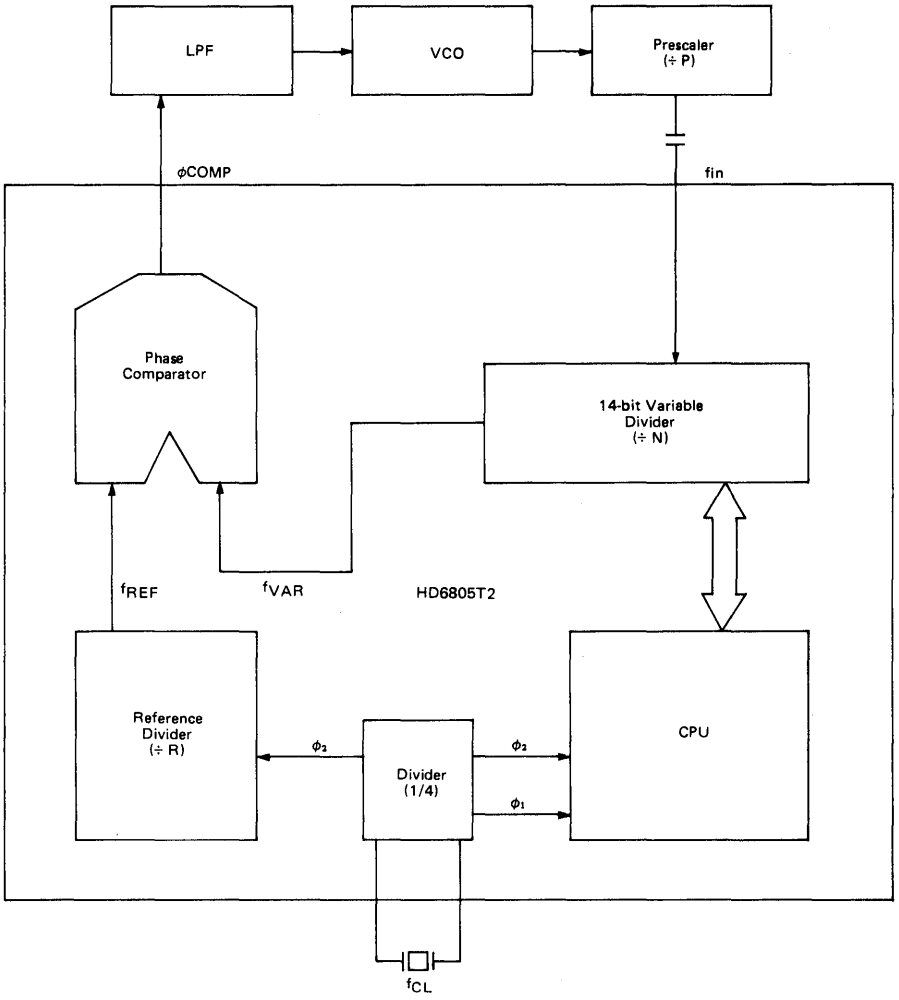


Figure 15 Phase Lock Loop as RF Frequency Synthesizer

For a system in lock: $f_{VAR} = f_{REF}$
since $f_{in} = f_{VAR} \cdot N$, $f_{VCO} = f_{in} \cdot P$ (P = Prescaler diving ratio),
$f_{VCO} = f_{REF} \cdot P \cdot N$

Minimum frequency step $= \frac{\Delta f_{VCO}}{\Delta N} = f_{REF} \cdot P$

e.g.: f_{CL} = 4.00 MHz, R = 2^{10} (R=the reference dividing ratio), P=64

$\frac{\Delta f_{VCO}}{\Delta N}$ = 62.5 kHz, f_{REF} = 976.5 Hz

Figure 16 Principal PLL Equations

■ VARIABLE DIVIDER

The variable devider, a 14-bit binary down counter, sends/receives data to/from the CPU through read/write registers which are located at $00A in the case of the Least Significant (LS) byte and $00B in that of the Most Significant (MS) byte. "1" is always read from the high-order two bits of the $00B register. The variable divider counting zero will generate a preset pulse f_{VAR}. Fig. 17 provides a PLL block diagram where the 14-bit latch is reloaded to the variable divider.

When the $00A register is being written into, data is transferred from $00A and $00B registers to the latch, outside the preset time. Assume that 6-bit data is transferred to $00B register. The data is transferred to the variable divider only when $0A register is next written into. An errorless data transfer in the fine tuning operation is shown in Fig. 18.

The 14-bit latch is activated synchronously with the communication between the CPU and the variable divider, which are asynchronous devices.

When switching on, the PLL registers and the variable divider are fixed to "1".

The variable frequency input pin, f_{in}, is self biased requiring an ac coupled signal with a normal swing of 1.2V. As the input frequency of f_{in} varies with the appropriate prescaler, the whole TV frequency spectrum may be included.

■ REFERENCE DIVIDER

A reference frequency, f_{REF}, is made by the 10-phase binary counter and is compared with the variable divider output. This reference divider is mask optional and the user can select any frequency as shown in Fig. 17.

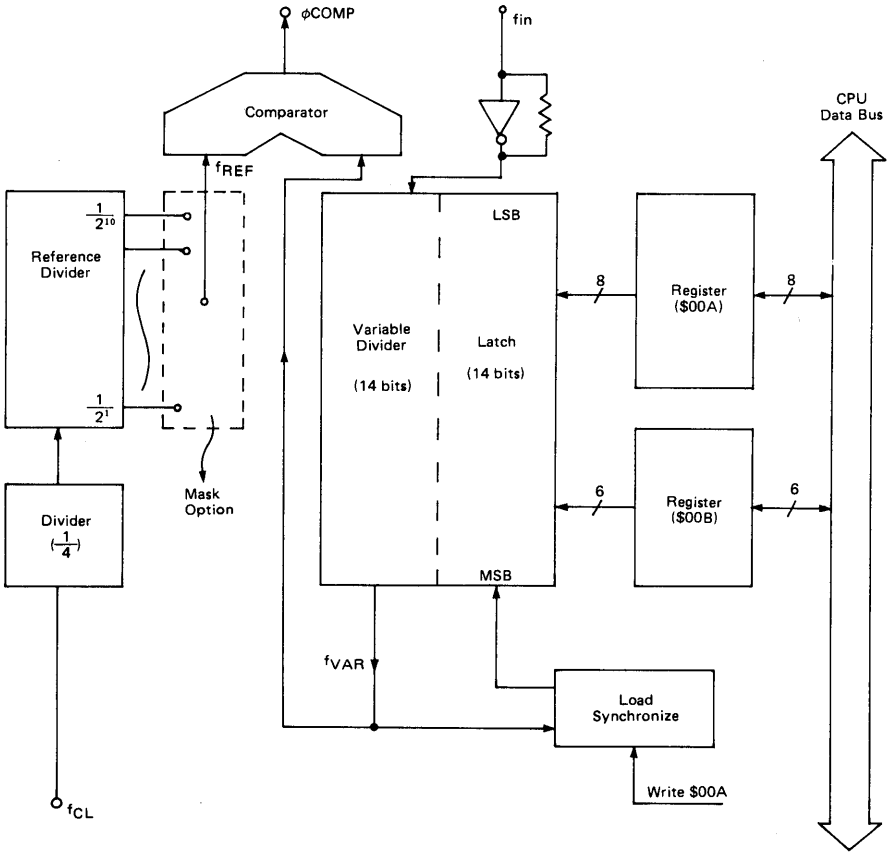


Figure 17 PLL Block Diagram

```
CTD     TST     PLLA    Check if Low byte = "00"
        BNE     CTD1    If not decrement only Low byte
        DEC     PLLB    Decrement High byte
CTD1    DEC     PLLA    Decrement Low byte
                .
                .
                .
CTU     LDA     PLLA
        INCA            Check if Low byte = "FF"
        BNE     CTU1    If not increment only Low byte
        INC     PLLB    Increment High byte
CTU1    INC     PLLA    Increment Low byte
```

Figure 18 Typical Fine Tune Example

■ PHASE COMPARATOR

Both the frequency and phase of f_{VAR} and f_{REF} are compared by the phase comparator, whose relation will generate ϕCOMP, a three-level output, shown in Figs. 19 and 20. ϕCOMP is combined, amplified and the dc voltage is supplied to the voltage control oscillator.

Internal transfer delays will prevent linear features in the stable area. Non-linear characteristics are shown by the phase comparators and this leads to a "backlash" effect which will cause sideband and FM distortion. Insertion of a very short pulse into the device will prevent this. On insertion, the loop tries to cancel the pulse to carry the phase comparator to the linear area shown in Fig. 21.

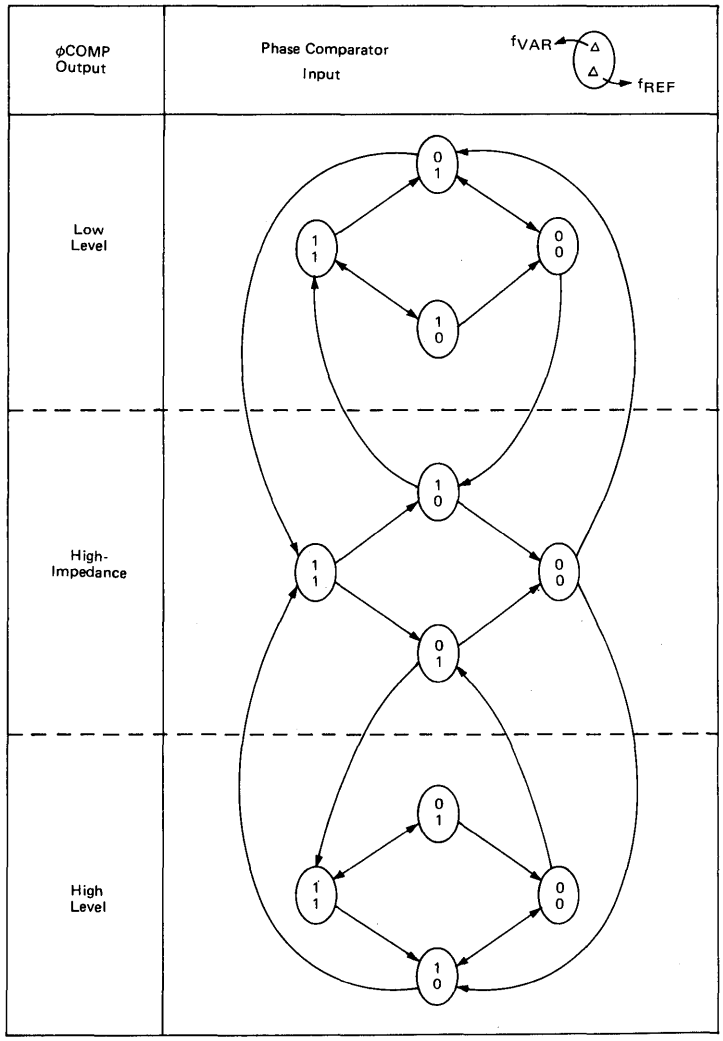


Figure 19 State Diagram (Phase Comparator)

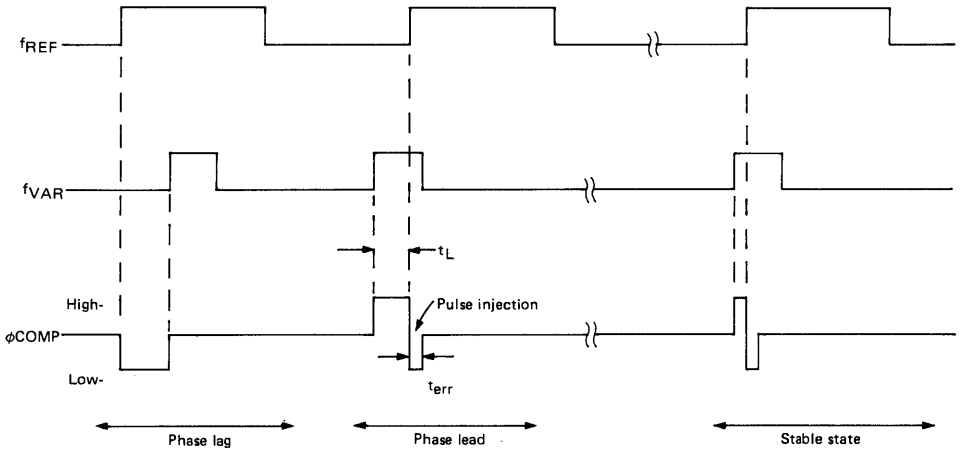


Figure 20 Input/Output Waveform (Phase Comparator)

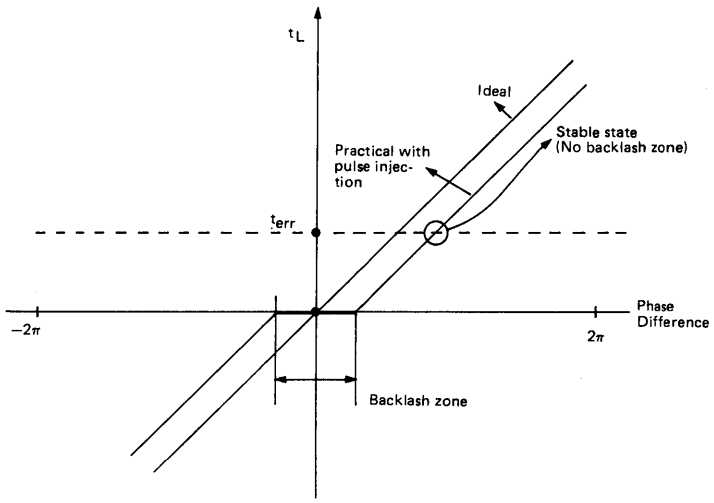


Figure 21 Phase Comparator with Pulse Injection

■ BIT MANIPULATION

With one instruction (BSET, BCLR) the HD6805T2 MCU can set or clear any single bit of the RAM and of the I/O ports (except the Data Direction Registers: — See Caution in the "INPUT/OUTPUT" paragraph). Except for the DDRs, any RAM or I/O bit in page zero including ROM, can be tested with the BRSET and BRCLR instructions, and the program can branch as a result of the state. The carry bit is equal to the value of the bit referenced by BRSET and BRCLR. A rotate instruction is used to pile up serial input data in a RAM location or register. Since the MCU deal with any bit in RAM, ROM and I/O, the user can use the bits in RAM as flags and handle I/O bits as control lines.

See Figure 22. It shows the usefulness of the bit manipulation and test instructions. The program is made up, on the assumption that the MCU is to communicate with an external serial device. The external serial device is provided with a data busy signal, a data input line, and a clock line to clock data one bit at a time, LSB first, out of the MCU.

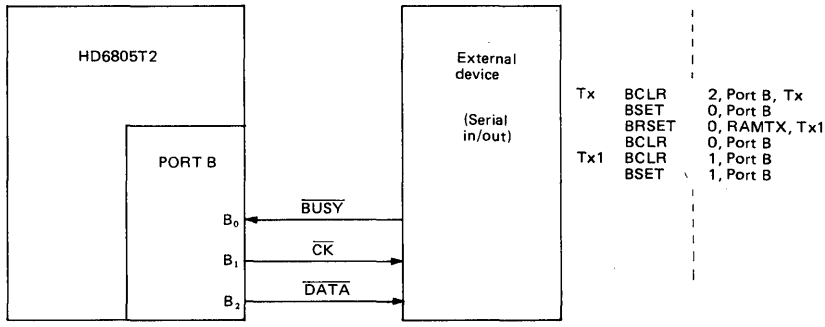


Figure 22 Bit Manipulation Example

■ ADDRESSING MODES

The MCU is provided with ten addressing modes for programming. The following describes them briefly.

• Immediate

Refer to Figure 23. The immediate addressing mode accesses constants which don't change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

• Direct

Refer to Figure 24. In direct addressing, the address of the operand is contained in the second byte, and the user can directly address the lowest 256 bytes in the memory. All RAM space, I/O registers and 128 bytes of ROM, are located in page zero, to take advantages of this efficient memory addressing mode.

• Extended

Refer to Figure 25. Extended addressing is used to reference any location in the memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instruction are three bytes long.

• Relative

Refer to Figure 26. The relative addressing mode applies only to branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch occurs. EA = (PC) + 2 + Rel. Rel shows the contents of the 7-bit signed bit location following the instruction opcode. If branching doesn't occur, Rel = 0. When a branch takes place, the program goes to somewhere within the range of +129 bytes to 126 of the present instruction. These instructions are two bytes long.

• Indexed (No offset)

Refer to Figure 27. This addressing mode accesses the lowest 256 bytes of the memory. These instruction are one byte long and their EA is the contents of the index register.

• Indexed (8-bit offset)

Refer to Figure 28. The EA is the sum of the contents of the byte following the opcode and the contents of the indexed register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

• Indexed (16-bit offset)

Refer to Figure 29. The EA is the sum of the contents of the two bytes following the opcode and the contents of the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

• Bit Set/Clear

Refer to Figure 30. This addressing mode applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode select the bit to be set or cleared. The byte following the opcode specifies the address in page zero.

• Bit Test and Branch

Refer to Figure 31. This addressing mode applies to instructions which can test any bit in the first 256 locations ($00 – $FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within the byte to be tested is addressed by the low-order three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the tested bit is written into the carry bit in the code register.

• Implied

Refer to Figure 32. The implied mode of addressing has not EA. All the information necessary to execute an instruction is included in the opcode. Direct operations on the accumulator and the index register are contained in this addressing mode. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are only one byte long.

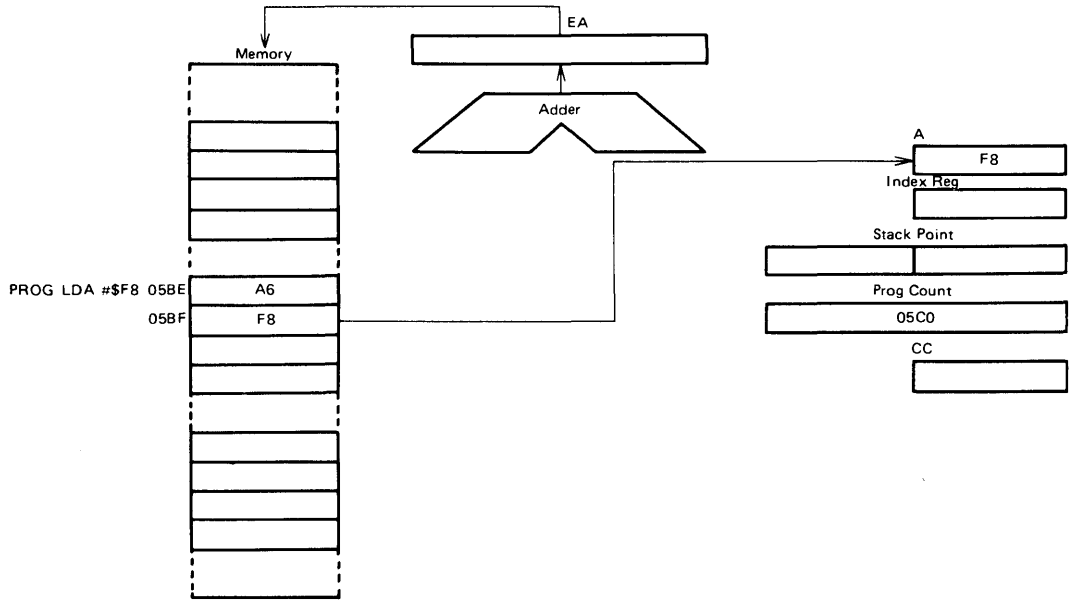


Figure 23 Immediate Addressing Example

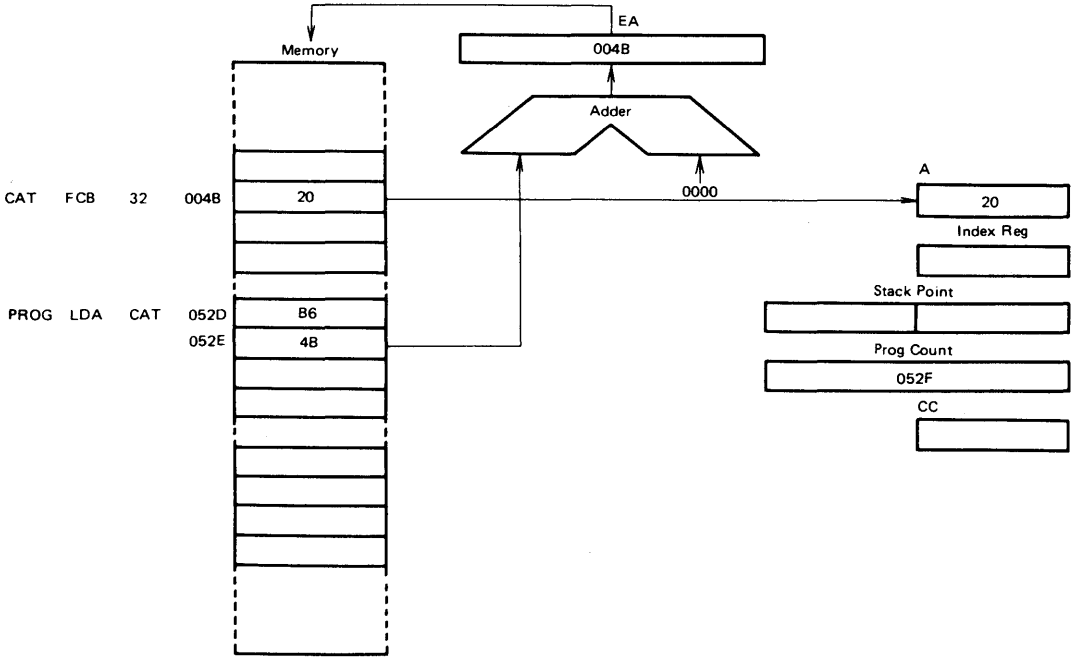


Figure 24 Direct Addressing Example

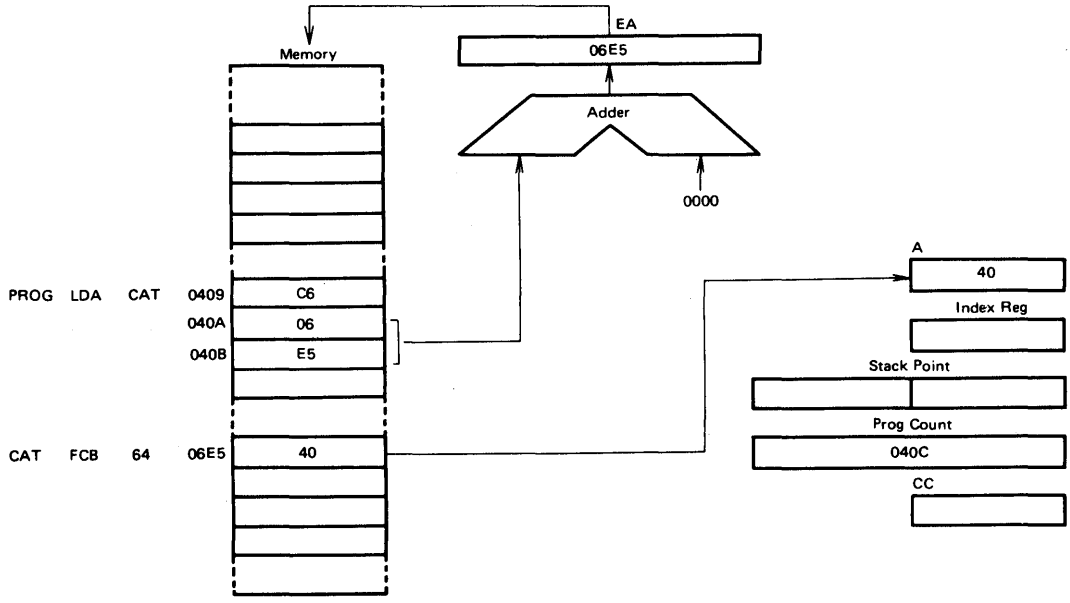


Figure 25 Extended Addressing Example

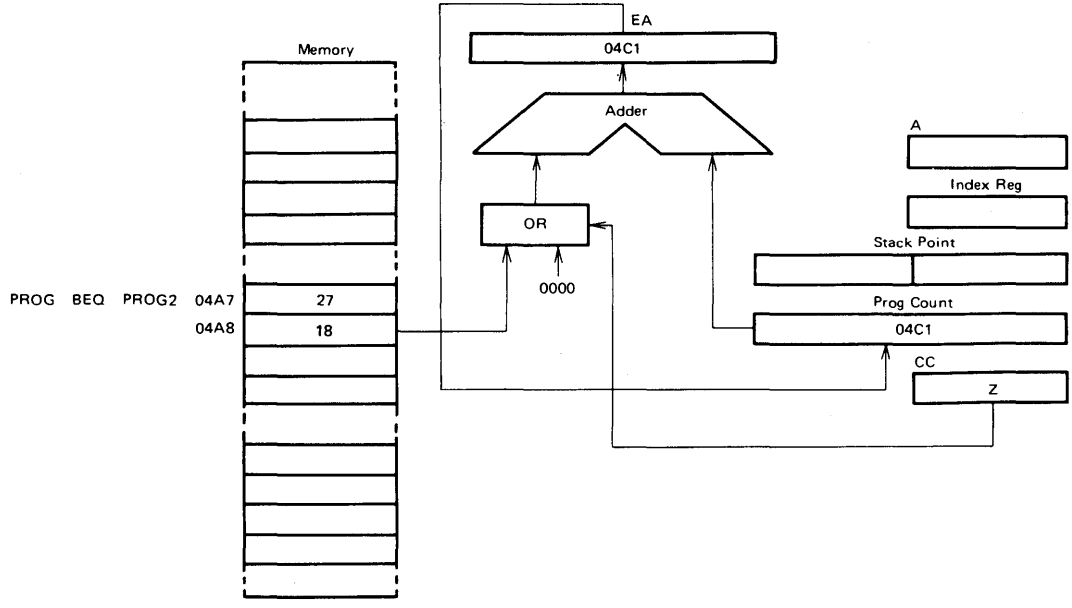


Figure 26 Relative Addressing Example

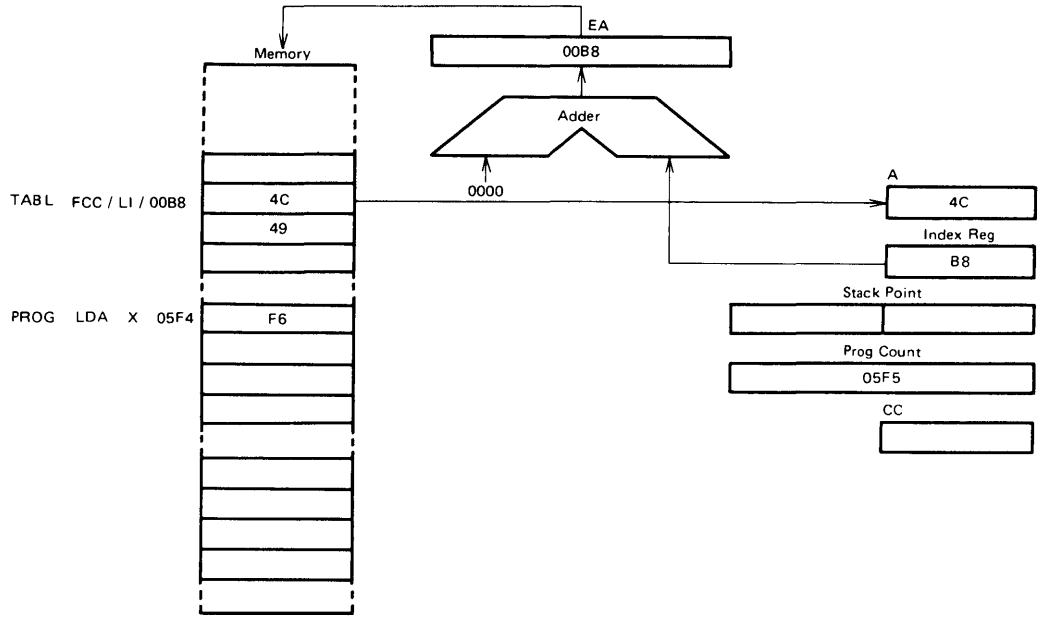


Figure 27 Indexed (No Offset) Addressing Example

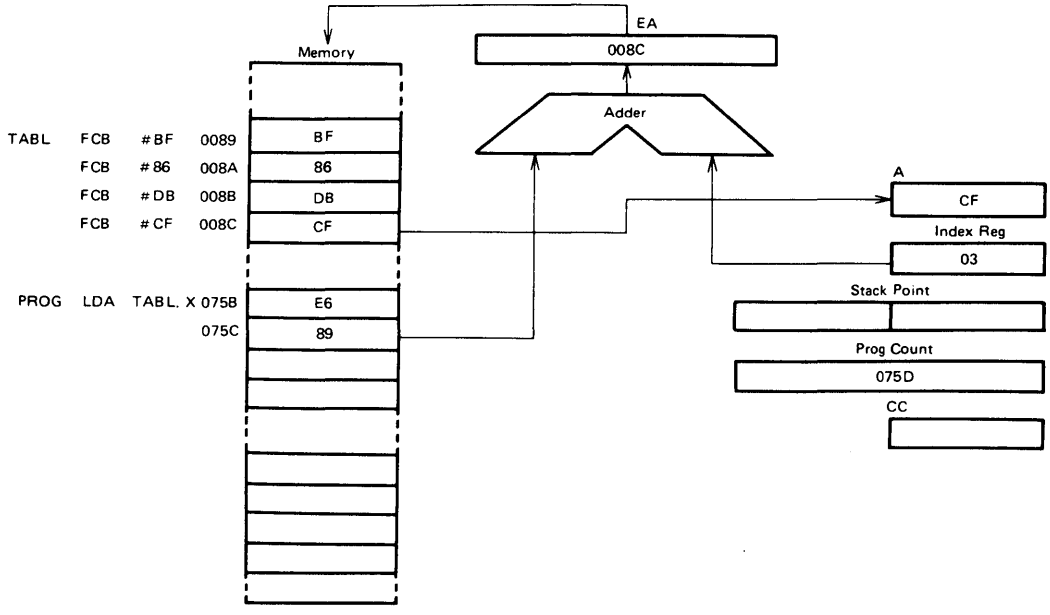


Figure 28 Indexed (8-Bit Offset) Addressing Example

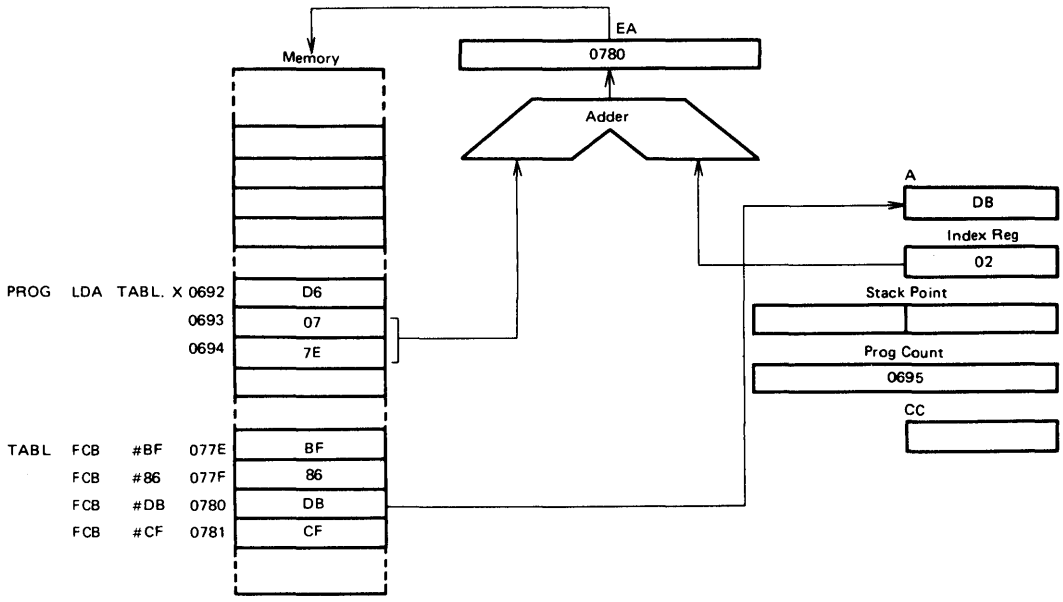


Figure 29 Indexed (16-Bit Offset) Addressing Example

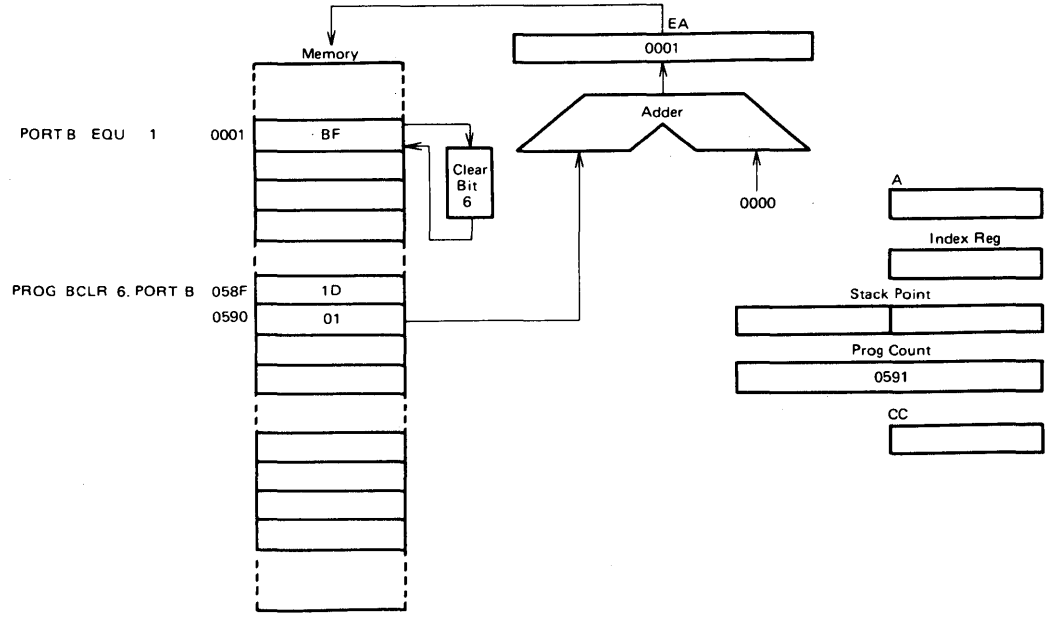


Figure 30 Bit Set/Clear Addressing Example

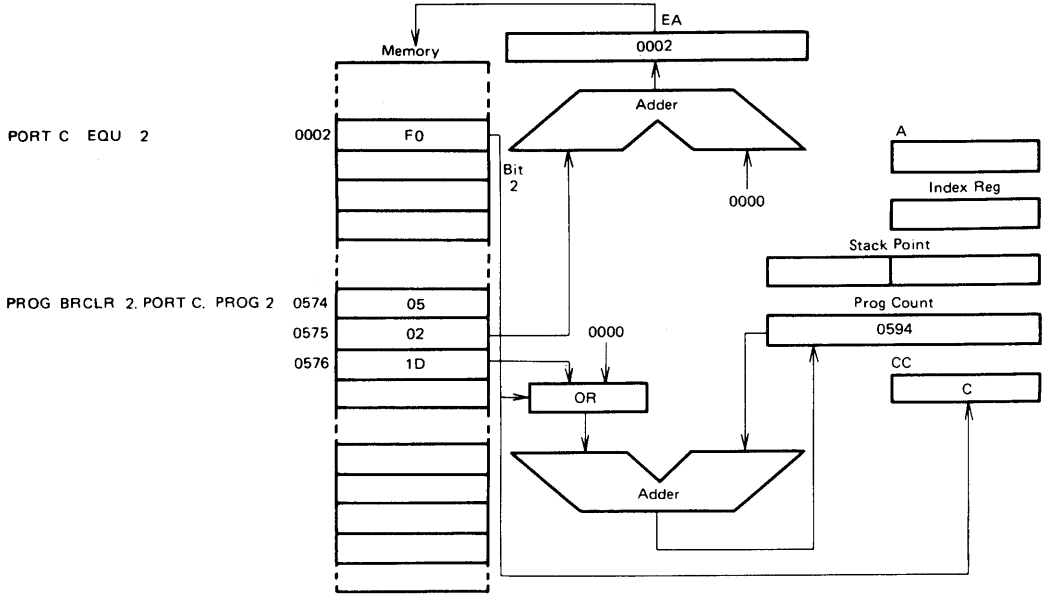


Figure 31 Bit Test and Branch Addressing Example

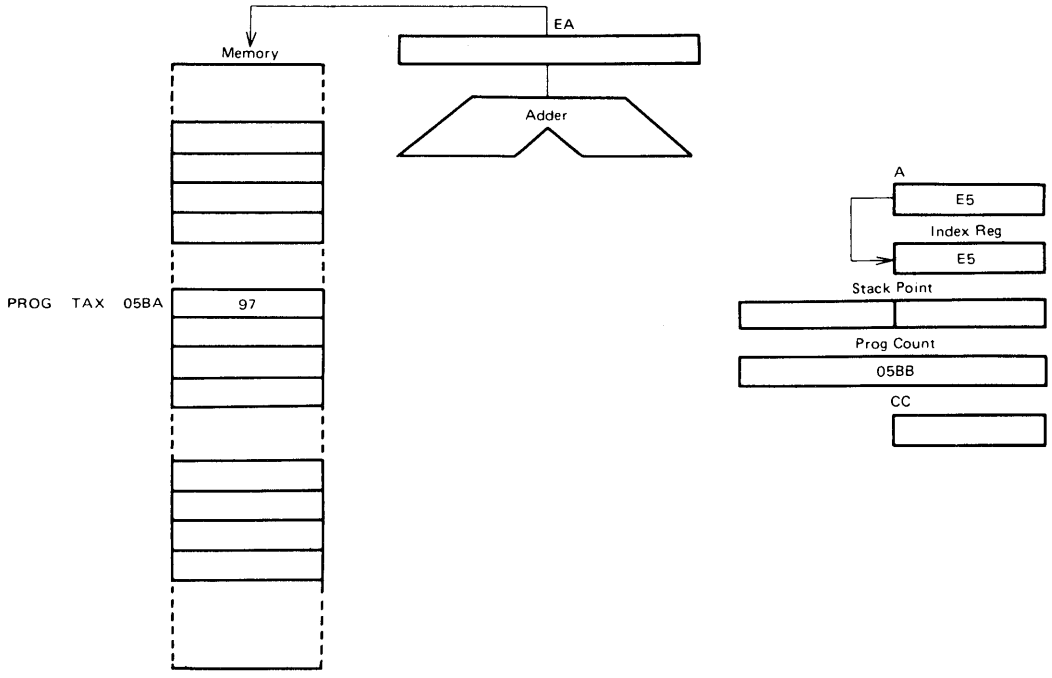


Figure 32 Implied Addressing Example

■ INSTRUCTION SET

There are 59 basic instructions classified into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following briefly describes each type. Individual tables present all the instruction in a given type.

• Register/Memory Instructions

Refer to Table 2. Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is got from the memory by using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand.

• Read/Modify/Write Instructions

Refer to Table 3. These instructions read a memory address or a register, modify or test its contents, and write the modified result back into the memory or to the zero (TST) instruction does not execute "write," it is an exception to these instructions.

• Branch Instructions

Refer to Table 4. These instructions cause a branch from the program when a certain condition is met.

• Bit Manipulation Instructions

Refer to Table 5. The bit manipulation instructions are applied to any bit in the first 256 bytes of the memory. Some of these instructions set or clear the bits. The others perform the test and branch operations.

• Control Instructions

Refer to Table 6. These instructions control the MCU operations during program execution.

• Alphabetical Listing

All the instructions above are listed in alphabetical order in Table 7.

• Opcode Map

Table 8 is an opcode map for the instructions used on the MCU.

Table 2 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 3 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 4 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 5 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 6 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		○	○	○		○	○	○			∧	•	∧	∧	∧
ADD		○	○	○		○	○	○			∧	•	∧	∧	∧
AND		○	○	○		○	○	○			•	•	∧	∧	•
ASL	○		○			○	○				•	•	∧	∧	∧
ASR	○		○			○	○				•	•	∧	∧	∧
BCC					○						•	•	•	•	•
BCLR									○		•	•	•	•	•
BCS					○						•	•	•	•	•
BEQ					○						•	•	•	•	•
BHCC					○						•	•	•	•	•
BHCS					○						•	•	•	•	•
BHI					○						•	•	•	•	•
BHS					○						•	•	•	•	•
BIH					○						•	•	•	•	•
BIL					○						•	•	•	•	•
BIT		○	○	○		○	○	○			•	•	∧	∧	•
BLO					○						•	•	•	•	•
BLS					○						•	•	•	•	•
BMC					○						•	•	•	•	•
BMI					○						•	•	•	•	•
BMS					○						•	•	•	•	•
BNE					○						•	•	•	•	•
BPL					○						•	•	•	•	•
BRA					○						•	•	•	•	•
BRN					○						•	•	•	•	•
BRCLR										○	•	•	•	•	∧
BRSET										○	•	•	•	•	∧
BSET									○		•	•	•	•	•
BSR					○						•	•	•	•	•
CLC	○										•	•	•	•	0
CLI	○										•	0	•	•	•
CLR	○		○			○	○				•	•	0	1	•
CMP		○	○	○		○	○	○			•	•	∧	∧	∧
COM	○		○			○	○				•	•	∧	∧	1
CPX		○	○	○		○	○	○			•	•	∧	∧	∧
DEC	○		○			○	○				•	•	∧	∧	•
EOR		○	○	○		○	○	○			•	•	∧	∧	•
INC	○		○			○	○				•	•	∧	∧	•
JMP			○	○		○	○	○			•	•	•	•	•
JSR			○	○		○	○	○			•	•	•	•	•
LDA		○	○	○		○	○	○			•	•	∧	∧	•
LDX		○	○	○		○	○	○			•	•	∧	∧	•

(to be continued)

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	○		○			○	○				•	•	∧	∧	∧
LSR	○		○			○	○				•	•	0	∧	∧
NEG	○		○			○	○				•	•	∧	∧	∧
NOP	○										•	•	•	•	•
ORA		○	○	○		○	○	○			•	•	∧	∧	•
ROL	○		○			○	○				•	•	∧	∧	∧
ROR	○		○			○	○				•	•	∧	∧	∧
RSP	○										•	•	•	•	•
RTI	○										?	?	?	?	?
RTS	○										•	•	•	•	•
SBC		○	○	○		○	○	○			•	•	∧	∧	∧
SEC	○										•	•	•	•	1
SEI	○										•	1	•	•	•
STA			○	○		○	○	○			•	•	∧	∧	•
STX			○	○		○	○	○			•	•	∧	∧	•
SUB		○	○	○		○	○	○			•	•	∧	∧	∧
SWI	○										•	1	•	•	•
TAX	○										•	•	•	•	•
TST	○		○			○	○				•	•	∧	∧	•
TXA	○										•	•	•	•	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

Table 8 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(–3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. "–" is an undefined operation code.

2. The numbers in the lowermost row represent the number of bytes and cycles required (number of bytes/number of cycles). The number of cycles for the mnemonics asterisked (*) are as follows.

RT1	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. Add the parenthesized numbers to the number or cycle of an instruction.

MASK OPTION LIST

Item	Option	Symbol	Check	Remarks
Timer Clock Source	Internal clock	TIMER I	☐	
	External clock	TIMER E	☐	
Timer Prescaler	Divide by 1	TPR 1	☐	
	Divide by 2	TPR 2	☐	
	Divide by 4	TPR 4	☐	
	Divide by 8	TPR 8	☐	
	Divide by 16	TPR 16	☐	
	Divide by 32	TPR 32	☐	
	Divide by 64	TPR 64	☐	
	Divide by 128	TPR 128	☐	
Reference Divider Ratio	Divide by 2	RDR 2	☐	
	Divide by 4	RDR 4	☐	
	Divide by 8	RDR 8	☐	
	Divide by 16	RDR 16	☐	
	Divide by 32	RDR 32	☐	
	Divide by 64	RDR 64	☐	
	Divide by 128	RDR 128	☐	
	Divide by 256	RDR 256	☐	
	Divide by 512	RDR 512	☐	
	Divide by 1024	RDR 1024	☐	
Low Voltage Inhibit	Disable	LVID	☐	
	Enable	LVIE	☐	
Output Port A	CMOS	CMOS	☐	
	TTL	TTL	☐	

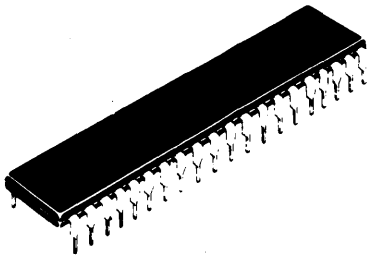
HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

HD6805U1
MCU (Microcomputer Unit)

The HD6805U1 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, I/O and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD6800-based instruction set.

The following are some of the hardware and software highlights of the MCU.

HD6805U1P



(DP-40)

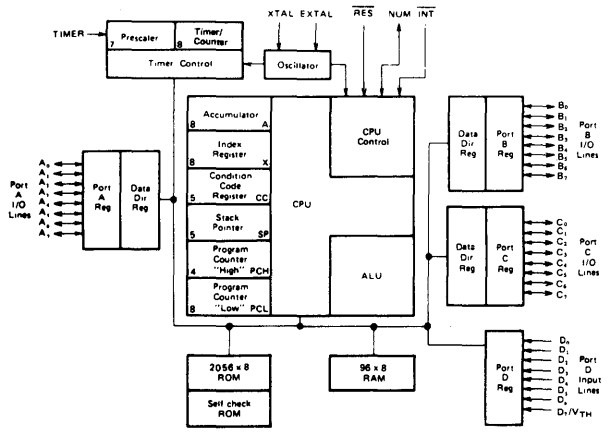
■ HARDWARE FEATURES

- 8-Bit Architecture
- 96 Bytes of RAM
- Memory Mapped I/O
- 2056 Bytes of User ROM
- Internal 8-Bit Timer with 7-Bit Prescaler
- Vectored Interrupts – External and Timer
- 24 I/O Ports + 8 Input Port
  (8 Lines LED Compatible; 7 Bits Comparator Inputs)
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- Easy for System Development and Debugging
- 5 Vdc Single Supply

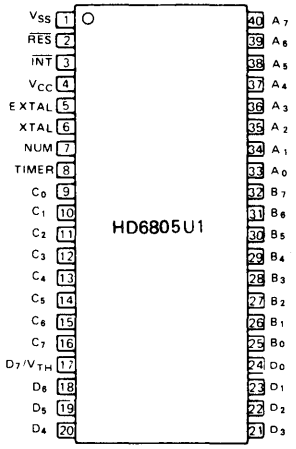
■ SOFTWARE FEATURES

- Similar to HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible Instruction Set with MC6805P2

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in} *	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +12.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT}$			3.0	–	V_{CC}	V
	All Other			2.0	–	V_{CC}	V
Input "High" Voltage (Timer)	Timer Mode			2.0	–	V_{CC}	V
	Self-Check Mode			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT}$			-0.3	–	0.8	V
	EXTAL(Crystal Mode)			-0.3	–	0.6	V
	All Other			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	700	mW
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT}$			-50	–	50	μA
	EXTAL(Crystal Mode)			-1200	–	0	μA

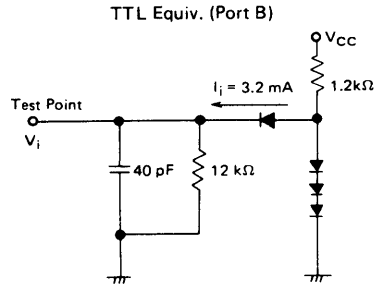
• AC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0 ~ +70°C, unless otherwise noted.)

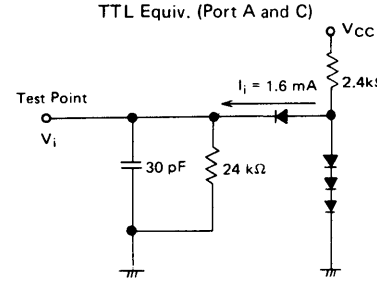
Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
Oscillation Frequency (External Resistor Mode)		f_{EXT}	R_{CP}=15.0kΩ±1%	–	3.4	–	MHz
$\overline{INT}$ Pulse Width		t_{IWL}		t_{cyc}+250	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		t_{cyc}+250	–	–	ns
TIMER Pulse Width		t_{TWL}		t_{cyc}+250	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20%, R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2 μF	100	–	–	ms
Input Capacitance	XTAL	C_{in}	V_{in}=0V	–	–	35	pF
	All Other			–	–	10	pF

• PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = −10 μA	3.5	–	–	V
			I_{OH} = −100 μA	2.4	–	–	V
	Port B		I_{OH} = −200 μA	2.4	–	–	V
			I_{OH} = −1 mA	1.5	–	–	V
	Port C		I_{OH} = −100 μA	2.4	–	–	V
Output "Low" Voltage	Port A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.4	V
	Port B		I_{OL} = 3.2 mA	–	–	0.4	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Port A, B, C, and D*	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	−500	–	–	μA
			V_{in} = 2V	−300	–	–	μA
	Port B, C, and D		V_{in} = 0.4V ~ V_{CC}	− 20	–	20	μA
Input "High" Voltage	Port D** (D_0 ~ D_6)	V_{IH}		–	V_{TH}+0.2	—	V
Input "Low" Voltage	Port D** (D_0 ~ D_6)	V_{IL}		–	V_{TH}−0.2	—	V
Threshold Voltage	Port D**(D_7)	V_{TH}		0	—	0.8×V_{CC}	V

* Port D as digital input
** Port D as analog input





(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074Ⓗ or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

• V_{CC} and V_{SS}

Power is supplied to the MCU using these two pins. V_{CC} is +5.25V ±0.5V. V_{SS} is the ground connection.

• $\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

• XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4 MHz maximum), a resistor or an external signal can be connected to these pins to provide a system clock with various stability/cost tradeoffs. Refer to INTERNAL OSCILLATOR OPTIONS for recommendations about these inputs.

• TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER for additional information about the timer circuitry.

• $\overline{RES}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to RESETS for additional information.

• NUM

This pin is not for user application and should be connected to V_{SS}.

- **Input/Output Lines ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$)**

These 24 lines are arranged into three 8-bit ports (A, B and C). All lines are programmable as either inputs or outputs under software control of the Data Direction Register (DDR). Refer to INPUT/OUTPUT for additional information.

- **Input Lines ($D_0 \sim D_7$)**

These are 8-bit input lines, which has two functions. Firstly, these are TTL compatible inputs, in location $003. The other function is 7 bits comparator, in location $007. Refer to INPUT for more details.

■ MEMORY

The MCU memory is configured as shown in Figure 2. During the processing of an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Figure 3. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

Caution: – Self Test ROM Address Area
Self test ROM locations can not be used for a user program. If the user's program is in this location, it will be removed when manufacturing mask for production.

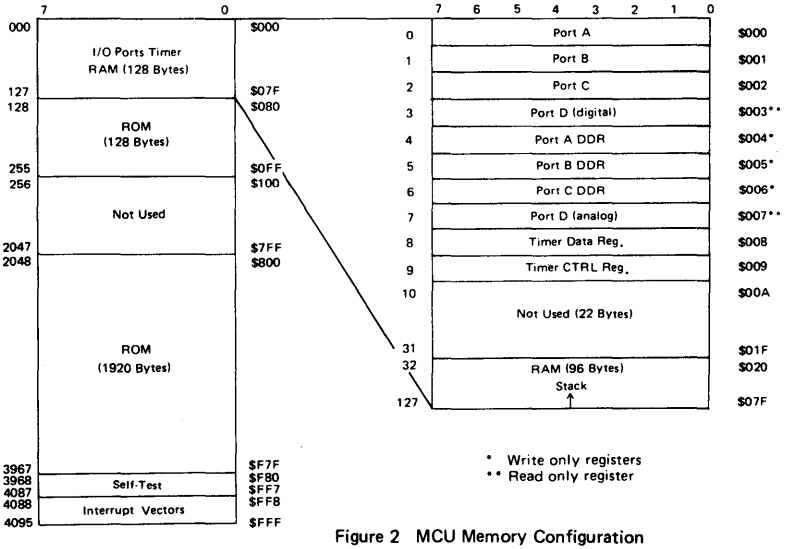


Figure 2 MCU Memory Configuration

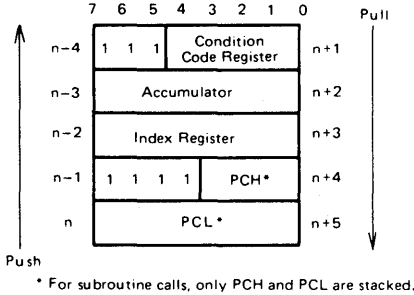


Figure 3 Interrupt Stacking Order

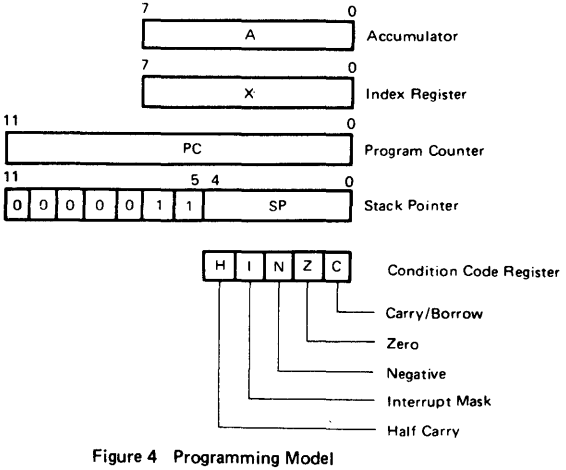


Figure 4 Programming Model

■ REGISTERS

The CPU has five registers available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

- **Accumulator (A)**

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

- **Index Register (X)**

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

- **Program Counter (PC)**

The program counter is a 12-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The six most significant bits of the stack pointer are permanently set to 0000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 which allows the programmer to use up to 15 levels of subroutine calls.

- **Condition Code Register (CC)**

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

Used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in result equal to a logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

■ TIMER

The MCU timer circuitry is shown in Figure 5. The 8-bit counter, the Timer Data Register (TDR), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR) is set. The CPU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the Condition Code Register also prevents a time interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When the internal ϕ_2 signal is selected as the input source, the node a is connected to b (see Fig. 5). In case of the external source, the node b connects with c. When the ϕ_2 signal is used as the source, the clock signal is input to the prescaler while the TIMER input is "High". The source of the clock input is one of the options that has to be specified before manufac-

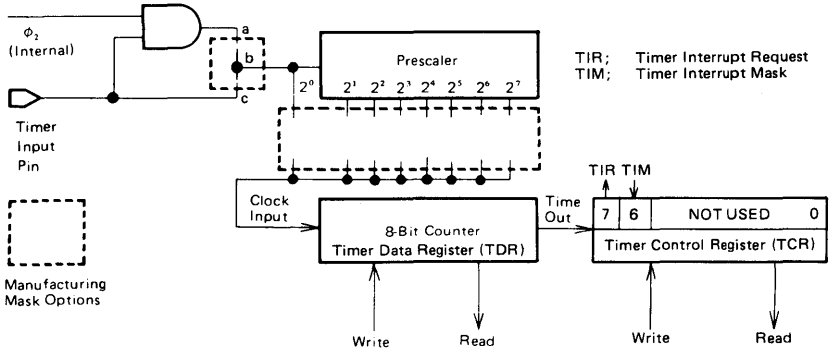


Figure 5 Timer Block Diagram

ture of the MCU. A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter (TDR). The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter (TDR) can be read at any time by reading the TDR. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

The TDR is 8-bit Read/Write Register in location $008. At power-up or reset, the TDR and the prescaler are initialized with all logical ones.

The Timer Interrupt Request bit (bit 7 of the TCR) is set by hardware when timer count reaches zero, and is cleared by program or by hardware reset. The bit 6 of the TCR is writable by program. Both of those bits can be read by CPU.

(NOTE) If the MCU Timer is not used, the TIMER input pin must be grounded.

■ SELF CHECK

The self-check capability of the MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 6 and monitor the output of port C bit 3 for an oscillation of approximately 3Hz. ROM, RAM, TIMER, Interrupts, I/O of Port A, B and C are checked by this capability.

■ RESETS

The MCU can be reset three ways; by initial power-up, by the external reset input ($\overline{RES}$) and by an optional internal low voltage inhibit circuit, see Figure 7. All the I/O port are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum of 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High".

This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 8, typically provides sufficient delay.

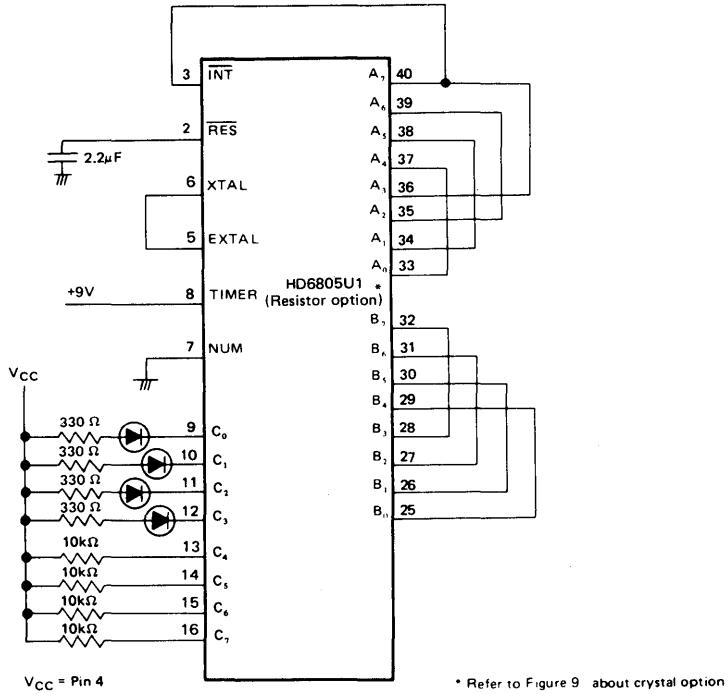


Figure 6 Self Check Connections

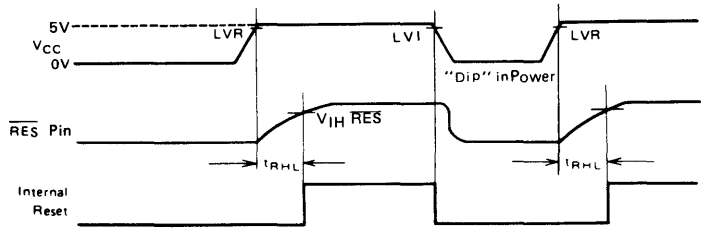


Figure 7 Power Up and $\overline{RES}$ Timing

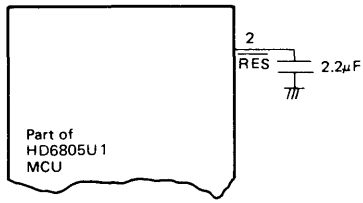


Figure 8 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR OPTIONS

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoff. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different connection methods are shown in Figure 9. Crystal specifications are given in Figure 10. A resistor selection graph is given in Figure 11.

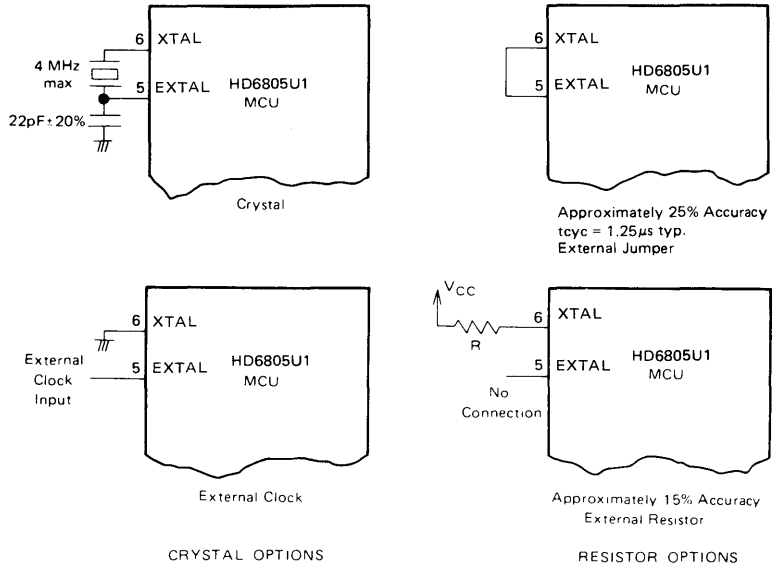
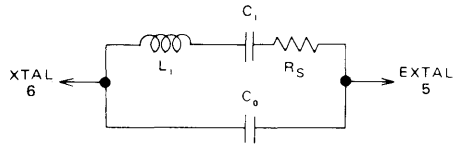


Figure 9 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz
R_S = 60 Ω max.

Figure 10 Crystal Parameters

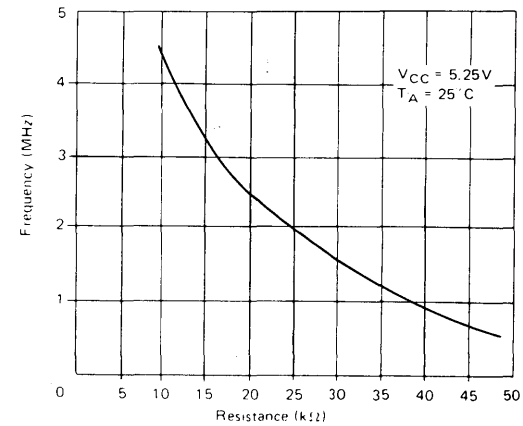


Figure 11 Typical Resistor Selection Graph

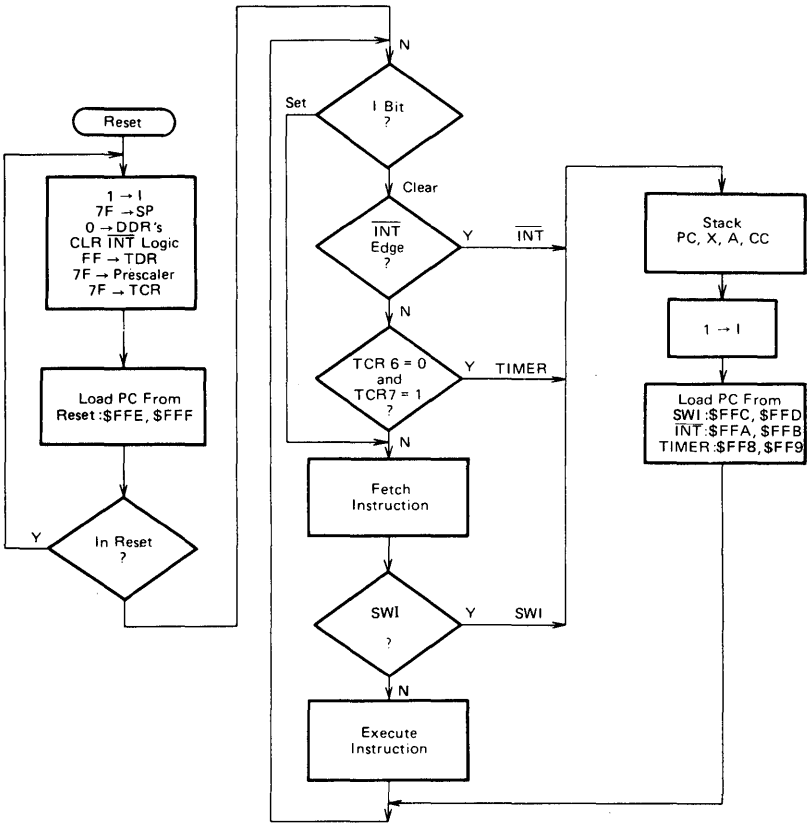
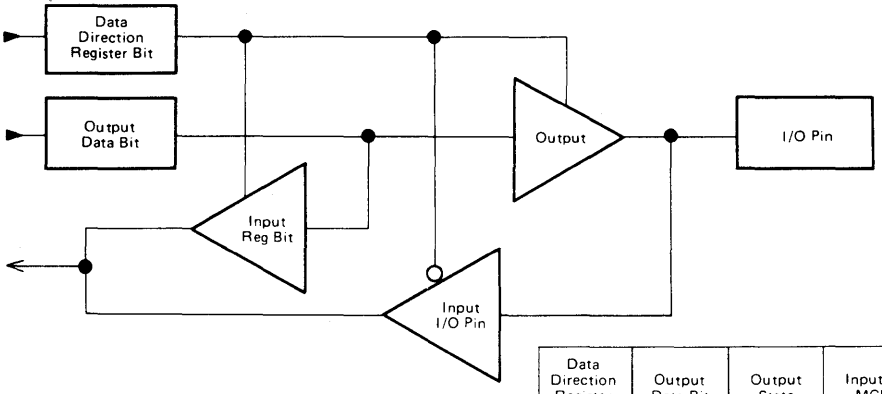


Figure 12 Interrupt Processing Flowchart



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

Figure 13 Typical Port I/O Circuitry

■ INTERRUPTS

The CPU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires 11 cycles. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. Table 1 provides a listing of the interrupts, their priority, and the vector address that contain the starting address of the appropriate interrupt routine.

A flowchart of the interrupt processing sequence is given in Fig. 12.

Table 1 Interrupt Priorities

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$FFE and $FFF
SWI	2	$FFC and $FFD
$\overline{INT}$	3	$FFA and $FFB
TIMER	4	$FF8 and $FF9

■ INPUT/OUTPUT

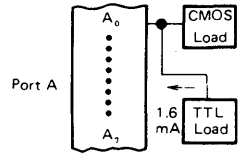
There are 24 input/output pins. All pins are programmable as either inputs or outputs under software control of the corresponding Data Direction Register (DDR). When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (see Fig. 13). When Port B is programmed for outputs, it is capable of sinking 10mA on each pin (V_{OL} = 1V max). All input/output lines are TTL compatible as both inputs and outputs. Port A is CMOS compatible as outputs, and Port B and C lines are CMOS compatible as inputs. Figure 14 provides some examples of port connections.

■ INPUT

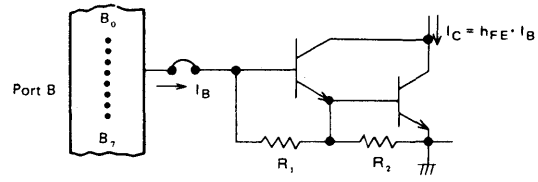
Port D can be used as either 8 TTL compatible inputs or 1 threshold input and 7 analog inputs pins. Fig. 15 (a) shows the construction of port D. The Port D register at location $003 stores TTL compatible inputs, and those in location $007 store the result of comparison D_0 to D_6 inputs with D_7 threshold input. Port D has not only the conventional function as inputs but also voltage-comparison function. Applying the latter, can easily check that 7 analog input electric potential max. exceeds the limit with the construction shown in Fig. 15 (b). Also, using one output pin of MCU, after external capacity is discharged at the preset state, charge the CR circuit of long enough time constant, apply the charging curve to the D_7 pin. The construction described above is shown in Fig. 15 (c). The compared result of D_0 to D_6 is regularly monitored, which gives the analog input electric potential applied to D_0 to D_6 pins from inverted time. This method enables 7 inputs to be converted from analog to digital. Furthermore, combination of two functions gives 3 level voltages from D_0 to D_6. Fig. 15 (d) provides the example when V_{TH} is set to 3.5V.

■ BIT MANIPULATION

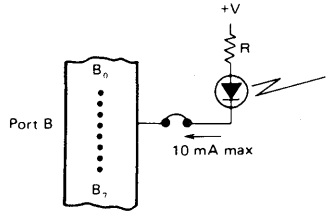
The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 16 illustrates the usefulness of the bit manipulation and test



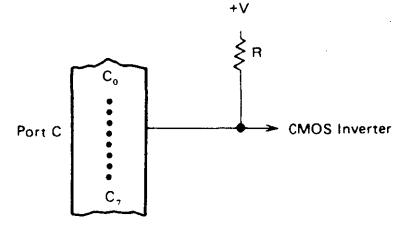
Port A Programmed as output(s),driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s),driving Darlington base directly.
(b)



Port B Programmed as output(s),driving LED(s) directly.
(c)

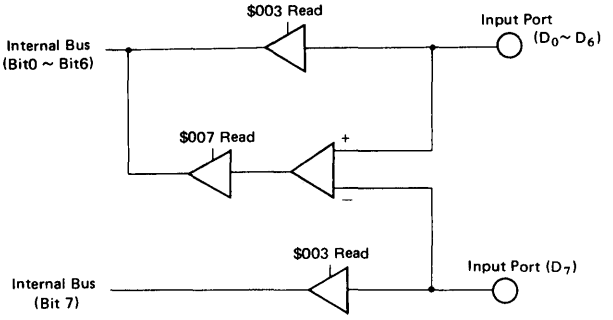


Port C Programmed as output(s), driving CMOS loads, using external pull-up
(d)

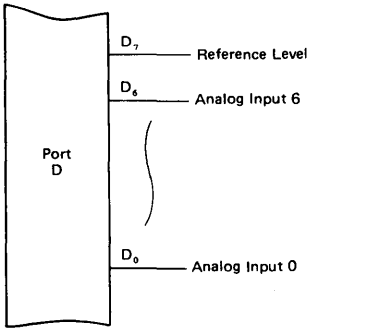
Figure 14 Typical Port Connections

instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which power the controlled hardware.

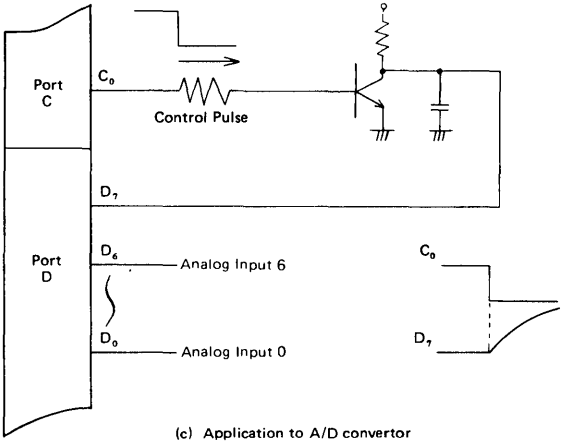
This program, which uses only seven ROM locations, provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer could also be incorporated to provide turn-on at some later time which would permit pulse-width modulation of the controlled power.



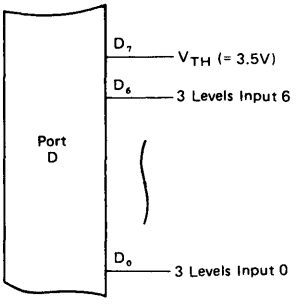
(a) The logic configuration of Port D



(b) Seven analog inputs and a reference level input of Port D



(c) Application to A/D convertor



(d) Application to 3 levels input

Input Voltage	($003)	($007)
0V ~ 0.8V	0	0
2.0V ~ 3.3V	1	0
3.7V ~ V_{CC}	1	1

Figure 15 Configuration and Application of Port D

```
SELF 1    BRCLR 0, PORT A, SELF 1
          BSET 1, PORT A
          BCLR 1, PORT A
```

Figure 16 Bit Manipulation Example

■ ADDRESSING MODES

The CPU has ten addressing modes available for use by the programmer. They are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 17. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 18. In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 19. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 20. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA=(PC)+2+Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken Rel=0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 21. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 22. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 23. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 24. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 25. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($00-$FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 26. The implied mode of addressing has no EA. All the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are one byte long.

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

- **Register/Memory Instructions**

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 2.

- **Read/Modity/Write Instructions**

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 3.

- **Branch Instructions**

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

- **Bit Manipulation Instructions**

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 5.

- **Control Instructions**

The control instructions control the MCU operations during program execution. Refer to Table 6.

- **Alphabetical Listing**

The complete instruction set is given in alphabetical order in Table 7.

- **Opcode Map**

Table 8 is an opcode map for the instructions used on the MCU.

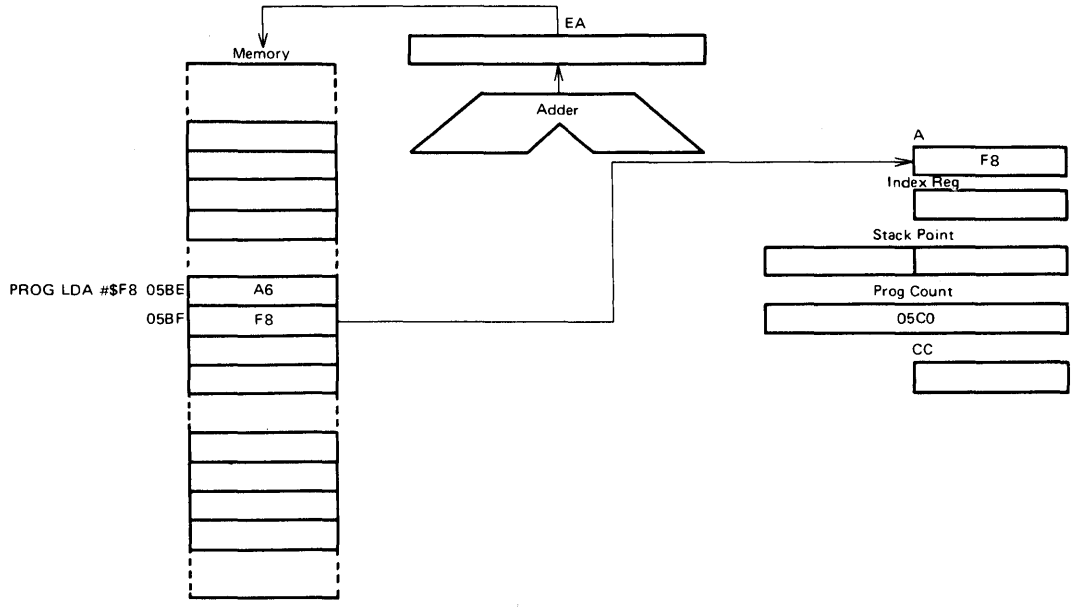


Figure 17 Immediate Addressing Example

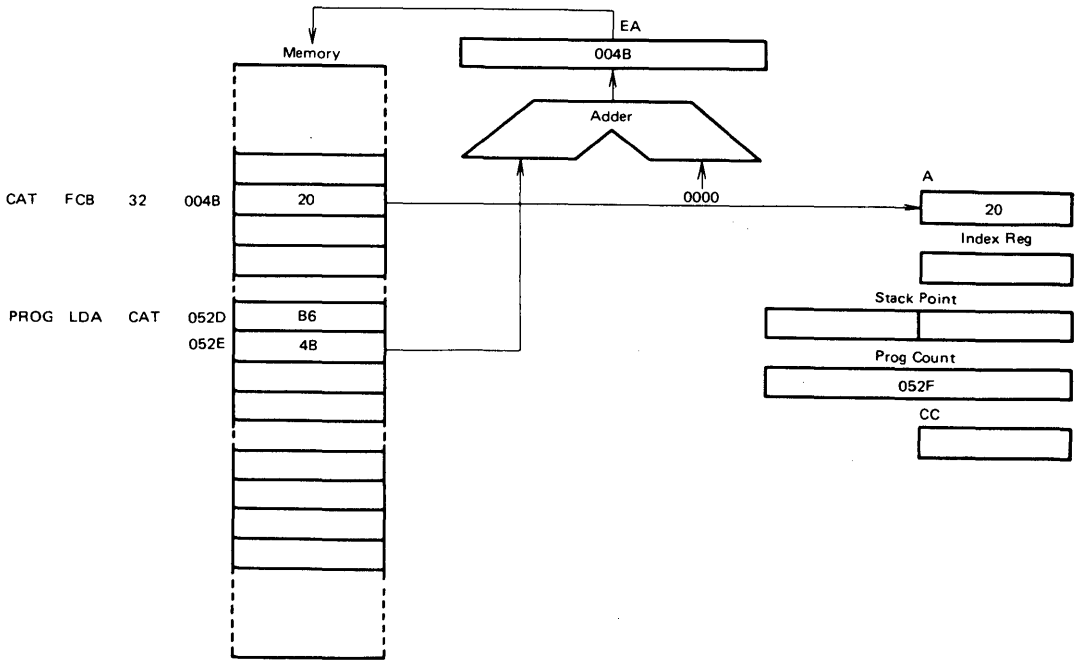


Figure 18 Direct Addressing Example

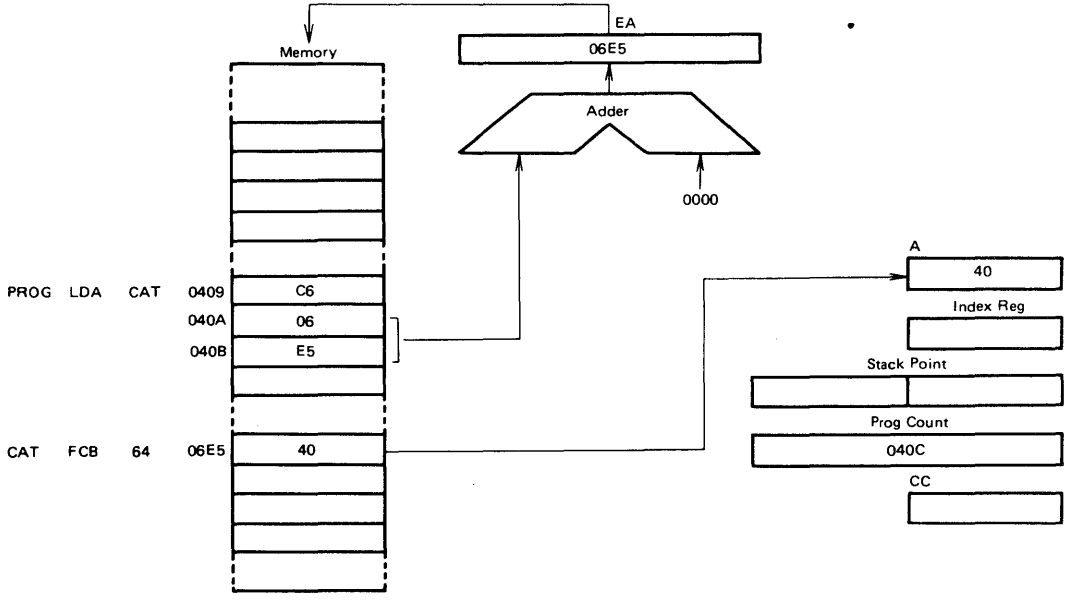


Figure 19 Extended Addressing Example

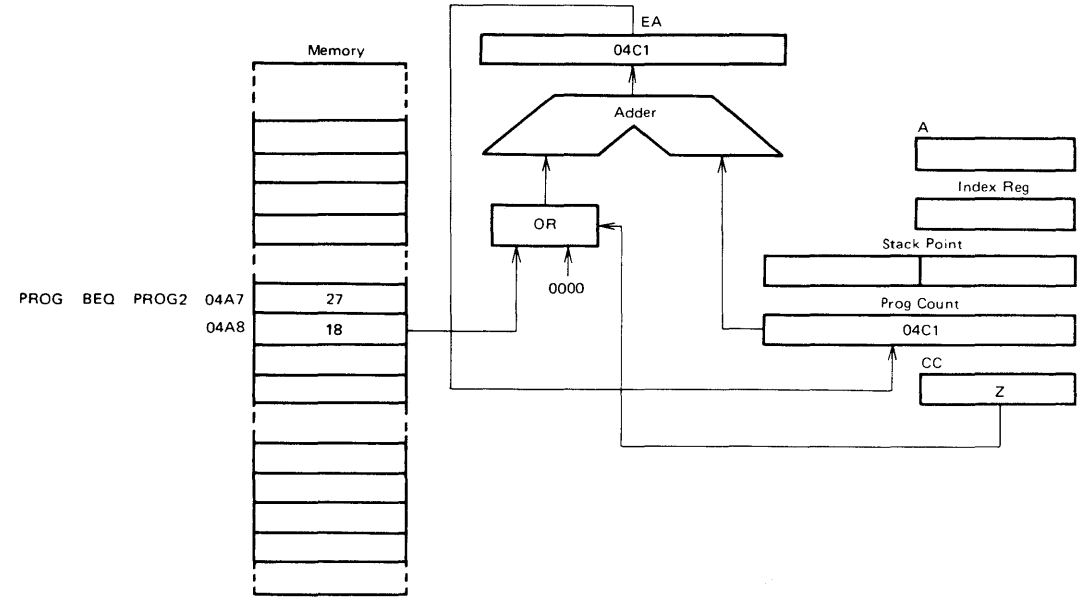


Figure 20 Relative Addressing Example

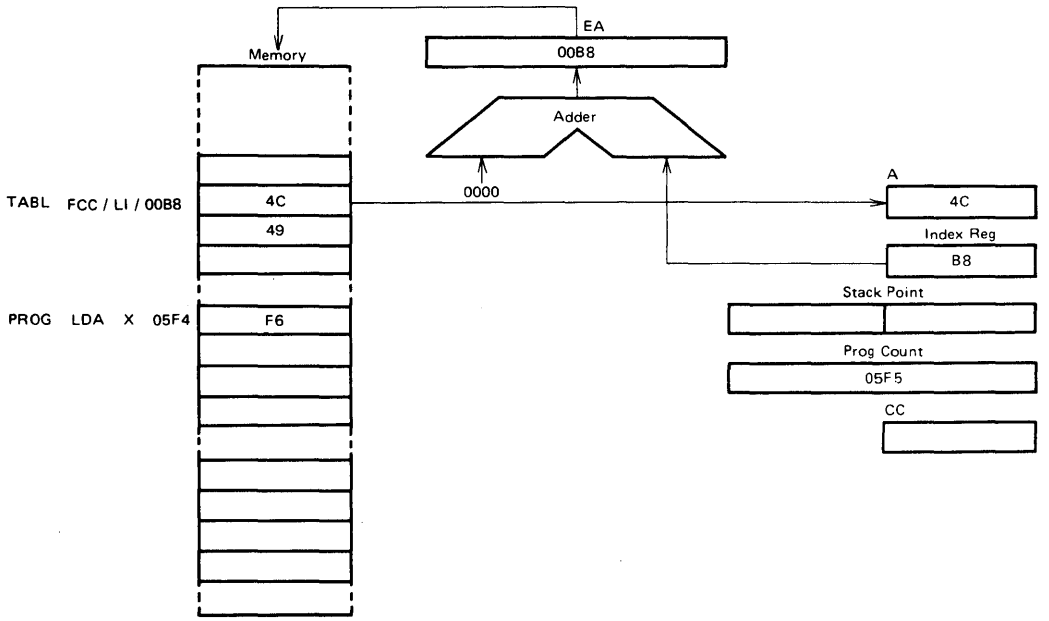


Figure 21 Indexed (No Offset) Addressing Example

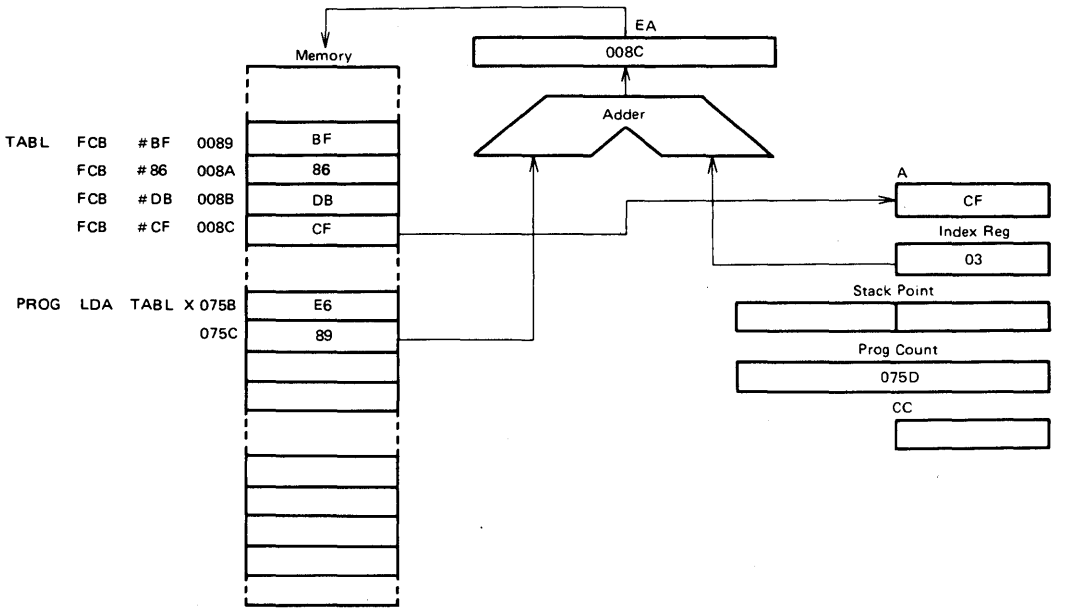


Figure 22 Indexed (8-Bit Offset) Addressing Example

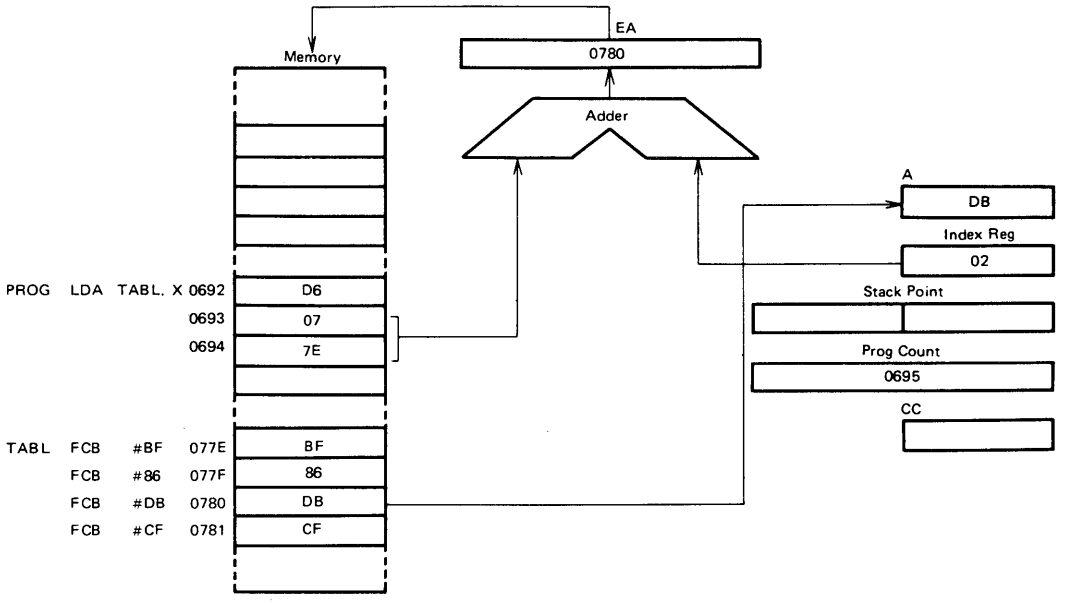


Figure 23 Indexed (16-Bit Offset) Addressing Example

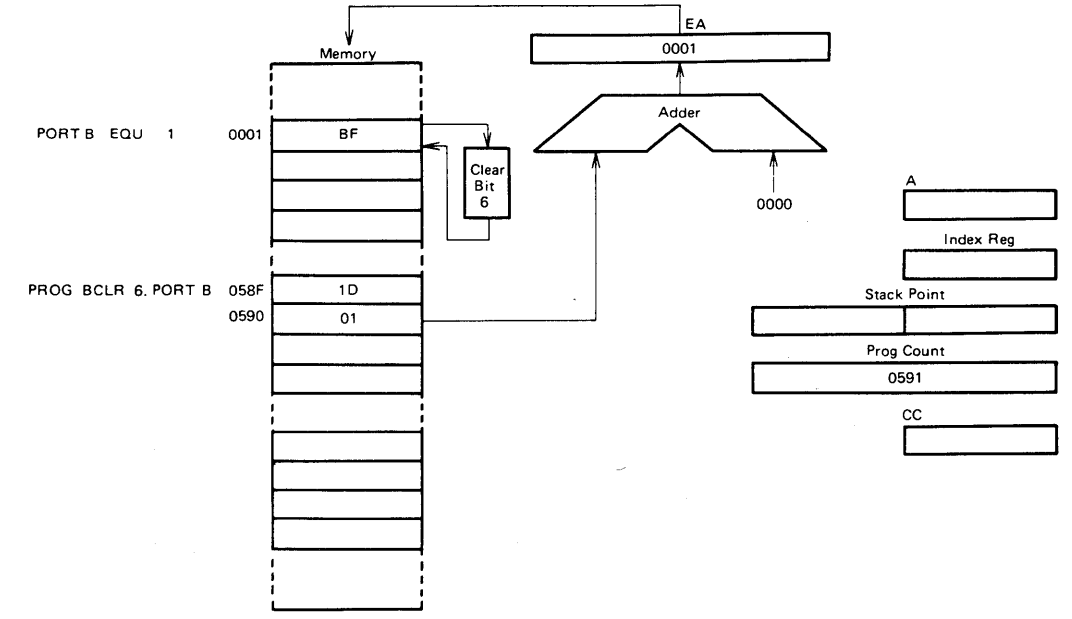


Figure 24 Bit Set/Clear Addressing Example

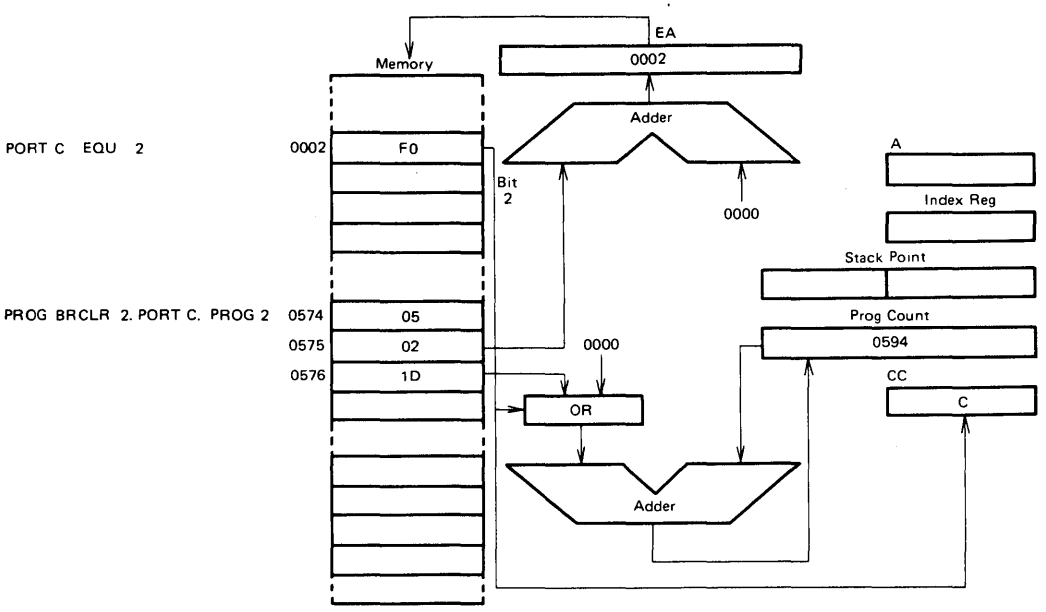


Figure 25 Bit Test and Branch Addressing Example

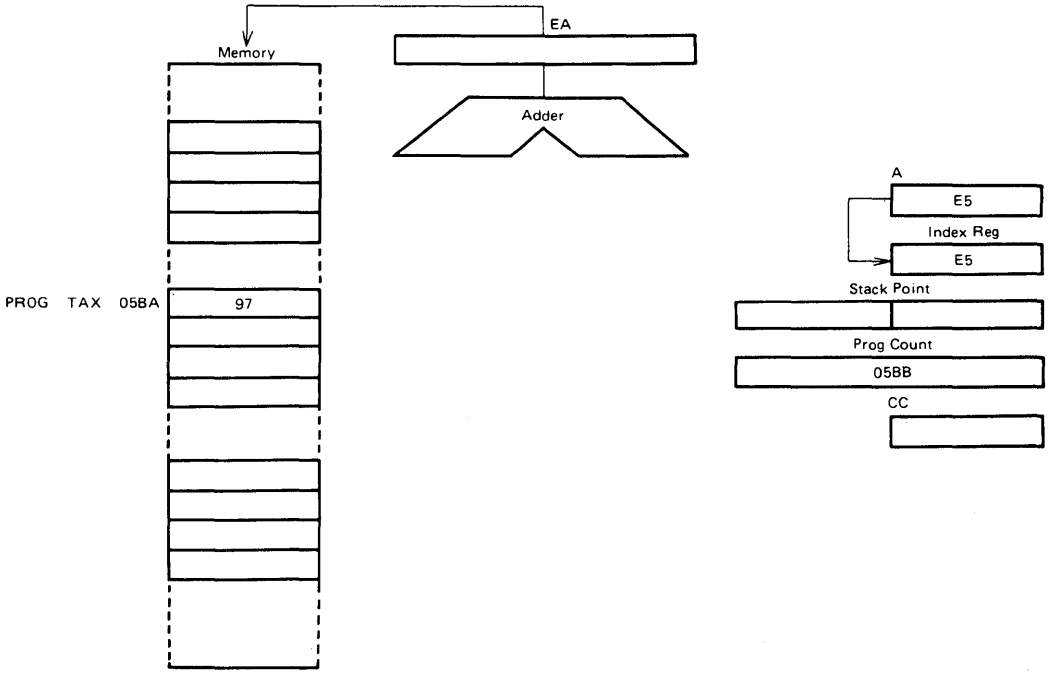


Figure 26 Implied Addressing Example

Table 2 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 3 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 4 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 5 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 6 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			^	•	^	^	^
ADD		x	x	x		x	x	x			^	•	^	^	^
AND		x	x	x		x	x	x			•	•	^	^	•
ASL	x		x			x	x				•	•	^	^	^
ASR	x		x			x	x				•	•	^	^	^
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	^	^	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	^
BRSET										x	•	•	•	•	^
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	^	^	^
COM	x		x			x	x				•	•	^	^	1
CPX		x	x	x		x	x	x			•	•	^	^	^
DEC	x		x			x	x				•	•	^	^	•
EOR		x	x	x		x	x	x			•	•	^	^	•
INC	x		x			x	x				•	•	^	^	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	^	^	•
LDX		x	x	x		x	x	x			•	•	^	^	•

(to be continued)

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	^	^	^
LSR	x		x			x	x				•	•	0	^	^
NEG	x		x			x	x				•	•	^	^	^
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	^	^	•
ROL	x		x			x	x				•	•	^	^	^
ROR	x		x			x	x				•	•	^	^	^
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	^	^	^
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	^	^	•
STX			x	x		x	x	x			•	•	^	^	•
SUB		x	x	x		x	x	x			•	•	^	^	^
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	^	^	•
TXA	x										•	•	•	•	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ^ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

Table 8 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(-1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and the number of cycles required (Bytes/Cycles).
Mnemonics followed by a "*" require a different number of cycles as follows:

RTI	9
RTS	6
SWI	11
BSR	8

3. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

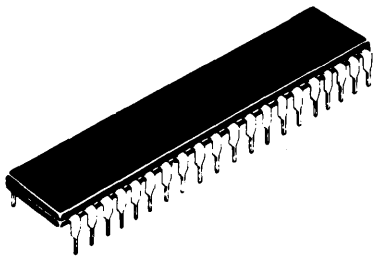
HD6805V1

MCU (Microcomputer Unit)

The HD6805V1 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, I/O and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD6800-based instruction set.

The following are some of the hardware and software highlights of the MCU.

HD6805V1P



(DP-40)

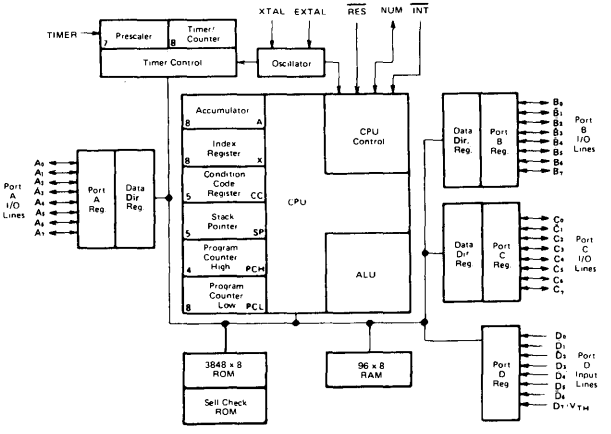
■ HARDWARE FEATURES

- 8-Bit Architecture
- 96 Bytes of RAM
- Memory Mapped I/O
- 3848 Bytes of User ROM
- Internal 8-Bit Timer with 7-Bit Prescaler
- Vectored Interrupts – External and Timer
- 24 I/O Ports + 8 Input Port
  (8 Lines LED Compatible; 7 Bits Comparator Inputs)
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- Easy for System Development and Debugging
- 5 Vdc Single Supply

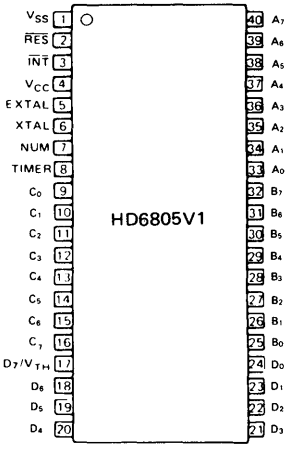
■ SOFTWARE FEATURES

- Similar to HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible Instruction Set with MC6805P2

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in} *	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +12.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT}$			3.0	–	V_{CC}	V
	All Other			2.0	–	V_{CC}	V
Input "High" Voltage (Timer)	Timer Mode			2.0	–	V_{CC}	V
	Self-Check Mode			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT}$			-0.3	–	0.8	V
	EXTAL(Crystal Mode)			-0.3	–	0.6	V
	All Other			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	700	mW
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT}$			-50	–	50	μA
	XTAL(Crystal Mode)			-1200	–	0	μA

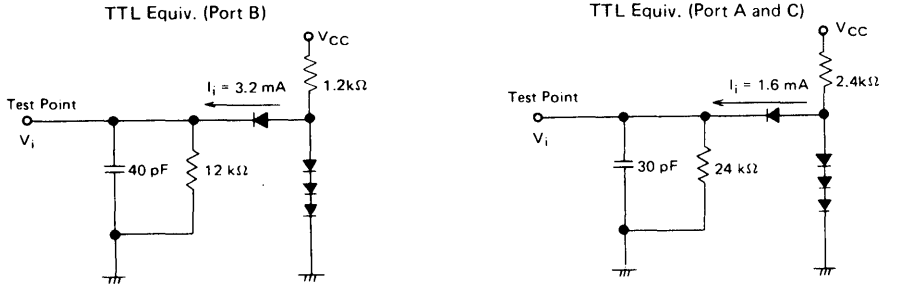
• AC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
Oscillation Frequency (External Resistor Mode)		f_{EXT}	R_{CP}=15.0kΩ±1%	–	3.4	–	MHz
$\overline{INT}$ Pulse Width		t_{IWL}		t_{cyc}+250	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		t_{cyc}+250	–	–	ns
TIMER Pulse Width		t_{TWL}		t_{cyc}+250	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20%, R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2 μF	100	–	–	ms
Input Capacitance	XTAL	C_{in}	V_{in}=0V	–	–	35	pF
	All Other			–	–	10	pF

● PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = −10 μA	3.5	–	–	V
			I_{OH} = −100 μA	2.4	–	–	V
	Port B		I_{OH} = −200 μA	2.4	–	–	V
			I_{OH} = −1 mA	1.5	–	–	V
	Port C		I_{OH} = −100 μA	2.4	–	–	V
Output "Low" Voltage	Port A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.4	V
	Port B		I_{OL} = 3.2 mA	–	–	0.4	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Port A, B, C, and D*	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	−500	–	–	μA
			V_{in} = 2V	−300	–	–	μA
	Port B, C, and D		V_{in} = 0.4V ~ V_{CC}	− 20	–	20	μA
Input "High" Voltage	Port D** (D_0 ~ D_6)	V_{IH}		–	V_{TH}+0.2	—	V
Input "Low" Voltage	Port D** (D_0 ~ D_6)	V_{IL}		–	V_{TH}-0.2	—	V
Threshold Voltage	Port D**(D_7)	V_{TH}		0	—	0.8 xV_{CC}	V

* Port D as digital input
** Port D as analog input



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074Ⓗ or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

● V_{CC} and V_{SS}

Power is supplied to the MCU using these two pins. V_{CC} is +5.25V ±0.5V. V_{SS} is the ground connection.

● $\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

● XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4 MHz maximum), a resistor or an external signal can be connected to these pins to provide a system clock with various stability/cost tradeoffs. Refer to INTERNAL OSCILLATOR OPTIONS for recommendations about these inputs.

● TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER for additional information about the timer circuitry.

● $\overline{RES}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to RESETS for additional information.

● NUM

This pin is not for user application and should be connected to V_{SS}.

- **Input/Output Lines ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$)**

These 24 lines are arranged into three 8-bit ports (A, B and C). All lines are programmable as either inputs or outputs under software control of the Data Direction Register (DDR). Refer to INPUT/OUTPUT for additional information.

- **Input Lines ($D_0 \sim D_7$)**

These are 8-bit input lines, which has two functions. Firstly, these are TTL compatible inputs, in location $003. The other function is 7 bits comparator in location $007. Refer to INPUT for more details.

■ MEMORY

The MCU memory is configured as shown in Figure 2. During the processing of an interrupt, the contents of the CPU registers are pushed onto the stack in the order shown in Figure 3. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

Caution: – Self Test ROM Address Area
Self test ROM locations can not be used for a user program. If the user's program is in this location, it will be removed when manufacturing mask for production.

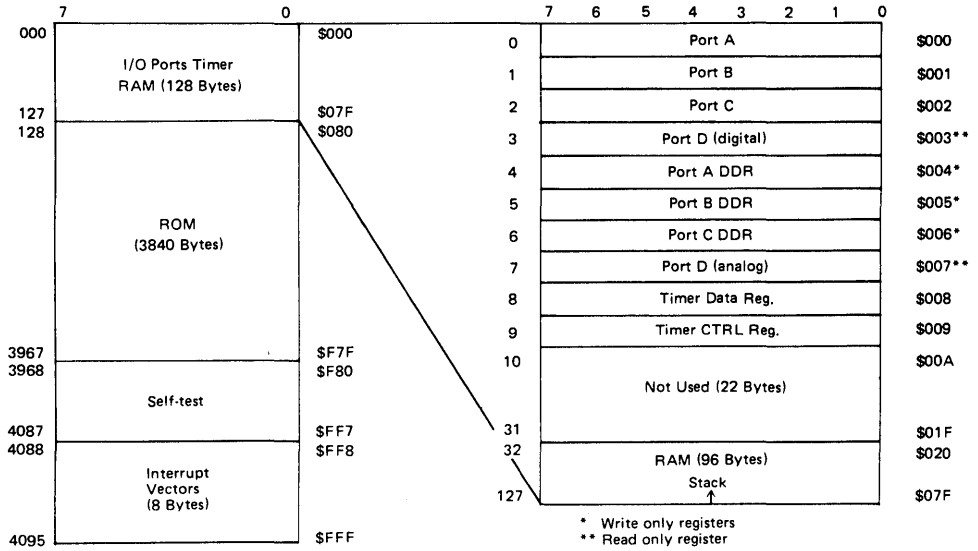


Figure 2 MCU Memory Configuration

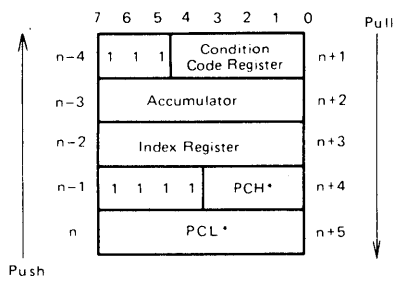


Figure 3 Interrupt Stacking Order

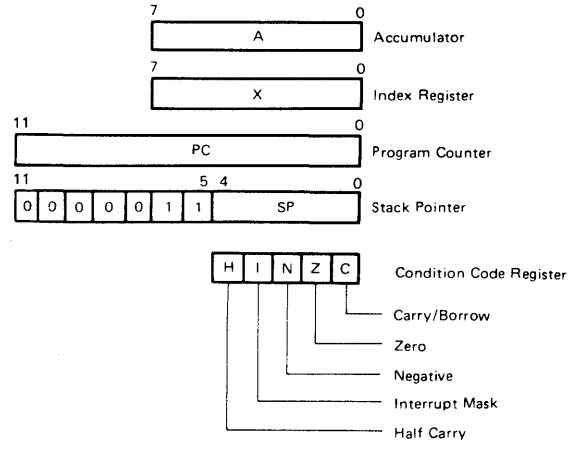


Figure 4 Programming Model

■ REGISTERS

The CPU has five registers available to the programmer. They are shown in Figure 4 and are explained in the following paragraphs.

- **Accumulator (A)**

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

- **Index Register (X)**

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

- **Program Counter (PC)**

The program counter is a 12-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The six most significant bits of the stack pointer are permanently set to 0000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 which allows the programmer to use up to 15 levels of subroutine calls.

- **Condition Code Register (CC)**

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

Used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in result equal to a logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

■ TIMER

The MCU timer circuitry is shown in Figure 5. The 8-bit counter, the Timer Data Register (TDR), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR) is set. The CPU responds to this interrupt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the Condition Code Register also prevents a timer interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When the internal ϕ_2 signal is selected as the input source, the node a is connected to b (see Fig. 5). In case of the external source, the node b connects with c. When the ϕ_2 signal is used as the source, the clock signal is input to the prescaler while the TIMER input is "High". The source of the clock input is one of the options that has to be specified before manufacture of the MCU. A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128

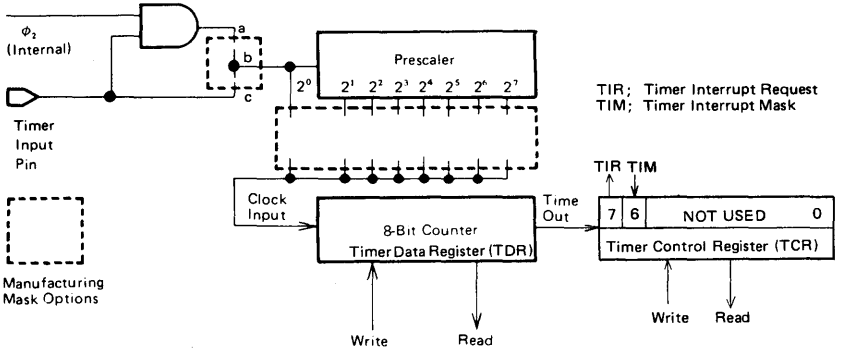


Figure 5 Timer Block Diagram

counts before decrementing the counter (TDR). The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter (TDR) can be read at any time by reading the TDR. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

The TDR is 8-bit Read/Write Register in location $008. At power-up or reset, the TDR and the prescaler are initialized with all logical ones.

The Timer Interrupt Request bit (bit 7 of the TCR) is set by hardware when timer count reaches zero, and is cleared by program or by hardware reset. The bit 6 of the TCR is writable by program. Both of those bits can be read by CPU.

(NOTE) If the MCU Timer is not used, the TIMER input pin must be grounded.

■ SELF CHECK

The self-check capability of the MCU provides an internal check to determine if the part is functional. Connect the MCU as shown in Figure 6 and monitor the output of port C bit 3 for an oscillation of approximately 3Hz. ROM, RAM, TIMER, Interrupts, I/O of Port A, B and C are checked by this capability.

■ RESETS

The MCU can be reset three ways; by initial power-up, by the external reset input ($\overline{RES}$) and by an optional internal low voltage inhibit circuit, see Figure 7. All the I/O port are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum of 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High".

This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 8, typically provides sufficient delay.

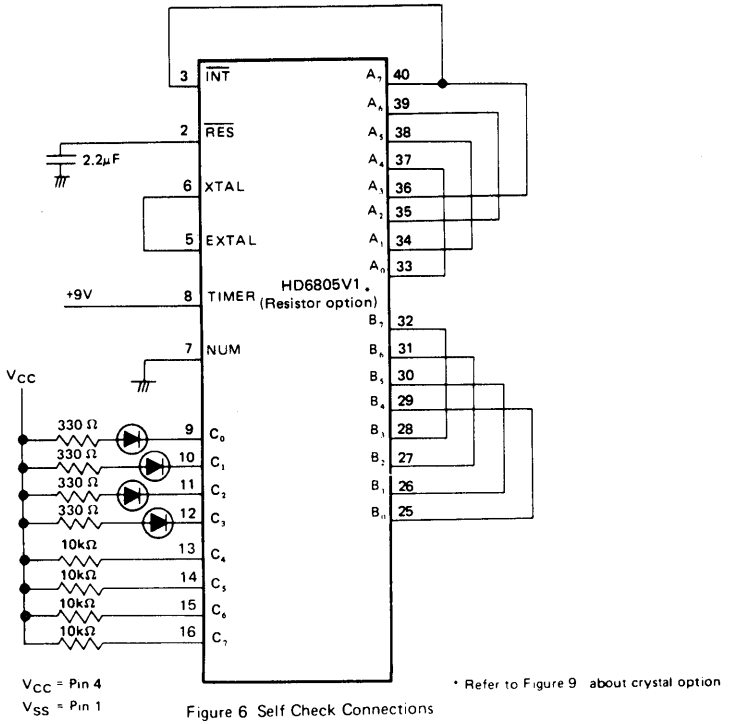


Figure 6 Self Check Connections

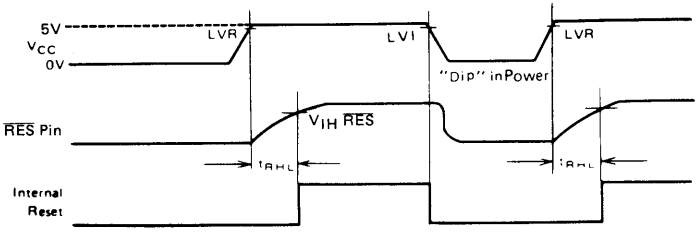


Figure 7 Power Up and $\overline{RES}$ Timing

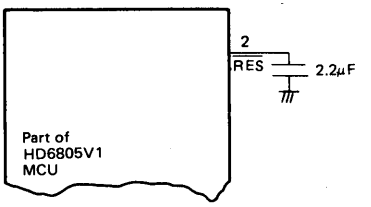


Figure 8 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR OPTIONS

The internal oscillator circuit is designed to require a minimum of external components. A crystal, a resistor, a jumper wire, or an external signal may be used to generate a system clock with various stability/cost tradeoff. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. The different connection methods are shown in Figure 9. Crystal specifications are given in Figure 10. A resistor selection graph is given in Figure 11.

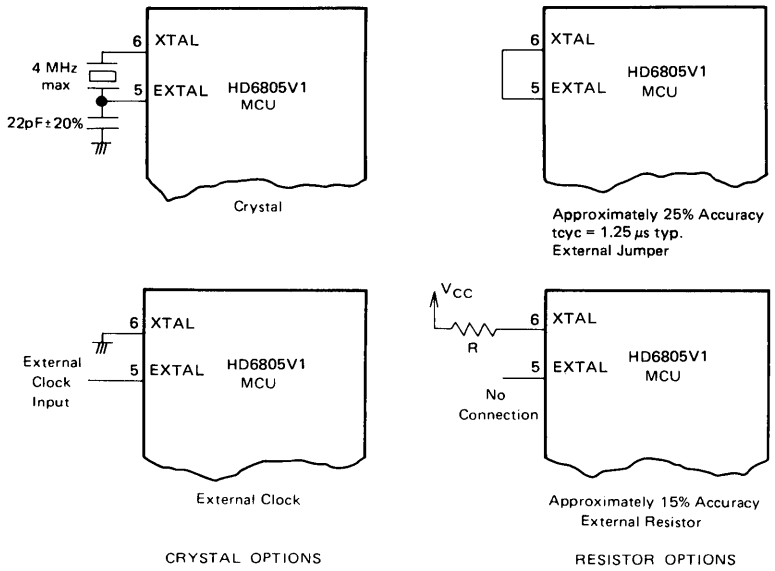
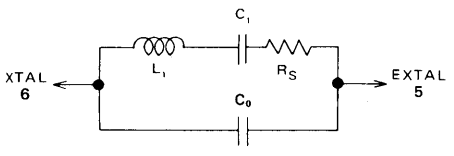


Figure 9 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz
R_S = 60 Ω max.

Figure 10 Crystal Parameters

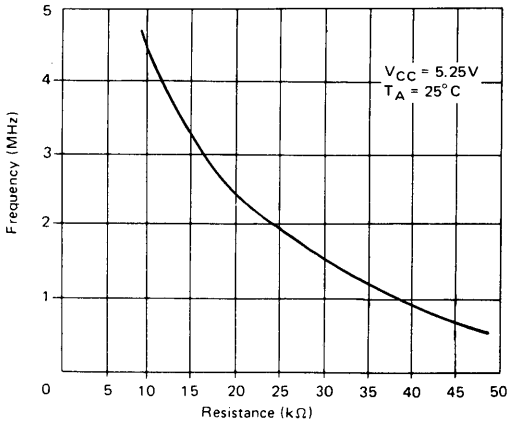


Figure 11 Typical Resistor Selection Graph

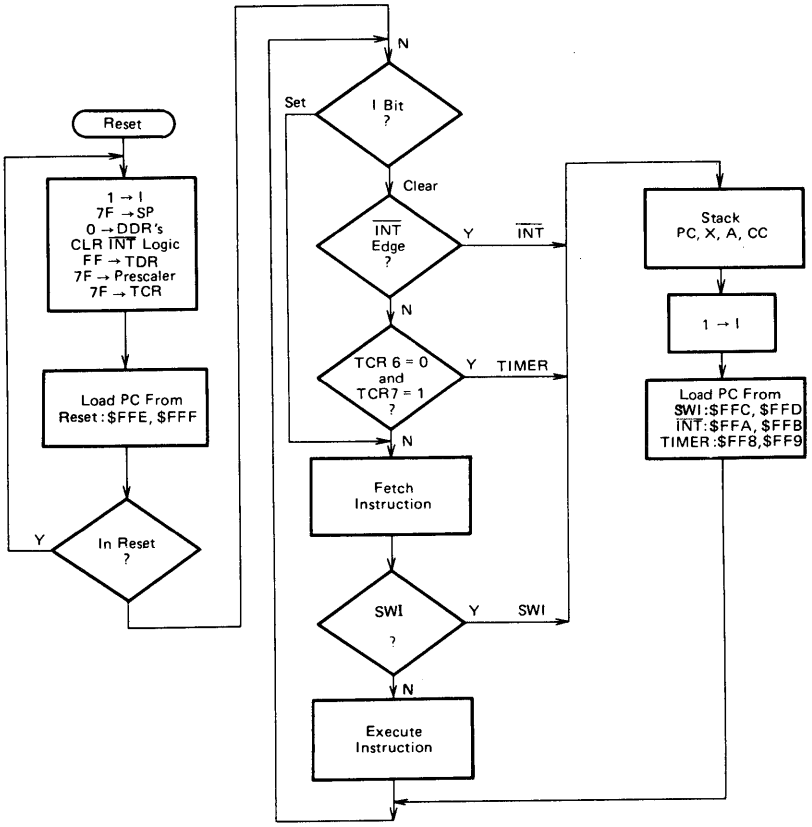
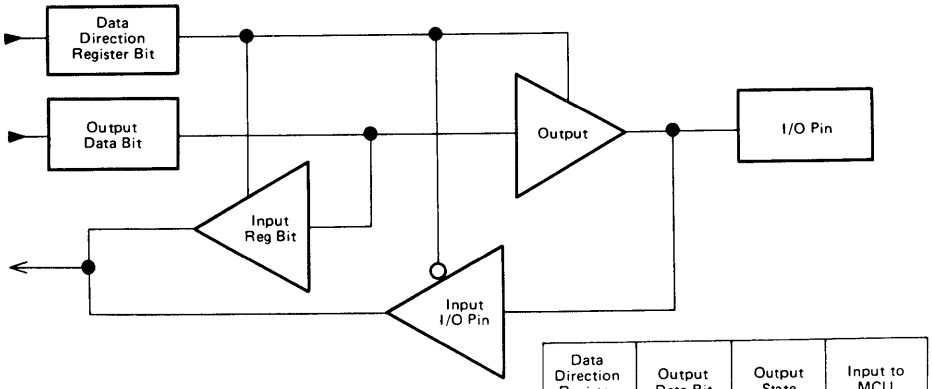


Figure 12 Interrupt Processing Flowchart



Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

Figure 13 Typical Port I/O Circuitry

■ INTERRUPTS

The CPU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU state is pushed onto the stack, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires 11 cycles. The Interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. Table 1 provides a listing of the interrupts, their priority, and the vector address that contain the starting address of the appropriate interrupt routine.

A flowchart of the interrupt processing sequence is given in Fig. 12.

Table 1 Interrupt Priorities

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$FFE and $FFF
SWI	2	$FFC and $FFD
$\overline{INT}$	3	$FFA and $FFB
TIMER	4	$FF8 and $FF9

■ INPUT/OUTPUT

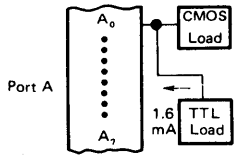
There are 24 input/output pins. All pins are programmable as either inputs or outputs under software control of the corresponding Data Direction Register (DDR). When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (see Fig. 13). When Port B is programmed for outputs it is capable of sinking 10mA on each pin (V_{OL} = 1V max). All input/output lines are TTL compatible as both inputs and outputs. Port A is CMOS compatible as outputs, and Port B and C lines are CMOS compatible as inputs. Figure 14 provides some examples of port connections.

■ INPUT

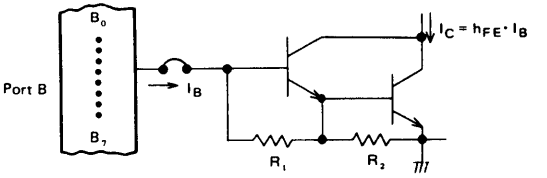
Port D can be used as either 8 TTL compatible inputs or 1 threshold input and 7 analog inputs pins. Fig. 15 (a) shows the construction of port D. The Port D register at location $003 stores TTL compatible inputs, and those in location $007 store the result of comparison D_0 to D_6 inputs with D_7 threshold input. Port D has not only the conventional function as inputs but also voltage-comparison function. Applying the latter, can easily check that 7 analog input electric potential max. exceeds the limit with the construction shown in Fig. 15 (b). Also, using one output pin of MCU, after external capacity is discharged at the preset state, charge the CR circuit of long enough time constant, apply the charging curve to the D_7 pin. The construction described above is shown in Fig. 15 (c). The compared result of D_0 to D_6 is regularly monitored, which gives the analog input electric potential applied to D_0 to D_6 pins from inverted time. This method enables 7 inputs to be converted from analog to digital. Furthermore, combination of two functions gives 3 level voltages from D_0 to D_6. Fig. 15 (d) provides the example when V_{TH} is set to 3.5V.

■ BIT MANIPULATION

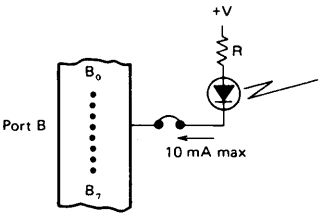
The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 16 illustrates the usefulness of the bit manipulation and test



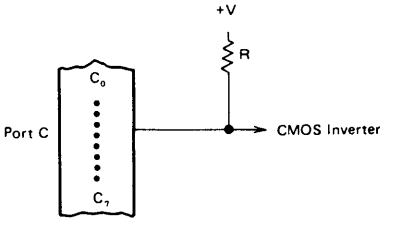
Port A Programmed as output(s), driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s), driving Darlington base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)

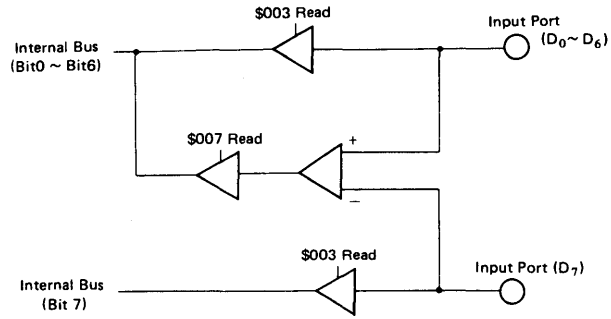


Port C Programmed as output(s), driving CMOS loads, using external pull-u resistors.
(d)

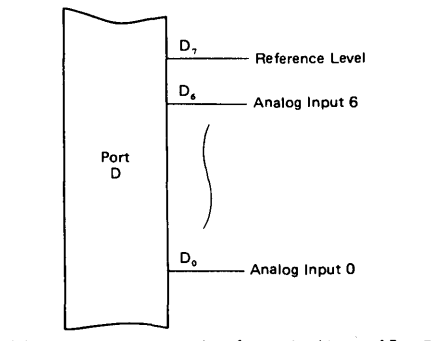
Figure 14 Typical Port Connections

instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which power the controlled hardware.

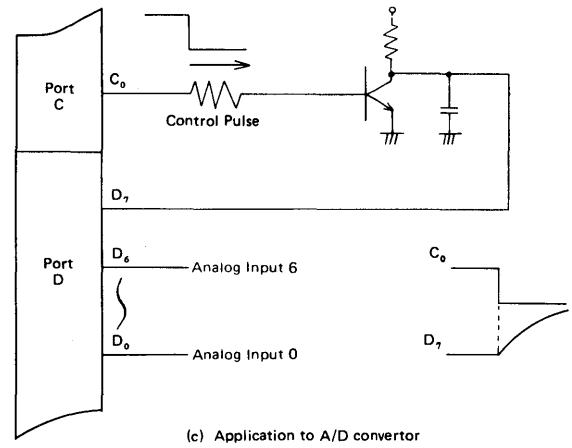
This program, which uses only seven ROM locations, provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer could also be incorporated to provide turn-on at some later time which would permit pulse-width modulation of the controlled power.



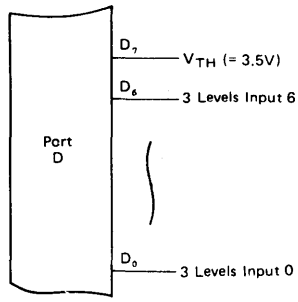
(a) The logic configuration of Port D



(b) Seven analog inputs and a reference level input of Port D



(c) Application to A/D convertor



(d) Application to 3 levels input

Input Voltage	($003)	($007)
0V ~ 0.8V	0	0
2.0V ~ 3.3V	1	0
3.7V ~ V_{CC}	1	1

Figure 15 Configuration and Application of Port D

```
SELF 1    BRCLR 0, PORT A, SELF 1
          BSET 1, PORT A
          BCLR 1, PORT A
```

Figure 16 Bit Manipulation Example

■ ADDRESSING MODES

The CPU has ten addressing modes available for use by the programmer. They are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 17. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 18. In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 19. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 20. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA=(PC)+2+Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken Rel=0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 21. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 22. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 23. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 24. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 25. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($00-$FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 26. The implied mode of addressing has no EA. All the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are one byte long.

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

- **Register/Memory Instructions**

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 2.

- **Read/Modity/Write Instructions**

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 3.

- **Branch Instructions**

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 4.

- **Bit Manipulation Instructions**

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 5.

- **Control Instructions**

The control instructions control the MCU operations during program execution. Refer to Table 6.

- **Alphabetical Listing**

The complete instruction set is given in alphabetical order in Table 7.

- **Opcode Map**

Table 8 is an opcode map for the instructions used on the MCU.

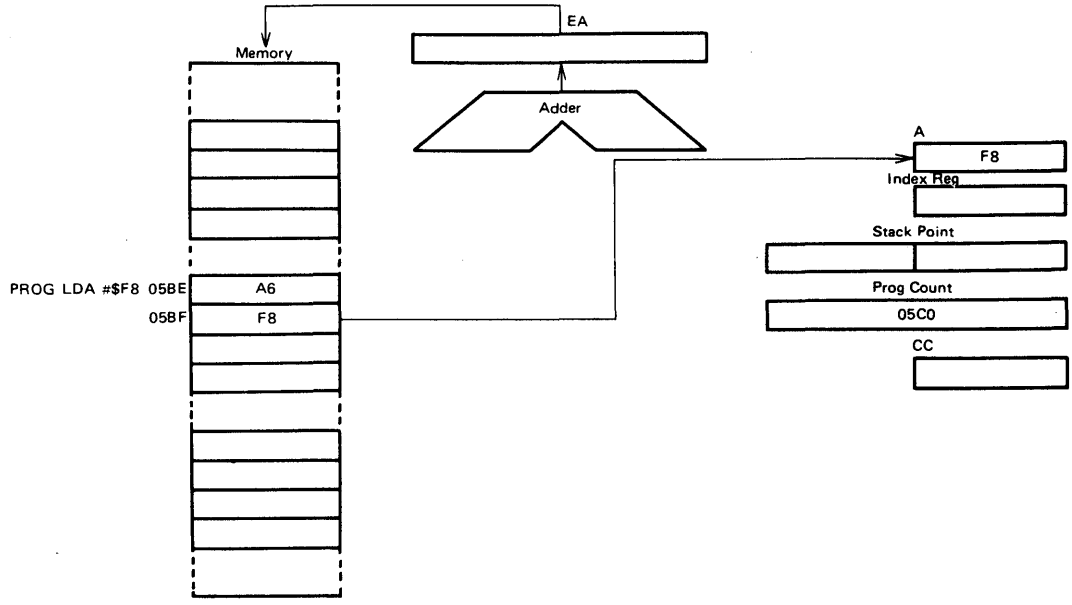


Figure 17 Immediate Addressing Example

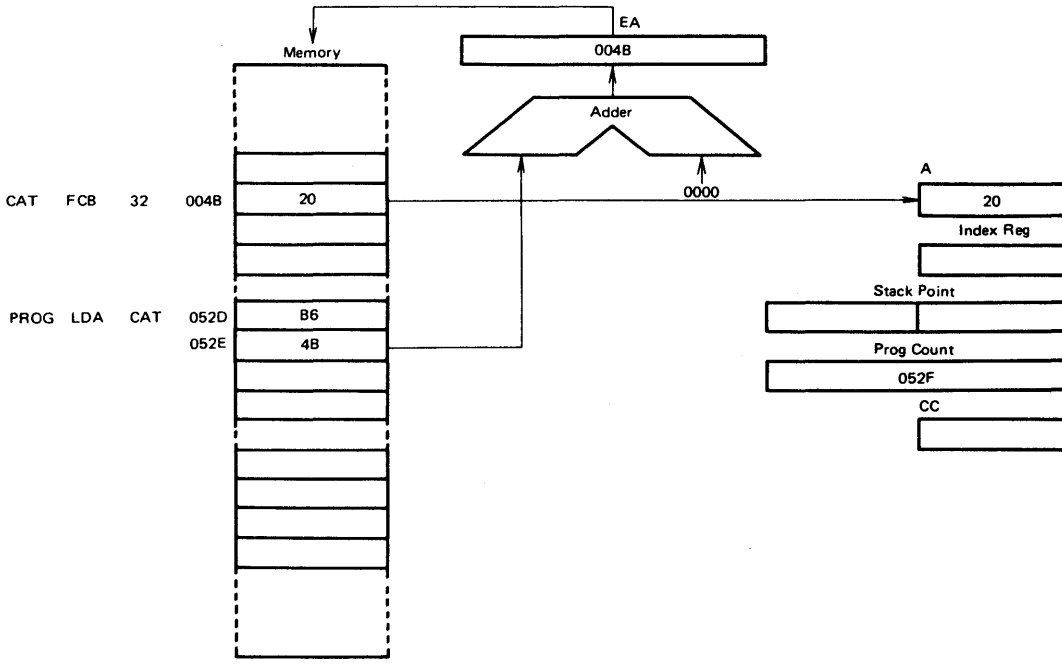


Figure 18 Direct Addressing Example

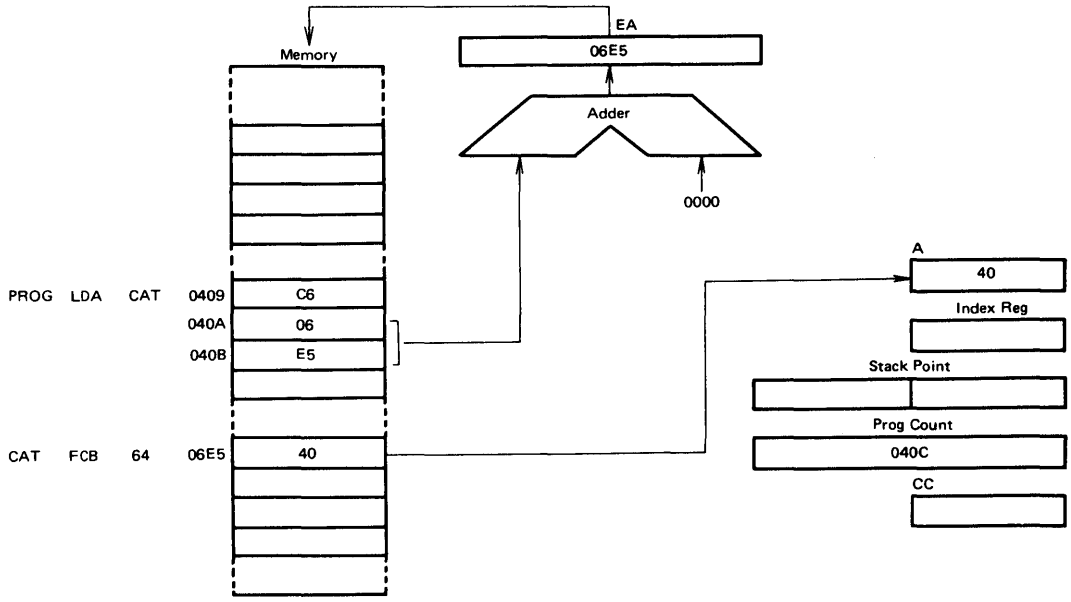


Figure 19 Extended Addressing Example

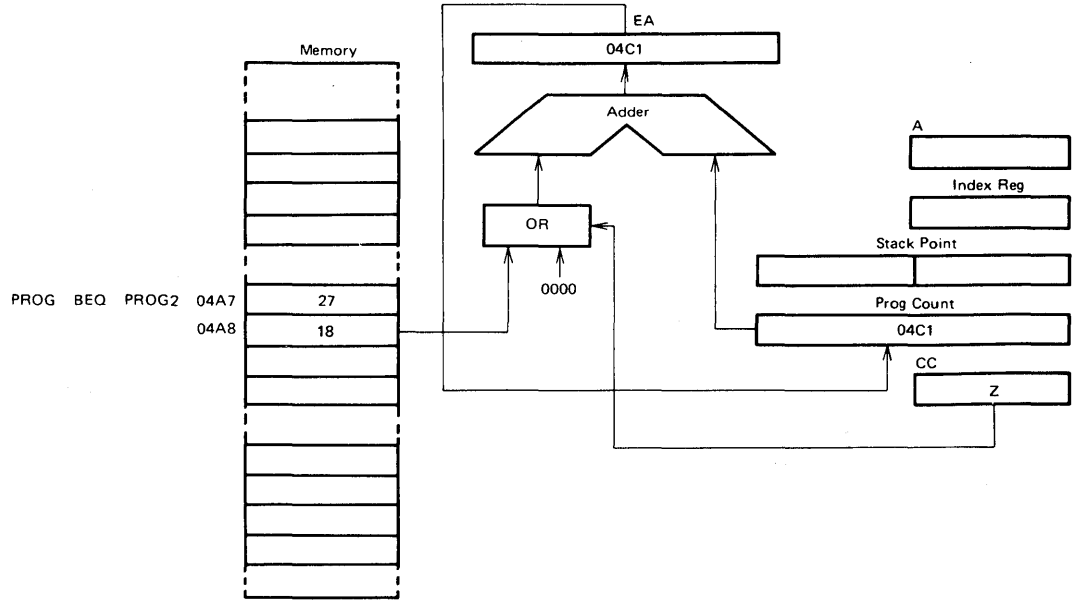


Figure 20 Relative Addressing Example

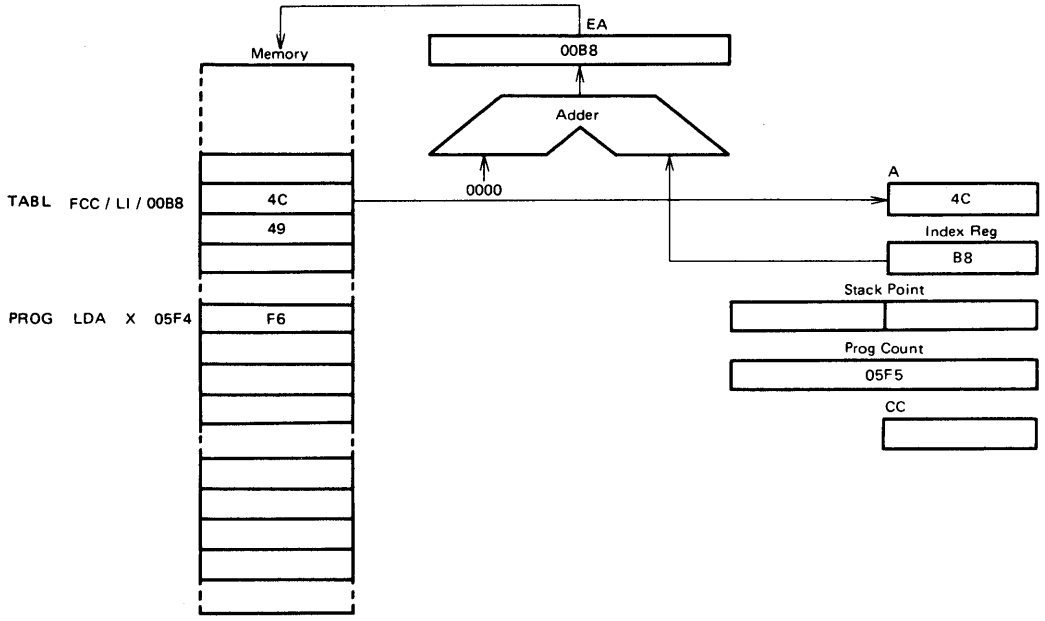


Figure 21 Indexed (No Offset) Addressing Example

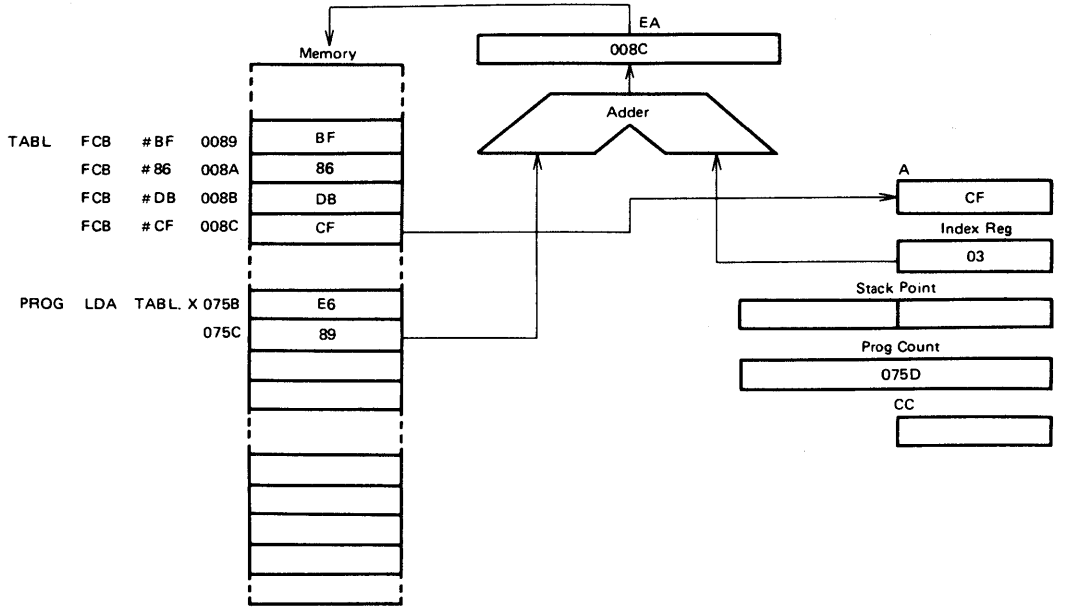


Figure 22 Indexed (8-Bit Offset) Addressing Example

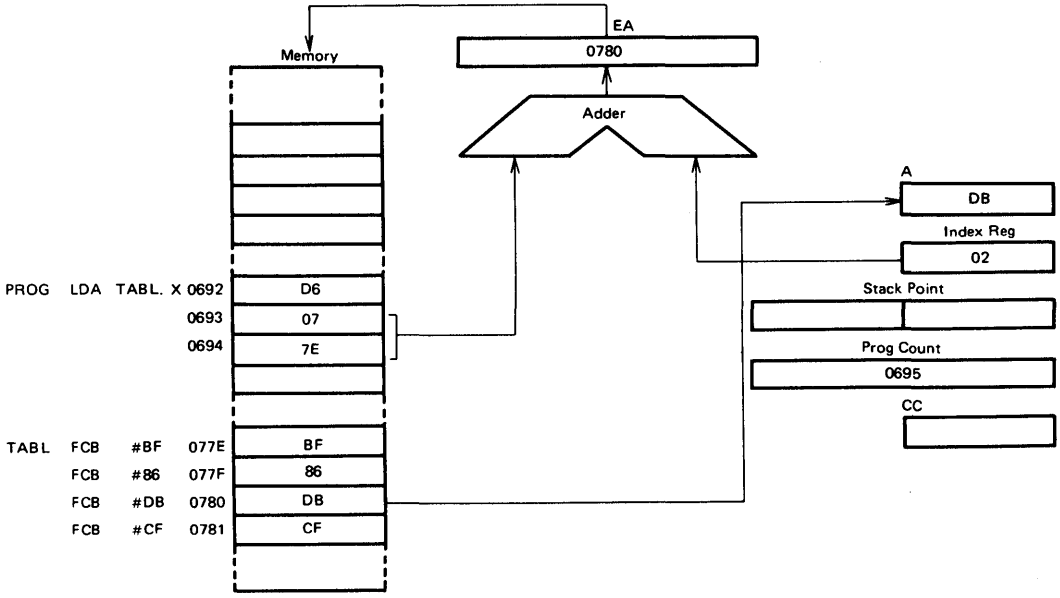


Figure 23 Indexed (16-Bit Offset) Addressing Example

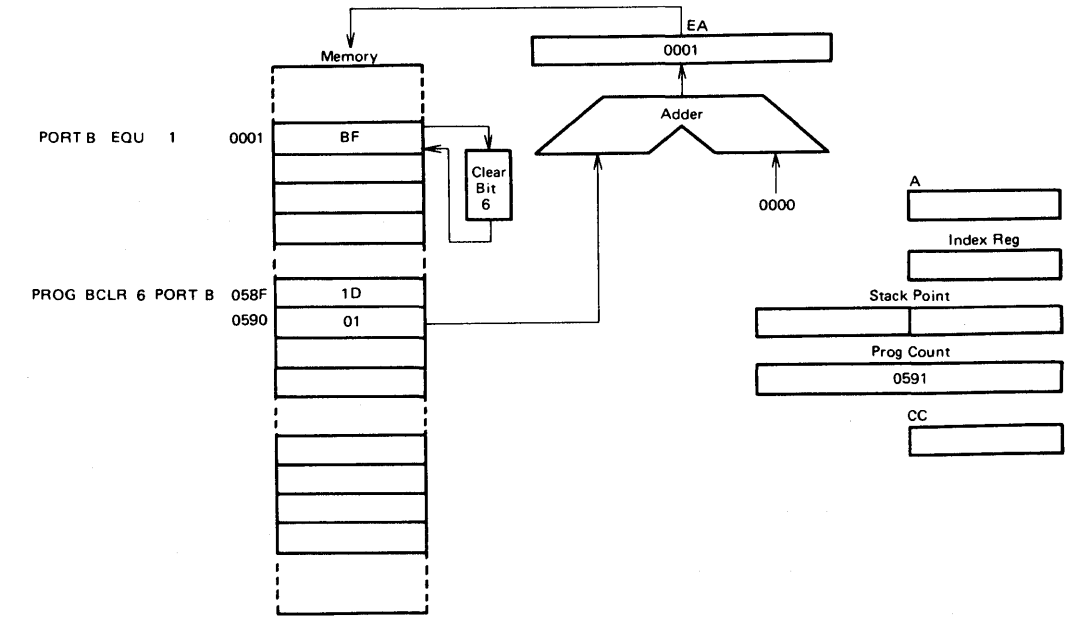


Figure 24 Bit Set/Clear Addressing Example

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

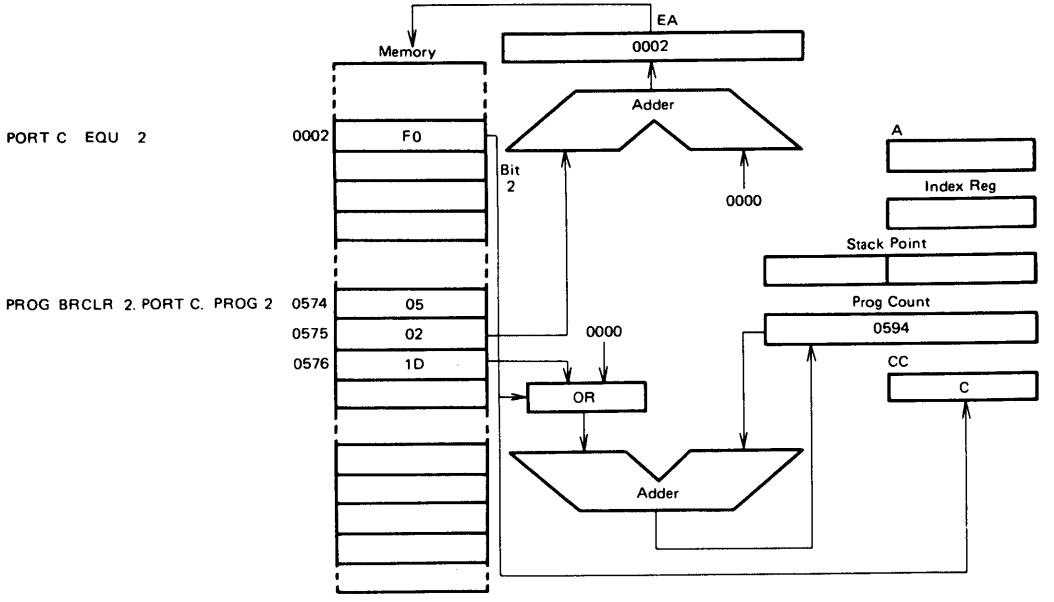


Figure 25 Bit Test and Branch Addressing Example

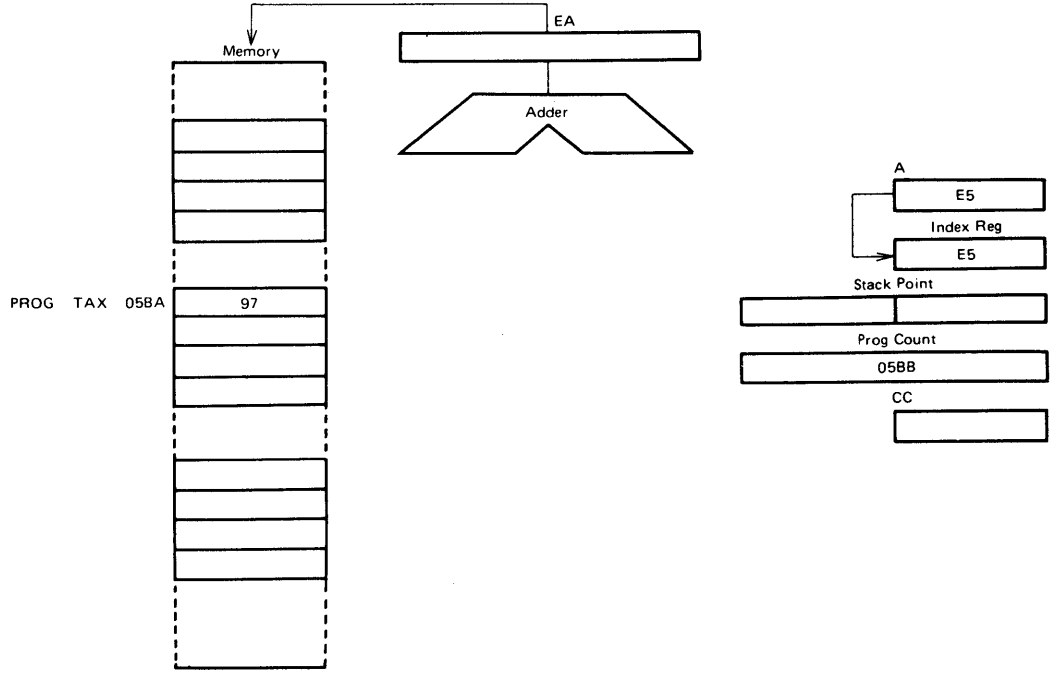


Figure 26 Implied Addressing Example

Table 2 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 3 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 4 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 5 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 6 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•

(to be continued)

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 7 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 8 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and the number of cycles required (Bytes/Cycles).
Mnemonics followed by a "*" require a different number of cycles as follows:

RTI	9
RTS	6
SWI	11
BSR	8

3. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

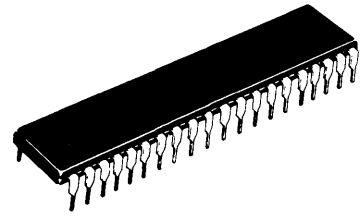
HD6805W1

MCU (Microcomputer Unit)

The HD6805W1 is an 8-bit microcomputer unit (MCU) which contains a CPU, on-chip clock, ROM, RAM, standby RAM, an A/D Converter, I/O and two timers. It is a member of the HD6805 family which is designed for user who needs an economical microcomputer with proven capabilities of the HD6800-based instruction set.

The following are some of the hardware and software highlights of the MCU.

HD6805W1P



(DP-40)

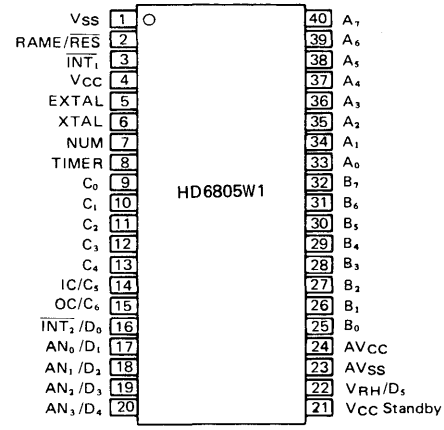
■ HARDWARE FEATURES

- 8-Bit Architecture
- 96 Bytes of RAM
  (8 bytes are standby RAM functions)
- Memory Mapped I/O
- 3848 Bytes of User ROM
- Internal 8-Bit Timer (Timer 1) with 7-Bit Prescaler
- Internal 8-Bit Programmable Timer (Timer 2)
- Interrupts – 2 External and 4 Timers
- 23 TTL/CMOS compatible I/O Lines; 8 Lines Directly Drive LEDs.
- On-Chip 8-Bit, 4-Channel A/D Converter
- On-Chip Clock Circuit
- Self-Check Mode
- Master Reset
- Low Voltage Inhibit
- Easy for System Development and Debugging
- 5 Vdc Single Supply

■ SOFTWARE FEATURES

- Similar to HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible with MC6805P2, HD6805S1 and HD6805V1

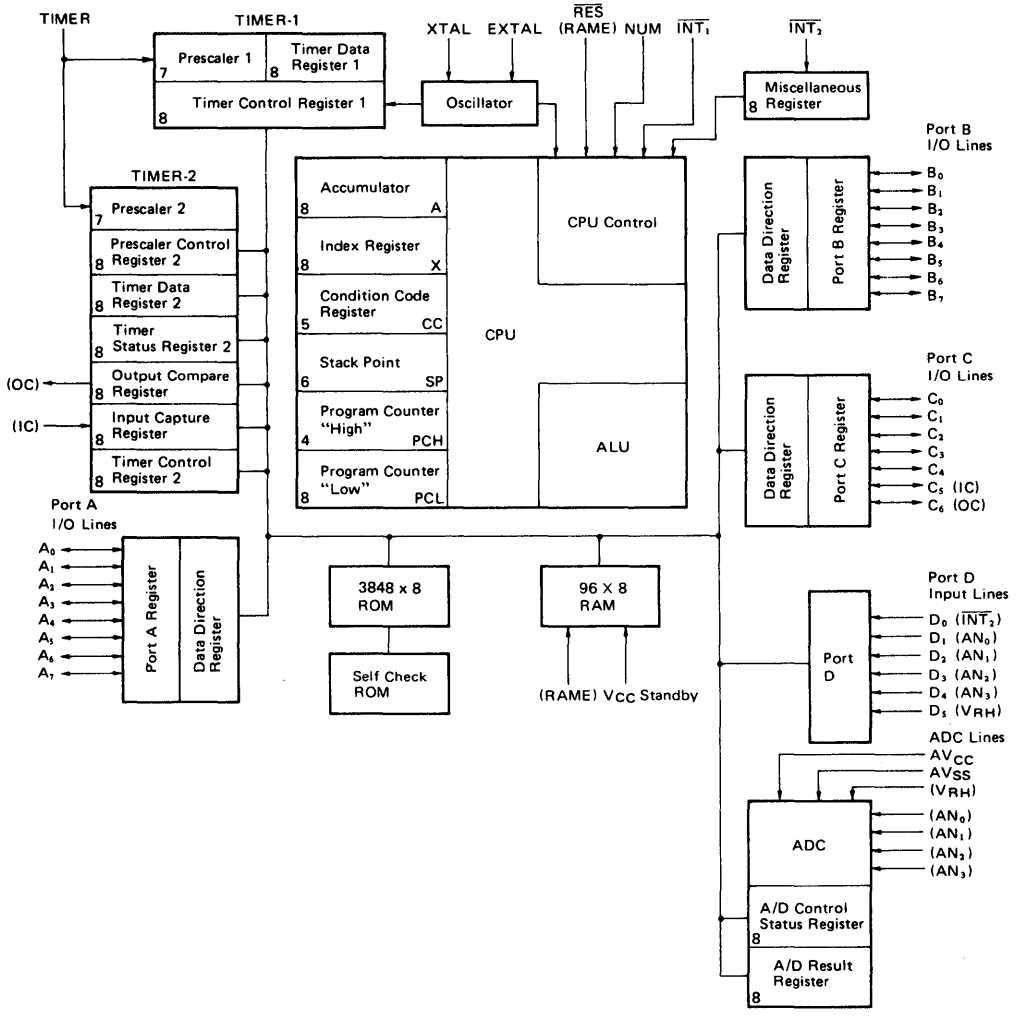
■ PIN ARRANGEMENT



(Top View)

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



(NOTE) The contents of ( ) items can be changed by software.

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in}	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +12.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This device has an input protection circuit for high quiescent voltage and field, however, be careful not to impress a high input voltage than the insulation maximum value to the high input impedance circuit. To insure normal operation, the following are recommended for V_{in} and V_{out}:
$V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT_1}$, $\overline{INT_2}$			3.0	–	V_{CC}	V
	All Others			2.0	–	V_{CC}	V
Input "High" Voltage (Timer)	Timer Mode			2.0	–	V_{CC}	V
	Self-Check Mode			9.0	–	11.0	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT_1}$, $\overline{INT_2}$			-0.3	–	0.8	V
	EXTAL(Crystal Mode)			-0.3	–	0.6	V
	All Others			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	750	mW
Low Voltage Recover		LVR		–	–	4.75	V
Low Voltage Inhibit		LVI		–	4.0	–	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT_1}$, $\overline{INT_2}$			-50	–	50	μA
	EXTAL(Crystal Mode)			-1200	–	0	μA
Standby Voltage	Nonoperation Mode	V_{SBB}		4.0	–	V_{CC}	V
	Operation Mode	V_{SB}		4.75	–	V_{CC}	
Standby Current	Nonoperation Mode	I_{SBB}	V_{SBB}=4.0V	–	–	3	mA

• AC CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

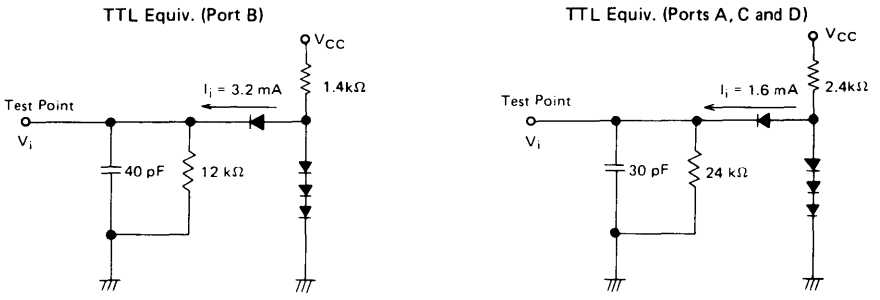
Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
Oscillation Frequency (External Resistor Mode)		f_{EXT}	R_{CP}=15.0kΩ±1%	–	3.4	–	MHz
$\overline{INTi}$ Pulve Width		t_{IWL}		t_{cyc}+250	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		t_{cyc}+250	–	–	ns
TIMER Pulse Width		t_{TWL}		t_{cyc}+250	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20% R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2μF	100	–	–	ms
Input Capacitance	XTAL, V_{RH}/Ds	C_{in}	V_{in}=0V	–	–	35	pF
	All Others			–	–	10	pF

• **PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = -10 μA	3.5	–	–	V
			I_{OH} = -100 μA	2.4	–	–	V
	Port B		I_{OH} = -200 μA	2.4	–	–	V
			I_{OH} = -1 mA	1.5	–	–	V
	Port C		I_{OH} = -100 μA	2.4	–	–	V
Output "Low" Voltage	Ports A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.5	V
	Port B		I_{OL} = 3.2 mA	–	–	0.5	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Ports A, B, C and D	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		-0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	-500	–	–	μA
			V_{in} = 2V	-300	–	–	μA
	Ports B, C and D		V_{in} = 0.4V ~ V_{CC}	-20	–	20	μA

• **A/D CONVERTER ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V±0.5V, V_{SS} = AV_{SS} = GMD, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	min	typ	max	Unit
Analog Power Supply Voltage	AV_{CC}		4.75	5.25	5.75	V
Analog Input Voltage	AV_{in}		0	–	V_{RH}	V
Reference Voltage	V_{RH}	4.75V ≦ V_{CC} ≦ 5.25V	4.0	–	V_{CC}	V
		5.25V < V_{CC} ≦ 5.75V	4.0	–	5.25	V
Analog Multiplexer Input Capacitance			–	–	7.5	pF
Resolution Power			–	8	–	Bit
Conversion Time		at 4MHz	76	76	76	t_{cyc}
Input Channels			4	4	4	Channel
Absolute Accuracy		Ta = 25°C	–	–	±1.5	LSB



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074(H) or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

• V_{CC} and V_{SS}

Voltage is supplied to the MCU using these two pins. V_{CC} is 5.25V ±0.5V. V_{SS} is the ground connection.

• $\overline{INT_1}$/$\overline{INT_2}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

• XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4 MHz maximum), a resistor or an external signal can be connected to these pins to provide a system clock with various stability/cost tradeoffs. Refer to INTERNAL OSCILLATOR OPTIONS for recommendations about these inputs.

• TIMER

This pin allows an external input to be used to count for the internal timer circuitry. Refer to TIMER 1 and TIMER 2 for additional information about the timer circuitry. When this pin isn't used, it must be grounded.

• $\overline{RES}$

This pin allows resetting of the MCU at times other than the automatic resetting capability already in the MCU. Refer to RESETS for additional information.

• NUM

This pin is not for user application and should be connected to V_{SS}.

• I/O Lines ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_6$)

There 23 lines are arranged into three ports (A, B and C). All lines are programmable as either inputs or outputs under software control of the Data Direction Register (DDR). Refer to INPUT/OUTPUT for additional information.

• Input Lines ($D_0 \sim D_5$)

These are TTL compatible input lines, in location $003. These also allow analog inputs to be used for an A/D converter and interrupt. Refer to INPUT and A/D converter for additional information.

• V_{CC} Standby

V_{CC} Standby provides power to the standby portion of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the MCU is in a powerup or powerdown state. In the powerup state, the power supply should provide V_{CC} and must reach V_{SB} before $\overline{RES}$ reaches 4.0V. During powerdown, V_{CC} standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} Standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation shown Figure 2.

To sustain the standby RAM during powerdown, the following software or hardware are needed.

(1) Software

When clearing the RAM Enable bit (RAME) which is bit 6 of the RAM Control Register at location $001F, the RAM is disabled.

V_{CC} Standby must remain above V_{SBB} (min).

(2) Hardware

When RAME pin is "Low" before powerdown, the RAM is disabled. V_{CC} Standby must remain above V_{SBB} (min).

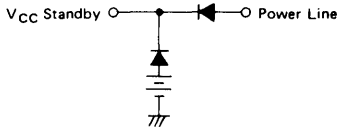


Figure 2 Battery Backup for V_{CC} Standby

• RAME

This pin is used for the external control of the RAM. When it is "Low" before powerdown, the RAM is disabled. If V_{CC} Standby remains above V_{SBB} (min), the standby RAM is sustained.

- **AV_{CC}**

This pin is used for the power supply of the A/D converter. When high accuracy is required, a different power source from V_{CC} should be impressed.

Connect to V_{CC} for all other cases. AV_{SS} corresponds to AV_{CC} as a GND terminal.

- **$AN_0 \sim AN_3$**

These pins allow analog inputs to be used for an A/D converter. These inputs are switched by the internal multiplexer and selected by bit 0 and 1 of the A/D Control Status Register (ADCSR: $00E).

- **V_{RH} and AV_{SS}**

The input terminal reference voltage for the A/D converter is "High" (V_{RH}) or "Low" (AV_{SS}). AV_{SS} is fixed at 0V.

- **Input Capture (IC)**

This pin is used for input of Timer 2 control, in this case, Port C_5 should be configured as input. Refer to TIMER 2 for more details.

- **Output Compare (OC)**

This pin is used for output of Timer 2 when the Output Compare Register is matched with the Timer Data Register 2. In this case, Port C_6 should be configured as an output. Refer to TIMER 2 for more details.

■ MEMORY

The MCU memory is configured as shown in Figure 3. During the interrupt processing, the contents of the CPU registers are pushed onto the stack in the order shown in Figure 4. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order three bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

Caution: – Self Test ROM Address Area

Self test ROM locations can not be used for a user program. If the user's program is in this location, it will be removed when manufacturing mask for production.

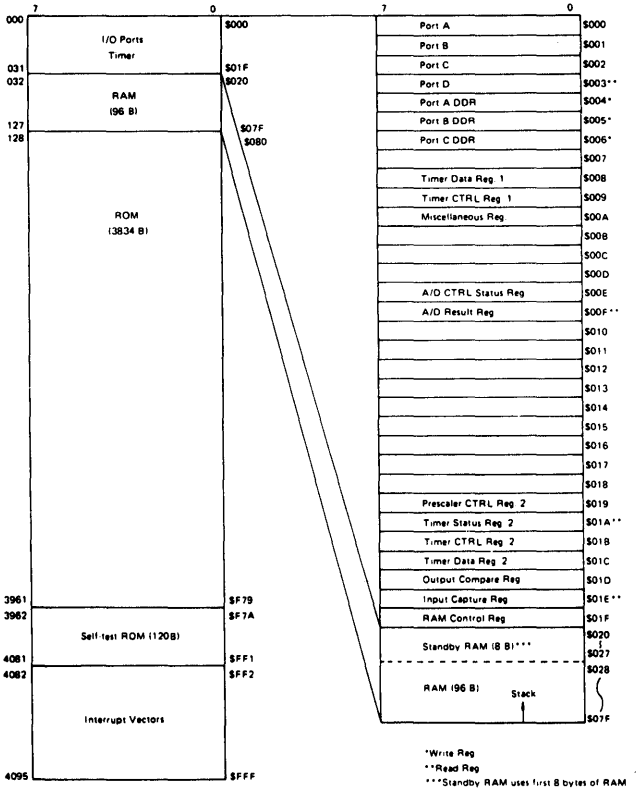


Figure 3 MCU Memory Structure

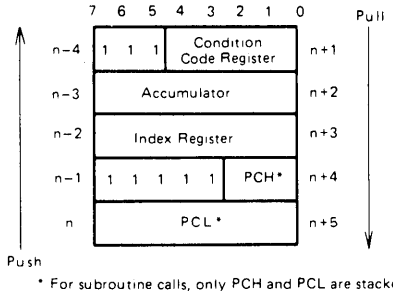


Figure 4 Interrupt Stacking Order

■ REGISTERS

The CPU has five registers available to the programmer, as shown in Figure 5 and explained below.

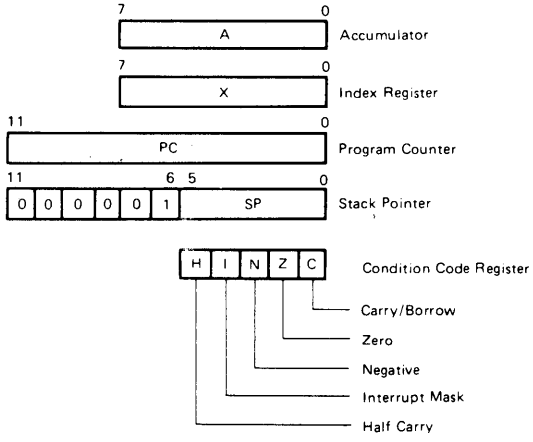


Figure 5 Programming Model

• Accumulator (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

• Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode and contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations or data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

• Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented while data is being pulled from the stack. The six most significant bits of the stack pointer are permanently set to 000001. During an MCU reset or reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $041 which allows the programmer to use up to 31 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained below.

Half Carry (H)

The half carry bit is used during arithmetic operations (ADD or ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask everything. If an interrupt occurs while this bit is set, it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

The negative bit is used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in a result equal to a logical one).

Zero (Z)

Zero is used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Carry/borrow is used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts and rotates.

■ TIMER 1

The MCU timer circuitry is shown in Figure 6. The 8-bit counter, Timer Data Register 1 (TDR1), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the TDR1 reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register 1 (TCR1) is set. The CPU responds to this interrupt by saving the present CPU state in the stack, fetching the timer 1 interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine. The timer 1 interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR1. The interrupt bit (I bit) in the Condition Code Register also prevents a timer 1 interrupt from being processed.

The clock input to the timer 1 can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When ϕ_2 is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. When the TIMER input pin doesn't control the ϕ_2 signal, connect the TIMER input pin to V_{CC}. The timer 1 continues to count past zero, falling through to $FF from zero and then continuing the

count. Thus, the counter (TDR1) can be read at any time by reading the TDR1. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and the counter (TDR1) are initialized with all logical ones; the timer 1 interrupt request bit (bit 7) is cleared and the timer 1 interrupt mask bit (bit 6) is set. In order to release the timer 1 interrupt, bit 7 of the TCR1 must be cleared by software.

(NOTE) If the MCU Timer 1 and Timer 2 are not used, the TIMER input pin must be grounded.

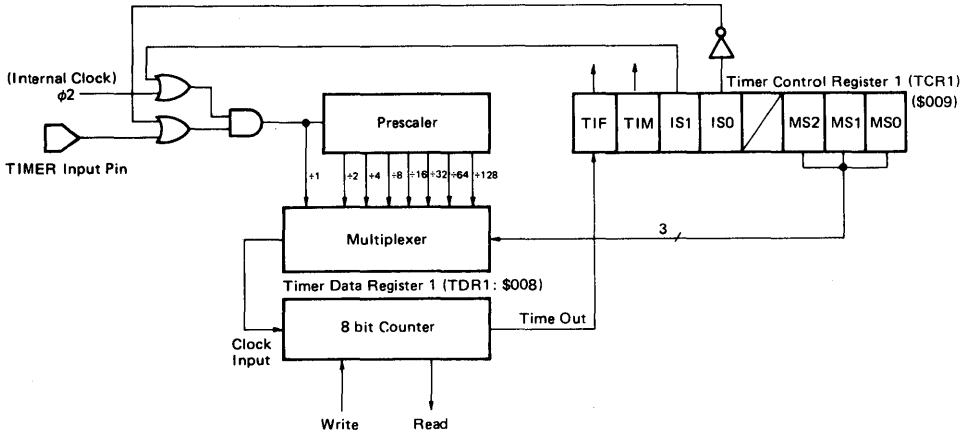
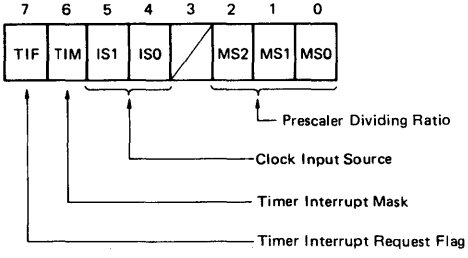


Figure 6 Timer Clock

- **Timer Control Register 1 (TCR1: $009)**

The Timer Control Register 1 (TCR1) can control selection of clock input source and prescaler dividing ratio and timer interrupt.

Timer Control Register 1 (TCR1: $009)



As shown in Table 1, the selection of the clock input source is ISO and IS1 in the TCR1 (bit 4 and bit 5) and 3 kinds of input are selectable. At reset, internal clock ϕ_2 controlled by the TIMER input (bit 4=1, bit5=0) is selected.

The prescaler dividing ratio is selected by MS0, MS1, and MS2 in the TCR1 (bit 0, bit 1, bit 2) as shown in Table 2. The dividing ratio is selectable from eight ways (÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128). At reset, ÷1 mode is selected. The prescaler is initialized by writing in the TDR1.

Timer 1 interrupt mask bit (TIM) allows the Timer 1 into interrupt at "0" and masks at "1". Timer 1 interrupt causes Timer 1 interrupt request bit (TIF) to be set. TIF must be cleared by software.

Table 1 Selection of Clock Input Source

TCR1		Clock Input Source
Bit 5	Bit 4	
0	0	Internal Clock ϕ_2 *
0	1	ϕ_2 Controlled by TIMER Input
1	0	
1	1	Event Input From TIMER

* The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock input.

Table 2 Selection of Prescaler Dividing Ratio

TCR1			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

■ TIMER 2

The HD6805W1 includes an 8-bit programmable timer (Timer 2) which can not only measure the input waveform but also generate the output waveform. The pulse width for both input and output waveform can be varied from several microseconds to several seconds.

(NOTE) If the MCU Timer 1 and Timer 2 are not used, the TIMER input pin must be grounded.

Timer 2 hardware consists of the followings.

- an 8-bit control register 2
- an 8-bit status register 2
- an 8-bit timer data register 2
- an 8-bit output compare register
- an 8-bit input capture register
- a 5-bit prescaler control register 2
- a 7-bit prescaler 2

A block diagram of the timer 2 is shown in Figure 7.

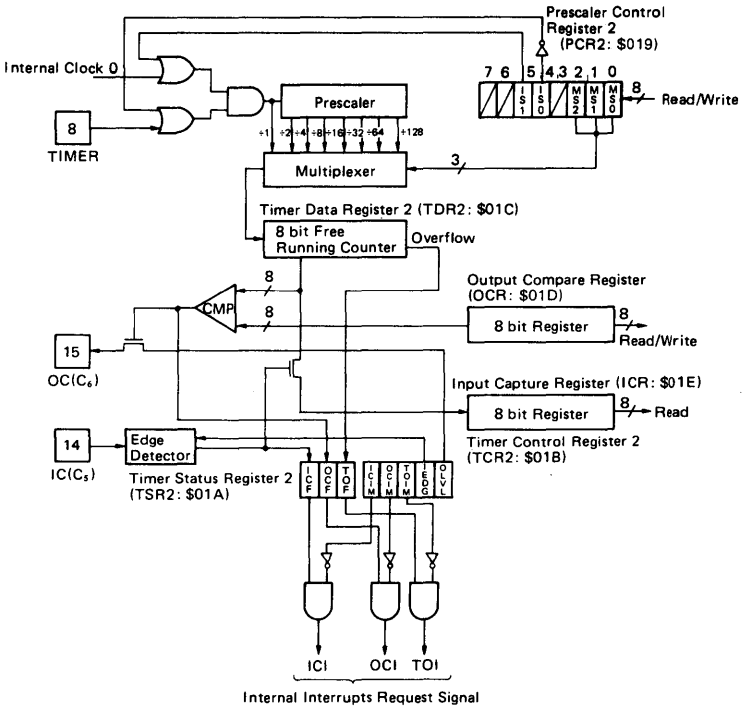


Figure 7 Block Diagram of Timer 2

● Timer Data Register 2 (TDR2: $01C)

The main part of the Timer 2 is the 8-bit Timer Data Register 2 (TDR2) as free-running counter, which is driven by internal clock ϕ_2 or the TIMER input and increments the value. The values in the counter is always readable by software.

The Timer Data Register 2 is Read/Write register and is cleared at reset.

● Output Compare Register (OCR: $01D)

The Output Compare Register (OCR) is an 8-bit Read/Write register used to control an output waveform. The contents of this register are always compared with those of the TDR2. When these two contents conform to each other, the flag (OCF) in the Timer Status Register 2 (TCR 2) is set and the value of the output level bit (OLVL) in the TCR2 is transferred to Port C_6 (OC).

If Port C_6's Data Direction Register (DDR) is "1" (output), this value will appear at Port C_6 (OC). Then the values of OCF and OLVL can be changed for the next compare. The OCR is set to $FF at reset.

● Input Capture Register (ICR: $01E)

The Input Capture Register (ICR) is an 8-bit Read-only register used to store the value of the TDR2 when Port C_5 (IC) input transition occurs as defined by the input edge bit (IEDG) of the TCR2.

In order to apply Port C_5 (IC) input to the edge detect circuit, the DDR of Port C_5 should be cleared ("0").*

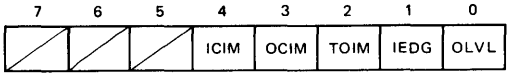
To ensure an input capture under all condition, Port C_5 (IC) input pulse width should be 2 Enable-cycles at least.

*The edge detect circuit always senses Port C_5 (IC) even if the DDR is set with Port C_5 output.

- **Timer Control Register 2 (TCR2: $01B)**

The Timer Control Register 2 (TCR2) consists of an 5-bit register of which all bits can be read and written.

Timer Control Register 2 (TCR2: $01B)



Bit 0 OLVL Output Level

This bit will appear at Port C_6 when the value in the TDR2 equals the value in the OCR, if the DDR of Port C_6 is set. It is cleared by reset.

Bit 1 IEDG Input Edge

This bit determines which level transition of Port Cs (IC) input will trigger a data store to ICR from the TDR2. When this function is used, it is necessary to clear DDR of Port Cs. When IEDG = 0, the negative edge triggers ("High" to "Low" transition). When IEDG = 1, the positive edge triggers ("Low" to "High" transition). It is cleared by reset.

Bit 2 TOIM Timer Overflow Interrupt Mask

When this bit is cleared, internal interrupt (TOI) is enabled by TOF interrupt but when set, interrupt is inhibited.

Bit 3 OCIM Output Compare Interrupt Mask

When this bit is cleared, internal interrupt (OCI) by OCF interrupt occurs. When set, interrupt is inhibited.

Bit 4 ICIM Input Capture Interrupt Mask

When this bit is cleared, internal interrupt (ICI) by ICF interrupt occurs. When set, interrupt is inhibited.

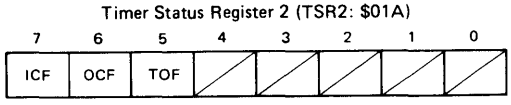
- **Timer Status Register 2 (TSR2: $01A)**

The Timer Status Register 2 (TSR2) is an 8-bit read-only register which indicates that;

(1) A proper level transition has been detected on the input pin with a subsequent transfer of the TDR2 value to the ICR (ICF).
(2) A match has been found between the TDR2 and the OCR (OCF).
(3) The TDR2 is zero (TOF).

Each of the event can generate 3 kinds of internal interrupt request and is controlled by an individual inhibit bits in the TCR2. If the I bit in the Condition Code Register is cleared, priority vectors are generated in response to clearing each interrupt mask bit. Each bit is described below.

Timer Status Register 2 (TSR2: $01A)



Bit 5 TOF Timer Overflow Flag

This read-only bit is set when the TDR2 contains $00. It is cleared by reading the TSR2 followed by reading of the TDR2.

Bit 6 OCF Output Compare Flag

This read-only bit is set when a match is found between the OCR and the TDR2. It is cleared by reading the TSR2 and then writing to the OCR.

Bit 7 ICF Input Capture Flag

This read-only bit is set to indicate a proper level transition and cleared by reading the TSR2 and then reading the TCR2.

CAUTION

The flag of the TSR2 will be occasionally cleared when manipulating or testing the TSR2 by Read/Modify/Write instruction shown in Table 3. Don't use these instructions for read/write/test operation of the TSR2 flags.

Table 3

		Bit Manipulation		Read/Modify/Write
		Test & Branch	Set/Clear	DIR
High →		0	1	3
	0	BRSET 0	BSET 0	NEG
	1	BRCLR 0	BCLR 0	–
	2	BRSET 1	BSET 1	–
	3	BRCLR 1	BCLR 1	COM
L	4	BRSET 2	BSET 2	LSR
o	5	BRCLR 2	BCLR 2	–
w	6	BRSET 3	BSET 3	ROR
	7	BRCLR 3	BCLR 3	ASR
	8	BRSET 4	BSET 4	LSL/ASL
	9	BRCLR 4	BCLR 4	ROL
	A	BRSET 5	BSET 5	DEC
	B	BRCLR 5	BCLR 5	–
	C	BRSET 6	BSET 6	INC
	D	BRCLR 6	BCLR 6	TST
	E	BRSET 7	BSET 7	–
	F	BRCLR 7	BCLR 7	CLR
		3/10	2/7	2/6

[Note] 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and cycles required.

User can write into port C_6 by software.

Accordingly, after port C_6 has output by hardware and is immediately write into by software, simultaneous cyclic pulse control with a short width is easy.

- **Prescaler Control Register 2 (PCR2: $019)**

The selection of clock input source and prescaler dividing ratio are performed by the Prescaler Control Register 2 (PCR2).

The selection of clock input source is performed in three different ways by bit 4 and bit 5 of the PCR2, as shown in Table 3. At reset, internal clock ϕ_2 controlled by the TIMER input (bit 4 = 1, bit 5 = 0) is selected.

The prescaler dividing ratio is selected by three bits in the PCR2 (bits 0, 1, 2), as shown in Table 4. The dividing ratio can be selected in 8 ways (÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128). At reset, ÷1 (bit 0 = bit 1 = bit 2 = 0) is selected.

When writing into the PCR2, or when writing into the TDR2,

prescaler is initialized to $FF.

Prescaler Control Register 2 (PCR2: $019)

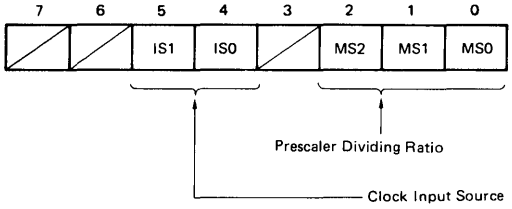


Table 4 Selection of Clock Input Source

PCR2		Clock Input Source
Bit 5	Bit 4	
0	0	Internal Clock ϕ_2 *
0	1	ϕ_2 Controlled by TIMER Input
1	0	
1	1	Event Input from TIMER

* The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock input.

Table 5 Selection of Prescaler Dividing Ratio

PCR2			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

■ SELF CHECK

The MCU self check easily determines whether the LSI functions normally or not. When the MCU is connected as shown in Fig. 8, the outputs of port C_3 (LED) flicker in normal operation. ROM, RAM, TIMER 1, $\overline{INT}_1$, Interrupt, I/O of Port A, B and C are checked by this capability.

■ RESETS

The MCU can be reset three ways; by initial power-up, by the external reset input ($\overline{RES}$) and by an optional internal low voltage detect circuit, see Figure 9. All the I/O ports are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High". This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 10, typically provides sufficient delay.

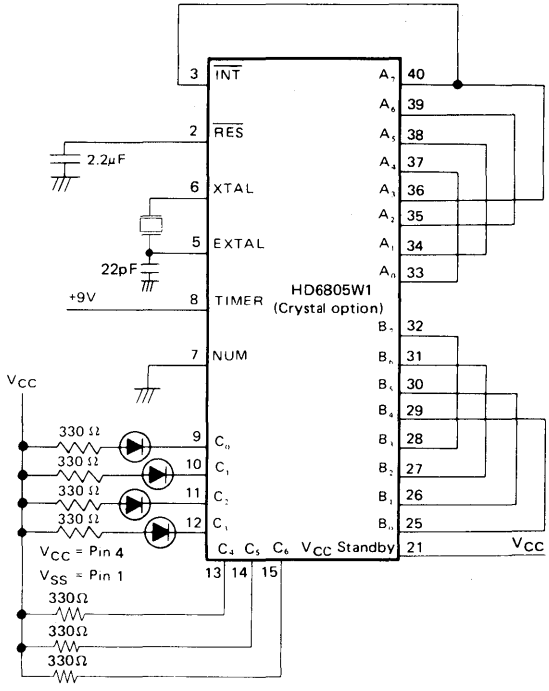


Figure 8 Self Check Connections

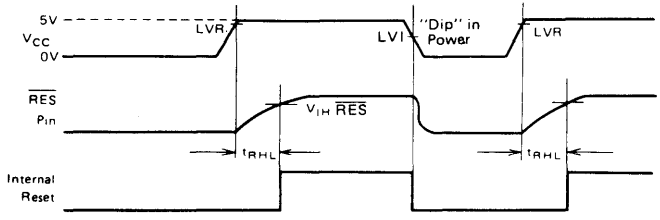


Figure 9 Power Up and Reset Timing

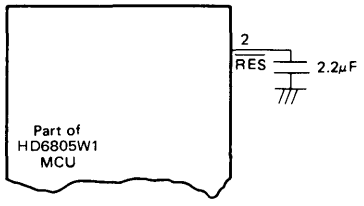


Figure 10 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR OPTIONS

The internal oscillator circuit is designed to require a minimum of external components. A crystal (AT cut, 4 MHz max), a resistor, a jumper wire or an external signal may be used to generate a system clock with various stability/cost tradeoffs. A manufacturing mask option is required to select either the crystal oscillator or the RC oscillator circuit. Four different connection methods are shown in Figure 11. Crystal specifications are given in Figure 12. A resistor selection graph is shown in Figure 13. EXTAL may be driven with a duty cycle of 50% with XTAL connected to ground.

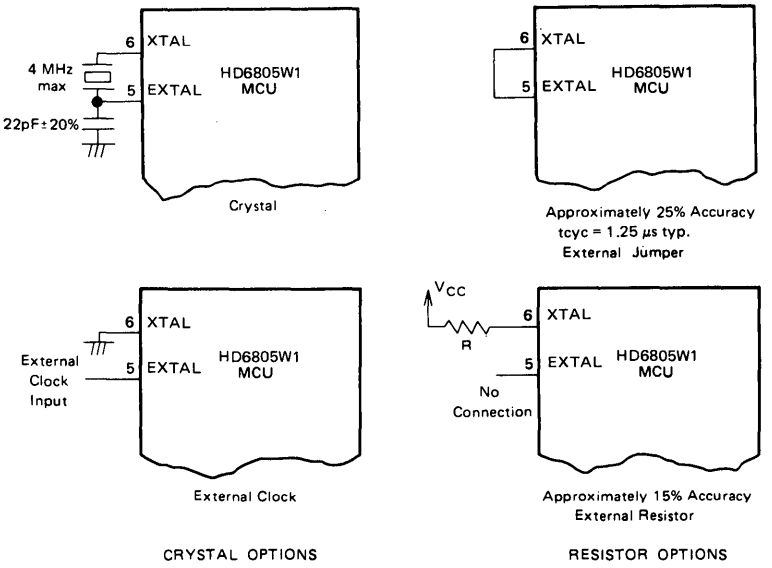
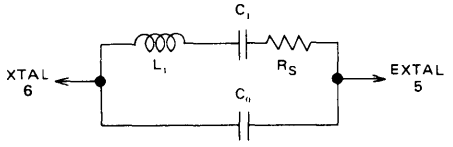


Figure 11 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz (C_1 = 22pF ± 20%)
R_S = 60 Ω max.

Figure 12 Crystal Parameters

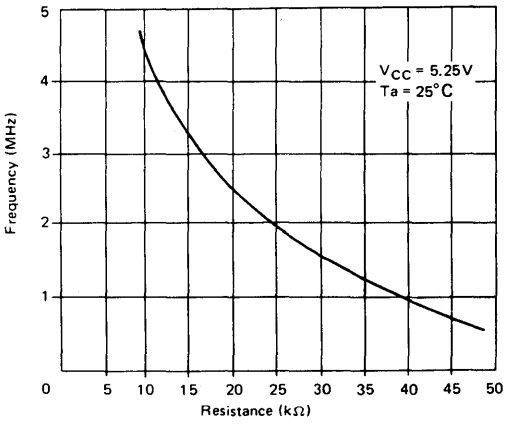


Figure 13 Typical Resistor Selection Graph

■ INTERRUPTS

The MCU can be interrupted in seven different ways: through external interrupt input pin ($\overline{INT_1}$ and $\overline{INT_2}$), internal timer interrupt request (Timer 1, ICI, OCI and OFI) and a software interrupt instruction (SWI). $\overline{INT_2}$ and Timer 1 are generated by the same vector address. When interrupt occurs, processing of the program is suspended, the present CPU state is pushed onto the stack in the order shown in Figure 4. The interrupt mask bit (I) of the Condition Code Register is set and the external routine priority address is achieved from the special external vector address. After that, the external interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires 11 cycles. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. The priority interrupts are shown in Table 6 with the vector address that contains the starting address of the appropriate interrupt routine. The interrupt sequence is shown as a flowchart in Figure 14.

Table 6 Interrupt Priorities

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$FFE, $FFF
SWI	2	$FFC, $FFD
$\overline{INT_1}$	3	$FFA, $FFB
TIMER$_1$/$\overline{INT_2}$	4	$FF8, $FF9
ICI	5	$FF6, $FF7
OCI	6	$FF4, $FF5
OFI	7	$FF2, $FF3

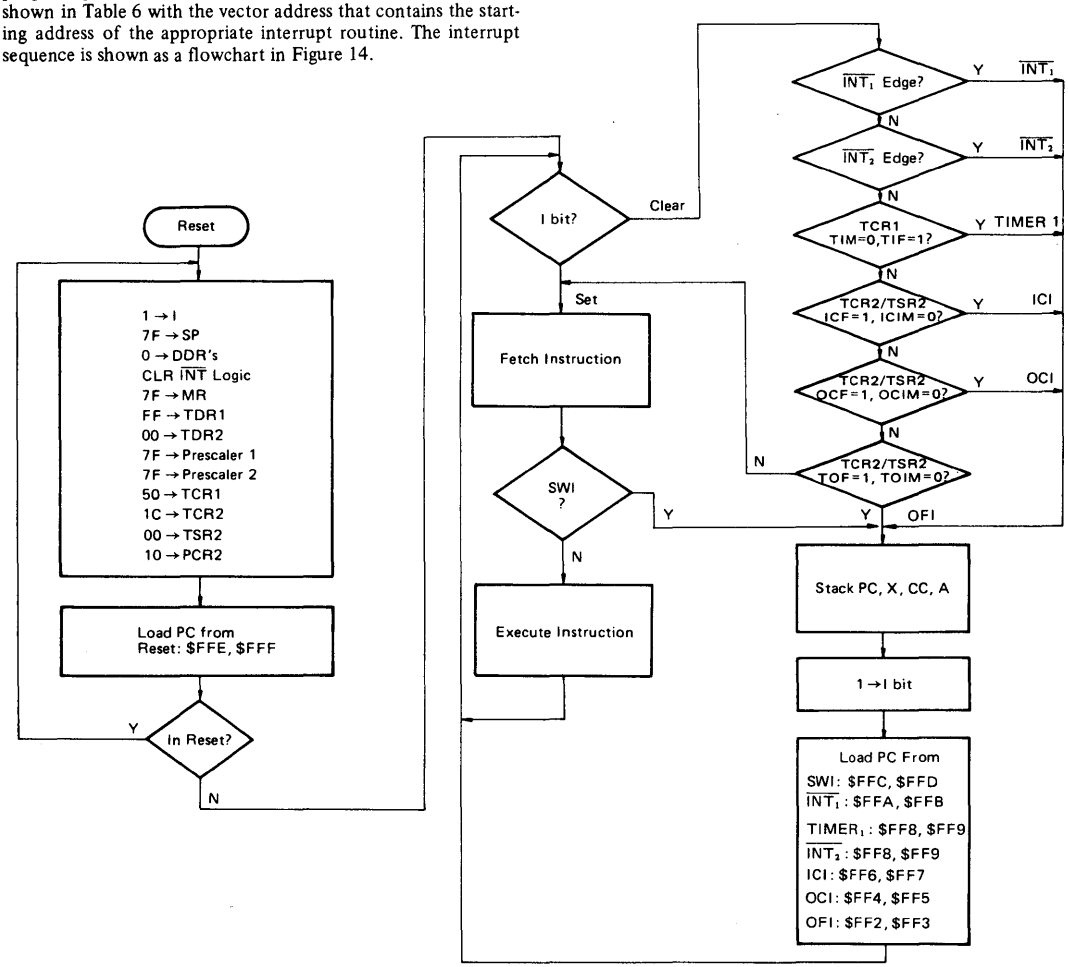


Figure 14 Interrupt Flowchart

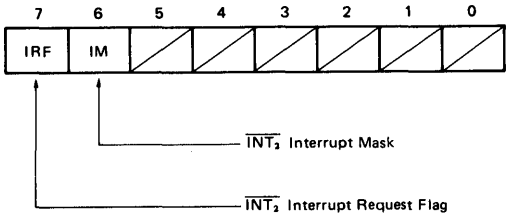
- **Miscellaneous Register (MR: $00A)**

The vector address generated by the external interrupt ($\overline{INT_2}$) is the same as that of TIMER1 as shown in Table 6. The Miscellaneous Register (MR) controls the $\overline{INT_2}$ interrupt.

Bit 7 (IRF) of the MR is used as an $\overline{INT_2}$ interrupt request flag. $\overline{INT_2}$ interrupt occurs at the $\overline{INT_2}$ negative edge, and IRF is set. $\overline{INT_2}$ interrupt or not can be proved by checking IRF by software in the interrupt routine of the vector address ($FF8, $FF9). IRF should be reset by software (BCLR instruction).

Bit 6 (IM) of the MR is an $\overline{INT_2}$ interrupt mask bit. When $\overline{IM}$ is set, $\overline{INT_2}$ interrupt is disabled. $\overline{INT_2}$ interrupt is also disabled by bit (I) of the Condition Code Register (CC) like other interrupts.

Miscellaneous Register (MR: $00A)



IRF is available for both read and write. However, IRF is not writable by software. Therefore, $\overline{INT_2}$ interrupt cannot be requested by software. At reset, IRF is cleared and IM is set.

- **INPUT/OUTPUT**

There are 23 input/output pins. All pins (port A, B, and C) are programmable as either inputs or outputs under software control of the corresponding Data Direction Register (DDR). The port I/O programming is accomplished by writing the corresponding bit in the port DDR to a logic "1" for outpt or a logic "0" for input. On reset, all the DDRs are initialized to a logic "0" state to put the ports in the input mode. The port output registers are not initialized on reset but may be written to before setting the DDR bits to avoid undefined levels. When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading; see Figure 15. When port B is programmed for outputs, it is capable of sinking 10 mA and sourcing 1 mA on each pin.

All input/output lines are TTL compatible as both inputs and outputs. Ports B and C are CMOS compatible as inputs. Port A is CMOS compatible as outputs. Figure 16 provides some examples of port connections.

Port C_5 and C_6 are also used for Timer 2.

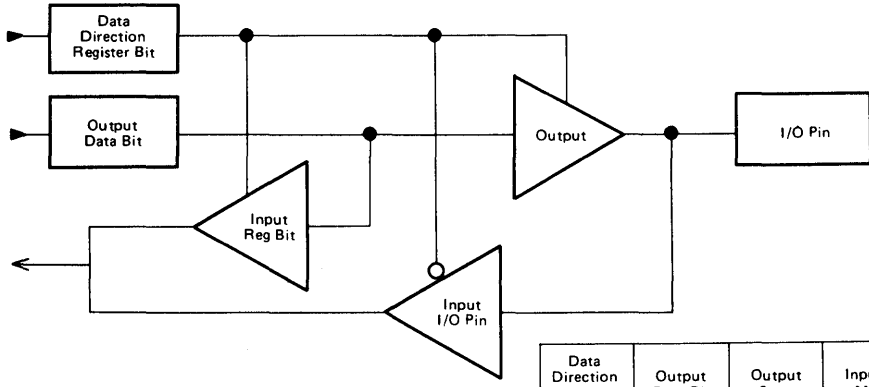
When Port C_5 is used as Timer 2 Input Capture (IC), Port C_5's DDR should be cleared (Port C_5 as input) and bit 4 (ICIM) in the Timer Control Register 2 (TCR2) should be cleared too. The Input Capture Register (ICR) stores the TDR2 when a Port C_5 input transition occurs as defined by bit 1 (IDEG) of the TCR2.

When Port C_6 is used as Timer 2 Output Compare (OC), Port C_6's DDR should be set (Port C_6 as output). When the Output Compare Register (OCR) matches the TDR2, bit 0 (OLVL) in the TCR2 is set and OLVL will appear at Port C_6. Port C_6 is writable by software. But the writing by software is unavailable when a match between the TDR2 and the OCR is found at the same time.

- **INPUT**

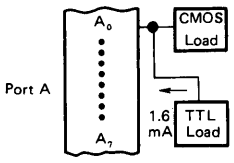
Port D is usable as either TTL compatible inputs or a 4-channel input for an A/D converter. Figure 17 shows port D logic configuration.

The Port D register at location $003 stores TTL compatible inputs. When using as analog inputs for an A/D converter, refer to A/D CONVERTER. D_0 can be used as the $\overline{INT_2}$ interrupt input pin.

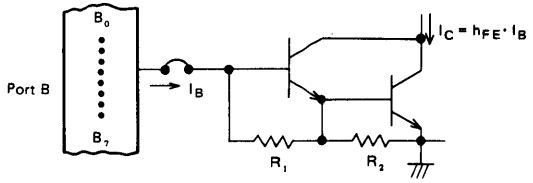


Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

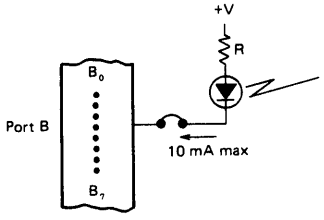
Figure 15 Typical Port I/O Circuitry



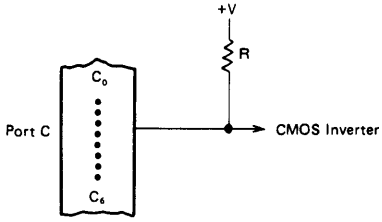
Port A Programmed as output(s), driving CMOS and TTL load directly.
(a)



Port B Programmed as output(s), driving Darlington base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)



Port C Programmed as output(s), driving CMOS loads, using external pull-up resistors. (d)

Figure 16 Typical Port Connections

■ A/D CONVERTER

The HD6805W1 has an internal 8-bit A/D converter. The A/D converter, shown in Figure 18, includes 4 analog inputs (AN_0 to AN_3), the Result Register (ADRR) and the Control Status Register (ADCSR).

• Analog Input (AN_0 to AN_3)

Analog inputs AN_0 to AN_3 accept analog voltages of 0V to 5V. The resolution is 8-bit (256 divisions) with a conversion time of 76 μs at 1 MHz. Analog conversion starts selecting analog inputs by bit 0 and bit 1 of the ADCSR analog input. Since the CPU is not required during conversion, other user programs can be executed.

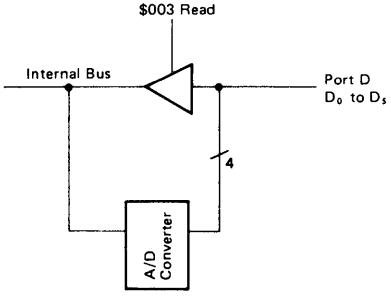


Figure 17 Port D

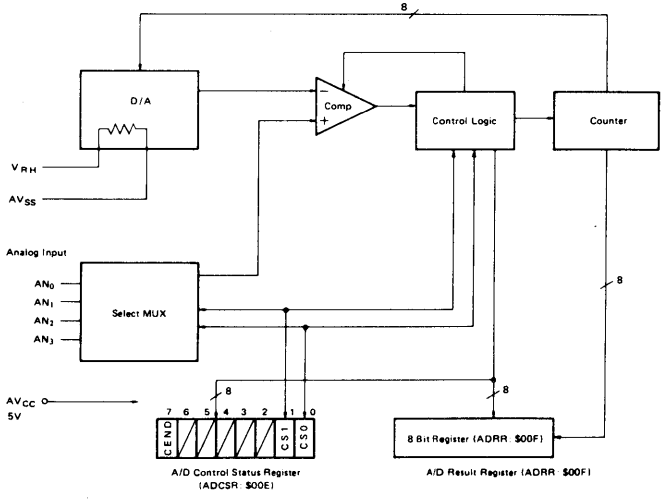


Figure 18 A/D Converter Block Diagram

CAUTION

The MCU has circuitry to protect the inputs against damage due to high static voltages or electric field; however, the design of the input circuitry for the A/D converter, $AN_0 \sim AN_3$, V_{RH} and AV_{CC}, does not offer the same level of protection. Precautions should be taken to avoid applications of any voltage higher than maximum-rated voltage or handled in any environment producing high-static voltages.

Table 7 Analog Input Selection

ADCSR		Analog Input Signal
Bit 1	Bit 0	
0	0	AN_0
0	1	AN_1
1	0	AN_2
1	1	AN_3

- **A/D Control Status Register (ADCSR: $00E)**

The Control Status Register (ADCSR) is used to select an analog input pin and confirm A/D conversion termination. An analog input pin is selected by bit 0 and bit 1 as shown in Table 7.

A/D conversion begins when the data is written into bit 0 and bit 1 of the ADCSR. When A/D conversion ends, bit 7 (CEND) is set. Bit 7 is reset after the ADRR is read. Even if bit 7 is set, A/D conversion execution still continues. To end the A/D conversion, the A/D Result Register (ADRR) stores the most current value. During A/D conversion execution, new data is written into the ADCSR selecting the input channel and the A/D conversion execution at that time is suspended. CEND is reset and new A/D conversion begins.

- **A/D Result Register (ADRR: $00F)**

When the A/D conversion ends, the result is set in the A/D Result Register ($00F). When CEND of the ADCSR is set, converted result is obtained by reading the ADRR. Furthermore, CEND is cleared.

■ STANDBY RAM

The portion from $020 to $027 of the RAM can be used for the standby RAM.

When using the standby RAM, V_{CC} Standby should remain above V_{SBB} (min) during powerdown. Consequently, power is provided only to the standby RAM and STBY PWR bit of the RAM Control Register. 8 byte RAM is sustained with small power dissipation. The RAM including the standby RAM is controlled by the RAM Control Register (RCR) or RAME pin.

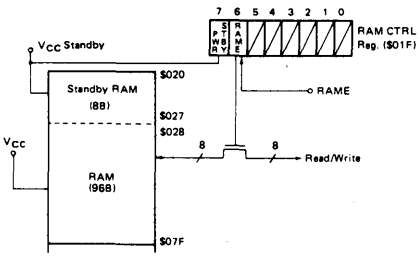
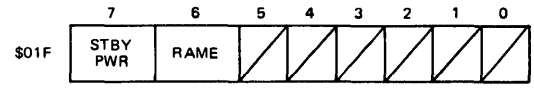


Figure 19 Standby RAM

- **RAM Control Register (RCR: $01F)**

This register at location $01F gives the status information about the RAM. When RAM Enable bit (RAME) is "0", the RAM is disabled. When V_{CC} Standby is greater than V_{SBB}, Standby Power bit (STBY PWR) is set and the standby RAM is sustained during powerdown.

RAM Control Register



Bit 6 RAM Enable

RAME bit is set or cleared by either software or hardware. When the MCU is reset, RAME bit is set and the RAM is enabled. If RAME bit is cleared, the user can neither read nor write the RAM.

When the RAM is disabled (logic "0"), the RAM address is invalid.

Bit 7 Standby Power

STBY PWR bit is cleared whenever V_{CC} standby decreases below V_{SBB} (min). This bit is a read/write status bit that the user can read. When this bit is set, it indicates that the standby power is applied and data in the standby RAM is valid. This bit is set by software and not affected by reset.

- **RAME Signal**

RAME bit in the RCR can be cleared when RAME pin goes "Low" by hardware (RAM is disabled). To make standby mode by hardware, set RAME pin "Low" during V_{CC} Standby remains above V_{SBB} (min) and powerdown sequence should be as shown in Fig. 20.

When RAME pin gets "Low" in the powerup state, RAME bit of the RCR is cleared and the RAM is disabled. During powerdown, RAME bit is sustained by V_{CC} Standby. When RAME pin gets "High" in the powerup state, RAME bit of the RCR is set and the RAM is enabled.

RAME pin can be used to control the RAM externally without software.

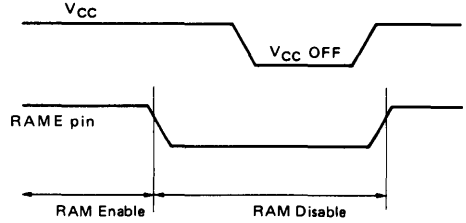


Figure 20 RAM Control Signal (RAME)

■ BIT MANIPULATION

The MCU has the ability to set or clear any single RAM or input/output port (except the data direction registers) with a single instruction (BSET and BCLR). Any bit in the page zero read only memory can be tested by using the BRSET and

BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 21 shows the usefulness of the bit manipulation and test instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which powers the controlled hardware.

This program, which uses only seven bytes of ROM provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer is also incorporated to provide turn-on at some later time which permits pulse-width modulation of the controlled power.

```
SELF 1    BRCLR0, PORTA, SELF 1
          BSET 1, PORTA
          BCLR 1, PORTA
```

Figure 21 Bit Manipulation Example

■ ADDRESSING MODES

The MCU has ten addressing modes available for use by the programmer. These modes are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 22. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 23. In direct addressing, the address of the operand is contained in the secondbyte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 24. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 25. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA = (PC) + 2 + Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken, Rel = 0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 bytes of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 26. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 27. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 28. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 29. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 30. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($00 through $FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit to be tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 31. The implied mode of addressing has no EA. All of the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI and RTI belong to this group. All implied addressing instructions are one byte long.

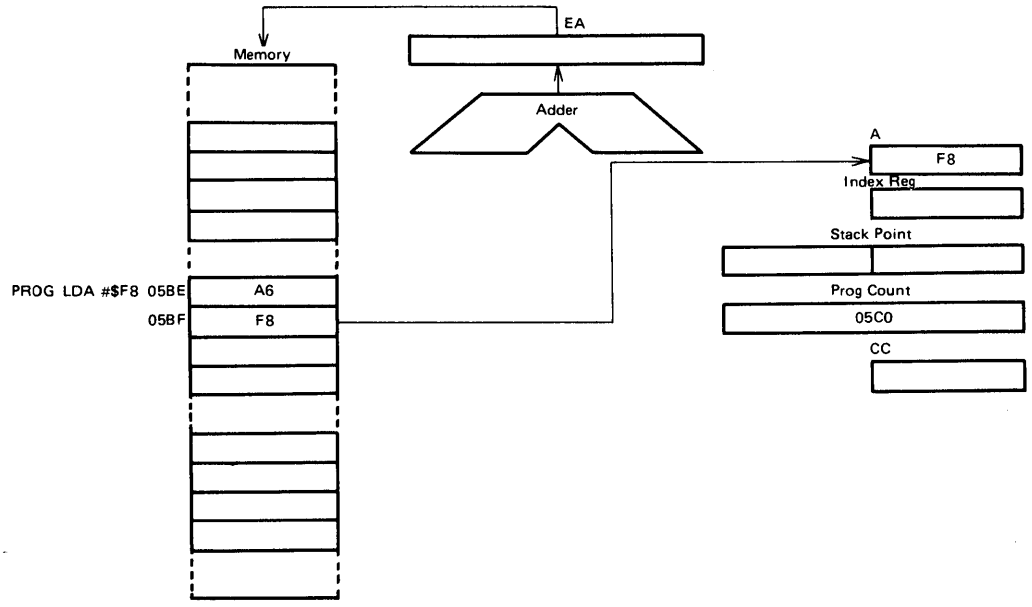


Figure 22 Immediate Addressing Example

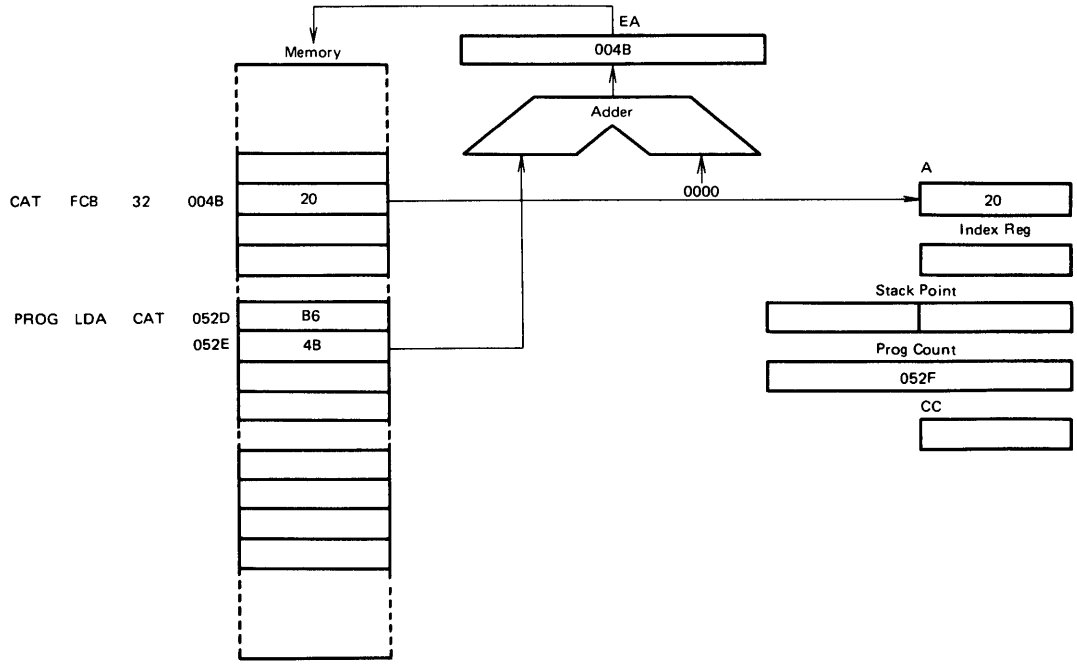


Figure 23 Direct Addressing Example

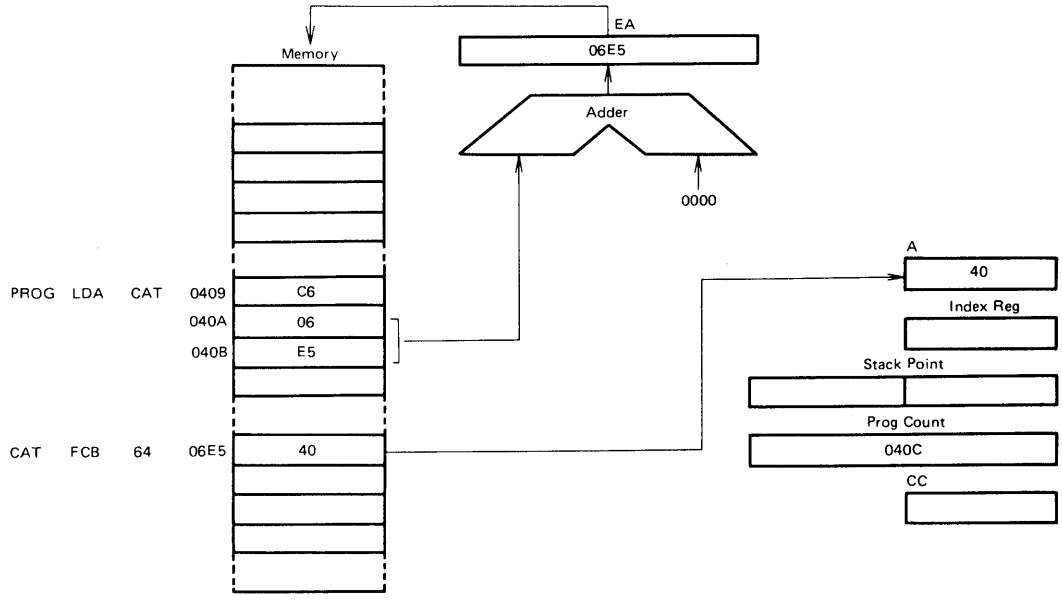


Figure 24 Extended Addressing Example

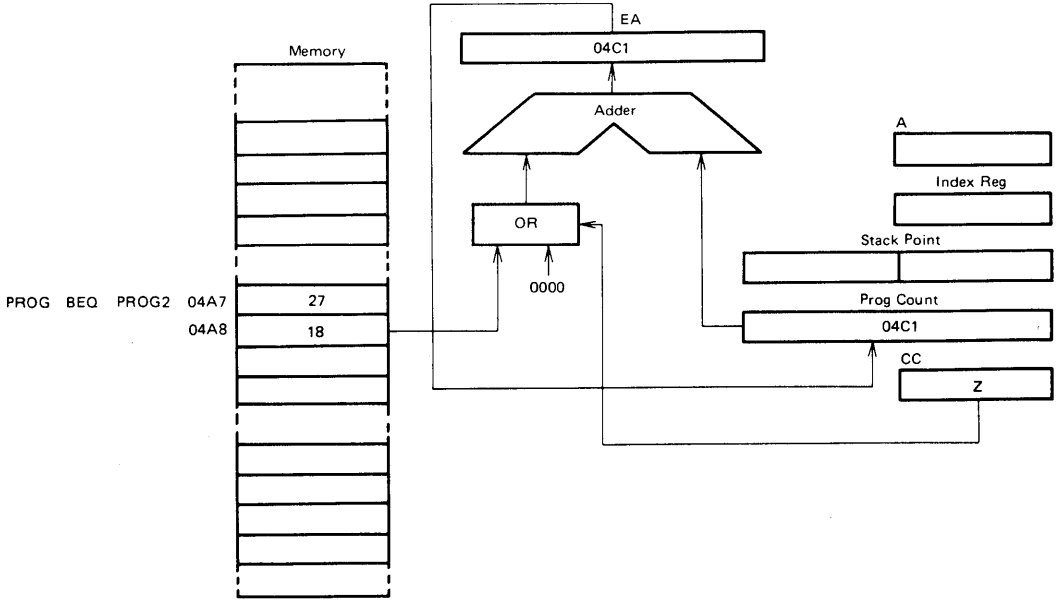


Figure 25 Relative Addressing Example

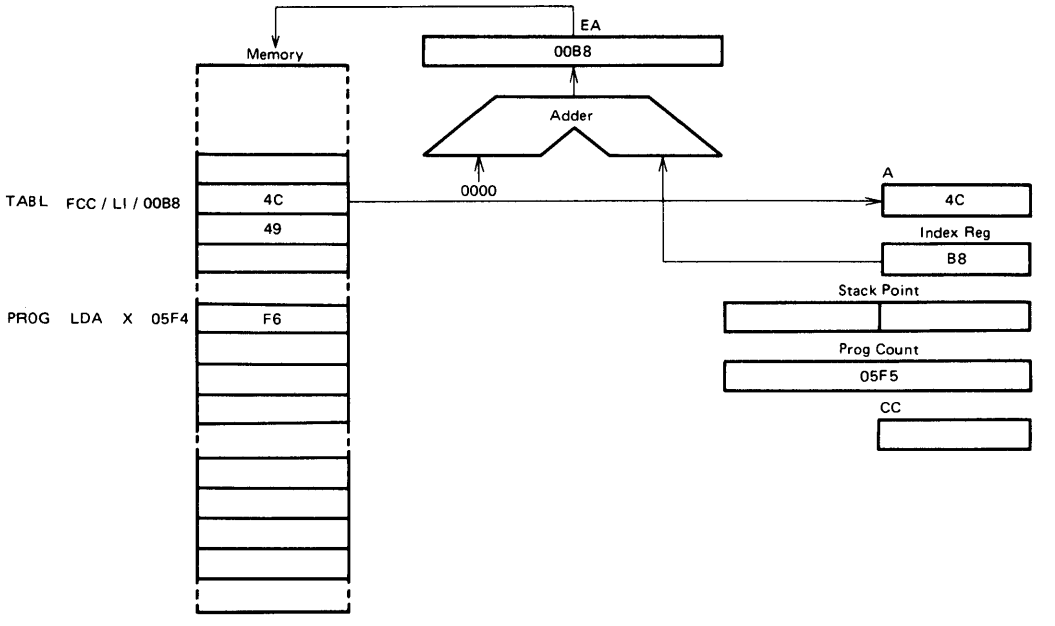


Figure 26 Indexed (No Offset) Addressing Example

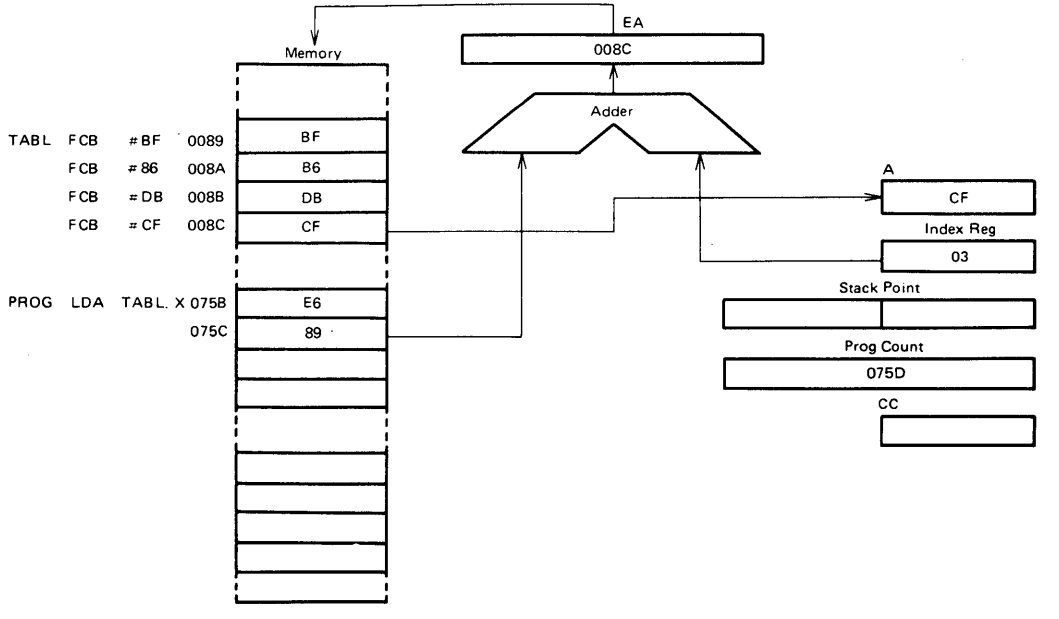


Figure 27 Indexed (8-Bit Offset) Addressing Example

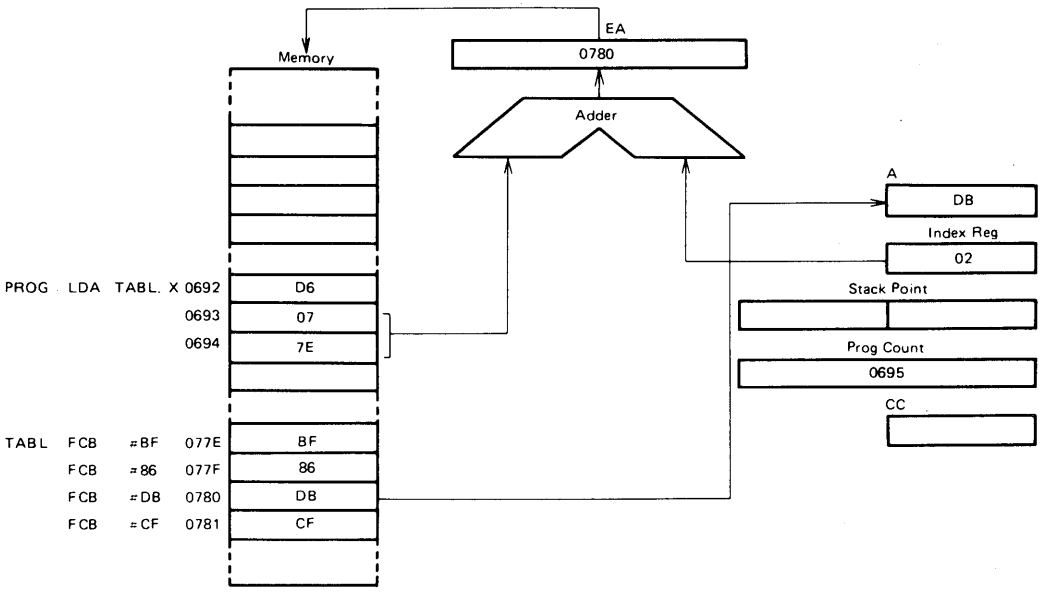


Figure 28 Indexed (16-Bit Offset) Addressing Example

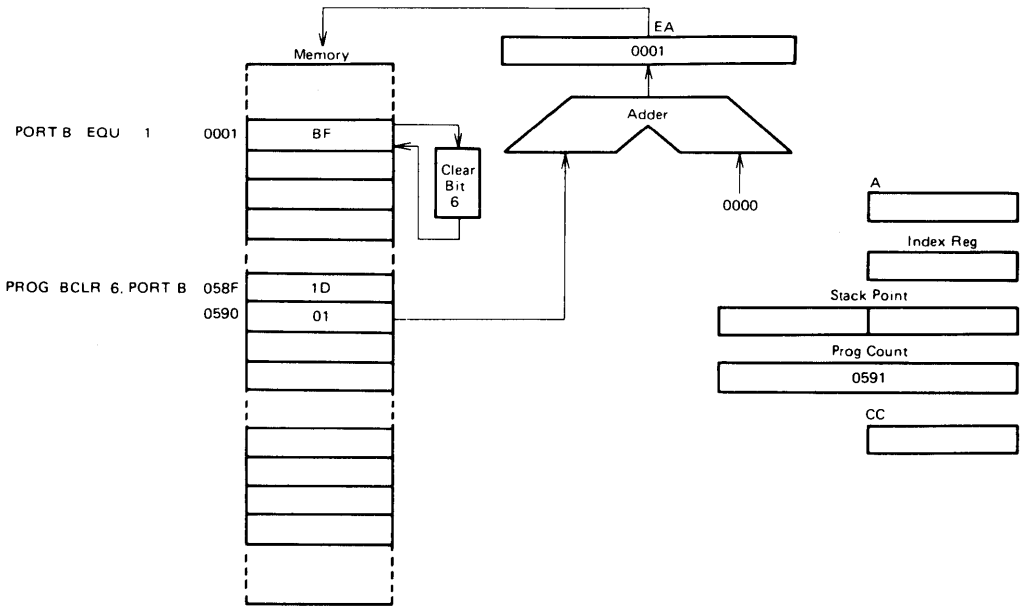


Figure 29 Bit Set/Clear Addressing Example

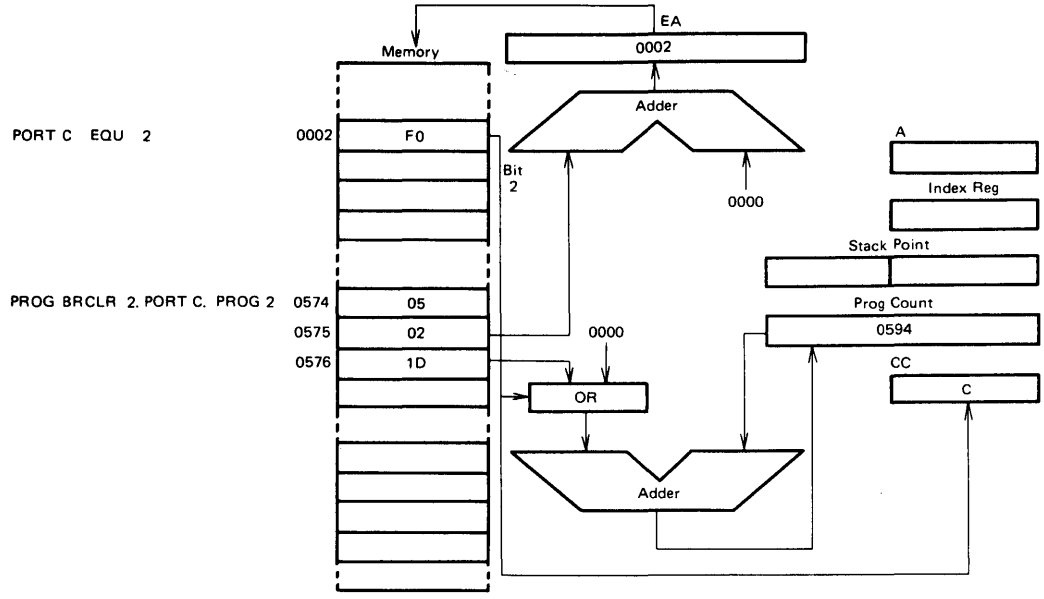


Figure 30 Bit Test and Branch Addressing Example

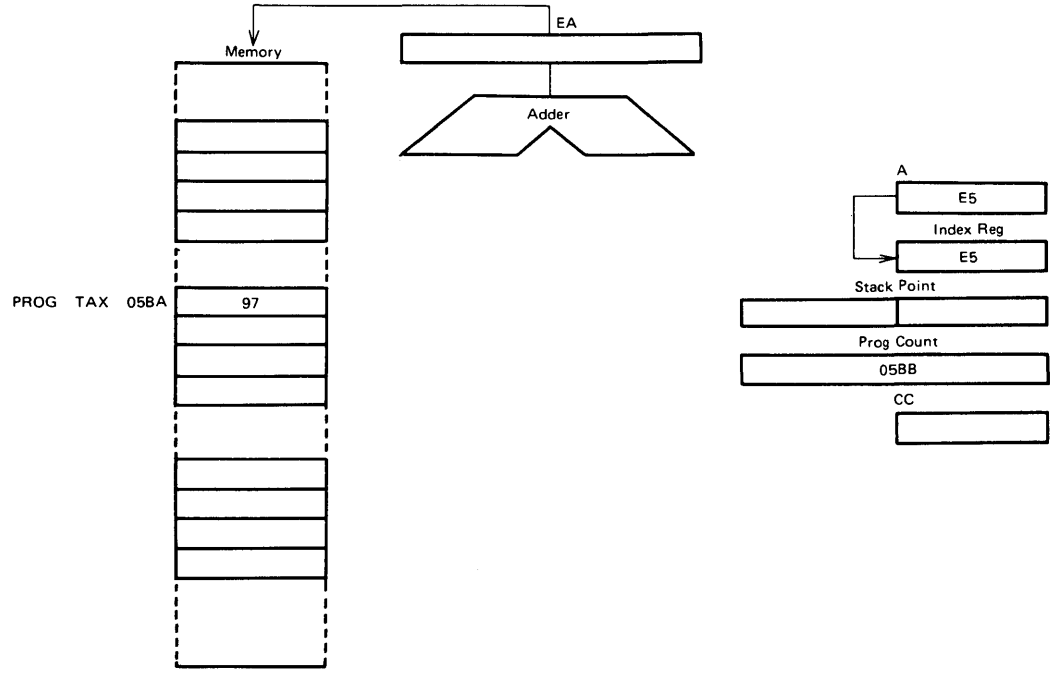


Figure 31 Implied Addressing Example

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. These instructions can be divided into five different types; register/memory, read/modify/write, branch, bit manipulation and control. Each instruction is breifly explained below. All of the instructions within a given type are presented in individual tables.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory by using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 8.

• Read/Modify/Write Instructions

These instructions read a memory location or a register, modify or test its contents and write the modified value back to the memory or register. The TST instruction for test of negative or zero is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 9.

• Branch Instructions

The branch instructional cause a branch from a program when a certain condition is met. Refer to Table 10.

• Bit Manipulation Instructions

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 11.

• Control Instructions

The control instructions control the MCU operations during program execution. Refer to Table 12.

• Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 13.

• Opcode Map

Table 14 is an opcode map for the instructions used on the MCU.

Table 8 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Symbols:
Op: Operation Abbreviation
: Instruction Statement

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 9 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Symbols:
Op: Operation Abbreviation
: Instruction Statement

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 10 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 11 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 12 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 13 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

(to be continued)

Table 13 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEQ	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

Table 14 Opcode Map

	Bit Manipulation		Brnch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	←HIGH
0	BRSET0	BSET0	BRA	NEQ					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(-1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

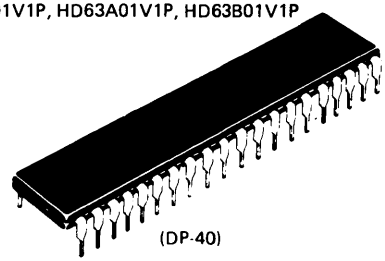
[NOTE] 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and the number of cycles required (Bytes/Cycles).
Mnemonics followed by a "•" require a different number of cycles as follows:
RTI 9
RTS 6
SWI 11
BSR 8
3. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

HD6301V1, HD63A01V1, HD63B01V1

CMOS MCU (Microcomputer Unit)

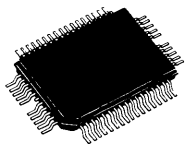
The HD6301V1 is an 8-bit CMOS single-chip microcomputer unit, Object Code compatible with the HD6801. 4kB ROM, 128 bytes RAM, Serial Communication Interface (SCI), parallel I/O ports and multi function timer are incorporated in the HD6301V1. It is bus compatible with HMCS6800. Execution time of key instructions are improved and several new instructions are added to increase system throughput. The HD6301V1 can be expanded up to 65k bytes. Like the HMCS6800 family, I/O level is TTL compatible with +5.0V single power supply. As HD6301V1 is fabricated by the advanced CMOS process technology, power dissipation is extremely reduced. In addition to that, HD6301V1 has Sleep Mode and Standby Mode at lower power dissipation mode. Therefore flexible low power consumption application is possible.

HD6301V1P, HD63A01V1P, HD63B01V1P



(DP-40)

HD6301V1F, HD63A01V1F, HD63B01V1F



(FP-54)

HD6301V1CG,HD63A01V1CG,HD63B01V1CG

(CG-40)

■ FEATURES

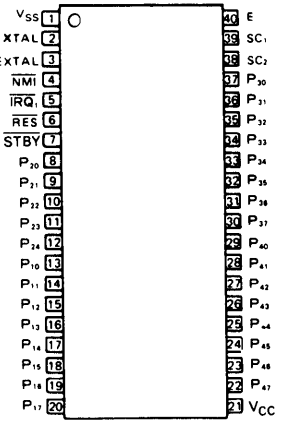
- Object Code Upward Compatible with HD6801 Family
- Abundant On-Chip Functions Compatible with HD6801V0; 4kB ROM,128 Bytes RAM,29 Parallel I/O Lines, 2 Lines of Data Strobe, 16-bit Timer, Serial Communication Interface
- Low Power Consumption Mode: Sleep Mode, Standby Mode
- Minimum Instruction Execution Time
  1μs (f=1MHz), 0.67μs (f=1.5MHz), 0.5μs (f=2MHz)
- Bit Manipulation, Bit Test Instruction
- Protection from System Upset: Address Trap, On-Code Trap
- Up to 65k Words Address Space
- Wide Operation Range
  V_{CC}=3 to 6V (f=0.1~0.5MHz),
  f=0.1 to 2.0MHz (V_{CC}=5V±10%)

■ TYPE OF PRODUCTS

Type No.	Bus Timing
HD6301V1	1 MHz
HD63A01V1	1.5 MHz
HD63B01V1	2 MHz

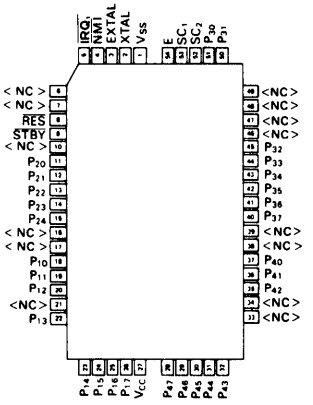
■ PIN ARRANGEMENT

- **HD6301V1P, HD63A01V1P, HD63B01V1P**



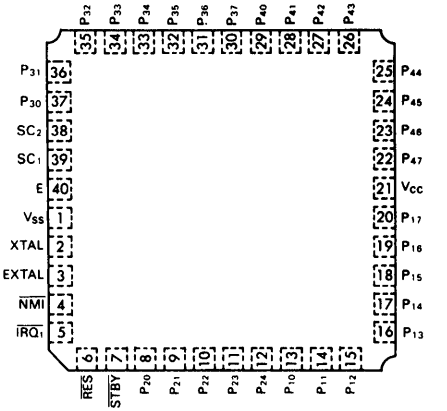
(Top View)

- **HD6301V1F, HD63A01V1F, HD63B01V1F**



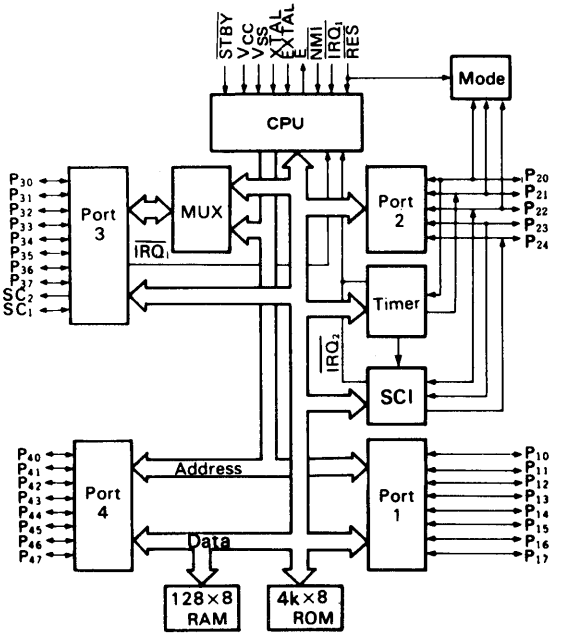
(Top View)

- **HD6301V1CG, HD63A01V1CG, HD63B01V1CG**



(Top View)

■ BLOCK DIAGRAM



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out} : $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}x0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$, $\overline{STBY}$	$\|I_{in}\|$	V_{in} =0.5~V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	P_{10}~P_{17}, P_{20}~P_{24}, P_{30}~P_{37}, P_{40}~P_{47}, $\overline{IS3}$	$\|I_{TSI}\|$	V_{in} =0.5~V_{CC} -0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input Capacitance	All Inputs	C_{in}	V_{in}=0V, f=1.0MHz, Ta = 25°C	–	–	12.5	pF
Standby Current	Non Operation	I_{CC}		–	2.0	15.0	μA
Current Dissipation*		I_{CC}	Operating (f=1MHz**)	–	6.0	10.0	mA
			Sleeping (f=1MHz**)	–	1.0	2.0	
RAM Stand-By Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at f = x MHz operation are decided according to the following formula;

typ. value (f = x MHz) = typ. value (f = 1MHz) x x
max. value (f = x MHz) = max. value (f = 1MHz) x x
(both the sleeping and operating)

● **AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)**

BUS TIMING

Item		Symbol	Test Condition	HD6301V1 min	HD6301V1 typ	HD6301V1 max	HD63A01V1 min	HD63A01V1 typ	HD63A01V1 max	HD63B01V1 min	HD63B01V1 typ	HD63B01V1 max	Unit
Cycle Time		t_{cyc}		1	–	10	0.666	–	10	0.5	–	10	μs
Address Strobe Pulse Width "High" *		PW_{ASH}		220	–	–	150	–	–	110	–	–	ns
Address Strobe Rise Time		t_{ASr}		–	–	20	–	–	20	–	–	20	ns
Address Strobe Fall Time		t_{ASf}		–	–	20	–	–	20	–	–	20	ns
Address Strobe Delay Time *		t_{ASD}		60	–	–	40	–	–	20	–	–	ns
Enable Rise Time		t_{Er}		–	–	20	–	–	20	–	–	20	ns
Enable Fall Time		t_{Ef}		–	–	20	–	–	20	–	–	20	ns
Enable Pulse Width "High" Level *		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level *		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address Strobe to Enable Delay Time*		t_{ASED}		60	–	–	40	–	–	20	–	–	ns
Address Delay Time		t_{AD1}		–	–	250	–	–	190	–	–	160	ns
		t_{AD2}	Fig. 1	–	–	250	–	–	190	–	–	160	ns
Address Delay Time for Latch *		t_{ADL}	Fig. 2	–	–	250	–	–	190	–	–	160	ns
Data Set-up Time	Write	t_{DSW}		230	–	–	150	–	–	100	–	–	ns
	Read	t_{DSR}		80	–	–	60	–	–	50	–	–	ns
Data Hold Time	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	20	–	–	ns
Address Set-up Time for Latch *		t_{ASL}		60	–	–	40	–	–	20	–	–	ns
Address Hold Time for Latch		t_{AHL}		30	–	–	20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	20	–	–	ns
A_0 ~ A_7 Set-up Time Before E *		t_{ASM}		200	–	–	110	–	–	60	–	–	ns
Peripheral Read Access Time	Non-Multiplexed Bus *	(t_{ACCN})		–	–	650	–	–	395	–	–	270	ns
	Multiplexed Bus *	(t_{ACCM})		–	–	650	–	–	395	–	–	270	ns
Oscillator stabilization Time		t_{RC}	Fig. 10	20	–	–	20	–	–	20	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 11	200	–	–	200	–	–	200	–	–	ns

*These timings change in approximate proportion to tcyc. The figures in this characteristics represent these when tcyc is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD6301V1 min	HD6301V1 typ	HD6301V1 max	HD63A01V1 min	HD63A01V1 typ	HD63A01V1 max	HD63B01V1 min	HD63B01V1 typ	HD63B01V1 max	Unit
Peripheral Data Set-up Time	Port 1, 2, 3, 4	t_{PDSU}	Fig. 3	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200	–	–	200	–	–	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		t_{OSD1}	Fig. 5	–	–	300	–	–	300	–	–	300	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		t_{OSD2}	Fig. 5	–	–	300	–	–	300	–	–	300	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4	t_{PWD}	Fig. 4	–	–	300	–	–	300	–	–	300	ns
Input Strobe Pulse Width		t_{PWIS}	Fig. 6	200	–	–	200	–	–	200	–	–	ns
Input Data Hold Time	Port 3	t_{IH}	Fig. 6	150	–	–	150	–	–	150	–	–	ns
Input Data Setup Time	Port 3	t_{IS}	Fig. 6	0	–	–	0	–	–	0	–	–	ns

* Except P_{21}

TIMER, SCI TIMING

Item	Symbol	Test Condition	HD6301V1			HD63A01V1			HD63B01V1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Timer Input Pulse Width	t_{PWT}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	t_{Scyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING

Item	Symbol	Test Condition	HD6301V1			HD63A01V1			HD63B01V1			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{RES}$ "Low" Pulse Width	PW_{RSTL}	Fig. 8	3	–	–	3	–	–	3	–	–	t_{cyc}
Mode Programming Set-up Time	t_{MPS}		2	–	–	2	–	–	2	–	–	t_{cyc}
Mode Programming Hold Time	t_{MPH}		150	–	–	150	–	–	150	–	–	ns

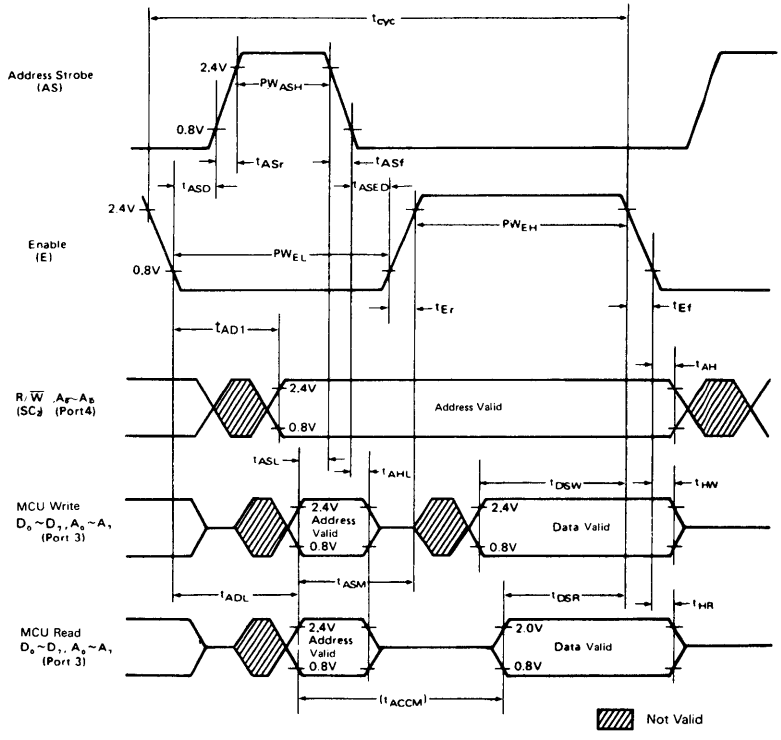


Figure 1 Expanded Multiplexed Bus Timing

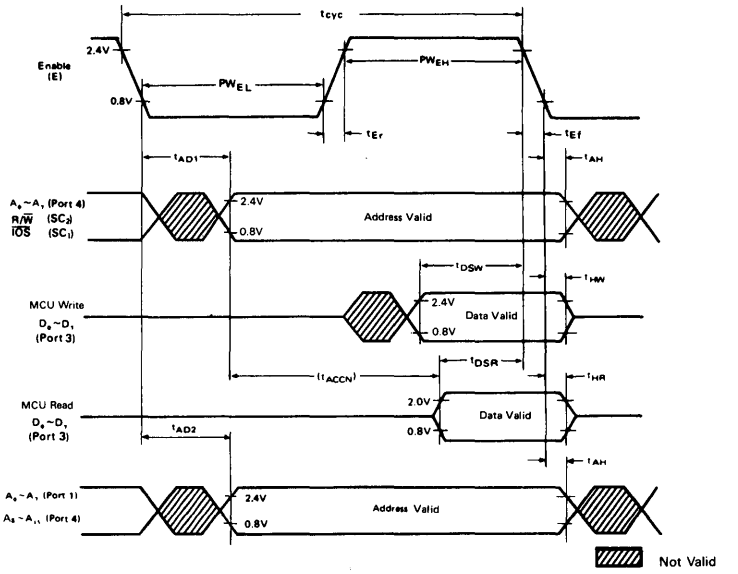
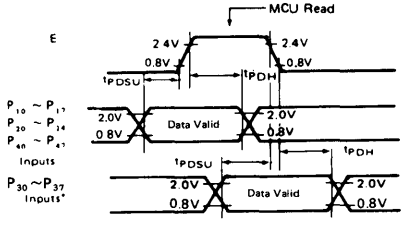
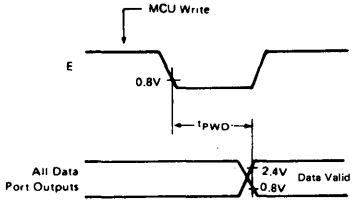


Figure 2 Expanded Non-Multiplexed Bus Timing



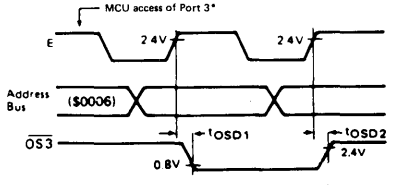
*Port 3 Non-Latched Operation

Figure 3 Port Data Set-up and Hold Times (MCU Read)



Note) Port 2: Except P_{21}

Figure 4 Port Data Delay Times (MCU Write)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

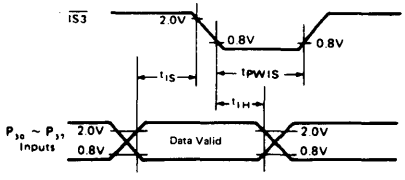


Figure 6 Port 3 Latch Timing (Single Chip Mode)

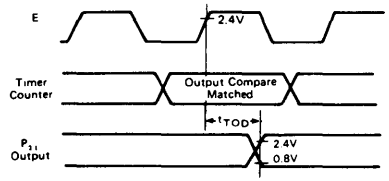


Figure 7 Timer Output Timing

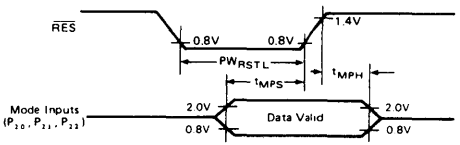
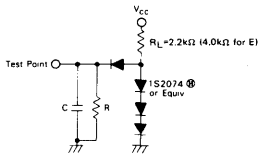


Figure 8 Mode Programming Timing



C = 90pF for P_{30} ~ P_{37}, P_{40} ~ P_{47}, SC_1, SC_2
= 30pF for P_{10} ~ P_{17}, P_{20} ~ P_{24}
= 40pF for E
R = 12kΩ

Figure 9 Bus Timing Test Loads (TTL Load)

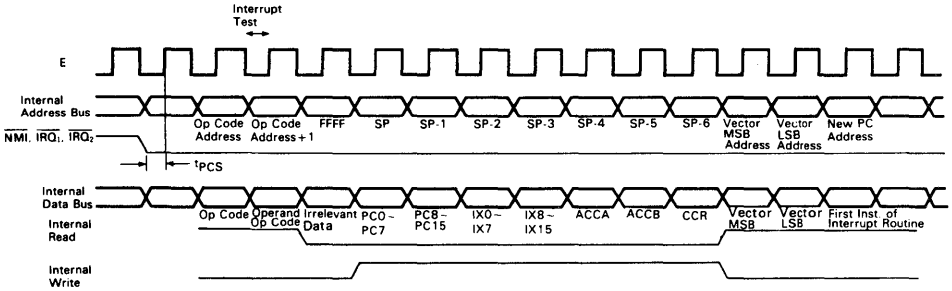


Figure 10 Interrupt Sequence

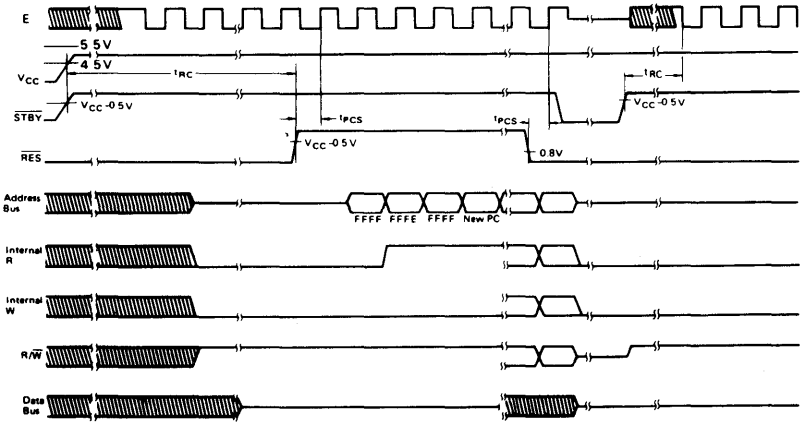


Figure 11 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

These two pins are used for power supply and GND. Recommended power supply voltage is 5V ±10%. 3 to 6V can be used for low speed operation (100 ~ 500 kHz).

• XTAL, EXTAL

These two pins are connected with parallel resonant fundamental crystal, AT cut. For instance, in order to obtain the system clock 1MHz, a 4MHz resonant fundamental crystal is used because the devide-by-4 circuitry is included. An example of the crystal interface is shown in Fig. 12. EXTAL accepts an external clock input of duty 45% to 55% to drive, then internal clock is a quarter the frequency of an external clock. External driving frequency will be less than 4 times as maximum internal clock. For external clock, XTAL pin should be open.

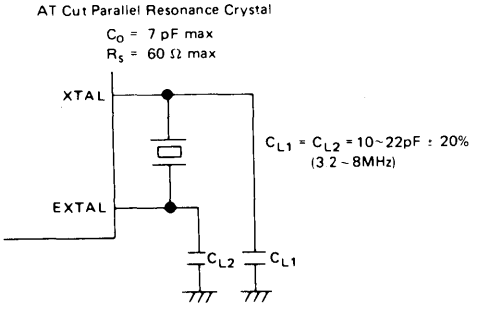


Figure 12 Crystal Interface

• Standby ($\overline{STBY}$)

This pin is used to place the MCU in the Standby mode. If this goes to "Low" level, the oscillation stops, the internal clock is tied to V_{SS} or V_{CC} and the MCU is reset. In order to retain information in RAM during standby mode, write "0" into RAM enable bit (RAME). RAME is bit 6 of the RAM Control Register at address $0014. This disables the RAM, so the contents of RAM is guaranteed. For details of the standby mode, see the Standby section.

• Reset ($\overline{RES}$)

This input resets the MCU. $\overline{RES}$ must be held "Low" for at least 20ms when the power starts up. It should be noted that, before clock generator stabilize, the internal state and I/O ports are uncertain, because MCU can not be reset without clock. To reset the MCU during system operation, it must be held "Low" for at least 3 system clock cycles. From the third cycle, all address buses become "High-impedance" and it continues while $\overline{RES}$ is "Low". If $\overline{RES}$ goes to "High", CPU does the following.

(1) I/O Port 2 bits, 2,1,0 are latched into bits PC2, PC1, PC0 of program control register.
(2) The contents of the two Start Addresses, $FFFE, $FFFF are brought to the program counter, from which program starts (see Table 1).
(3) The interrupt mask bit is set. In order to have the CPU recognize the maskable interrupts $\overline{IRQ_1}$ and $\overline{IRQ_2}$, clear it before those are used.

• Enable (E)

This output pin supplies system clock. Output is a single-phase, TTL compatible and 1/4 of the crystal oscillation frequency. It will drive two LS TTL load and 40pF.

• Non maskable Interrupt ($\overline{NMI}$)

When the falling edge of the input signal of this pin is recognized, NMI sequence starts. The current instruction is continued to complete, even if $\overline{NMI}$ signal is detected. Interrupt mask bit in Condition Code Register has no effect on NMI detection. In response to NMI interrupt, the information of Program Counter, Index Register, Accumulators, and Condition Code Register are stored on the stack. On completion of this sequence, vectoring address $FFFC and $FFFD are generated to load the contents to the program counter. Then the CPU branch to a non maskable interrupt service routine.

• Interrupt Request ($\overline{IRQ_1}$)

This level sensitive input requests maskable interrupt sequence. When $\overline{IRQ_1}$ goes to "Low", the CPU waits until it completes the current instruction that is being executed. Then, if the interrupt mask bit in Condition Code Register is not set, CPU begins interrupt sequence; otherwise, interrupt request is neglected.

Once the sequence has started, the information of Program Counter, Index Register, Accumulators, Condition Code Register are stored on the stack. Then the CPU sets the interrupt mask bit so that no further maskable interrupts may be responded.

Table 1 Interrupt Vectoring memory map

	Vector		Interrupt
	MSB	LSB	
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$ (or $\overline{IS3}$)
	FFF6	FFF7	ICF (Timer Input Capture)
	FFF4	FFF5	OCF (Timer Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

At the end of the cycle, the CPU generates 16 bit vectoring addresses indicating memory addresses $FFF8 and $FFF9, and load the contents to the Program Counter, then branch to an interrupt service routine.

The Internal Interrupt will generate signal ($\overline{IRQ_2}$) which is quite the same as $\overline{IRQ_1}$ except that it will use the vector address $FFF0 to $FFF7.

When $\overline{IRQ_1}$ and $\overline{IRQ_2}$ are generated at the same time, the former precede the latter. Interrupt Mask Bit in the condition code register, if being set, will keep the both interrupts off.

$\overline{IRQ_1}$ has no internal latch. Therefore, if $\overline{IRQ_1}$ is removed during suspension, that $\overline{IRQ_1}$ is ignored.

On occurrence of Address error or Op-code error, TRAP interrupt is invoked. This interrupt has priority next to $\overline{RES}$. Regardless of the Interrupt Mask Bit condition, the CPU will

start an interrupt sequence. The vector for this interrupt will be $FFEE, $FFEF.

The following pins are available only in single chip mode.

- **Input Strobe ($\overline{IS3}$) (SC_1)**

This signal controls $\overline{IS3}$ interrupt and the latch of Port 3. When the falling edge of this signal is detected, the flag of Port 3 Control Status Register is set.

For detailed explanation of Port 3 Control Status Register, see the I/O PORT 3 CONTROL STATUS REGISTER section.

- **Output Strobe ($\overline{OS3}$) (SC_2)**

This signal is used to send a strobe to an external device, indicating effective data is on the I/O pins. The timing chart for Output Strobe are shown in Figure 5.

The following pins are available for Expanded Modes.

- **Read/Write (R/$\overline{W}$) (SC_2)**

This output signal indicates peripheral and memory devices whether CPU is in Read ("High"), or in Write ("Low"). The normal stand-by state is Read ("High"). Its output will drive one TTL load and 90pF.

- **I/O Strobe ($\overline{IOS}$) (SC_1)**

In expanded non multiplexed mode 5 of operation, $\overline{IOS}$ goes to "Low" only when A_9 through A_{15} are "0" and A_8 is "1" This allows external access up to 256 addresses from $0100 to $01FF in memory. The timing chart is shown in Figure 2.

- **Address Strobe (AS) (SC_1)**

In the expanded multiplexed mode, address strobe signal appears at this pin. It is used to latch the lower 8 bits addresses multiplexed with data at Port 3. The 8-bit latch is controlled by address strobe as shown in Figure 18. Thereby, I/O Port 3 can become data bus during E pulse. The timing chart of this signal is shown in Figure 1.

Address Strobe (AS) is sent out even if the internal address area is accessed.

■ PORTS

There are four I/O Ports on HD6301V1 MCU (three 8-bit ports and one 5-bit port). 2 control pins are connected to one of the 8-bit port. Each port has an independent write-only data direction register to program individual I/O pins for input or output.*

When the bit of associated Data Direction Register is "1". I/O pin is programmed for output, if "0", then programmed for an input.

There are four ports: Port 1, Port 2, Port 3, and Port 4. Addresses of each port and associated Data Direction Registers are shown in Table 2.

* Only one exception is bit 1 of Port 2 which becomes either a data input or a timer output. It cannot be used as an output port.

$\overline{RES}$ does not affect I/O port Data Register. Therefore, just after $\overline{RES}$, Data Register is uncertain. Data Direction Registers are reset.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

- **I/O Port 1**

This is an 8-bit port, each bit being defined individually as input or outputs by associated Data Direction Register. The 8-bit output buffers have three-state capability, maintaining in high impedance state when they are used for input. In order to be read accurately, the voltage on the input lines must be more than 2.0V for logic "1" and less than 0.8 V for logic "0".

These are TTL compatible. After the MCU has been reset, all I/O lines of Port 1 are configured as inputs in all modes except mode 1. In all modes except Mode 1, Port 1 is always parallel I/O. In mode 1, Port 1 will be output line for lower order address lines (A_0 to A_7).

- **I/O Port 2**

This port has five lines, whose I/O direction depends on its data direction register. The 5-bit output buffers have three-state capability, going high impedance state when used as inputs. In order to be read accurately, the voltage on the input lines must be more than 2.0V for logic "1" and less than 0.8V for logic "0". After the MCU has been reset, I/O lines are configured as inputs. These pins of Port 2 (pins P_{20}, P_{21}, P_{22} of the chip) are used to program the mode of operation during reset. The values of these three pins during reset are latched into the upper 3 bits (bit 7, 6 and 5) of Port 2 Data Register, which is explained in the MODE SELECTION section.

In all modes, Port 2 can be configured as I/O lines. This port also provides access to the Serial I/O and the Timer. However, note that bit 1 (P_{21}) is the only pin restricted to data input or Timer output.

- **I/O Port 3**

This is an 8-bit port which can be configured as I/O lines, a data bus, or an address bus multiplexed with data bus. Its function depends on hardware operation mode programmed by the user using 3 bits of Port 2 during Reset. Port 3 as a data bus is bi-directional. For an input from peripherals, regular TTL level must be supplied, that is greater than 2.0V for a logic "1" and less than 0.8V for a logic "0". This TTL compatible three-state buffer can drive one TTL load and 90pF capacitance. In the expanded Modes, data direction register will be inhibited after Reset and data direction will depend on the state of the R/$\overline{W}$ line. Function of Port 3 is shown below.

Single Chip Mode (Mode 7)

Parallel Inputs/Outputs as programmed by its corresponding Data Direction Register.

There are two control lines associated with this port in this mode, an input strobe ($\overline{IS3}$) and an output strobe ($\overline{OS3}$), both being used for handshaking. They are controlled by I/O Port 3 Control/Status Register. Function of these two control lines of Port 3 are summarized as follows:

(1) Port 3 input data can be latched using $\overline{IS3}$ (SC_1) as a input strobe signal.
(2) $\overline{OS3}$ can be generated by CPU read or write to Port 3's data register.
(3) $\overline{IRQ_1}$ interrupt can be generated by an $\overline{IS3}$ falling edge.

Port 3 strobe and latch timing is shown in Figs. 5 and 6 respectively.

I/O Port 3 Control/Status Register is explained as follows:

I/O Port 3 Control/Status Register

	7	6	5	4	3	2	1	0
$000F	$\overline{IS3}$ FLAG	$\overline{IS3}$ $\overline{IRQ_1}$ ENABLE	X	OSS	LATCH ENABLE	X	X	X

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.
Bit 3 LATCH ENABLE.

Bit 3 is used to control the input latch of Port 3. If the bit is set at "1", the input data on Port 3 is latched by the falling edge of $\overline{IS3}$. The latch is released by the MCU read to Port 3; now new data can be latched again by $\overline{IS3}$ falling edge. Bit 3 is cleared by a reset. If this bit is "0", $\overline{IS3}$ does not affect I/O Port 3 latch operation.

Bit 4 OSS (Output Strobe Select)

This bit identifies the cause of output strobe generation: a write operation or read operation to I/O Port 3. When the bit is cleared, the strobe will be generated by a read operation to Port 3. When the bit is not cleared, the strobe will be generated by a write operation. Bit 4 is cleared by a reset.

Bit 5 Not used.

Bit 6 $\overline{IS3}$ $\overline{IRQ_1}$ ENABLE.

If this bit is set, $\overline{IRQ_1}$ interrupt by $\overline{IS3}$ Flag is enabled. Otherwise the interrupt is disabled. The bit is cleared by a reset.

Bit 7 $\overline{IS3}$ FLAG.

Bit 7 is a read-only bit which is set by the falling edge of $\overline{IS3}$ (SC_1). It is cleared by a read of the Control/Status Register followed by a read/write of I/O Port 3. The bit is cleared by reset.

Expanded Non Multiplexed Mode (mode 1,5)

In this mode, Port 3 becomes data bus. ($D_0 \sim D_7$)

Expanded Multiplexed Mode (mode 0, 2, 4, 6)

Port 3 becomes both the data bus ($D_0 \sim D_7$) and lower bits of the address bus ($A_0 \sim A_7$). An address strobe output is "High" while the address is on the port.

- **I/O Port 4**

This is an 8-bit port that becomes either I/O or address outputs depending on the selected operation mode. In order to be read accurately, the voltage at the input lines must be greater than 2.0V for a logic "1", and less than 0.8V for a logic "0". For outputs, each line is TTL compatible and can drive one TTL load and 90pF. Function of Port 4 for each mode is explained below.

Single Chip Mode (Mode 7): Parallel Inputs/Outputs as programmed by its associated data direction register.

Expanded Non Multiplexed Mode (Mode 5): In this mode, Port 4 becomes the lower address lines (A_0 to A_7) by writing "1"s on the data direction register. After reset, this port becomes inputs. In order to use these pins as addresses, they should be programmed as outputs.

When all of the eight bits are not required as addresses, the remaining lines can be used as I/O lines (Inputs only).

Expanded Non Multiplexed Mode (Mode 1): In this mode, Port 4 becomes output for upper order address lines (A_8 to A_{15}) regardless of the value of the direction register.

Expanded Multiplexed Mode (Mode 6): In this mode, Port 4 becomes the upper address lines (A_8 to A_{15}). After reset, this port becomes inputs. In order to use these pins as addresses, they should be programmed as outputs. When all of the eight bits are not required, the remaining lines can be used as I/O lines (input only).

Expanded Multiplexed Mode (Mode 0, 2, 4): In this mode, Port 4 becomes output for upper order address lines (A_8 to A_{15}) regardless of the value of data direction register.

The relation between each mode and I/O Port 1 to 4 is summarized in Table 3.

■ MODE SELECTION

The operation mode after the reset must be determined by the user wiring the P_{20}, P_{21} and P_{22} pins externally. These three pins are lower order bits; I/O 0, I/O 1, I/O 2 of Port 2. They are latched into the control bits PC0, PC1, PC2 of I/O Port 2 register when reset goes "High". I/O Port 2 Register is shown below.

Port 2 DATA REGISTER

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

An example of external hardware used for Mode Selection is shown in Fig. 13. The HD14053B is used to separate the peripheral device from the MCU during reset. It is necessary if the data may conflict between peripheral device and Mode generation circuit.

No mode can be changed through software because the bits 5, 6, and 7 of Port 2 Data Register are read-only. The mode selection of the HD6301V1 is shown in Table 4.

The HD6301V1 operates in three basic modes: (1) Single Chip Mode; (2) Expanded Multiplexed Mode (compatible with the HMCS6800 peripheral family), (3) Expanded Non Multiplexed Mode (compatible with HMCS6800 peripheral family).

- **Single Chip Mode (Mode 7)**

In the Single Chip Mode, all ports will become I/O. This is shown in Figure 15. In this mode, SC_1, SC_2 pins are configured for control lines of Port 3 and can be used as input strobe (IS3) and output strobe ($\overline{OS3}$) for data handshaking.

- **Expanded Multiplexed Mode (Mode 0, 2, 4, 6)**

In this mode, Port 4 is configured for I/O (inputs only) or address lines. The data bus and the lower order address bus are multiplexed in Port 3 and can be separated by the Address Strobe.

Port 2 is configured for 5 parallel I/O or Serial I/O, or Timer, or any combination thereof. Port 1 is configured for 8 parallel I/O. In this mode, HD6301V1 is expandable up to 65k words (See Fig. 16).

- **Expanded Non Multiplexed Mode (Mode 1, 5)**

In this mode, the HD6301V1 can directly address HMCS6800 peripherals with no address latch. In mode 5, Port 3 becomes a data bus. Port 4 becomes A_0 to A_7 address bus or partial address bus and I/O (inputs only). Port 2 is configured for a parallel I/O, Serial I/O, Timer or any combination thereof.

Port 1 is configured as a parallel I/O only.

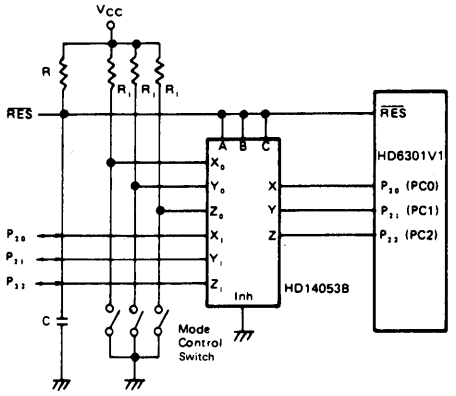
In this mode, HD6301V1 is expandable to 256 locations.

In mode 1, Port 3 becomes a data bus and Port 1 becomes A_0 to A_7 address bus, and Port 4 becomes A_8 to A_{15} address bus.

In this mode, the HD6301V1 is expandable to 65k bytes with no address latch. (See Fig. 17).

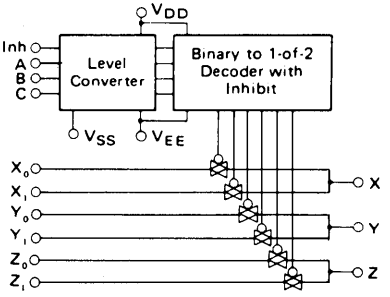
- **Lower Order Address Bus Latch**

Because the data bus is multiplexed with the lower order address bus in Port 3 in the expanded multiplexed mode, address bits must be latched. It requires the 74LS373 Transparent octal D-type to latch the LSB. Latch connection of the HD6301V1 is shown in Figure 18.



Note 1) Figure of Mode 7
2) RC≈Reset Constant
3) R_1=10kΩ

Figure 13 Recommended Circuit for Mode Selection



Truth Table

Control Input				On Switch
Inhibit	Select			
	C	B	A	HD14053B
0	0	0	0	Z_0 Y_0 X_0
0	0	0	1	Z_0 Y_0 X_1
0	0	1	0	Z_0 Y_1 X_0
0	0	1	1	Z_0 Y_1 X_1
0	1	0	0	Z_1 Y_0 X_0
0	1	0	1	Z_1 Y_0 X_1
0	1	1	0	Z_1 Y_1 X_0
0	1	1	1	Z_1 Y_1 X_1
1	X	X	X	

Figure 14 HD14053B Multiplexers/De-Multiplexers

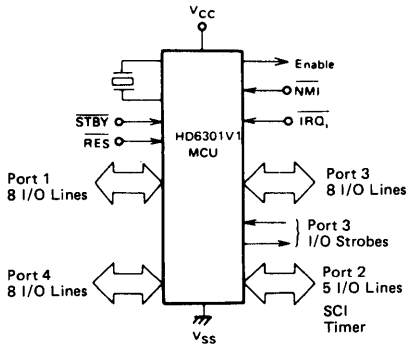


Figure 15 HD6301V1 MCU Single-Chip Mode

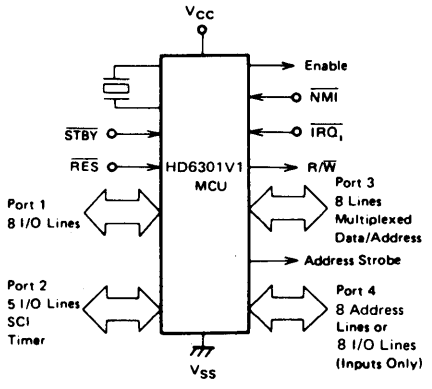


Figure 16 HD6301V1 MCU Expanded Multiplexed Mode

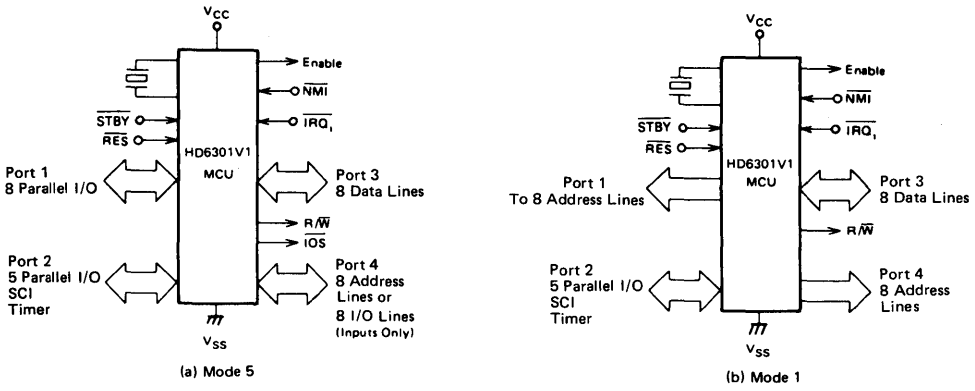
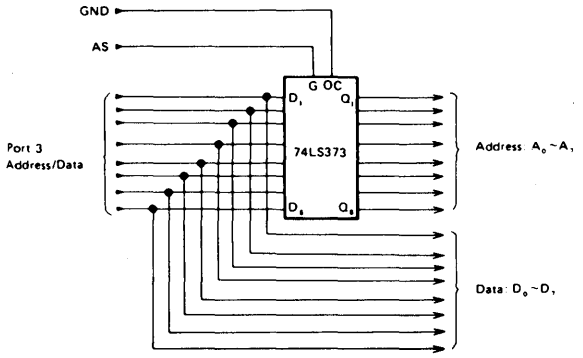


Figure 17 HD6301V1 MCU Expanded Non Multiplexed Mode



Function Table

Output Control (OC)	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 18 Latch Connection

- **Summary of Mode and MCU Signal**

This section gives a description of the MCU signals for the various modes. SC_1 and SC_2 are signals which vary with the mode.

Table 3 Feature of each mode and lines

MODE		PORT 1 Eight Lines	PORT 2 Five Lines	PORT 3 Eight Lines	PORT 4 Eight Lines	SC_1	SC_2
SINGLE CHIP (Mode 7)		I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)
EXPANDED MUX (Mode 0, 2, 4, 6)		I/O	I/O	ADDRESS BUS (A_0~A_7) DATA BUS (D_0~D_7)	ADDRESS BUS* (A_8~A_{15})	AS(O)	R/$\overline{W}$(O)
EXPANDED	(Mode 5)	I/O	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS* (A_0~A_7)	$\overline{IOS}$(O)	R/$\overline{W}$(O)
NON-MUX	(Mode 1)	ADDRESS BUS (A_0~A_7)	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS (A_8~A_{15})	Not Used	R/$\overline{W}$(O)

*These lines can be substituted for I/O (Input Only) (except Mode 0, 2, 4)

I = Input, O = Output, R/$\overline{W}$ = Read/Write, $\overline{IS3}$ = Input Strobe, $\overline{OS3}$ = Output Strobe, $\overline{IOS}$ = I/O Select, SC = Strobe Control, AS = Address Strobe

Table 4 Mode Selection Summary

Mode	P_{22} ($PC2$)	P_{21} ($PC1$)	P_{20} ($PC0$)	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(4)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(4)	Non-Multiplexed/Partial Decode
4	H	L	L	E(2)	I(1)	E	MUX	Multiplexed/RAM
3	L	H	H	—	—	—	—	Not Used
2	L	H	L	E(2)	I(1)	E	MUX	Multiplexed/RAM
1	L	L	H	E(2)	I	E	NMUX	Non-Multiplexed
0	L	L	L	I	I	I(3)	MUX	Multiplexed Test

LEGEND :
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

(NOTES)
1) Internal RAM is addressed at $0080.
2) Internal ROM is disabled.
3) Reset vector is external for 3 or 4 cycles after $\overline{RES}$ goes "high".
4) Idle lines of Port 4 address outputs can be assigned to Input Port.

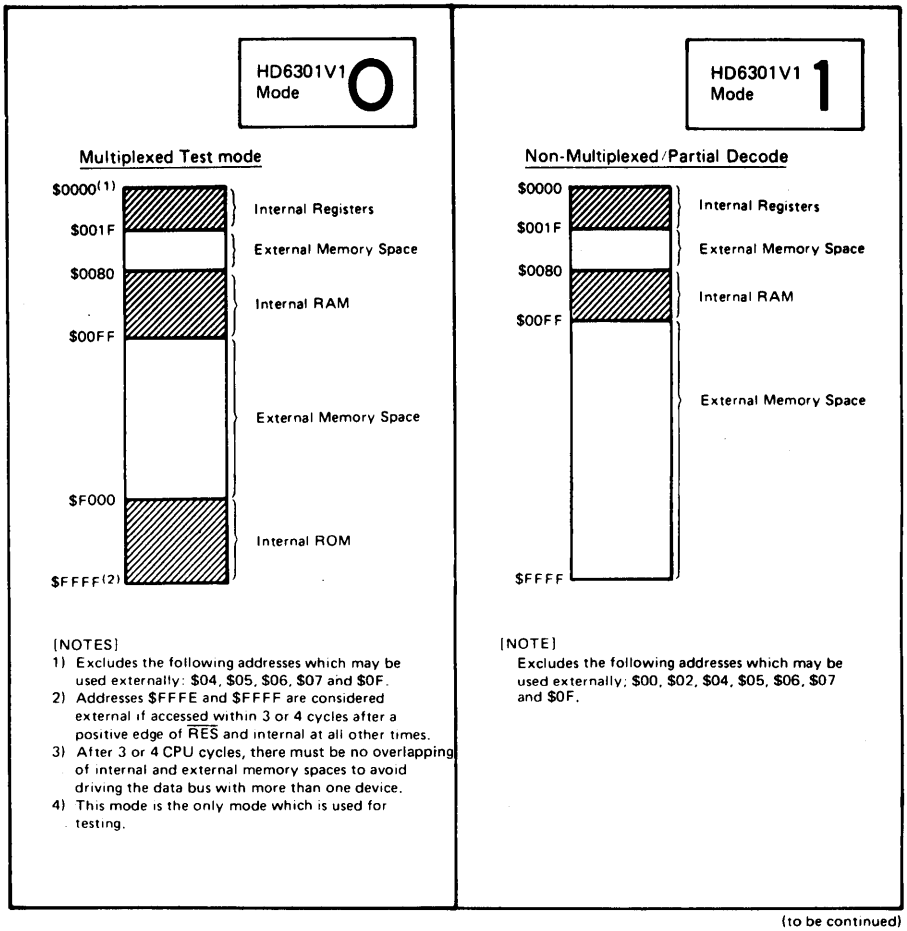
■ Memory Map

The MCU can provide up to 65k byte address space depending on the operating mode. Fig. 19 shows a memory map for each operating mode. The first 32 locations of each map are for the MCU's internal register only, as shown in Table 5.

Table 5 Internal Register Area

Register	Address
Port 1 Data Direction Register****	00*
Port 2 Data Direction Register****	01
Port 1 Data Register	02*
Port 2 Data Register	03
Port 3 Data Direction Register****	04**
Port 4 Data Direction Register****	05***
Port 3 Data Register	06**
Port 4 Data Register	07***
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F**
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External address in Mode 1
** External address in Modes 0, 1, 2, 4, 6; cannot be accessed in Mode 5
*** External address in Modes 0, 1, 2, 4
**** 1 = Output, 0 = Input

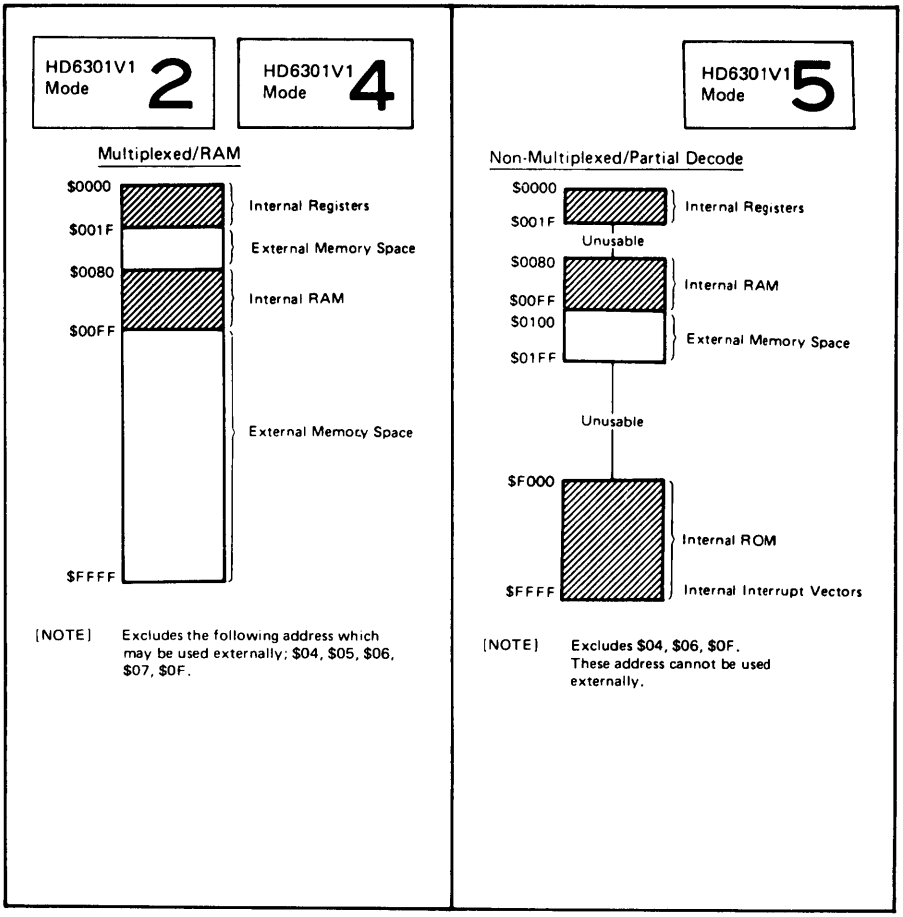


[NOTES]
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) Addresses $FFFE and $FFFF are considered external if accessed within 3 or 4 cycles after a positive edge of $\overline{RES}$ and internal at all other times.
3) After 3 or 4 CPU cycles, there must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which is used for testing.

[NOTE]
Excludes the following addresses which may be used externally: $00, $02, $04, $05, $06, $07 and $0F.

(to be continued)

Figure 19 HD6301V1 Memory Maps



(to be continued)

Figure 19 HD6301V1 Memory Maps

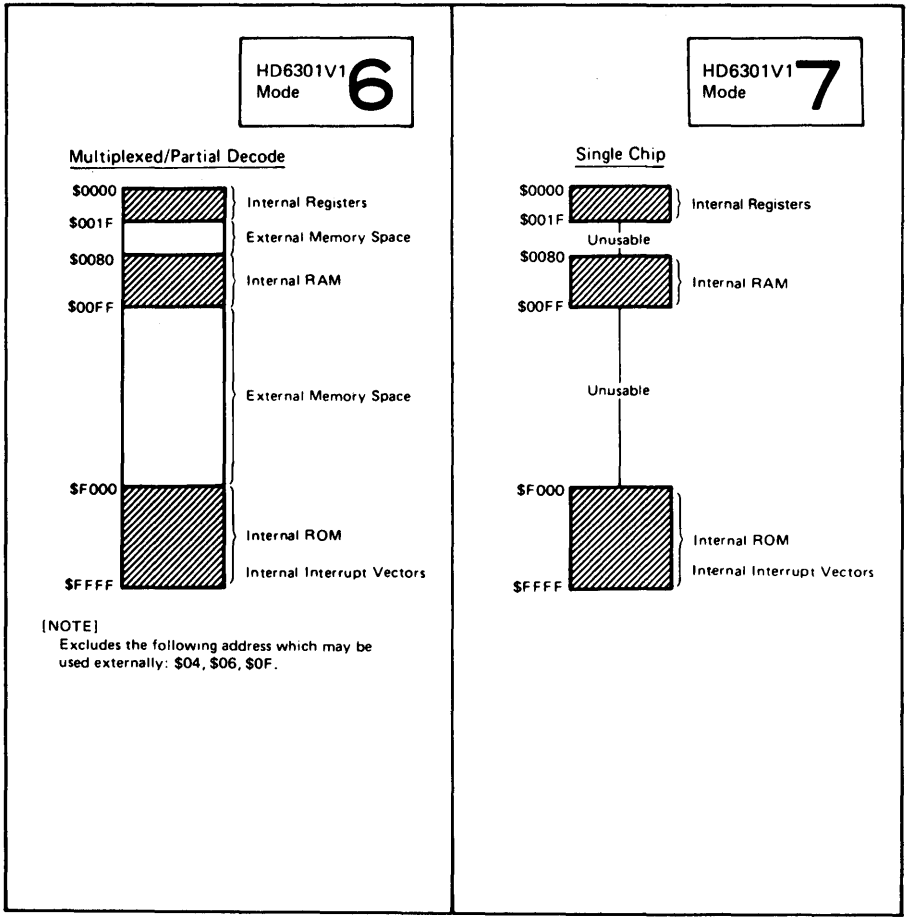


Figure 19 HD6301V1 Memory Maps

■ PROGRAMMABLE TIMER

The HD6301V1 contains 16-bit programmable timer which may be used to make measurement of input waveform. In addition to that it can generate an output waveform by itself. For both input and output waveform, the pulse width may vary from a few microseconds to several seconds.

The timer hardware consists of

- an 8-bit control and status register
- a 16-bit free running counter
- a 16-bit output compare register, and
- a 16-bit input capture register

A block diagram of the timer is shown in Figure 20.

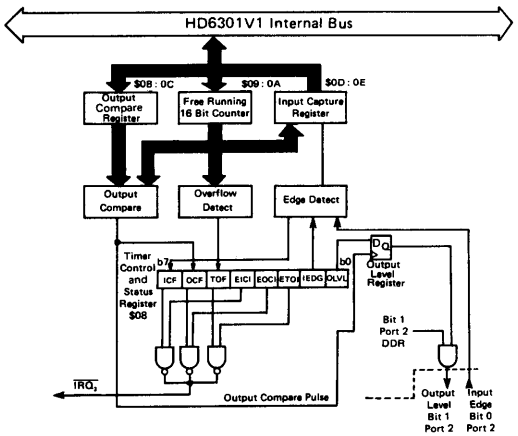


Figure 20 Programmable Timer Block Diagram

• Free Running Counter ($0009: $000A)

The key element in the programmable timer is a 16-bit free running counter, that is driven by an E (Enable) clock to increment its values. The counter value will be read out by the CPU software at any time with no effects on the counter. Reset will clear the counter.

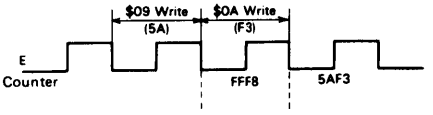
When the upper byte of this counter is read, the lower byte is stored in temporary latch. The data is fetched from this latch by the subsequent read of the lower byte. Thus consistent double byte data can be read from the counter.

When the CPU writes arbitrary data to the upper byte ($09), the value of $FFF8 is being pre-set to the counter ($09, $0A) regardless of the write data value. Then the CPU writes arbitray data to the lower byte ($0A), the data is set to the "Low" byte of the counter, at the same time, the data preceedingly written in the upper byte ($09) is set to "High" byte of the counter.

When the data is written to this counter, a double byte store instruction (ex. STD) must be used. If only the upper byte of counter is written, the counter is set to $FFF8.

The counter value written to the counter using the double byte store instruction is shown in Figure 21.

To write to the counter may disturb serial operations, so it should be inhibited during using the SCI in internal clock mode.



(5AF3 written to the counter)

Figure 21 Counter Write Timing

• Output Compare Register ($000B:$000C)

This is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly being compared with current value of the free running counter.

When the contents match with the value of the free running counter, a flag (OCF) in the timer control/status register (TCSR) is set and the current value of an output level Bit (OLVL) in the TCSR is transferred to Port 2 bit 1. When bit 1 of the Port 2 data direction register is "1" (output), the OLVL value will appear on the bit 1 of Port 2. Then, the value of Output Compare Register and Output level bit may be changed for the next compare.

The output compare register is set to $FFFF during reset.

The compare function is inhibited at the cycle of writing to the high byte of the output compare register and at the cycle just after that to ensure valid compare. It is also inhibited in same manner at writing to the free running counter.

In order to write a data to Output Compare Register, a double byte store instruction (ex. STD) must be used.

• Input Capture Register ($000D:$000E)

The input capture register is a 16-bit read-only register used to hold the current value of free running counter when the proper transition of an external input signal occurs.

The input transition change required to trigger the counter transfer is controlled by the input edge bit (IEDG).

To allow the external input signal to go in the edge detect unit, the bit of the Data Direction Register corresponding to bit 0 of Port 2 must have been cleared (to zero).

To insure input capture in all cases, the width of an input pulse requires at least 2 Enable cycles.

• Timer Control/Status Register (TCSR) ($0008)

This is an 8-bit register. All 8 bits are readable and the lower 5-bit may be written. The upper 3 bits are read-only, indicating the timer status information as is shown below.

(1) A proper transition has been detected on the input pin (ICF).
(2) A match has been found between the value in the free running counter and the output compare register (OCF).
(3) When counting up to $0000 (TOF).

Each flag has an individual enable bit in TCSR which determines whether or not an interrupt request may occur ($\overline{IRQ_2}$). If the I-bit in Condition Code Register has been cleared, a priority vectored address occurs corresponding to each flag. A description of each bit is as follows.

Timer Control / Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

Bit 0 OLVL (Output Level); When a match is found in the value between the counter and the output com-

pare register, this bit is transferred to the Port 2 bit 1. If the DDR corresponding to Port 2 bit 1 is set "1", the value will appear on the output pin of Port 2 bit 1.

Bit 1 IEDG (Input Edge): This bit control which transition of an input of Port 2 bit 0 will trigger the data transfer from the counter to the input capture register. The DDR corresponding to Port 2 bit 0 must be cleared in advance of using this function.
When IEDG = 0, trigger takes place on a negative edge ("High" to "Low" transition). When IEDG = 1, trigger takes place on a positive edge ("Low" to "High" transition).

Bit 2 ETOI (Enable Timer Overflow Interrupt); When set, this bit enables TOF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 3 EOCI (Enable Output Compare Interrupt); When set, this bit enables OCF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 4 EICI (Enable Input Capture Interrupt); When set, this bit enables ICF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 5 TOF (Timer Over Flow Flag); This read-only bit is set at the transition of $FFFF to $0000 of the counter. It is cleared by CPU read of TCSR (with TOF set) followed by an CPU read of the counter ($0009).

Bit 6 OCF (Output Compare Flag); This read-only bit is set when a match is found in the value between the output compare register and the counter. It is cleared by a read of TCSR (with OCF set) followed by an CPU write to the output compare register ($000B or $000C).

Bit 7 ICF (Input Capture Flag); The read-only bit is set by a proper transition on the input, and is cleared by a read of TCSR (with ICF set) followed by an CPU read of Input Capture Register ($000D).

Reset will clear each bit of Timer Control and Status Register.

■ SERIAL COMMUNICATION INTERFACE

The HD6301V1 contains a full-duplex asynchronous Serial Communication Interface (SCI). SCI may select the several kinds of the data rate. It consists of a transmitter and a receiver which operate independently but with the same data format and the same data rate. Both the transmitter and receiver communicate with the CPU via the data bus and with the outside world through Port 2 bit 2, 3 and 4. Description of hardware, software and register is as follows.

• Wake-Up Feature

In typical multiprocessor applications the software protocol will usually have the designated address at the initial byte of the message. The purpose of Wake-Up feature is to have the non-selected MCU neglect the remainder of the message. Thus the non-selected MCU can inhibit the all further interrupt process until the next message begins.

Wake-Up feature is re-enabled by a ten consecutive "1"s which indicates an idle transmit line. Therefore software protocol must put an idle period between the messages and must prevent it within the message.

With this hardware feature, the non-selected MCU is re-enabled or ("waked-up") by the next message.

• Programmable Options

The HD6301V1 has the following programmable features.
- data format; standard mark/space (NRZ)
- clock source; external or internal
- baud rate; one of 4 rates per given E clock frequency or 1/8 of external clock
- wake-up feature; enabled or disabled
- interrupt requests; enabled or masked individually for transmitter and receiver
- clock output; internal clock enabled or disabled to Port 2 bit 2
- Port 2 (bits 3, 4); dedicated or not dedicated to serial I/O individually

• Serial Communication Hardware

The serial communications hardware is controlled by 4 registers as shown in Figure 22. The registers include:
- an 8-bit control/status register
- a 4-bit rate/mode control register (write-only)
- an 8-bit read-only receive data register
- an 8-bit write-only transmit data register

Besides these 4 registers, Serial I/O utilizes Port 2 bit 3 (input) and bit 4 (output). Port 2 bit 2 can be used when an option is selected for the internal-clock-out or the external-clock-in.

• Transmit/Receive Control Status Register (TRCSR)

TRCS Register consists of 8 bits which all may be read while only bits 0 to 4 may be written. The register is initialized to $20 on $\overline{RES}$. The bits of the TRCS register are explained below.

Transmit / Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	ADDR $0011

Bit 0 WU (Wake Up); Set by software and cleared by hardware on receipt of ten consecutive "1"s. While this bit is "1", RDRF and ORFE flags are not set even if data are received or errors are detected. Therefore received data are ignored. It should be noted that RE flag must have already been set in advance of WU flag's set.

Bit 1 TE (Transmit Enable); This bit enables transmitter. When this bit is set, bit 4 of Port 2 DDR is also forced to be set. It remains set even if TE is cleared. Preamble of ten consecutive "1"s is transmitted just after this bit is set, and then transmitter becomes ready to send data.
If this bit is cleared, the transmitter is disabled and serial I/O affects nothing on Port 2 bit 4.

Bit 2 TIE (Transmit Interrupt Enable); When this bit is set, TDRE (bit 5) causes an $\overline{IRQ_2}$ interrupt. When cleared TDRE interrupt is masked.

Bit 3 RE (Receive Enable); When set, Port 2 bit 3 can be used as an input of receive regardless of DDR value for this bit. When cleared, the receiver is disabled.

Bit 4 RIE (Receive Interrupt Enable); When this bit is set, RDRF (bit 7) or ORFE (bit 6) cause an $\overline{IRQ_2}$ interrupt. When cleared, this interrupt is masked.

Bit 5 TDRE (Transmit Data Register Empty); When the data is transferred from the Transmit Data Register to Output Shift Register, this bit is set by hardware. The bit is cleared by reading the status register followed by writing the next new data into the Transmit Data Register. TDRE is initialized to 1 by $\overline{RES}$.

Bit 6 ORFE (Over Run Framing Error); When overrun or framing error occurs (receive only), this bit is set by hardware. Over Run Error occurs if the attempt is made to transfer the new byte to the receive data register while the RDRF is "1". Framing Error occurs when the bit counter is not synchronized with the boundary of the byte in the receiving bit stream. When Framing Error is detected, RDRF is not set. Therefore Framing Error can be distinguished from Overrun Error. That is, if ORFE is "1" and RDRF is "1", Overrun Error is detected. Otherwise Framing Error occurs. The bit is cleared by reading the status register followed by reading the receive data register, or by $\overline{RES}$.

Bit 7 RDRF (Receive Data Register Full); This bit is set by hardware when the data is transferred from the receive shift register to the receive data register. It is cleared by reading the status register followed by reading the receive data register, or by $\overline{RES}$.

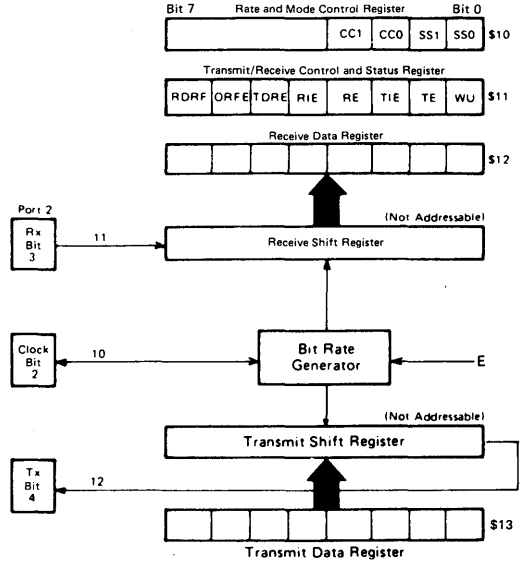
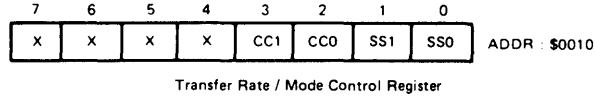


Figure 22 Serial I/O Register



Transfer Rate / Mode Control Register

Table 6 SCI Bit Times and Transfer Rates

	XTAL	2.4576 MHz	4.0 MHz	4.9152MHz
SS1 : SS0	E	614.4 kHz	1.0 MHz	1.2288MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800Baud
0 1	E ÷ 128	208μs/4,800 Baud	128 μs/7812.5 Baud	104.2μs/ 9,600Baud
1 0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud	833.3μs/ 1,200Baud
1 1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.333ms/ 300Baud

Table 7 SCI Format and Clock Source Control

CC1: CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used***	••	••
1 0	NRZ	Internal	Output*	••	••
1 1	NRZ	External	Input	••	••

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS.
Bit 4 is used for serial output if TE = "1" in TRCS.
*** This pin can be used as I/O port.

- **Transfer rate/Mode Control Register (RMCR)**

The register controls the following serial I/O functions:
- Bauds rate
- data format
- clock source
- Port 2 bit 2 feature

It is 4-bit write-only register, cleared by $\overline{RES}$. The 4 bits are considered as a pair of 2-bit fields. The lower 2 bits control the bit rate of internal clock while the upper 2 bits control the format and the clock select logic.

Bit 0 SS0, Bit 1 SS1 — Speed Select

These bits select the Baud rate for the internal clock. The rates selectable are function of E clock frequency of the CPU. Table 6 lists the available Baud Rates.

Bit2 CC0, Bit 3 CC1 — Clock Control/Format Select

They control the data format and the clock select logic. Table 7 defines the bit field.

- **Internally Generated Clock**

If the user wish to use externally an internal clock of the serial I/O, the following requirements should be noted.
- CC1, CC0 must be set to "10"
- The maximum clock rate must be E/16.
- The clock rate is equal to the bit rate.
- The values of RE and TE have no effect.

- **Externally Generated Clock**

If the user wish to supply an external clock to the Serial I/O, the following requirements should be noted.
- The CC1, CC0 must be set to "11" (See Table 7).
- The external clock must be set to 8 times of the desired baud rate.
- The maximum external clock frequency is E/2 clock.

- **Serial Operations**

The serial I/O hardware must be initialized by the software before operation. The sequence will be normally as follows.
- Writing the desired operation control bits of the Rate and Mode Control Register.
- Writing the desired operation control bits of the TRCS register.

If Port 2 bit 3, 4 are used for serial I/O, TE, RE bits may be kept set. When TE, RE bit are cleared during SCI operation, and subsequently set again, it should be noted that TE, RE must be kept "0" for at least one bit time of the current baud rate. If TE, RE are set again within one bit time, there may be the case where the initializing of internal function for transmitter and receiver does not take place correctly.

- **Transmit Operation**

Data transmission is enabled by the TE bit in the TRCS register. When set, the output of the transmit shift register is connected with Port 2 bit 4 which is unconditionally configured as an output.

After $\overline{RES}$, the user should initialize both the RMC register and the TRCS register for desired operation. Setting the TE bit causes a transmission of ten-bit preamble of "1"s. Following the preamble, internal synchronization is established and the transmitter is ready to operate. Then either of the following states exists.

(1) If the transmit data register is empty (TDRE = 1), the consecutive "1"s are transmitted indicating an idle states.
(2) If the data has been loaded into the Transmit Data Register (TDRE = 0), it is transferred to the output shift register and data transmission begins.

During the data transfer, the start bit ("0") is first transferred. Next the 8-bit data (beginning at bit 0) and finally the stop bit ("1"). When the contents of the Transmit Data Register is transferred to the output shift register, the hardware sets the TDRE flag bit: If the CPU fails to respond to the flag within the proper time, TDRE is kept set and then a continuous string of 1's is sent until the data is supplied to the data register.

- **Receive Operation**

The receive operation is enabled by the RE bit. The serial input is connected with Port 2 bit 3. The receiver operation is determined by the contents of the TRCS and RMC register. The received bit stream is synchronized by the first "0" (start bit). During 10-bit time, the data is strobed approximately at the center of each bit. If the tenth bit is not "1" (stop bit), the system assumes a framing error and the ORFE is set.

If the tenth bit is "1", the data is transferred to the receive data register, and the RDRF flag is set. If the tenth bit of the next data is received and still RDRF is preserved set, then ORFE is set indicating that an overrun error has occurred.

After the CPU read of the status register as a response to RDRF flag or ORFE flag, followed by the CPU read of the receive data register, RDRF or ORFE will be cleared.

■ RAM CONTROL REGISTER

The register assigned to the address $0014 gives a status information about standby RAM.

RAM Control Register

	7	6	5	4	3	2	1	0
$0014	STBY PWR	RAME	x	x	x	x	x	x

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.

Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 RAM Enable.

Using this control bit, the user can disable the RAM. RAM Enable bit is set on the positive edge of $\overline{RES}$ and RAM is enabled. The program can write "1" or "0". If RAME is cleared, the RAM address becomes external address and the CPU may read the data from the outside memory.

Bit 7 Standby Bit

This bit can be read or written by the user program. It is cleared when the V_{CC} voltage is removed. Normally this bit is set by the program before going into stand-by mode. When the CPU recovers from stand-by mode, this bit should be checked. If it is "1", the data of the RAM is retained during stand-by and it is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD6301V1 has an upward object code compatible with the HD6801 to utilize all instruction sets of the HMCS6800. The execution time of the key instruction is reduced to increase the system through-put. In addition, the bit operation instruction, the exchange instruction between the index and the accumulator, the sleep instruction are added. This section describes:

- CPU programming model (See Fig. 23)
- Addressing modes
- Accumulator and memory manipulation instructions (See Table 8)
- New instructions
- Index register and stack manipulation instructions (See Table 9)
- Jump and branch instructions (See Table 10)
- Condition code register manipulation instructions (See Table 11)
- Op-code map (See Table 12)
- Cycle-by-Cycle Operation (See Table 13)

• CPU Programming Model

The programming model for the HD6301V1 is shown in Figure 23. The double accumulator is physically the same as the accumulator A concatenated with the accumulator B, so that the contents of A and B is changed with executing operation of an accumulator D.

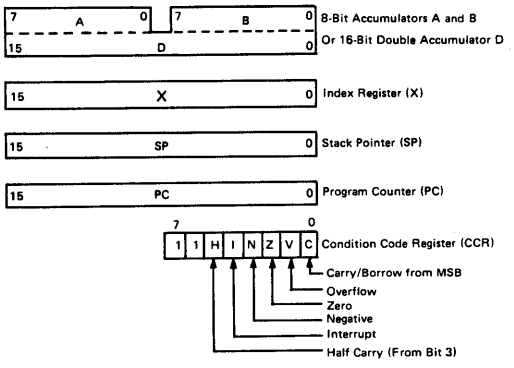


Figure 23 CPU Programming Model

• CPU Addressing Modes

The HD6301V1 has seven address modes which depend on both of the instruction type and the code. The address mode for every instruction is shown along with execution time given in terms of machine cycles (Table 8 to 12). When the clock frequency is 4 MHz, the machine cycles will be microseconds.

Accumulator (ACCX) Addressing

Only the accumulator (A or B) is addressed. Either accumulator A or B is specified by one-byte instructions.

Immediate Addressing

In this mode, the operand is stored in the second byte of the instruction except that the operand in LDS and LDX, etc are stored in the second and the third byte. These are two or three-byte instructions.

Direct Addressing

In this mode, the second byte of instruction indicates the address where the operand is stored. Direct addressing allows the user to directly address the lowest 256 Bytes in the machine locations zero through 255. Improved execution times are achieved by storing data in these locations. For system configuration, it is recommended that these locations should be RAM and be utilized preferably for user's data realm. These are two-byte instructions except the AIM, OIM, EIM and TIM which have three- byte.

Extended Addressing

In this mode, the second byte indicates the upper 8 bits addresses where the operand is stored, while the third byte indicates the lower 8 bits. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the Index Register. For each of AIM, OIM, EIM and TIM instructions, the contents of the third byte are added to the lower 8 bits in the Index Register. In addition, the resulting "carry" is added to the upper 8 bits in the Index Register. The result is used for addressing memory. Because the modified address is held in the Temporary Address Register, there is no change to the Index Register. These are two-byte instructions but AIM, OIM, EIM, TIM have three-byte.

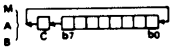
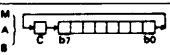
Implied Addressing

In this mode, the instruction itself gives the address; stack pointer, index register, etc. These are 1-byte instructions.

Relative Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the program counter. The resulting carry or borrow is added to the upper 8 bits. This helps the user to address the data within a range of −126 to +129 bytes of the current execution instruction. These are two-byte instructions.

Table 8 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

Note) Condition Code Register will be explained in Note of Table 11.

(to be continued)

Table 8 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑤	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑤	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑤	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	ACC A/ ACC B	•	•	↕	↕	⑤	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑤	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑤	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑤	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑤	↕
	LSRA													44	1	1	A	•	•	R	↕	⑤	↕
	LSRB													54	1	1	B	•	•	R	↕	⑤	↕
Double Shift Right Logical	LSRD													04	1	1	ACC A/ ACC B	•	•	R	↕	⑤	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

Note) Condition Code Register will be explained in Note of Table 11.

- **New Instructions**

In addition to the HD6801 Instruction Set, the HD6301V1 has the following new instructions:

AIM----(M) · (IMM) → (M)
Evaluates the AND of the immediate data and the memory, places the result in the memory.

OIM----(M) + (IMM) → (M)
Evaluates the OR of the immediate data and the memory, places the result in the memory.

EIM----(M) ⊕ (IMM) → (M)
Evaluates the EOR of the immediate data and the contents of memory, places the result in memory.

TIM----(M) · (IMM)
Evaluates the AND of the immediate data and the memory, changes the flag of associated condition code register

Each instruction has three bytes; the first is op-code, the second is immediate data, the third is address modifier.

XGDX--(ACCD) ↔ (IX)
Exchanges the contents of accumulator and the index register.

SLP----The CPU is brought to the sleep mode. For sleep mode, see the "sleep mode" section.

Table 9 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD↔IX	•	•	•	•	•	•

Note) Condition Code Register will be explained in Note of Table 11.

Table 10 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——		⑨		——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

Note) *WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 11.

Table 11 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	———			10		———
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

[NOTE 1] Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 9? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕C=1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1 of ACCB?

[NOTE 2] CLI instructions and interrupt.
If interrupt mask-bit is set (I="1") and interrupt is requested ($\overline{IRQ_1}$ = "0" or $\overline{IRQ_2}$ = "0"),
and then CLI instruction is executed, the CPU responds as follows.
1 the next instruction of CLI is one-machine cycle instruction.
Subsequent two instructions are executed before the interrupt is responded.
That is, the next and the next of the next instruction are executed.
2 the next instruction of CLI is two-machine cycle (or more) instruction.
Only the next instruction is executed and then the CPU jump to the interrupt routine.
Even if TAP instruction is used, instead of CLI, the same thing occurs.

Table 12 OP-Code Map

OP CODE						ACC A	ACC B	IND	EXT / DIR*	ACCA or SP IMM	DIR	IND	EXT	ACCB or X IMM	DIR	IND	EXT	
HI		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only for instructions of AIM, OIM, EIM, TIM

- **Instruction Execution Cycles**

In the HMCS6800 series, the execution cycle of each instruction is the number of cycles between the start of the current instruction fetch and just before the start of the subsequent instruction fetch.

The HD6301V1 uses a mechanism of the pipeline control for the instruction fetch and the subsequent instruction fetch is performed during the current instruction being executed.

Therefore, the method to count instruction cycles used in the HMCS6800 series cannot be applied to the instruction cycles such as MULT, PULL, DAA and XGDX in the HD6301V1.

Table 13 provides the information about the relationship among each data on the Address Bus, Data Bus, and R/$\overline{W}$ status in cycle-by-cycle basis during the execution of each instruction.

Table 13 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMMEDIATE					
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	2	1 2	Op Code Address+1 Op Code Address+2	1 1	Operand Data Next Op Code
ADDD CPX LDD LDS LDX SUBD	3	1 2 3	Op Code Address+1 Op Code Address+2 Op Code Address+3	1 1 1	Operand Data (MSB) Operand Data (LSB) Next Op Code
DIRECT					
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	3	1 2 3	Op Code Address+1 Address of Operand Op Code Address+2	1 1 1	Address of Operand (LSB) Operand Data Next Op Code
STA	3	1 2 3	Op Code Address+1 Destination Address Op Code Address+2	1 0 1	Destination Address Accumulator Data Next Op Code
ADDD CPX LDD LDS LDX SUBD	4	1 2 3 4	Op Code Address+1 Address of Operand Address of Operand+1 Op Code Address+2	1 1 1 1	Address of Operand (LSB) Operand Data (MSB) Operand Data (LSB) Next Op Code
STD STS STX	4	1 2 3 4	Op Code Address+1 Destination Address Destination Address+1 Op Code Address+2	1 0 0 1	Destination Address (LSB) Register Data (MSB) Register Data (LSB) Next Op Code
JSR	5	1 2 3 4 5	Op Code Address+1 FFFF Stack Pointer Stack Pointer−1 Jump Address	1 1 0 0 1	Jump Address (LSB) Restart Address (LSB) Return Address (LSB) Return Address (MSB) First Subroutine Op Code
TIM	4	1 2 3 4	Op Code Address+1 Op Code Address+2 Address of Operand Op Code Address+3	1 1 1 1	Immediate Data Address of Operand (LSB) Operand Data Next Op Code
AIM EIM OIM	6	1 2 3 4 5 6	Op Code Address+1 Op Code Address+2 Address of Operand FFFF Address of Operand Op Code Address+3	1 1 1 1 0 1	Immediate Data Address of Operand (LSB) Operand Data Restart Address (LSB) New Operand Data Next Op Code

– Continued –

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
INDEXED					
JMP	3	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Jump Address	1	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	Op Code Address+2	1	Next Op Code
STA	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Accumulator Data
		4	Op Code Address+2	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data (MSB)
		4	IX+Offset+1	1	Operand Data (LSB)
		5	Op Code Address+2	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Register Data (MSB)
		4	IX+Offset+1	0	Register Data (LSB)
		5	Op Code Address+2	1	Next Op Code
JSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	IX+Offset	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	IX+Offset	0	New Operand Data
		6	Op Code Address+2	1	Next Op Code
TIM	5	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	IX+Offset	0	00
		5	Op Code Address+2	1	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	FFFF	1	Restart Address (LSB)
		6	IX+Offset	0	New Operand Data
		7	Op Code Address+3	1	Next Op Code

– Continued –

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
EXTEND					
JMP	3	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	Jump Address	1	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Op Code Address+3	1	Next Op Code
STA	4	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Accumulator Data
		4	Op Code Address+3	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data (MSB)
		4	Address of Operand+1	1	Operand Data (LSB)
		5	Op Code Address+3	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Register Data (MSB)
		4	Destination Address+1	0	Register Data (LSB)
		5	Op Code Address+3	1	Next Op Code
JSR	6	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	Stack Pointer	0	Return Address (LSB)
		5	Stack Pointer−1	0	Return Address (MSB)
		6	Jump Address	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	Address of Operand	0	New Operand Data
		6	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Address of Operand	0	00
		5	Op Code Address+3	1	Next Op Code

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Accumulator Data
		4	Op Code Address+1	1	Next Op Code
PULX	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Index Register (LSB)
		4	Stack Pointer−1	0	Index Register (MSB)
		5	Op Code Address+1	1	Next Op Code
RTS	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Return Address (MSB)
		4	Stack Pointer+2	1	Return Address (LSB)
		5	Return Address	1	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	FFFF	1	Restart Address (LSB)
		5	FFFF	1	Restart Address (LSB)
		6	FFFF	1	Restart Address (LSB)
		7	FFFF	1	Restart Address (LSB)

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
WAI	9	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	Index Register (LSB)
		6	Stack Pointer−3	0	Index Register (MSB)
		7	Stack Pointer−4	0	Accumulator A
		8	Stack Pointer−5	0	Accumulator B
		9	Stack Pointer−6	0	Conditional Code Register
RTI	10	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer + 1	1	Conditional Code Register
		4	Stack Pointer + 2	1	Accumulator B
		5	Stack Pointer + 3	1	Accumulator A
		6	Stack Pointer + 4	1	Index Register (MSB)
		7	Stack Pointer + 5	1	Index Register (LSB)
		8	Stack Pointer + 6	1	Return Address (MSB)
		9	Stack Pointer + 7	1	Return Address (LSB)
		10	Return Address	1	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer − 1	0	Return Address (MSB)
		5	Stack Pointer − 2	0	Index Register (LSB)
		6	Stack Pointer − 3	0	Index Register (MSB)
		7	Stack Pointer − 4	0	Accumulator A
		8	Stack Pointer − 5	0	Accumulator B
		9	Stack Pointer − 6	0	Conditional Code Register
		10	Vector Address FFFA	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		↑	FFFF		High Impedance-Non MPX Mode
		Sleep			Address Bus -MPX Mode
		↓			↓
		3	FFFF		Restart Address (LSB)
		4	Op Code Address+1		Next Op Code

– Continued –

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
RELATIVE					
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	Branch Offset
		2	FFFF	1	Restart Address (LSB)
		3	Branch Address·····Test="1" Op Code Address+1···Test="0"	1	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Branch Address	1	First Op Code of Subroutine

■ LOW POWER CONSUMPTION MODE

The HD6301V1 has two low power consumption modes; sleep and standby mode.

• Sleep Mode

On execution of SLP instruction, the MCU is brought to the sleep mode. In the sleep mode, the CPU sleeps (the CPU clock becomes inactive), but the contents of the registers in the CPU are retained. In this mode, the peripherals of CPU will remain active. So the operations such as transmit and receive of the SCI data and counter may keep in operation. In this mode, the power consumption is reduced to about 1/6 the value of a normal operation.

The escape from this mode can be done by interrupt, $\overline{RES}$, $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode (This will be mentioned later). When interrupt is requested to the CPU and accepted, the sleep mode is released, then the CPU is brought in the operation mode and jumps to the interrupt routine. When the CPU has masked the interrupt, after recovering from the sleep mode, the next instruction of SLP starts to execute. However, in such a case that the timer interrupt is inhibited on the timer side, the sleep mode cannot be released due to the absence of the interrupt request to the CPU.

This sleep mode is available to reduce an average power consumption in the applications of the HD6301V1 which may not be always running.

• Standby Mode

Bringing $\overline{STBY}$ "Low", the CPU becomes reset and all clocks of the HD6301V1 become inactive. It goes into the standby mode. This mode remarkably reduces the power consumptions of the HD6301V1.

In the standby mode, if the HD6301V1 is continuously supplied with power, the contents of RAM is retained. The standby mode should escape by the reset start. The following is the typical application of this mode.

First, $\overline{NMI}$ routine stacks the MCU's internal information and the contents of SP in RAM, disables RAME bit of RAM control register, sets the Standby bit, and then goes into the standby mode. If the Standby bit keeps set on reset start, it means that the power has been kept during standby mode and the contents of RAM is normally guaranteed. The system recovery may be possible by returning SP and bringing into the condition before the standby mode has started. The timing relation for each line in this application is shown in Figure 24.

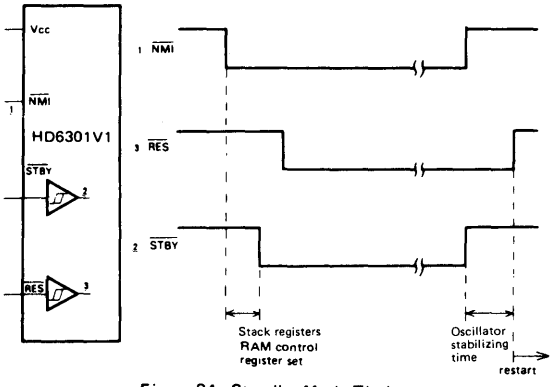


Figure 24 Standby Mode Timing

■ ERROR PROCESSING

When the HD6301V1 fetches an undefined instruction or fetches an instruction from unusable memory area, it generates the highest priority internal interrupt, that may protect from system upset due to noise or a program error.

• Op-Code Error

Fetching an undefined op-code, the HD6301V1 will stack the CPU register as in the case of a normal interrupt and vector to the TRAP ($FFEE, $FFEF), that has a second highest priority ($\overline{RES}$ is the highest).

• Address Error

When an instruction is fetched from other than a resident ROM, RAM, or an external memory area, the CPU starts the same interrupt as op-code error. In the case which the instruction is fetched from external memory area and that area is not usable, the address error cannot be detected.

The addresses which cause address error in particular mode are shown in Table 14.

This feature is applicable only to the instruction fetch, not to normal read/write of data accessing.

Table 14 Address Error

Mode	0	1	2, 4	5	6	7
Address	$0000 ↓ $001F	$0000 ↓ $001F	$0000 ↓ $001F	$0000 ↓ $007F $0200 ↓ $EFFF	$0000 ↓ $001F	$0000 ↓ $007F $0100 ↓ $EFFF

System Flow chart of HD6301V1 is shown in Fig. 25.

Transitions among the active mode, sleep mode, standby mode and reset are shown in Fig. 26.

Figures 27, 28, 29 and 30 shows a system configuration.

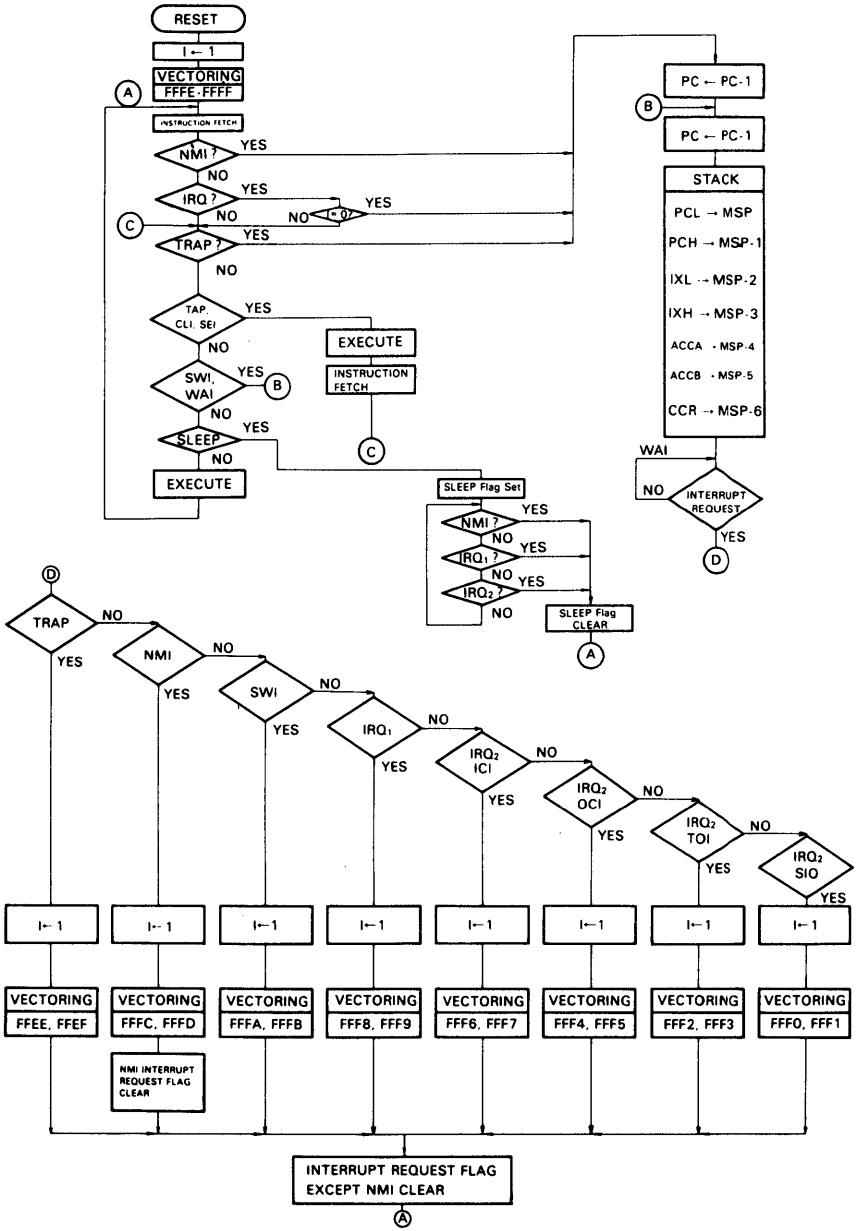


Figure 25 HD6301V1 System Flow Chart

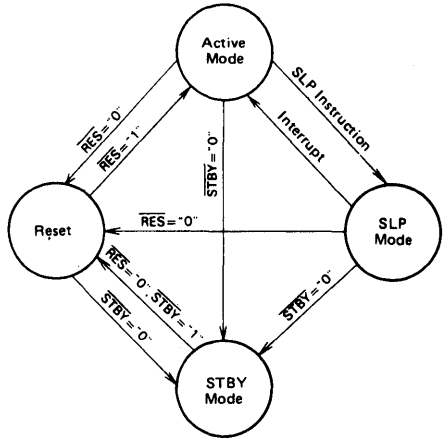


Figure 26 Transitions among Active Mode, Standby Mode, Sleep Mode, and Reset

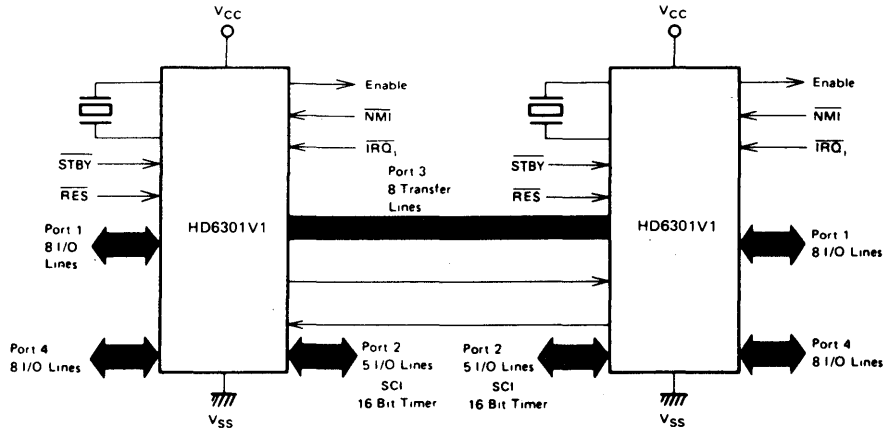


Figure 27 HD6301V1 MCU Single-Chip Dual Processor Configuration

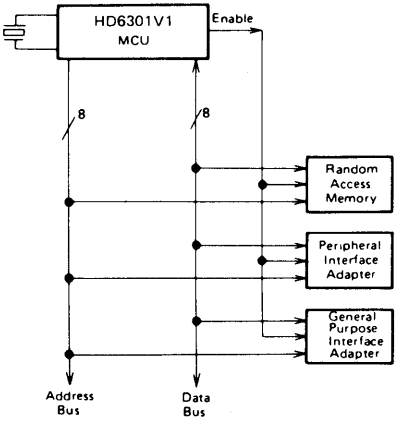


Figure 28 HD6301V1 MCU Expanded Non-Multiplexed Mode (Mode 5)

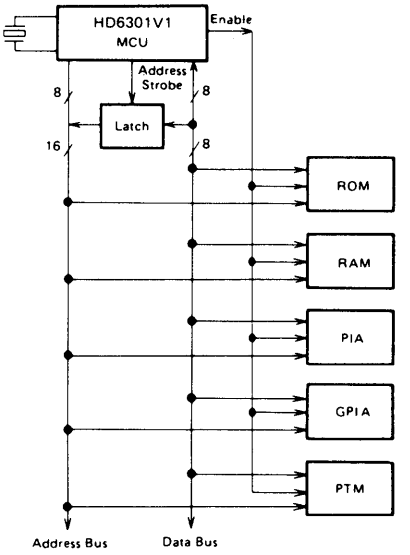


Figure 29 HD6301V1 MCU Expanded Multiplexed Mode

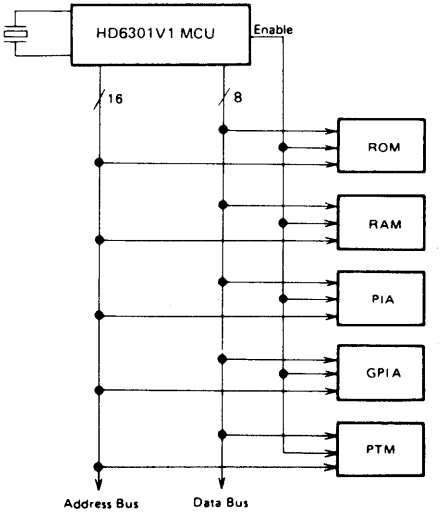
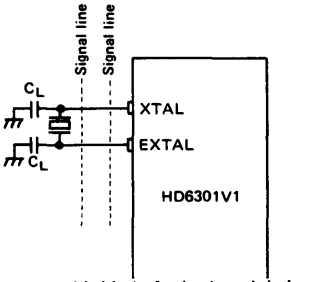


Figure 30 HD6301V1 MCU Expanded Non-Multiplexed Mode (Mode 1)

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 31, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD6301V1 as possible.



Do not use this kind of print board design.

Figure 31 Precaution to the boad design of oscillation circuit

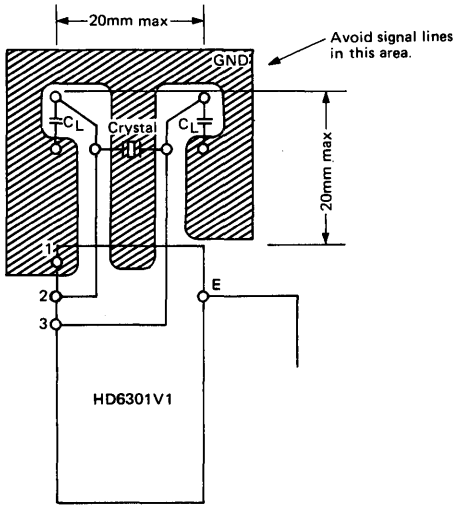


Figure 32 Example of Oscillation Circuits in Board Design

Table 15 Pin Condition in Sleep Mode

Pin \ Mode		0	1	2, 4	5	6	7
Port 1 P_{10}~P_{17}	Function	I/O Port	Lower Address Bus	I/O Port	←	←	←
	Condition	Keep the condition just before sleep	Output "1"	Keep the condition just before sleep	←	←	←
Port 2 P_{20}~P_{24}	Function	I/O Port	←	←	←	←	←
	Condition	Keep the condition just before sleep	←	←	←	←	←
Port 3 P_{30}~P_{37}	Function	$\overline{E}$: Lower Address Bus E: Data Bus	Data Bus	$\overline{E}$: Lower Address Bus E: Data Bus	Data Bus	$\overline{E}$: Lower Address Bus E: Data Bus	I/O Port
	Condition	$\overline{E}$: Output "1" E: High Impedance	High Impedance	$\overline{E}$: Output "1" E: High Impedance	High Impedance	$\overline{E}$: Output "1" E: High Impedance	Keep the condition just before sleep
Port 4 P_{40}~P_{47}	Function	Upper Address	←	←	Lower Address Bus or Input Port	Upper Address Bus or Input Port	I/O Port
	Condition	Output "1"	←	←	Address Bus: Output "1" Port: Keep the condition just before sleep	←	Keep the condition just before sleep
SC_2		Output "1" (Read Condition)	←	←	←	←	Output "1"
SC_1		Output Address Strobe	←	←	Output "1"	Output Address Strobe	Input Pin

Table 16 Pin Condition during RESET

Pin \ Mode	0	1	2, 4	5	6	7
Port 1 $P_{10} \sim P_{17}$	high impedance (input)	←	←	←	←	←
Port 2 $P_{20} \sim P_{24}$	high impedance (input)	←	←	←	←	←
Port 3 $P_{30} \sim P_{37}$	$\overline{E}$: "1" output E: "1" output[Note] (high impedance)	high impedance	$\overline{E}$: "1" output E: "1" output[Note] (high impedance)	high impedance	$\overline{E}$:"1" output E: "1" output[Note] (high impedance)	high impedance (input)
Port 4 $P_{40} \sim P_{47}$	high impedance (input)	←	←	←	←	←
SC_2	"1" output (READ)	←	←	←	←	"1" output
SC_1	$\overline{E}$: "1" output E: high impedance	←	←	"1" output	$\overline{E}$: "1" output E: high impedance	high impedance (input)

[Note] In mode 0, 2, 4, 6, port 3 is set to "1" output state during E = "1" and it causes the conflict with the output of external memory. Following 1 and 2 should be done to avoid the conflict;
(1) Construct the system that disables the external memory during reset.
(2) Add 4.7kΩ pull-down resistance to the SC_1 pin (AS) to make SC_1 pin "0" level during E = "1".
This operation makes port 3 high impedance state.

■ PIN CONDITIONS AT SLEEP AND STANDBY STATE

● Sleep State

The conditions of power supply pins, clock pins, input pins and E clock pin are the same as those of operation. Refer to Table 15 for the other pin conditions.

● Standby State

Only power supply pins and $\overline{STBY}$ pin are active. As for the clock pin EXTAL, its input is fixed internally so the MCU is not influenced by the pin conditions. XTAL is in "1" output. All the other pins are in high impedance.

■ DIFFERENCE BETWEEN HD6301V0 and HD6301V1

The HD6301V1 is an upgraded version of the HD6301V0. The difference between HD6301V0 and HD6301V1 is shown in Table 17.

Table 17 Difference between HD6301V0 and HD6301V1

Item	HD6301V0	HD6301V1
Operating Mode	Mode 2: Not defined	Mode 2: Expanded Multiplexed Mode (Equivalent to Mode 4)
Electrical Characteristics	The electrical characteristics of 2MHz version (B version) are not specified.	Some characteristics are improved. The 2MHz version is guaranteed.
Timer	Has problem in output compare function. (Can be avoided by software.)	The problem is solved.

■ APPLICATION NOTE FOR HIGH SPEED SYSTEM DESIGN USING THE HD6301V1

This note describes the solutions of the potential problem caused by noise generation in the system using the HD6301V1.

The CMOS ICs and LSIs featured by low power consumption and high noise immunity are generally considered to be enough with simply designed power source and the GND line.

But this does not apply to the applications configured of high speed system or of high speed parts. Such high speed system may have a chance to work incorrectly because of the noise by the transient current generated during switching. The noise generation owing to the over current (Sometimes it may be several hundreds mA for peak level.) during switching may cause data write error.

This noise problem may be observed only at the Expanded Mode (Mode 1, 2, 4, 5 and 6) of the HD6301V1. The Single Chip Mode (Mode 7) of the HD6301V1 has no such a problem.

Assuming the HD6301V1 is used as CPU in a system.

1. Noise Occurrence

If the HD6301V1 is connected to high speed RAM, a write error may occur. As shown in Fig. 33, the noise is generated in address bus during write cycle and data is written into an unexpected address from the HD6301V1. This phenomenon causes random failures in systems whose data bus load capacitance exceeds the specification value (90 pF max.) and/or the impedance of the GND line is high.

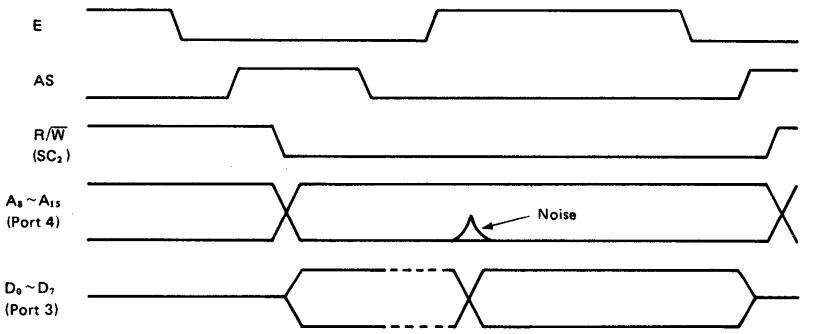


Fig. 33 Noise Occurrence in address bus during write cycle

If the data bus $D_0 \sim D_7$ changes from "FF" to "00", extremely large transient current flows through the GND line. Then the noise is generated on the LSI's V_{SS} pins proportioning to the transient current and to the impedance [Zg] of the GND line.

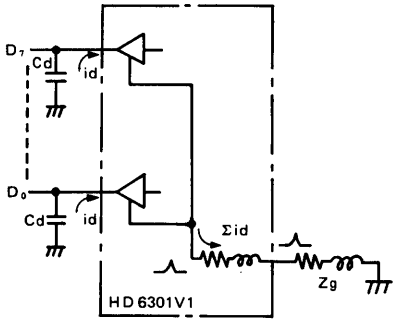
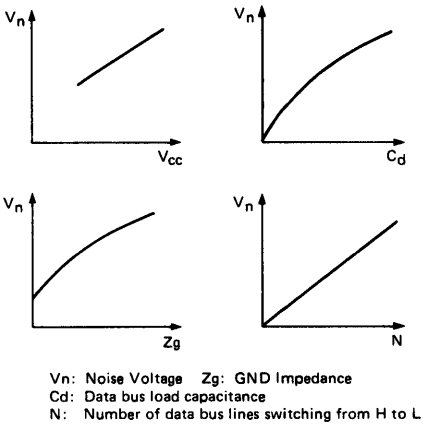


Fig. 34 Noise Source

This noise level, V_n, appears on all output pins on the LSI including the address bus.

Fig. 35 shows the dependency of the noise voltage on the each parameter.



Vn: Noise Voltage Zg: GND Impedance
Cd: Data bus load capacitance
N: Number of data bus lines switching from H to L

Fig. 35 Dependency of the noise voltage on each parameter

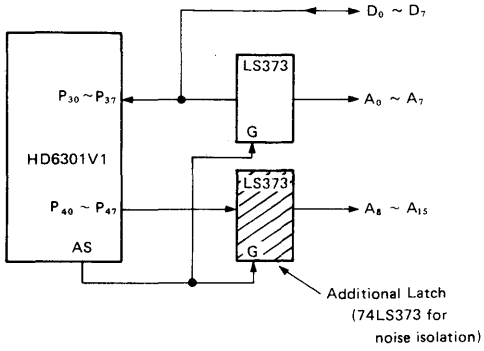
II. Noise Protection

To avoid the noise on the address bus during the system operation mentioned before, there are two solutions as follows:

The one method is to isolate the HD6301V1 from peripheral devices so that peripherals are not affected by the noise. The other is to reduce noise level to the extent of not affecting peripherals using analog method.

1. Noise Isolation

Addresses should be latched at the negative edge of the AS signal or at the positive edge of the E signal. The 74LS373 is often used in this case.



2. Noise Reduction

As the noise level depends on each parameter such Cd, V_{CC}, Zg, the noise level can be reduced to the allowable level by controlling those analog parameters.

(a) Transient Current Reduction

(1) Reduce the data bus load capacitance. If large load capacitance is expected, a bus buffer should be inserted.

(2) Lower the power supply voltage V_{CC} within specification.

(3) Increase a time constant at transient state by inserting a resistor (100 ~ 200Ω) to Data Buses in series to keep noise level down.
Table 18 shows the relationship between a series resistors and noise level or a resistor and DC/AC characteristics.

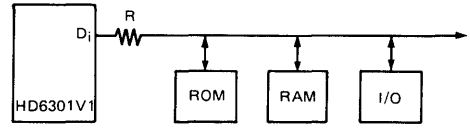


Table 18.

Item		Resistor	No	100Ω	200Ω
Noise Voltage Level			See Fig. 36		
DC Characteristics		I_{OL}	1.6 mA	1.6 mA	1.0 mA
AC Characteristics	f = 1 MHz		No change		
	f = 1.5 MHz	t_{ADL}	190 ns	190 ns	210 ns
		t_{ACCM}	395 ns	395 ns	375 ns
	f = 2 MHz	t_{ADL}	160 ns	180 ns	200 ns
		t_{ASL}	20 ns	20 ns	0 ns
		t_{ACCM}	270 ns	250 ns	230 ns

Fig. 36 shows an example of the dependency of the noise voltage on the load capacitance of the data bus.*

*Note: The value of series resistor should be carefully selected because it heavily depends on each parameter of actual application system.

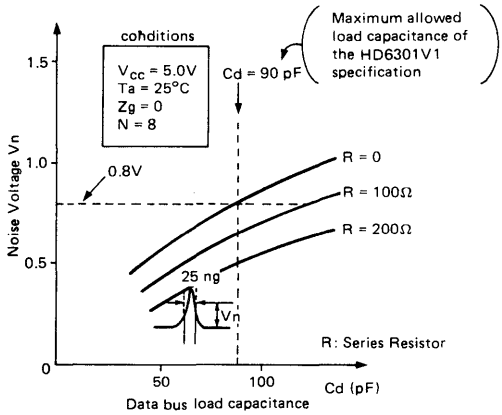


Fig. 36

Fig 37 shows the typical wave form of the noise.

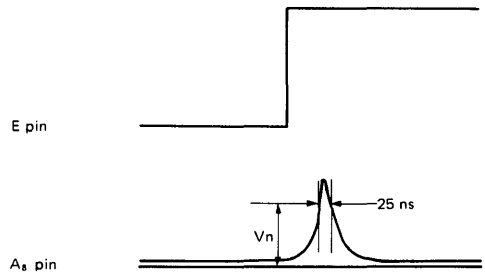


Fig. 37

(b) Reduction of GND line impedance
(1) Widen the GND line width on the PC board.
(2) Place theHD6301V1 close by power source.
(3) Insert a bypass capacitor between the V_{CC} line and the GND of the HD6301V1. A tantalum capacitor (about 0.1μF) is effective on the reduction.

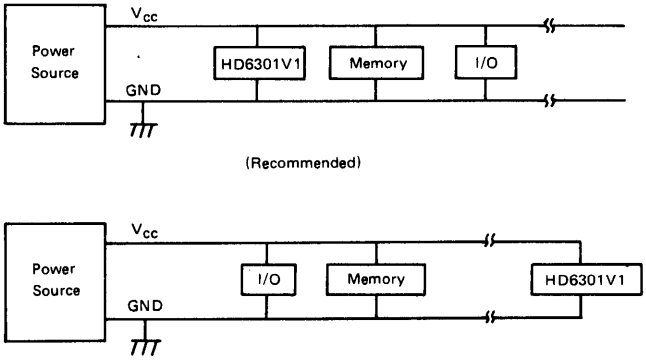


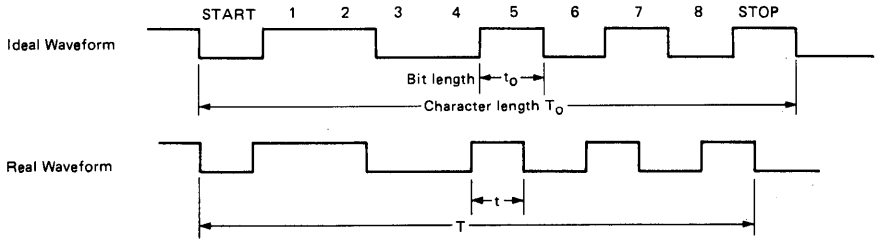
Fig. 38 Layout of the HD6301V1 on the PC board

■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD6301V1 is shown in Table 19.
Note: SCI = Serial Communication Interface

Table 19

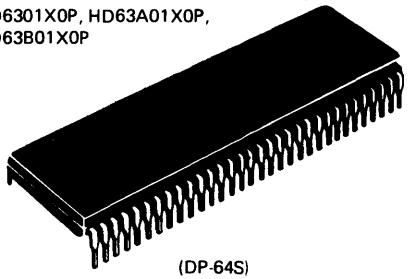
Bit distortion tolerance (t−to) /to	Character distortion tolerance (T−To) /To
±37.5%	+3.75% -2.5%



HD6301X0, HD63A01X0, HD63B01X0
CMOS MCU (Microcomputer Unit)

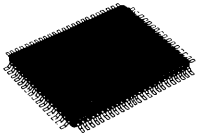
The HD6301X0 is a CMOS single-chip microcomputer unit (MCU) which includes a CPU compatible with the HD6301V1, 4k bytes of ROM, 192 bytes of RAM, 53 parallel I/O pins, a Serial Communication Interface (SCI) and two timers on chip.

HD6301X0P, HD63A01X0P, HD63B01X0P



(DP-64S)

HD6301X0F, HD63A01X0F, HD63B01X0F



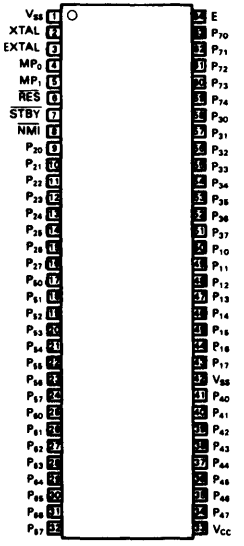
(FP-80)

- **FEATURES**
- Instruction Set Compatible with the HD6301V1
- Abundant On-chip Functions
  - 4k Bytes of ROM, 192 Bytes of RAM
  - 53 Parallel I/O Ports
  - 16-Bit Programmable Timer
  - 8-Bit Reloadable Timer
  - Serial Communication Interface
  - Memory Ready
  - Halt
  - Error-Detection (Address Trap, Op Code Trap)
- Interrupts . . . 3 External, 7 Internal
- Operation Mode
  - Mode 1 . . . Expanded (Internal ROM Inhibited)
  - Mode 2 . . . Expanded (Internal ROM Valid)
  - Mode 3 . . . Single-chip Mode
- Low Power Dissipation Mode
  - Sleep
  - Standby
- Wide Range of Operation
  - V_{CC} = 3 ~ 6V (f = 0.1 ~ 0.5MHz)
  - V_{CC} = 5V±10% (f = 0.1 ~ 1.0MHz; HD6301X0 / f = 0.1 ~ 1.5MHz; HD63A01X0 / f = 0.1 ~ 2.0MHz; HD63B01X0)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

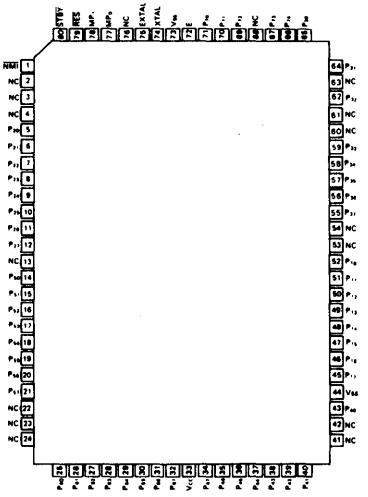
■ PIN ARRANGEMENT

- HD6301X0P, HD63A01X0P, HD63B01X0P



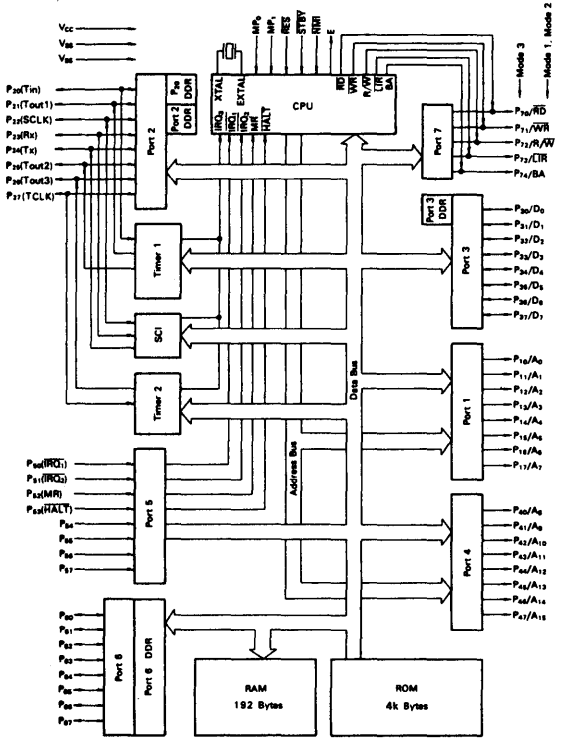
(Top View)

- HD6301X0F, HD63A01X0F, HD63B01X0F



(Top View)

■ BLOCK DIAGRAM



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out}: $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}x0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$, MP_0, MP_1, Port 5	$\|I_{in}\|$	V_{in} = 0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	Ports 1, 2, 3, 4, 6, 7	$\|I_{TSI}\|$	V_{in} = 0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
Darlington Drive Current	Ports 2, 6	$-I_{OH}$	Vout = 1.5V	1.0	–	10.0	mA
Input Capacitance	All Inputs	C_{in}	V_{in} = 0V, f = 1MHz, Ta = 25°C	–	–	12.5	pF
Standby Current	Non Operation	I_{STB}		–	3.0	15.0	μA
Current Dissipation*		I_{SLP}	Sleeping (f = 1MHz**)	–	1.5	3.0	mA
			Sleeping (f = 1.5MHz**)	–	2.3	4.5	mA
			Sleeping (f = 2MHz**)	–	3.0	6.0	mA
		I_{CC}	Operating (f = 1MHz**)	–	7.0	10.0	mA
			Operating (f = 1.5MHz**)	–	10.5	15.0	mA
			Operating (f = 2MHz**)	–	14.0	20.0	mA
RAM Standby Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V (All output terminals are at no load.)

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at x MHz operation are decided according to the following formula;

typ. value (f = x MHz) = typ. value (f = 1MHz) × x
max. value (f = x MHz) = max. value (f = 1MHz) × x
(both the sleeping and operating)

- **AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)**

BUS TIMING

Item		Symbol	Test Condition	HD6301X0			HD63A01X0			HD63B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time		t_{Er}		–	–	25	–	–	25	–	–	25	ns
Enable Fall Time		t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*		t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time	Write	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Set-up Time	Read	t_{DSR}		80	–	–	70	–	–	70	–	–	ns
Address, R/$\overline{W}$ Hold Time*		t_{AH}		80	–	–	50	–	–	35	–	–	ns
Data Hold Time	Write*	t_{HW}		80	–	–	50	–	–	40	–	–	ns
	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
$\overline{RD}$, $\overline{WR}$ Pulse Width*		PW_{RW}		450	–	–	300	–	–	220	–	–	ns
$\overline{RD}$, $\overline{WR}$ Delay Time		t_{RWD}		–	–	40	–	–	40	–	–	40	ns
$\overline{RD}$, $\overline{WR}$ Hold Time		t_{HRW}		–	–	30	–	–	30	–	–	25	ns
$\overline{LIR}$ Delay Time		t_{DLR}		–	–	200	–	–	160	–	–	120	ns
$\overline{LIR}$ Hold Time		t_{HLR}		10	–	–	10	–	–	10	–	–	ns
MR Set-up Time*		t_{SMR}	Fig. 2	400	–	–	280	–	–	230	–	–	ns
MR Hold Time*		t_{HMR}		–	–	90	–	–	40	–	–	0	ns
E Clock Pulse Width at MR		PW_{EMR}		–	–	9	–	–	9	–	–	9	μs
Processor Control Set-up Time		t_{PCS}	Fig. 3, 10, 11	200	–	–	200	–	–	200	–	–	ns
Processor Control Rise Time		t_{PCr}	Fig. 2, 3	–	–	100	–	–	100	–	–	100	ns
Processor Control Fall Time		t_{PCf}		–	–	100	–	–	100	–	–	100	ns
BA Delay Time		t_{BA}	Fig. 3	–	–	250	–	–	190	–	–	160	ns
Oscillator Stabilization Time		t_{RC}	Fig. 11	20	–	–	20	–	–	20	–	–	ms
Reset Pulse Width		PW_{RST}		3	–	–	3	–	–	3	–	–	t_{cyc}

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD6301X0			HD63A01X0			HD63B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set-up Time	Ports 2, 3, 5, 6	t_{PDSU}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Ports 2, 3, 5, 6	t_{PDH}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Delay Time (Enable Negative Transition to Peripheral Data Valid)	Ports 1, 2, 3, 4, 6, 7	t_{PWD}	Fig. 6	–	–	300	–	–	300	–	–	300	ns

TIMER, SCI TIMING

Item		Symbol	Test Condition	HD6301X0			HD63A01X0			HD63B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Output)		t_{TOD}	Fig. 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Async. Mode	t_{Scyc}	Fig. 8	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Clock Sync.		Fig. 4, 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Clock Sync. Mode)		t_{TXD}	Fig. 4	–	–	200	–	–	200	–	–	200	ns
SCI Receive Data Set-up Time (Clock Sync. Mode)		t_{SRX}		290	–	–	290	–	–	290	–	–	ns
SCI Receive Data Hold Time (Clock Sync. Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 8	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}		200	–	–	200	–	–	200	–	–	ns
Timer 1•2, SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1•2, SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

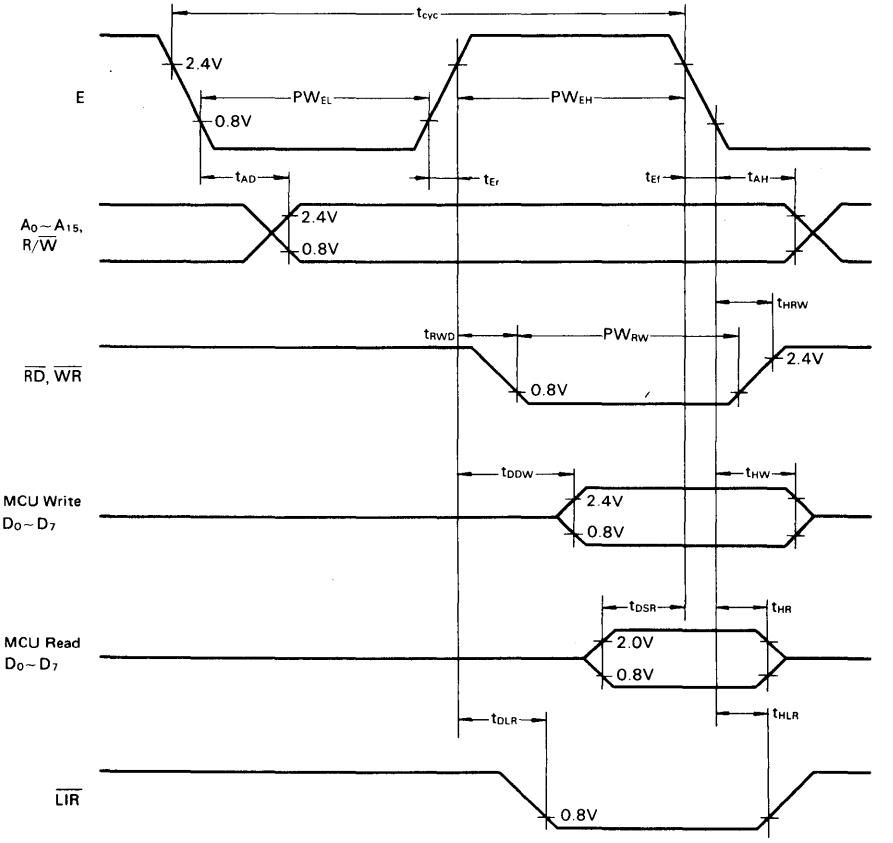


Figure 1 Mode 1, Mode 2 Bus Timing

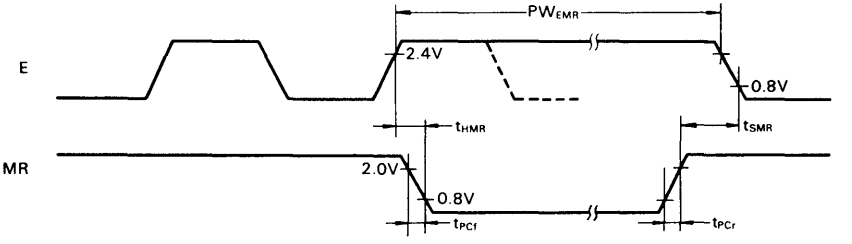


Figure 2 Memory Ready and E Clock Timing

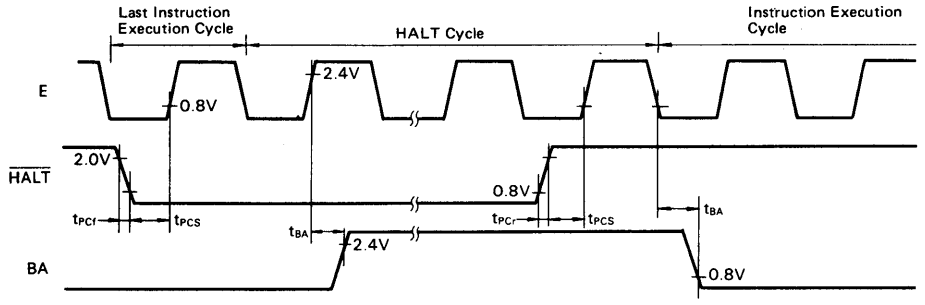


Figure 3 $\overline{\text{HALT}}$ and BA Timing

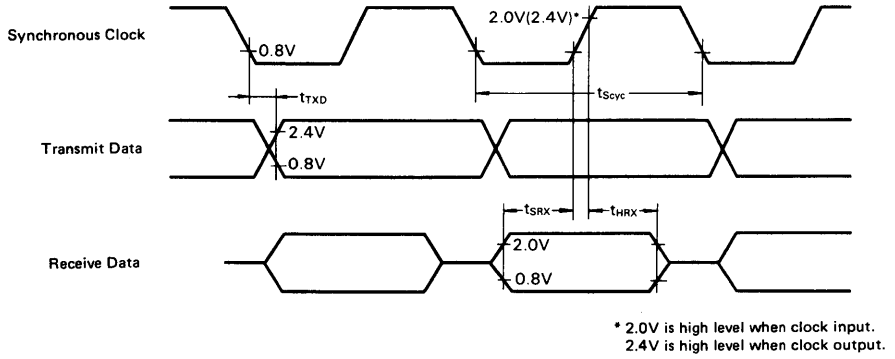


Figure 4 SCI Clocked Synchronous Timing

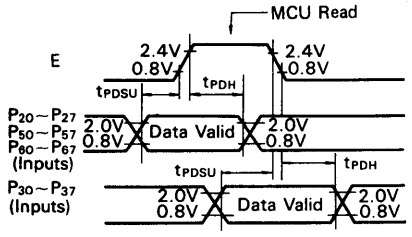


Figure 5 Port Data Set-up and Hold Times (MCU Read)

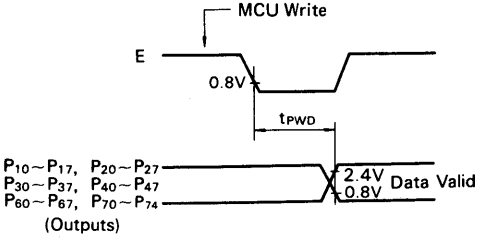
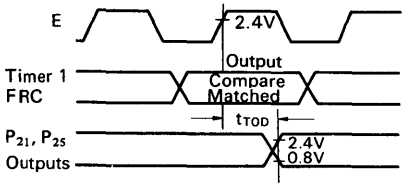
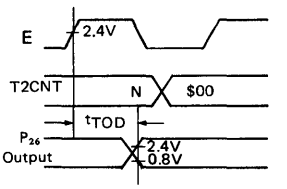


Figure 6 Port Data Delay Times (MCU Write)

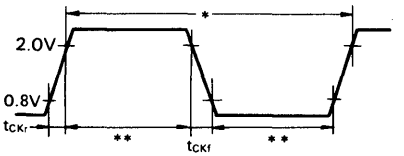


(a) Timer 1 Output Timing



(b) Timer 2 Output Timing

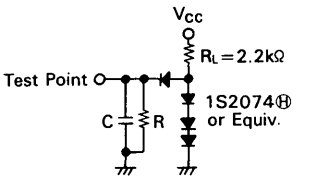
Figure 7 Timer Output Timing



* Timer 2; t_{TCYC}
SCI ; t_{SCYC}

** Timer 1; t_{PWT}
Timer 2; t_{PWTCK}
SCI ; t_{PWSCK}

Figure 8 Timer 1·2, SCI Input Clock Timing



C=90pF for Port 1, Port 3, Port 4, E
=30pF for Port 2, Port 6, Port 7
R=12kΩ for Port 1 ~ Port 4, Port 6, Port 7, E

Figure 9 Bus Timing Test Loads (TTL Load)

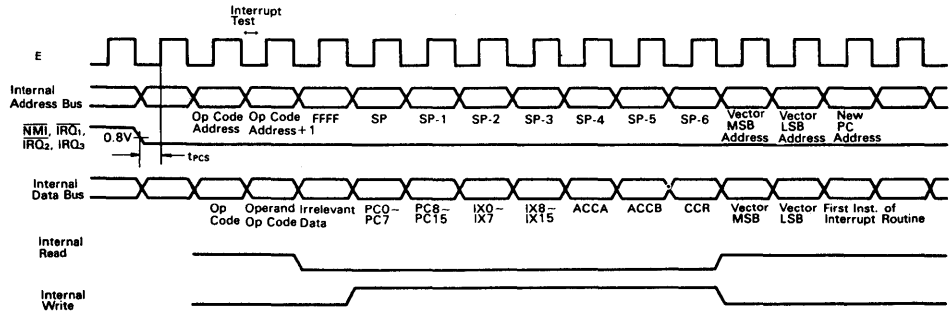


Figure 10 Interrupt Sequence

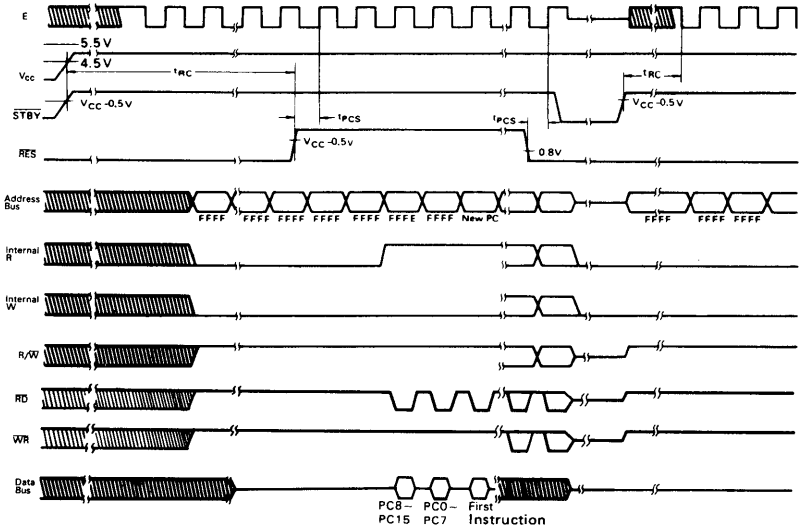


Figure 11 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

V_{CC} and V_{SS} provide power to the MCU with 5V±10% supply. In the case of low speed operation (fmax = 500kHz), the MCU can operate with three through six volts. Two V_{SS} pins should be tied to ground.

• XTAL, EXTAL

These two pins interface with an AT-cut parallel resonant crystal. Divide-by-four circuit is on chip, so if 4MHz crystal oscillator is used, the system clock is 1MHz for example.

EXTAL pin can be drived by the external clock of 45 to 55% duty, and one fourth frequency of the external clock is produced in the LSI. The external clock frequency should be less than four times of the maximum operable frequency. When using the external clock, XTAL pin should be open. Fig. 12 shows an example of the crystal interface. The crystal and C_{L1}, C_{L2} should be mounted as close as possible to XTAL and EXTAL pins. Any line must not cross the line between the crystal oscillator and XTAL, EXTAL.

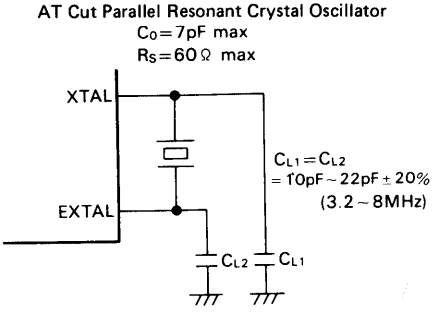


Figure 12 Crystal Interface

• STBY

This pin makes the MCU standby mode. In "Low" level, the oscillation stops and the internal clock is stabilized to make reset condition. To retain the contents of RAM at standby mode, "0" should be written into RAM enable bit (RAME). RAME is the bit 6 of the RAM/port 5 control register at $0014. RAM is disabled by this operation and its contents is sustained.

Refer to "LOW POWER DISSIPATION MODE" for the standby mode.

• Reset (RES)

This pin resets the MCU from power OFF state and provides a startup procedure. During power-on, RES pin must be held "Low" level for at least 20 ms.

The CPU registers (accumulator, index register, stack pointer, condition code register except for interrupt mask bit), RAM and the data register of a port are not initialized during reset, so their contents are unknown in this procedure.

To reset the MCU during operation, RES should be held "Low" for at least 3 system-clock cycles. At the 3rd cycle during "Low" level, all the address buses become "High". When RES remains "Low", the address buses keep "High". If RES becomes "High", the MCU starts the next operation.

(1) Latch the value of the mode program pins; MP_0 and MP_1.
(2) Initialize each internal register (Refer to Table 5).
(3) Set the interrupt mask bit. For the CPU to recognize the maskable interrupts $\overline{IRQ_1}$, $\overline{IRQ_2}$ and IRQ_3, this bit should be cleared in advance.
(4) Put the contents (= start address) of the last two addresses

($FFFE, $FFFF) into the program counter and start the program from this address. (Refer to Table 1).

* The MCU is unable to accept a reset input until the clock becomes normal oscillation after power on (max. 20ms). During this transient time, the MCU and I/O pins are undefined. Please be aware of this for system designing.

- **Enable (E)**

This pin provides a TTL-compatible system clock to external circuits. Its frequency is one fourth that of the crystal oscillator or external clock. This pin can drive one TTL load and 90pF capacitance.

- **Non-Maskable Interrupt ($\overline{NMI}$)**

When the falling edge of the input signal is detected at this pin, the CPU begins non-maskable interrupt sequence internally. As well as the IRQ mentioned below, the instruction being executed at $\overline{NMI}$ signal detection will proceed to its completion. The interrupt mask bit of the condition code register doesn't affect non-maskable interrupt at all.

When starting the acknowledge to the $\overline{NMI}$, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack. Upon completion of this sequence, a vector is fetched from $FFFC and $FFFD to transfer their contents into the program counter and branch to the non-maskable interrupt service routine.

(Note)

After reset start, the stack pointer should be initialized on an appropreate memory area and then the falling edge be input to $\overline{NMI}$ pin.

- **Interrupt Request ($\overline{IRQ_1}$, $\overline{IRQ_2}$)**

These are level-sensitive pins which request an internal interrupt sequence to the CPU. At interrupt request, the CPU will complete the current instruction before its request acknowledgement. Unless the interrupt mask in the condition code register is set, the CPU starts an interrupt sequence; if set, the interrupt request will be ignored. When the sequence starts, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack, then the CPU sets the interrupt mask bit and will not acknowledge the maskable request. During the last cycle, the CPU fetches vectors depicted in Table 1 and transfers their contents to the program counter and branches to the service routine.

The CPU uses the external interrupt pins, $\overline{IRQ_1}$ and $\overline{IRQ_2}$, also as port pins P_{50} and P_{51}, so it provides an enable bit to Bit 0 and 1 of the RAM port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for the details.

When one of the internal interrupts, ICI, OCI, TOI, CMI or SIO is generated, the CPU produces internal interrupt signal (IRQ_3). IRQ_3 functions just the same as $\overline{IRQ_1}$ or $\overline{IRQ_2}$ except for its vector address. Fig. 13 shows the block diagram of the interrupt circuit.

Table 1 Interrupt Vector Memory Map

Priority	Vector		Interrupt
	MSB	LSB	
Highest	FFFE	FFFF	$\overline{RES}$
↑	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	$\overline{IRQ_1}$
	FFF6	FFF7	ICI (Timer 1 Input Capture)
	FFF4	FFF5	OCI (Timer 1 Output Compare 1, 2)
	FFF2	FFF3	TOI (Timer 1 Overflow)
	FFEC	FFED	CMI (Timer 2 Counter Match)
	FFEA	FFEB	$\overline{IRQ_2}$
Lowest	FFF0	FFF1	SIO (RDRF+ORFE+TDRE)

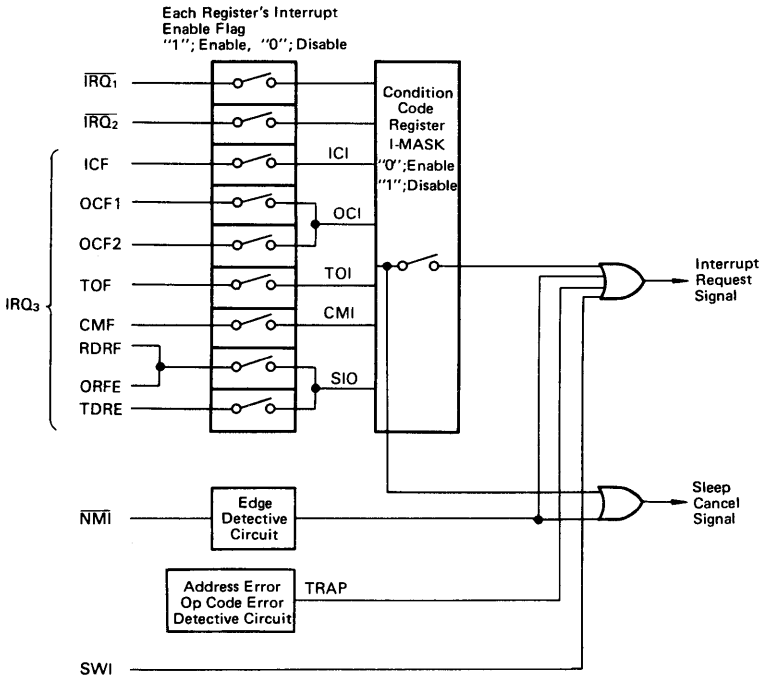


Figure 13 Interrupt Circuit Block Diagram

- **Mode Program (MP_0, MP_1)**

These two pins decide the operation mode. Refer to "MODE SELECTION" for mode details.

The following signal descriptions are applicable only for the expanded mode.

- **Read/Write (R/$\overline{W}$; P_{72})**

This signal, usually be in read state ("High"), shows whether the CPU is in read ("High") or write ("Low") state to the peripheral or memory devices. This can drive one TTL load and 30pF capacitance.

- **$\overline{RD}$, $\overline{WR}$ (P_{70}, P_{71})**

These signals show active low outputs when the CPU is reading/writing to the peripherals or memories. This enables the CPU easy to access the peripheral LSI with $\overline{RD}$ and $\overline{WR}$ input pins. These pins can drive one TTL load and 30pF capacitance.

- **Load Instruction Register ($\overline{LIR}$; P_{73})**

This signal shows the instruction opecode being on data bus (active low). This pin can drive one TTL load and 30pF capacitance.

- **Memory Ready (MR; P_{52})**

This is the input control signal which stretches the system clock's "High" period to access low-speed memories. During this signal is in "High", the system clock operates in normal sequence. But this signal in "Low", the "High" period of the system clock will be stretched depending on its "Low" level duration in integral multiples of the cycle time. This allows the CPU to interface with low-speed memories (See Fig. 2). Up to 9μs can be stretched.

During internal address space access or nonvalid memory access, MR is prohibited internally to prevent decrease of operation speed. Even in the halt state, MR can also stretch "High" period of system clock to allow peripheral devices to access low-speed memories. As this signal is used also as P_{52}, an enable bit is provided at bit 2 of the RAM/port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for more details.

- **Halt ($\overline{HALT}$; P_{53})**

This is an input control signal to stop instruction execution and to release buses. When this signal switches to "Low", the CPU stops to enter into the halt state after having executed the present instruction. When entering into the halt state, it makes BA (P_{74}) "High" and also an address bus, data bus, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ high impedance. When an interrupt is generated in the halt state, the CPU uses the interrupt handler after the halt is cancelled.

(Note)

Please don't switch the $\overline{HALT}$ signal to "Low" when the CPU executes the WAI instruction and is in the interrupt wait state to avoid the trouble of the CPU's operation after the halt is cancelled.

- **Bus Available (BA; P_{74})**

This is an output control signal which is normally "Low" but "High" when the CPU accepts $\overline{HALT}$ and releases the buses. The HD6800 and HD6802 make BA "High" and release the

buses at WAI execution, while the HD6301X0 doesn't make BA "High" under the same condition. But if the $\overline{HALT}$ becomes "Low" when the CPU is in the interrupt wait state after having executed the WAI, the CPU makes BA "High" and releases the buses. And when the $\overline{HALT}$ becomes "High", the CPU returns to the interrupt wait state.

■ PORT

The HD6301X0 provides seven I/O ports (six 8-bit ports and a 5-bit port). Table 2 gives the address of ports and the data direction register and Fig. 14 the block diagrams of each port.

Table 2 Port and Data Direction Register Address

Port	Port Address	Data Direction Register
Port 1	$0002	–
Port 2	$0003	$0001
Port 3	$0006	$0004
Port 4	$0007	–
Port 5	$0015	–
Port 6	$0017	$0016
Port 7	$0018	–

• Port 1

An 8-bit port for output only. In mode 3, port 1 goes to high impedance during reset and keeps the state even after accepting reset cancellation. It continues till a write operation is made to port 1. When a write operation is made to port 1, the high impedance state shifts to the output state and the written data will be output. Once port 1 gets in the output state, it operates as an output till reset occurs. The CPU can also read the value of the Port 1 data register, thus enables the CPU to use bit manipulation.

In mode 1 and 2, port 1 acts as lower address buses. This port can drive one TTL load and 90pF capacitance.

• Port 2

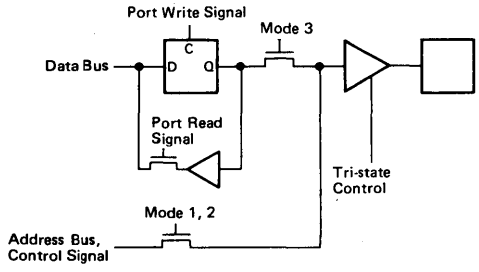
An 8-bit input/output port. The data direction register (DDR) of port 2 controls the I/O state. It provides two bits; bit 0 decides the I/O direction of P_{20} and bit 1 the I/O direction of P_{21} to P_{27} ("0" for input, "1" for output).

Port 2 is also used as an I/O pin for the timers and the SCI. When used as an I/O pin for the timers and the SCI, port 2 except P_{20} automatically becomes an input or an output depending on their functions regardless of the data direction register's value.

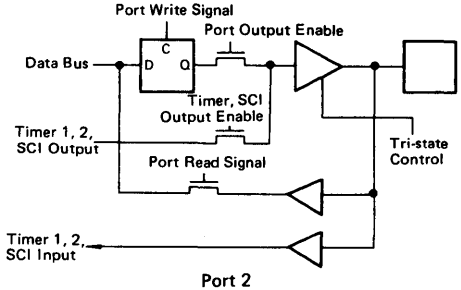
Port 2 Data Direction Register

7	6	5	4	3	2	1	0	
–	–	–	–	–	–	DDR 1–7	DDR 0	$0001

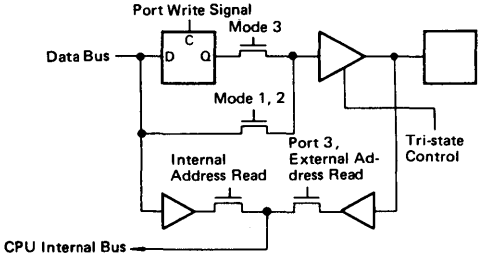
A reset clears the DDR of port 2 and configures port 2 as an input port. This port can drive one TTL and 30pF capacitance. In addition, it can produce 1mA current when Vout = 1.5V to drive directly the base of Darlington transistors.



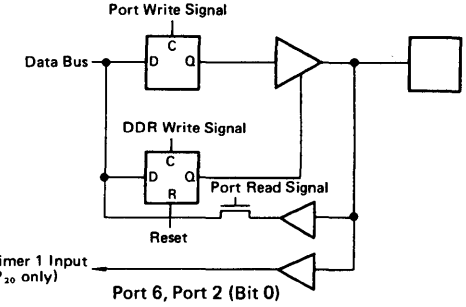
Port 1, Port 4, Port 7



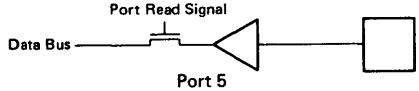
Port 2



Port 3



Port 6, Port 2 (Bit 0)



Port 5

Figure 14 Port Block Diagram

- **Port 3**

An 8-bit I/O port. The DDR of port 3 controls the I/O state. It provides only one bit which decides I/O state by the byte ("0" for input and "1" for output). It is cleared during reset. Port 3 can drive one TTL load and 90pF capacitance.

Port 3 Data Direction Register

7	6	5	4	3	2	1	0	
–	–	–	–	–	–	–	Port 3 DDR	$0004

- **Port 4**

An 8-bit port for output only like Port 1. In mode 1 and 2, "High" address will be produced.

- **Port 5**

An 8-bit port for input only. The lower four bits are also usable as input pins for interrupt, MR and $\overline{HALT}$.

- **Port 6**

An 8-bit I/O port. This port provides an 8-bit DDR corresponding to each bit and can specify input or output by the bit ("0" for input, "1" for output). This port can drive one TTL load and 30pF capacitance. A reset clears the DDR of port 6. In addition, it can produce 1mA current when Vout = 1.5V to drive directly the base of Darlington transistors.

- **Port 7**

A 5-bit port for output only. In mode 3, port 7 goes to high impedance during reset and keeps the state even after accepting reset cancellation. It continues till a write operation is made to port 7. When a write operation is made to port 7, the high impedance state shifts to the output state and the written data will be output. Once port 7 gets in the output state, it operates as an output till reset occurs. Port 7 can also read the value of the data register, thus enables the CPU to use the bit manipulation instruction. In this case $b_7 \sim b_5$ are "1".

In mode 1 and 2, port 7 acts as outputs for control signals ($\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$ and BA). This port can drive one TTL load and 30pF capacitance.

■ RAM/PORT 5 CONTROL REGISTER

The control register located at $0014 controls on-chip RAM and port 5.

RAM/Port 5 Control Register

7	6	5	4	3	2	1	0	
STBY PWR	RAME	–	–	HLTE	MRE	IRQ_2 E	IRQ_1 E	$0014

Bit 0, Bit 1 $\overline{IRQ_1}$, $\overline{IRQ_2}$ Enable Bit (IRQ_1E, IRQ_2E)

When using P_{50} and P_{51} as interrupt pins, write "1" in these bits. When "0", the CPU doesn't accept an external interrupt or a sleep cancellation by the external interrupt. These bits become "0" during reset.

Bit 2 Memory Ready Enable Bit (MRE)

When using P_{52} as an input for Memory Ready signal, write "1" in this bit. When "0", the memory ready function is prohibited. In mode 3, the memory ready function is prohibited regardless of the value of this bit and P_{52} can be used as the I/O port. This bit becomes "1" during reset.

Bit 3 Halt Enable bit (HLTE)

When using P_{53} as an input for Halt signal, write "1" in this bit. When "0", the halt function is prohibited. In mode 3, the halt function is prohibited regardless of the value of this bit and P_{53} can be used as the I/O port. This bit becomes "1" during reset.

(Note)

When using P_{52} and P_{53} as the input ports in mode 1 and 2, MRE and HLTE bit should be cleared just after the reset. Notice that memory ready and halt function is enable till MRE and HLTE bit is cleared.

Bit 4, Bit 5 Not Used.

Bit 6 RAM Enable (RAME)

On-chip RAM can be disabled by this control bit. By resetting the MCU, "1" is set to this bit, and on-chip RAM is enabled. This bit can be written "1" or "0" by software. When RAM is in disable condition (=logic "0"), on-chip RAM is invalid and the CPU can read data from external memory. This bit should be "0" before getting into the standby mode to protect on-chip RAM data.

Bit 7 Standby Power Bit (STBY PWR)

When V_{CC} is not provided in standby mode, this bit is cleared. This is a flag for both read/write by software. If this bit is set before standby mode, and remains set even after returning from standby mode, V_{CC} voltage is provided during standby mode and the on-chip RAM data is valid.

■ MODE SELECTION

Mode program pins, MP_0 and MP_1 determine the operation mode of the HD6301X0 as Table 3 gives.

- **Mode 1 (Expanded Mode)**

In this mode, port 3 is data bus and port 1 lower address bus and port 4 upper address bus to interface directly with the HMCS6800 buses. A control signal such as R/$\overline{W}$ is produced at port 7. In mode 1, on-chip ROM is disabled and 65k bytes of address space are externally expandable (refer to Fig. 15).

- **Mode 2 (Expanded Mode)**

This mode is also expandable as well as mode 1. But in this mode, on-chip ROM is enabled and the expandable address space is 61k bytes (refer to Fig. 16).

- **Mode 3 (Single-chip Mode)**

In this mode, all ports are available (refer to Fig. 17).

Table 3 Mode Selection

Mode	MP_1	MP_0	ROM	RAM	Interrupt Vector	Operation Mode
1	"L"	"H"	E	I*	E	Expanded Mode
2	"H"	"L"	I	I*	I	Expanded Mode
3	"H"	"H"	I	I	I	Single-chip Mode

"L" = Logic "0", "H" = Logic "1", I; Internal, E; External.
* The addressing RAM area can be external by clearing RAME bit at $0014.

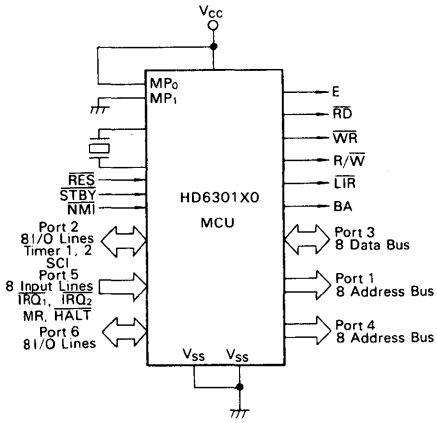


Figure 15 Mode 1

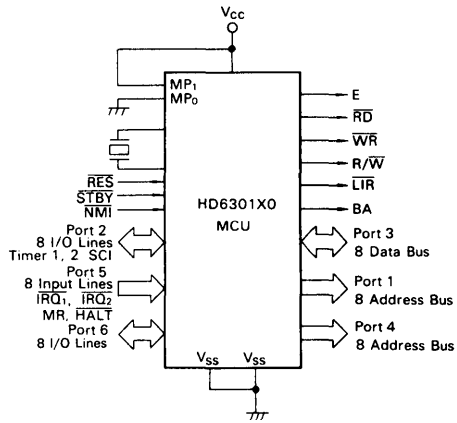


Figure 16 Mode 2

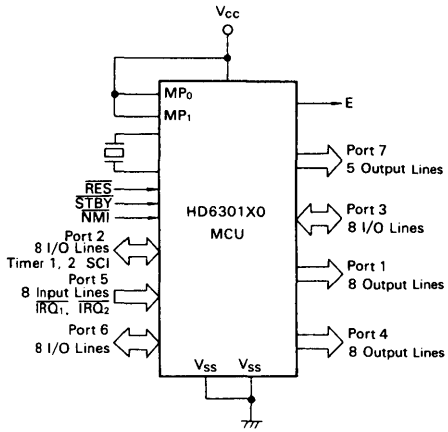


Figure 17 Mode 3

- **Mode and Port**

Table 4 shows MCU signals in each mode.

Table 4 MCU Signals in Each Mode

Port \ Mode	Mode 1	Mode 2	Mode 3
Port 1	Address Bus (A_0~A_7)	Address Bus (A_0~A_7)	Output Port
Port 2	I/O Port	I/O Port	I/O Port
Port 3	Data Bus (D_0~D_7)	Data Bus (D_0~D_7)	I/O Port
Port 4	Address Bus (A_8~A_{15})	Address Bus (A_8~A_{15})	Output Port
Port 5	Input Port	Input Port	Input Port
Port 6	I/O Port	I/O Port	I/O Port
Port 7	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	Output Port

■ MEMORY MAP

The MCU can address up to 65k bytes depending on its operation mode. Fig. 18 gives memory maps in each operation mode. 32 internal registers use addresses from "00" as shown in Table 5.

Table 5 Internal Register

Address	Registers	R/W***	Initialize at RESET
00	—	—	—
01	Port 2 Data Direction Register	W	$FC
02*	Port 1	R/W	Undefined
03	Port 2	R/W	Undefined
04*	Port 3 Data Direction Register	W	$FE
05	—	—	—
06*	Port 3	R/W	Undefined
07*	Port 4	R/W	Undefined
08	Timer Control/Status Register 1	R/W	$00
09	Free Running Counter ("High")	R/W	$00
0A	Free Running Counter ("Low")	R/W	$00
0B	Output Compare Register 1 ("High")	R/W	$FF
0C	Output Compare Register 1 ("Low")	R/W	$FF
0D	Input Capture Register ("High")	R	$00
0E	Input Capture Register ("Low")	R	$00
0F	Timer Control/Status Register 2	R/W	$10
10	Rate, Mode Control Register	R/W	$00
11	Tx/Rx Control Status Register	R/W	$20
12	Receive Data Register	R	$00
13	Transmit Data Register	W	$00
14	RAM/Port 5 Control Register	R/W	$7C or $FC
15	Port 5	R	—
16	Port 6 Data Direction Register	W	$00
17	Port 6	R/W	Undefined
18*	Port 7	R/W	Undefined
19	Output Compare Register 2 ("High")	R/W	$FF
1A	Output Compare Register 2 ("Low")	R/W	$FF
1B	Timer Control/Status Register 3	R/W	$20
1C	Time Constant Register	W	$FF
1D	Timer 2 Up Counter	R/W	$00
1E	—	—	—
1F**	Test Register	—	—

* External Address in Mode 1, 2.
** Test Register. Do not access to this register.
*** R : Read Only Register
W : Write Only Register
R/W: Read/Write Register

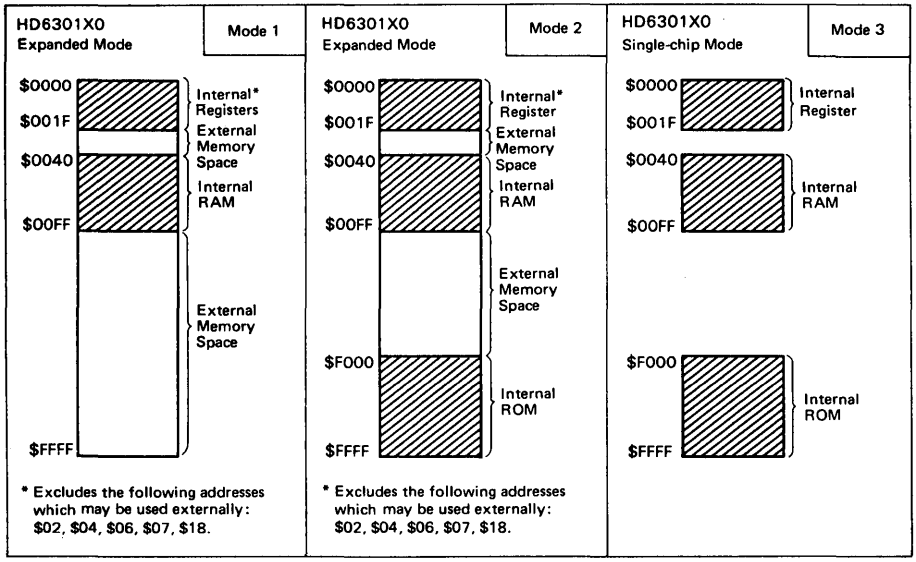


Figure 18 HD6301X0 Memory Map

■ TIMER 1

The HD6301X0 provides a 16-bit programmable timer which can simultaneously measure an input waveform and generate two independent output waveforms. The pulse widths of both input/output waveforms vary from microseconds to seconds.

Timer 1 is configurated as follows (refer to Fig. 20).

- Control/Status Register 1 (8 bit)
- Control/Status Register 2 (7 bit)
- Free Running Counter (16 bit)
- Output Compare Register 1 (16 bit)
- Output Compare Register 2 (16 bit)
- Input Capture Register (16 bit)

• Free-Running Counter (FRC) ($0009 : 000A)

The key timer element is a 16-bit free-running counter driven and incremented by system clock. The counter value is readable by software without affecting the counter. The counter is cleared by reset.

When writing to the upper byte ($09), the CPU writes the preset value ($FFF8) into the counter (address $09, $0A) regardless of the write data value. But when writing to the lower byte ($0A) after the upper byte writing, the CPU writes not only the lower byte data into lower 8 bit, but also the upper byte data into higher 8 bit of the FRC.

The counter will be as follows when the CPU writes to it by double store instructions (STD, STX etc.).

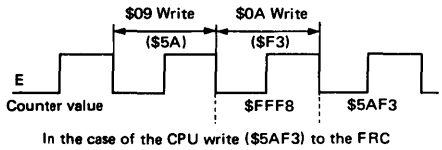


Figure 19 Counter Write Timing

• Output Compare Register (OCR) ($000B, $000C; OCR1) ($0019, $001A ; OCR2)

The output compare register is a 16-bit read/write register which can control an output waveform. The data of OCR is always compared with the FRC.

When the data matches, output compare flag (OCF) in the timer control/status register (TCSR) is set. If an output enable bit (OE) in the TCSR2 is "1", an output level bit (OLVL) in the TCSR will be output to bit 1 (Tout 1) and bit 5 (Tout 2) of port 2. To control the output level again by the next compare, the value of OCR and OLVL should be changed. The OCR is set to $FFFF at reset. The compare function is inhibited for a cycle just after a write to the OCR or to the upper byte of the FRC. This is to begin the comparison after setting the 16-bit value valid in the register and to inhibit the compare function at this cycle, because the CPU writes the upper byte to the FRC, and at the next cycle the counter is set to $FFF8.

* For data write to the FRC or the OCR, 2-byte transfer instruction (such as STX etc.) should be used.

• Input Capture Register (ICR) ($000D : 000E)

The input capture register is a 16-bit read only register which stores the FRC's value when external input signal transition generates an input capture pulse. Such transition is controlled by input edge bit (IEDG) in the TCSR1.

In order to input the external input signal to the edge detecter, a bit of the DDR corresponding to bit 0 of port 2 should be cleared ("0"). When an input capture pulse occurs by the external input signal transition at the next cycle of CPU's high-byte read of the ICR, the input capture pulse will be delayed by one cycle. In order to ensure the input capture operation, a CPU read of the ICR needs 2-byte transfer instruction. The input pulse width should be at least 2 system cycles. This register is cleared ($0000) during reset.

- **Timer Control/Status Register 1 (TCSR1) ($0008)**

The timer control/status register 1 is an 8-bit register. All bits are readable and the lower 5 bits are also writable. The upper 3 bits are read-only which indicate the following timer status.

Bit 5 The counter value reached to $0000 as a result of counting-up (TOF).

Bit 6 A match has occured between the FRC and the OCR 1 (OCF1).

Bit 7 Defined transition of the timer input signal causes the counter to transfer its data to the ICR (ICF).

The followings are each bit descriptions.

Timer Control/Status Register 1

7	6	5	4	3	2	1	0	
ICF	OCF1	TOF	EICI	EOCI1	ETOI	IEDG	OLVL1	$0008

Bit 0 OLVL1 Output Level 1

OLVL1 is transferred to port 2, bit 1 when a match occurs between the counter and the OCR1. If bit 0 of the TCSR2 (OE1) is set to "1", OLVL1 will appear at bit 1 of port 2.

Bit 1 IEDG Input Edge

This bit determines which edge, rising or falling of input signal of port 2, bit 0 will trigger data transfer from the counter to the ICR. For this function, the DDR corresponding to port 2, bit 0 should be cleared beforehand.

IEDG=0, triggered on a falling edge ("High" to "Low")

IEDG=1, triggered on a rising edge ("Low" to "High")

Bit 2 ETOI Enable Timer Overflow Interrupt

When this bit is set, an internal interrupt (IRQ_3) by TOI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 3 EOCI1 Enable Output Compare Interrupt 1

When this bit is set, an internal interrupt (IRQ_3) by OCI1 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 EICI Enable Input Capture Interrupt

When this bit is set, an internal interrupt (IRQ_3) by ICI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag

This read-only bit is set when the counter increments from SFFFF by 1. Cleared when the counter's the upper byte (S0009) is read by the CPU after the TCSR1 read.

Bit 6 OCF1 Output Compare Flag 1

This read-only bit is set when a match occurs between the OCR1 and the FRC. Cleared when writing to the OCR1 (S000B or S000C) after the TCSR1 or TCSR2 read.

Bit 7 ICF Input Capture Flag

This read-only bit is set when an input signal of port 2, bit 0 makes a transition as defined by IEDG and the FRC is transferred to the ICR. Cleared when reading the upper byte (S000D) of the ICR after the TCSR1 or TCSR2 read.

- **Timer Control/Status Register 2 (TCSR2) ($000F)**

The timer control/status register 2 is a 7-bit register. All bits are readable and the lower 4 bits are also writable. But the upper 3 bits are read-only which indicate the following timer status.

Bit 5 A match has occured between the FRC and the OCR2 (OCF2).

Bit 6 The same status flag as the OCF1 flag of the TCSR1, bit 6.

Bit 7 The same status flag as the ICF flag of the TCSR1, bit 7.

The followings are the each bit descriptions.

Timer Control/Status Register 2

7	6	5	4	3	2	1	0	
ICF	OCF1	OCF2	–	EOCI2	OLVL2	OE2	OE1	$000F

Bit 0 OE1 Output Enable 1

This bit enables the OLVL1 to appear at port 2, bit 1 when a match has occurred between the counter and the output compare register 1. When this bit is cleared, bit 1 of port 2 will be an I/O port. When set, it will be an output of OLVL1 automatically.

Bit 1 OE2 Output Enable 2

This bit enables the OLVL2 to appear at port 2, bit 5 when a match has occurred between the counter and the output compare register 2. When this bit is cleared, port 2, bit 5 will be an I/O port. When set, it will be an output of OLVL2 automatically.

Bit 2 OLVL2 Output Level 2

OLVL2 is transferred to port 2, bit 5 when a match has occurred between the counter and the OCR2. If bit 5 of the TCSR2 (OE2) is set to "1", OLVL2 will appear at port 2, bit 5.

Bit 3 EOCI2 Enable Output Compare Interrupt 2

When this bit is set, an internal interrupt (IRQ_3) by OCI2 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 Not Used

Bit 5 OCF2 Output Compare Flag 2

This read-only bit is set when a match has occurred between the counter and the OCR2. Cleared when writing to the OCR2 (S0019 or S001A) after the TCSR2 read.

Bit 6 OCF1 Output Compare Flag 1

Bit 7 ICF Input Capture Flag

OCF1 and ICF addresses are partially decoded. The CPU read of the TCSR1/TCSR2 makes it possible to read OCF1 and ICF into bit 6 and bit 7.

Both the TCSR1 and TCSR2 will be cleared during reset. (Note)

If OE1 or OE2 is set to "1" before the first output compare match occurs after reset restart, bit 1 or bit 5 of port 2 will produce "0" respectively.

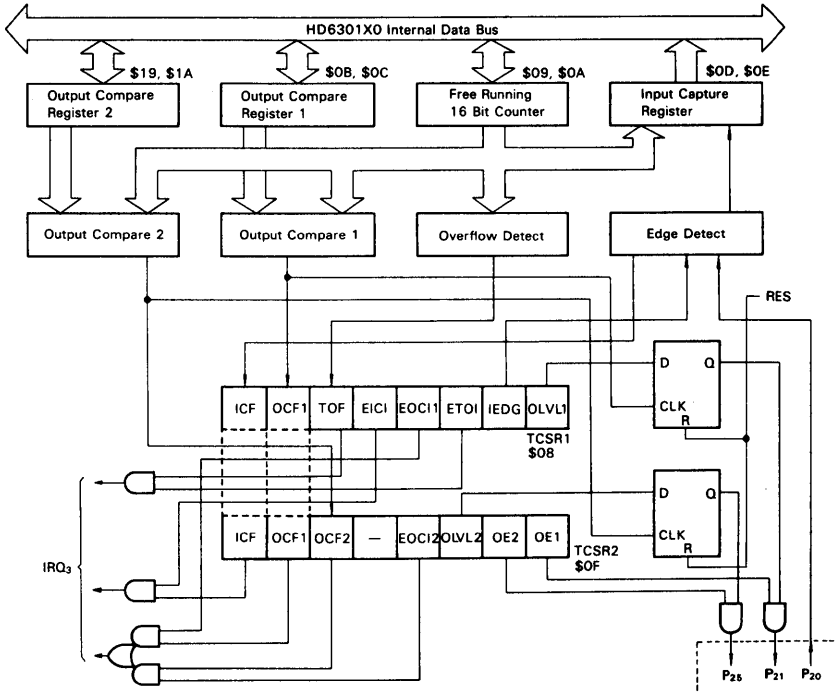


Figure 20 Timer 1 Block Diagram

■ TIMER 2

In addition to the timer 1, the HD6301X0 provides an 8-bit reloadable timer, which is capable of counting the external event. This timer 2 contains a timer output, so the MCU can generate three independent waveforms. (Refer to Fig. 21.)

The timer 2 is configured as follows:

Control/Status Register 3 (7 bit)
8-bit Up Counter
Time Constant Register (8 bit)

• Timer 2 Up Counter (T2CNT) ($001D)

This is an 8-bit up counter which operates with the clock decided by CKS0 and CKS1 of the TCSR3. The CPU can read the value of the counter without affecting the counter. In addition, any value can be written to the counter by software even during counting.

The counter is cleared when a match occurs between the counter and the TCONR or during reset.

If a write operation is made by software to the counter at the cycle of counter clear, it does not reset the counter but put the write data to the counter.

• Time Constant Register (TCONR) ($001C)

The time constant register is an 8-bit write only register. It is always compared with the counter.

When a match has occurred, counter match flag (CMF) of the timer control status register 3 (TCSR3) is set and the value selected by TOS0 and TOS1 of the TCSR3 will appear at port 2, bit 6. When CMF is set, the counter will be cleared simultaneously and then start counting from $00. This enables regular interrupts and waveform outputs without any software support. The TCONR is set to "$FF" during reset.

• Timer Control/Status Register 3 (TCSR3) ($001B)

The timer control/status register 3 is a 7-bit register. All bits are readable and 6 bits except for CMF can be written.

The followings are each pin descriptions.

Timer Control/Status Register 3

7	6	5	4	3	2	1	0	
CMF	ECMI	—	T2E	TOS1	TOS0	CKS1	CKS0	$001B

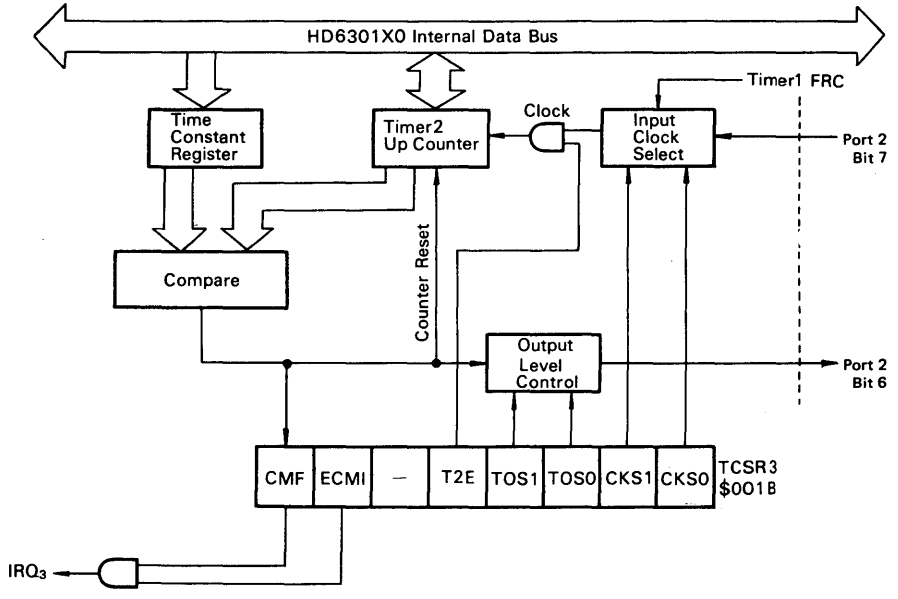


Figure 21 Timer 2 Block Diagram

Bit 0 CKS0 Input Clock Select 0
Bit 1 CKS1 Input Clock Select 1

Input clock to the counter is selected as shown in Table 6 depending on these two bits. When an external clock is selected, bit 7 of port 2 will be a clock input automatically. Timer 2 detects the rising edge of the external clock and increments the counter. The external clock is countable up to half the frequency of the system clock.

Table 6 Input Clock Select

CKS1	CKS0	Input Clock to the Counter
0	0	E clock
0	1	E clock/8*
1	0	E clock/128*
1	1	External clock

* These clocks come from the FRC of the timer 1. If one of these clocks is selected as an input clock to the up counter, the CPU should not write to the FRC of the timer 1.

Bit 2 TOS0 Timer Output Select 0
Bit 3 TOS1 Timer Output Select 1

When a match occurs between the counter and the TCONR timer 2 outputs shown in Table 7 will appear at port 2, bit 6 depending on these two bits. When both TOS0 and TOS1 are "0", bit 6 of port 2 will be an I/O port.

Table 7 Timer 2 Output Select

TOS1	TOS0	Timer Output
0	0	Timer Output Inhibited
0	1	Toggle Output*
1	0	Output "0"
1	1	Output "1"

* When a match occurs between the counter and the TCONR, timer 2 output level is reversed. This leads to production of a square wave with 50% duty to the external without any software support.

Bit 4 T2E Timer 2 Enable Bit

When this bit is cleared, a clock input to the up counter is prohibited and the up counter stops. When set to "1", a clock selected by CKS1 and CKS0 (Table 6) is input to the up counter.

(Note)

P_{26} outputs "0" when T2E bit cleared and timer 2 set in output enable condition by TOS1 or TOS0. It also outputs "0" when T2E bit set "1" and timer 2 set in output enable condition before the first counter match occurs.

Bit 5 Not Used
Bit 6 ECMI Enable Counter Match Interrupt

When this bit is set, an internal interrupt (IRQ_3) by CMI is enabled. When cleared, the interrupt is inhibited.

Bit 7 CMF Counter Match Flag

This read-only bit is set when a match occurs between the up counter and the TCONR. Cleared by writing "0" by software (unable to write "1" by software).

Each bit of the TCSR3 is cleared during reset.

■ SERIAL COMMUNICATION INTERFACE (SCI)

The HD6301X0 SCI contains two operation modes; one is an asynchronous mode by the NRZ format and the other is a clocked synchronous mode which transfers data synchronizing with the serial clock.

The SCI consists of the following registers as shown in Fig. 22 Block Diagram:

- Control/Status Register (TRCSR)
- Rate/Mode Control Register (RMCR)
- Receive Data Register (RDR)
- Receive Data Shift Register (RDSR)
- Transmit Data Register (TDR)
- Transmit Data Shift Register (TDSR)

The serial I/O hardware requires an initialization by software for operation. The procedure is usually as follows:

1) Write a desirable operation mode into each corresponding control bit of the RMCR.
2) Write a desirable operation mode into each corresponding control bit of the TRCSR.

When using bit 3 and 4 of port 2 for serial I/O only, there is no problem even if TE and RE bit are set. But when setting the baud rate and operation mode, TE and RE should be "0". When clearing TE and RE bit and setting them again, more than 1 bit cycle of the current baud rate is necessary. If set in less than 1 bit cycle, there may be a case that the internal transmit/receive initialization fails.

• Asynchronous Mode

An asynchronous mode contains the following two data formats:

1 Start Bit + 8 Bit Data + 1 Stop Bit
1 Start Bit + 9 Bit Data + 1 Stop Bit

In addition, if the 9th bit is set to "1" when making 9 bit data format, the format of

1 Start bit + 8 Bit Data + 2 Stop Bit

is also transferred.

Data transmission is enabled by setting TE bit of the TRCSR, then port 2, bit 4 will become a serial output independently of the corresponding DDR.

For data transmit, both the RMCR and TRCSR should be set under the desirable operating conditions. When TE bit is set during this process, 10 bit preamble will be sent in 8-bit data format and 11 bit in 9-bit data format. When the preamble is produced, the internal synchronization will become stable and the transmitter is ready to act.

The conditions at this stage are as follows.

1) If the TDR is empty (TDRE=1), consecutive 1's are produced to indicate the idle state.
2) If the TDR contains data (TDRE=0), data is sent to the transmit data shift register and data transmit starts.

During data transmit, a start bit of "0" is transmitted first. Then 8-bit or 9-bit data (starts from bit 0) and a stop bit "1" are transmitted.

When the TDR is "empty", hardware sets TDRE flag bit. If the CPU doesn't respond to the flag in proper timing (the TDRE is in set condition till the next normal data transfer starts from the transmit data register to the transmit sift register), "1" is transferred instead of the start bit "0" and continues to be transferred till data is provided to the data register. While the TDRE is "1", "0" is not transferred

Data receive is possible by setting RE bit. This makes port 2, bit 3 a serial input. The operation mode of data receive is decided by the contents of the TRCSR and RMCR. The first "0" (space) synchronizes the receive bit flow. Each bit of the following data will be strobed in the middle. If a stop bit is not "1", a framing error assumed and ORFE is set.

When a framing error occurs, receive data is transferred to the receive data register and the CPU can read error-generating data. This makes it possible to detect a line break.

If the stop bit is "1", data is transferred to the receive data register and an interrupt flag RDRF is set. If RDRF is still set when receiving the stop bit of the next data, ORFE is set to indicate overrun generation.

When the CPU read the receive data register as a response to RDRF flag or ORFE flag after having read TRCS, RDRF or ORFE is cleared.

(Note) Clock Source in Asynchronous Mode

If CC1: CC0 = 10, the internal bit rate clock is provided at P_{22} regardless of the values for TE or RE. Maximum clock rate is E ÷ 16.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P_{22} at sixteen times (16x) the desired bit rate, but not greater than E.

• Clocked Synchronous Mode

In the clocked synchronous mode, data transmit is synchronized with the clock pulse. The HD6301X0 SCI provides functionally independent transmitter and receiver which makes full duplex operation possible in the asynchronous mode. But in the clocked synchronous mode an SCI clock I/O pin is only P_{22}, so the simultaneous receive and transmit operation is not available. In this mode, TE and RE should not be in set condition ("1") simultaneously. Fig. 23 gives a synchronous clock and a data format in the clocked synchronous mode.

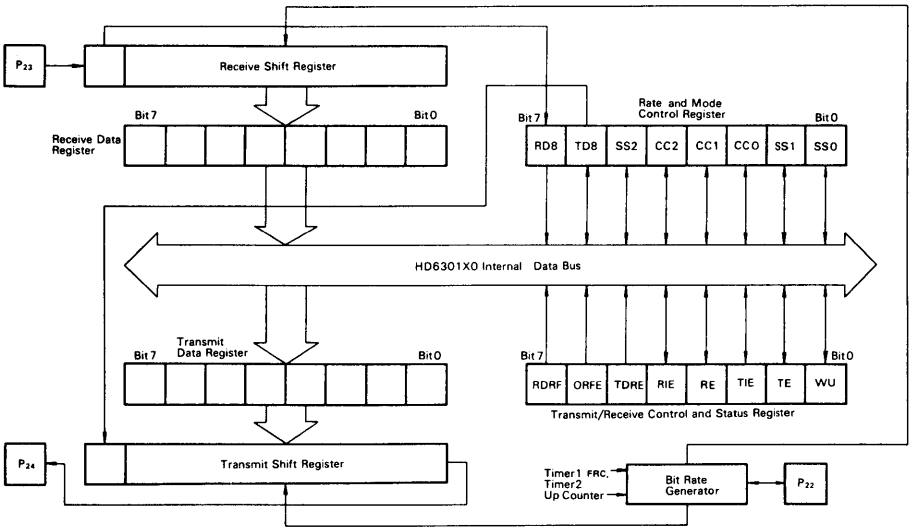


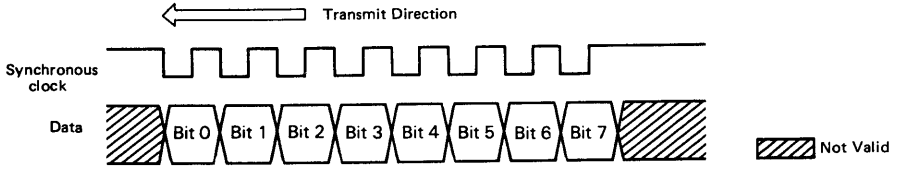
Figure 22 Serial Communication Interface Block Diagram

Data transmit is realized by setting TE bit in the TRCSR. Port 2, bit 4 becomes an output unconditionally independent of the value of the corresponding DDR.

Both the RMCR and TRCSR should be set in the desirable operating condition for data transmit.

When an external clock input is selected, data transmit is performed under the TDRE flag "0" from port 2, bit 4, synchronizing with 8 clock pulses input from external to port 2, bit 2.

Data is transmitted from bit 0 and the TDRE is set when the transmit data shift register is "empty". More than 9th clock pulse of external are ignored.



- Transmit data is produced from a falling edge of a synchronous clock to the next falling edge.
- Receive data is latched at the rising edge.

Figure 23 Clocked Synchronous Mode Format

When data transmit is selected to the clock output, the MCU produces transmit data and synchronous clock at TDRE flag clear.

Data receive is enabled by setting RE bit. Port 2, bit 3 will be a serial input. The operating mode of data receive is decided by the TRCSR and the RMCR.

If the external clock input is selected, RE bit should be set when P_{22} is "High". Then 8 external clock pulses and the synchronized receive data are input to port 2, bit 2 and bit 3 respectively. The MCU put receive data into the receive data shift register by this clock and set the RDRF flag at the termination of 8 bit data receive. More than 9th clock pulse of external input are ignored. When RDRF is cleared by reading the receive data register, the MCU starts receiving the next data. So RDRF should be cleared with P_{22} "High"

When data receive is selected to the clock output, 8 synchronous clocks are output to the external by setting RE bit. So receive data should be input from external, synchronously with this clock. When the first byte data is received, the RDRF flag is set. After the second byte, receive operation is performed and output the synchronous clock to the external by clearing the RDRF bit.

- **Transmit/Receive Control Status Register (TRCSR) ($0011)**

The TRCSR is composed of 8 bits which are all readable. Bits 0 to 4 are also writable. This register is initialized to $20 during reset. Each bit functions as follows.

Transmit/Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU Wake-up

In a typical multi-processor configuration, the software protocol provides the destination address at the first byte of the message. In order to make uninterested MCU ignore the remaining message, a wake-up function is available. By this, uninterested MCU can inhibit all further receive processing till the next message starts.

Then wake-up function is triggered by consecutive 1's with 1 frame length (10 bits for 8-bit data, 11 for 9-bit). The software protocol should provide the idle time between messages.

By setting this bit, the MCU stops data receive till the next message. The receive of consecutive "1" with one frame length wakes up and clears this bit and then the MCU restarts receive operation. However, the RE flag should be already set before setting this bit. In the clocked synchronous mode WU is not available, so this bit should not be set.

Bit 1 TE Transmit Enable

When this bit is set, transmit data will appear at port 2, bit 4 after one frame preamble in asynchronous mode, while in clocked synchronous mode it appears immediately. This is executed regardless of the value of the corresponding DDR. When TE is cleared, the serial I/O doesn't affect port 2, bit 4.

Bit 2 TIE Transmit Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when TDRE (bit 5) is set. When cleared, the interrupt is inhibited.

Bit 3 RE Receive Enable

When set, a signal is input to the receiver from port 2, bit 3 regardless of the value of the DDR. When RE is cleared, the serial I/O doesn't affect port 2, bit 3.

Bit 4 RIE Receive Interrupt Enable

When this bit is set, an internal interrupt, IRQ_3 is enabled when RDRF (bit 7) or ORFE (bit 6) is set. When cleared, the interrupt is inhibited.

Bit 5 TDRE Transmit Data Register Empty

TDRE is set when the TDR is transferred to the transmit data shift register in the asynchronous mode, while in clocked synchronous mode when the TDSR is "empty". This bit is reset by reading the TRCSR and writing new transmit data to the transmit data register. TDRE is set to "1" during reset.

(Note)

TDRE should be cleared in the transmittable state after the TE set.

Bit 6 ORFE Overrun Framing Error

ORFE is set by hardware when an overrun or a framing error is generated (during data-receive only). An overrun error occurs when new receive data is ready to be transferred to the RDR during RDRF still being set. A framing error occurs when a stop bit is "0". But in clocked synchronous mode, this bit is not affected. This bit is cleared when reading the TRCSR, then the RDR, or during reset.

Bit 7 RDRF Receive Data Register Full

RDRF is set by hardware when the RDSR is transferred to the RDR. Cleared when reading the TRCSR, then the RDR, or during reset.

(Note) When a few bits are set between bit 5 to bit 7 in the TRCSR, a read of the TRCSR is sufficient for clearing those bits. It is not necessary to read the TRCSR every-time to clear each bit.

- **Transmit Rate/Mode Control Register (RMCR)**

The RMCR controls the following serial I/O:

- Baud Rate
- Clock Source
- Data Format
- Port 2, Bit 2 Function

In addition, if 9-bit data format is set in the asynchronous mode, the 9th bit is put in this register. All bits are readable and writable except bit 7 (read only). This register is set to $00 during reset.

Transfer Rate/Mode Control Register

7	6	5	4	3	2	1	0	
RD8	TD8	SS2	CC2	CC1	CC0	SS1	SS0	$0010

Bit 0 SS0
Bit 1 SS1 } Speed Select
Bit 5 SS2

These bits control the baud rate used for the SCI. Table 8 lists the available baud rates. The timer 1 FRC (SS2=0) and the timer 2 up counter (SS2=1) provide the internal clock to the SCI. When selecting the timer 2 as a baud rate source, it functions as a baud rate generator. The timer 2 generates the baud rate listed in Table 9 depending on the value of the TCONR.

(Note) When operating the SCI with internal clock, do not perform write operation to the timer/counter which is the clock source of the SCI.

Bit 2 CC0
Bit 3 CC1 } Clock Control/Format Select*
Bit 4 CC2

These bits control the data format and the clock source (refer to Table 10).

* CC0, CC1 and CC2 are cleared during reset and the MCU goes to the clocked synchronous mode of the external clock operation. Then the MCU sets port 2, bit 2 into the clock input state. When using port 2, bit 2 as an output port, the DDR of port 2 should be set to "1" and CC1 and CC0 to "0" and "1" respectively.

Table 8 SCI Bit Times and Transfer Rates

(1) Asynchronous Mode

			XTAL	2.4576MHz	4.0MHz	4.9152MHz
SS2	SS1	SS0	E	614.4kHz	1.0MHz	1.2288MHz
0	0	0	E÷16	26μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

* When SS2 is "1", Timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32\,(N+1)} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

(2) Clocked Synchronous Mode *

			XTAL	4.0MHz	6.0MHz	8.0MHz
SS2	SS1	SS0	E	1.0MHz	1.5MHz	2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

* Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC ~ 1/2 system clock.

** The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate } (\mu s/\text{bit}) = \frac{4\,(N+1)}{f} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

Table 9 Baud Rate and Time Constant Register Example

Baud Rate (Baud) \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

* E/8 clock is input to the timer 2 up counter and E clock otherwise.

Table 10 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock Source	Port 2, Bit 2	Port 2, Bit 3	Port 2, Bit 4
0	0	0	8-bit data	Clocked Synchronous	External	Input	When the TRCSR, RE bit is "1", bit 3 is used as a serial input.	
0	0	1	8-bit data	Asynchronous	Internal	Not Used**		
0	1	0	8-bit data	Asynchronous	Internal	Output*		
0	1	1	8-bit data	Asynchronous	External	Input		
1	0	0	8-bit data	Clocked Synchronous	Internal	Output	When the TRCSR, TE bit is "1", bit 4 is used as a serial output.	
1	0	1	9-bit data	Asynchronous	Internal	Not Used**		
1	1	0	9-bit data	Asynchronous	Internal	Output*		
1	1	1	9-bit data	Asynchronous	External	Input		

* Clock output regardless of the TRCSR, bit RE and TE.
** Not used for the SCI.

Bit 6 TD8 Transmit Data Bit 8

When selecting 9-bit data format in the asynchronous mode, this bit is transmitted as the 9th data. In transmitting 9-bit data, write the 9th data into this bit then write data to the receive data register.

Bit 7 RD8 Receive Data Bit 8

When selecting 9-bit data format in the asynchronous mode, this bit stores the 9th bit data. In receiving 9-bit data, read this bit then the receive data register.

■ TIMER, SCI STATUS FLAG

Table 11 shows the set and reset conditions of each status flag in the timer 1, timer 2 and SCI.

As for Timer1 and Timer 2 status flag, if the set and reset condition occur simultaneously, the set condition is prior to the reset condition. But in case of SCI control status flag, the reset condition has priority. Especially as for OCF1 and OCF2 of Timer 1, the set signal is generated periodically whenever FRC matches OCR after the set, and which can cause the unclear of the flag. To clear surely, the method is necessary to avoid the occurence of the set signal between TCSR read and OCR write. For example, match the OCR value to FRC first, and next read TCSR, and then write OCR at once.

Table 11 Timer 1, Timer 2 and SCI Status Flag

		Set Condition	Reset Condition
Timer 1	ICF	FRC → ICR by edge input to P_{20}.	1. Read the TCSR1 or TCSR2 then ICRH, when ICF=1 2. $\overline{RES}$=0
	OCF1	OCR1=FRC	1. Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1=1 2. $\overline{RES}$=0
	OCF2	OCR2=FRC	1. Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2=1 2. $\overline{RES}$=0
	TOF	FRC=$FFFF+1 cycle	1. Read the TCSR1 then FRCH, when TOF=1 2. $\overline{RES}$=0
Timer 2	CMF	T2CNT=TCONR	1. Write "0" to CMF, when CMF=1 2. $\overline{RES}$=0
SCI	RDRF	Receive Shift Register → RDR	1. Read the TRCSR then RDR, when RDRF=1 2. $\overline{RES}$=0
	ORFE	1. Framing Error (Asynchronous Mode) Stop Bit = 0 2. Overrun Error (Asynchronous Mode) Receive Shift Register → RDR when RDRF=1	1. Read the TRCSR then RDR, when ORFE=1 2. $\overline{RES}$=0
	TDRE	1. Asynchronous Mode TDR → Transmit Shift Register 2. Clocked Synchronous Mode Transmit Shift Register is "empty" 3. $\overline{RES}$=0	Read the TRCSR then write to the TDR, when TDRE=1 (Note) TDRE should be reset after the TE set.

(Note) 1. → ; transfer
2. For example; "ICRH" means High byte of ICR.

■ LOW POWER DISSIPATION MODE

The HD6301X0 provides two low power dissipation modes; sleep and standby.

• Sleep Mode

The MCU goes to the sleep mode by SLP instruction execution. In the sleep mode, the CPU stops its operation, while the registers' contents are retained. In this mode, the peripherals except the CPU such as timers, SCI etc. continue their functions. The power dissipation of sleep-condition is one fifth that of operating condition.

The MCU returns from this mode by an interrupt, $\overline{RES}$ or $\overline{STBY}$; it goes to the reset state by $\overline{RES}$ and the standby mode by $\overline{STBY}$. When the CPU acknowledges an interrupt request, it cancels the sleep mode, returns to the operation mode and branches to the interrupt routine. When the CPU masks this interrupt, it cancels the sleep mode and executes the next instruction. However, for example if the timer 1 or 2 prohibits a timer interrupt, the CPU doesn't cancel the sleep mode because of no interrupt request.

This sleep mode is effective to reduce the power dissipation for a system with no need of the HD6301X0's consecutive operation.

• Standby Mode

The HD6301X0 stops all the clocks and goes to the reset state with $\overline{STBY}$ "low. In this mode, the power dissipation is reduced conspicuously. All pins except for the power supply, the $\overline{STBY}$ and XTAL are detached from the MCU internally and go to the high impedance state.

In this mode the power is supplied to the HD6301X0, so the contents of RAM is retained. The MCU returns from this mode during reset. The followings are typical usage of this mode.

Save the CPU information and SP contents on RAM by $\overline{NMI}$. Then disable the RAME bit of the RAM control register and set the STBY PWR bit to go to the standby mode. If the STBY PWR bit is still set at reset start, that indicates the power is supplied to the MCU and RAM contents are retained properly. So system can restore itself by returning their pre-standby informations to the SP and the CPU. Fig. 24 depicts the timing at each pin with this example.

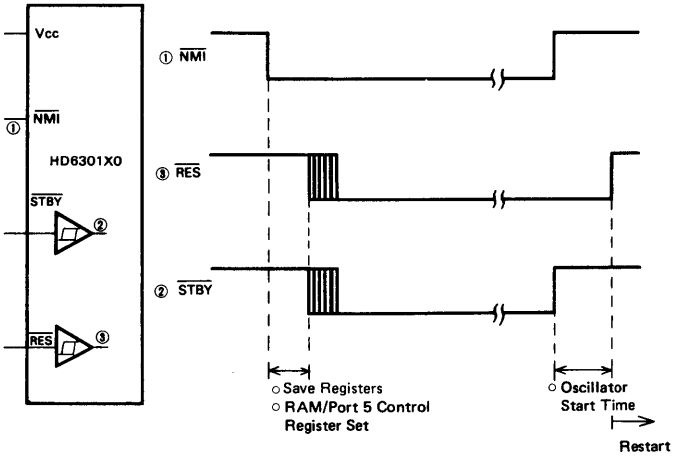


Figure 24 Standby Mode Timing

■ TRAP FUNCTION

The CPU generates an interrupt with the highest priority (TRAP) when fetching an undefined instruction or an instruction from non-memory space. The TRAP prevents the system-burst caused by noise or a program error.

• Op Code Error

When fetching an undefined op code, the CPU saves CPU registers as well as a normal interrupt and branches to the TRAP ($FFEE, $FFEF). This has the priority next to reset.

• Address Error

When an instruction fetch is made excluding internal ROM, RAM and external memory area, the MCU generates an interrupt as well as an op code error. But on the system with no memory in its external memory area, this function is not applicable if an instruction fetch is made from the external non-memory area. Table 12 provides addresses where an address error occurs to each mode.

This function is available only for an instruction fetch and is not applicable to the access of normal data read/write.

Table 12 Addresses Applicable to Address Errors

Mode	1	2	3
Address	$0000 ≀ $001F	$0000 ≀ $001F	$0000 ≀ $003F $0100 ≀ $EFFF

(Note) The TRAP interrupt provides a retry function differently from other interrupts. This is a program flow return to the address where the TRAP occurs when a sequence returns to a main routine from the TRAP interrupt routine by RTI. The retry can prevent the system burst caused by noise etc.
However, if another TRAP occurs, the program repeats the TRAP interrupt forever, so the consideration is necessary in programming.

■ INSTRUCTION SET

The HD6301X0 provides object code upward compatible with the HD6801 to utilize all instruction set of the HMCS6800. It also reduces the execution times of key instructions for throughput improvement.

Bit manipulation instruction, change instruction of the index register and accumulator and sleep instruction are also added.

The followings are explained here.

- CPU Programming Model (refer to Fig. 25)
- Addressing Mode
- Accumulator and Memory Manipulation Instruction (refer to Table 13)
- New Instruction
- Index Register and Stack Manipulation Instruction (refer to Table 14)
- Jump and Branch Instruction (refer to Table 15)
- Condition Code Register Manipulation (refer to Table 16)
- Op Code Map (refer to Table 17)

• Programming Model

Fig. 25 depicts the HD6301X0 programming model. The double accumulator D consists of accumulator A and B, so when using the accumulator D, the contents of A and B are destroyed.

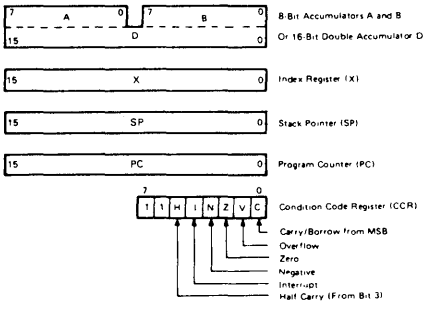


Figure 25 CPU Programming Model

• CPU Addressing Mode

The HD6301X0 provides 7 addressing modes. The addressing mode is decided by an instruction type and code. Table 13 through 17 show addressing modes of each instruction with the execution times counted by the machine cycle.

When the clock frequency is 4 MHz, the machine cycle time becomes microseconds directly.

Accumulator (ACCX) Addressing

Only an accumulator is addressed and the accumulator A or B is selected. This is a one-byte instruction.

Immediate Addressing

This addressing locates a data in the second byte of an instruction. However, LDS and LDX locate a data in the second and third byte exceptionally. This addressing is a 2 or 3-byte instruction.

Direct Addressing

In this addressing mode, the second byte of an instruction shows the address where a data is stored. 256 bytes ($0 through $255) can be addressed directly. Execution times can be reduced by storing data in this area so it is recommended to make it RAM for users' data area in configurating a system. This is a 2-byte instruction, while 3-byte with regard to AIM, OIM, EIM and TIM.

Extended Addressing

In this mode, the second byte shows the upper 8 bit of the data stored address and the third byte the lower 8 bit. This indicates the absolute address of 3-byte instruction in the memory.

Indexed Addressing

The second byte of an instruction and the lower 8 bit of the index register are added in this mode. As for AIM, OIM, EIM and TIM, the third byte of an instruction and the lower 8 bits of the index register are added.

This carry is added to the upper 8 bit of the index register and the result is used for addressing the memory. The modified address is retained in the temporary address register, so the contents of the index register doesn't change. This is a 2-byte instruction except AIM, OIM, EIM and TIM (3-byte instruction).

Implied Addressing

An instruction itself specifies the address. That is, the instruction addresses a stack pointer, index register etc. This is a one-byte instruction.

Relative Addressing

The second byte of an instruction and the lower 8 bits of the program counter are added. The carry or borrow is added to the upper 8 bit. So addressing from −126 to +129 byte of the current instruction is enabled. This is a 2-byte instruction.

(Note) CLI, SEI Instructions and Interrupt Operation

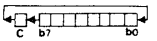
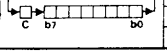
When accepting the IRQ at a preset timing with the CLI and SEI instructions, more than 2 cycles are necessary between the CLI and SEI instructions. For example, the following program (a) (b) don't accept the IRQ but (c) accepts it.

```
                 .          .          .
                 .          .          .
                 .          .          .
                 .          .        CLI
               CLI        CLI        NOP
               SEI        NOP        NOP
                .         SEI        SEI
                .          .          .
                .          .          .
                .          .          .
                .          .          .

               (a)        (b)        (c)
```

The same thing can be said to the TAP instruction instead of the CLI and SEI instructions.

Table 13 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

(Note) Condition Code Register will be explained in Note of Table 16.

(continued)

Table 13 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑤	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑤	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑤	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	ACC A/ ACC B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	1	1	A	•	•	R	↕	⑥	↕
	LSRB													54	1	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	0→ ACC A/ ACC B →C	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

(Note) Condition Code Register will be explained in Note of Table 16.

- **Additional Instruction**

In addition to the HD6801 instruction set, the HD6301X0 prepares the following new instructions.

AIM . . . . . (M)·(IMM) → (M)

Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM . . . . . (M) + (IMM) → (M)

Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM . . . . . (M) ⊕ (IMM) → (M)

Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM . . . . . (M) · (IMM)

Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These area 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX . . . . . (ACCD) ↔ (IX)

Exchanges the contents of accumulator and the index register.

SLP

Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 14 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes														Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD↔IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 16.

Table 15 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		—	—	⑧	—	—	—
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

(Note) * WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 16.

Table 16 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	⑩					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MCU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
• Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3 to bit 4
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from/to bit 7
R Reset Always
S Set Always
↕ Set if true after test or clear
• Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕C = 1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 17 OP-Code Map

OP CODE HI	LO	0000 0	0001 1	0010 2	0011 3	ACC A 0100 4	ACC B 0101 5	IND 0110 6	EXT DIR* 0111 7	ACCA or SP IMM 1000 8	DIR 1001 9	IND 1010 A	EXT 1011 B	ACCB or X IMM 1100 C	DIR 1101 D	IND 1110 E	EXT 1111 F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

■ CPU OPERATION

• CPU Instruction Flow

When operating, the CPU fetches an instruction from a memory and executes the required function. This sequence starts with $\overline{RES}$ cancel and repeats itself limitlessly if not affected by a special instruction or a control signal. SWI, RTI, $\overline{WAI}$ and SLP instructions change this operation, while $\overline{NMI}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, IRQ_3, $\overline{HALT}$ and $\overline{STBY}$ control it. Fig. 26 gives the CPU mode transition and Fig. 27 the CPU system flow chart. Table 18 shows CPU operating states and port states.

• Operation at Each Instruction Cycle

Table 19 shows the operation at each instruction cycle. By the pipeline control of the HD6301X0, MULT, PUL, DAA and XGDX instructions etc. prefetch the next instruction. So attention is necessary to the counting of the instruction cycles because it is different from the usual one ----- op code fetch to the next instruction of code.

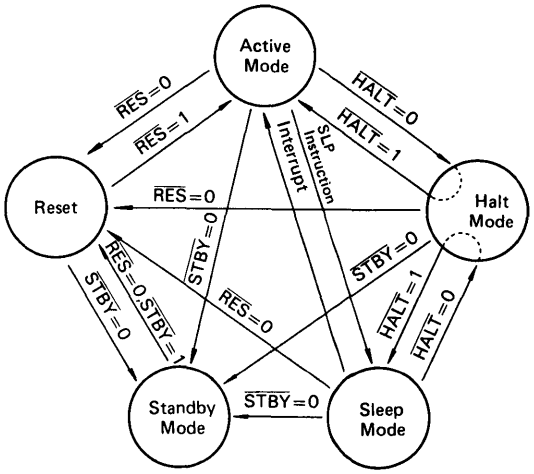


Figure 26 CPU Operation Mode Transition

Table 18 CPU Operation State and Port State

Port	Mode	Reset	STBY****	HALT***	Sleep
Port 1 ($A_0 \sim A_7$)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 2	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 3 ($D_0 \sim D_7$)	Mode 1, 2	T	T	T	T
	Mode 3				Keep
Port 4 ($A_8 \sim A_{15}$)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 5	Mode 1, 2	T	T	T	T
	Mode 3				
Port 6	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 7	Mode 1, 2	*	T	**	*
	Mode 3	T			Keep

H; High, L; Low, T; High Impedance
* $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$=H, BA=L
** $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$=T, $\overline{LIR}$, BA=H
*** $\overline{HALT}$ is unacceptable in mode 3.
**** E pin goes to high impedance state.

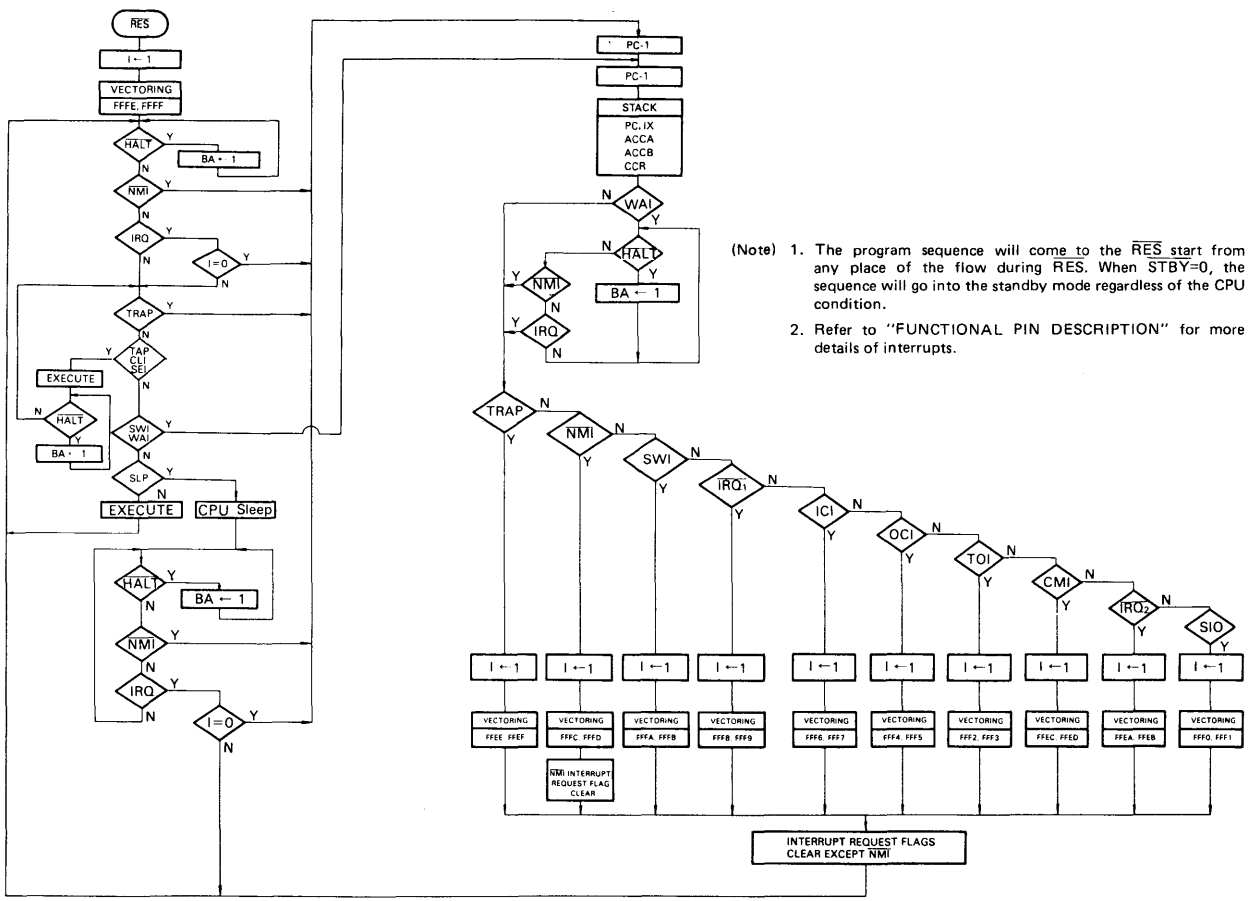


Figure 27 HD6301X0 System Flow Chart

Table 19 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMMEDIATE								
ADC ADD		1	Op Code Address+1	1	0	1	1	Operand Data
AND BIT		2	Op Code Address+2	1	0	1	0	Next Op Code
CMP EOR	2							
LDA ORA								
SBC SUB								
ADDD CPX		1	Op Code Address+1	1	0	1	1	Operand Data (MSB)
LDD LDS	3	2	Op Code Address+2	1	0	1	1	Operand Data (LSB)
LDX SUBD		3	Op Code Address+3	1	0	1	0	Next Op Code
DIRECT								
ADC ADD		1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
AND BIT		2	Address of Operand	1	0	1	1	Operand Data
CMP EOR	3	3	Op Code Address+2	1	0	1	0	Next Op Code
LDA ORA								
SBC SUB								
STA		1	Op Code Address+1	1	0	1	1	Destination Address
	3	2	Destination Address	0	1	0	1	Accumulator Data
		3	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX		1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
LDD LDS	4	2	Address of Operand	1	0	1	1	Operand Data (MSB)
LDX SUBD		3	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		4	Op Code Address+2	1	0	1	0	Next Op Code
STD STS		1	Op Code Address+1	1	0	1	1	Destination Address (LSB)
STX	4	2	Destination Address	0	1	0	1	Register Data (MSB)
		3	Destination Address+1	0	1	0	1	Register Data (LSB)
		4	Op Code Address+2	1	0	1	0	Next Op Code
JSR		1	Op Code Address+1	1	0	1	1	Jump Address (LSB)
		2	FFFF	1	1	1	1	Restart Address (LSB)
	5	3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM		1	Op Code Address+1	1	0	1	1	Immediate Data
	4	2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
AIM EIM		1	Op Code Address+1	1	0	1	1	Immediate Data
OIM		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
	6	3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
INDEXED								
JMP	3	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Accumulator Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data (MSB)
		4	IX+Offset+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Register Data (MSB)
		4	IX+Offset+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	IX+Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX+Offset	0	1	0	1	New Operand Data
		6	Op Code Address+2	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	IX+Offset	0	1	0	1	00
		5	Op Code Address+2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX+Offset	0	1	0	1	New Operand Data
		7	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
EXTEND								
JMP	3	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions		Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED									
ABA	ABX	1	1	Op Code Address+1	1	0	1	0	Next Op Code
ASL	ASLD								
ASR	CBA								
CLC	CLI								
CLR	CLV								
COM	DEC								
DES	DEX								
INC	INS								
INX	LSR								
LSRD	ROL								
ROR	NOP								
SBA	SEC								
SEI	SEV								
TAB	TAP								
TBA	TPA								
TST	TSX								
TXS									
DAA	XGDX	2	1	Op Code Address+1	1	0	1	0	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
PULA	PULB	3	1	Op Code Address+1	1	0	1	0	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer+1	1	0	1	1	Data from Stack
PSHA	PSHB	4	1	Op Code Address+1	1	0	1	1	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer	0	1	0	1	Accumulator Data
			4	Op Code Address+1	1	0	1	0	Next Op Code
PULX		4	1	Op Code Address+1	1	0	1	0	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer+1	1	0	1	1	Data from Stack (MSB)
			4	Stack Pointer+2	1	0	1	1	Data from Stack (LSB)
PSHX		5	1	Op Code Address+1	1	0	1	1	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer	0	1	0	1	Index Register (LSB)
			4	Stack Pointer−1	0	1	0	1	Index Register (MSB)
			5	Op Code Address+1	1	0	1	0	Next Op Code
RTS		5	1	Op Code Address+1	1	0	1	1	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer+1	1	0	1	1	Return Address (MSB)
			4	Stack Pointer+2	1	0	1	1	Return Address (LSB)
			5	Return Address	1	0	1	0	First Op Code of Return Routine
MUL		7	1	Op Code Address+1	1	0	1	0	Next Op Code
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	FFFF	1	1	1	1	Restart Address (LSB)
			4	FFFF	1	1	1	1	Restart Address (LSB)
			5	FFFF	1	1	1	1	Restart Address (LSB)
			6	FFFF	1	1	1	1	Restart Address (LSB)
			7	FFFF	1	1	1	1	Restart Address (LSB)

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
WAI	9	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
RTI	10	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Conditional Code Register
		4	Stack Pointer+2	1	0	1	1	Accumulator B
		5	Stack Pointer+3	1	0	1	1	Accumulator A
		6	Stack Pointer+4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer+5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer+6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer+7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		Sleep						
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address+1	1	0	1	0	Next Op Code

RELATIVE								
BCC	BCS	1	Op Code Address+1	1	0	1	1	Branch Offset
BEQ	BGE	2	FFFF	1	1	1	1	Restart Address (LSB)
BGT	BHI	3	Branch Address……Test="1"	1	0	1	0	First Op Code of Branch Routine
BLE	BLS		Op Code Address+1…Test="0"					Next Op Code
BLT	BMT							
BNE	BPL							
BRA	BRN							
BVC	BVS							
BSR		1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Branch Address	1	0	1	0	First Op Code of Subroutine

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 28, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD6301X0 as possible.

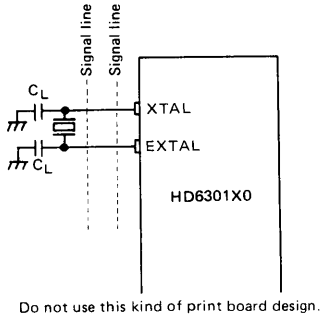


Figure 28 Precaution to the boad design of oscillation circuit

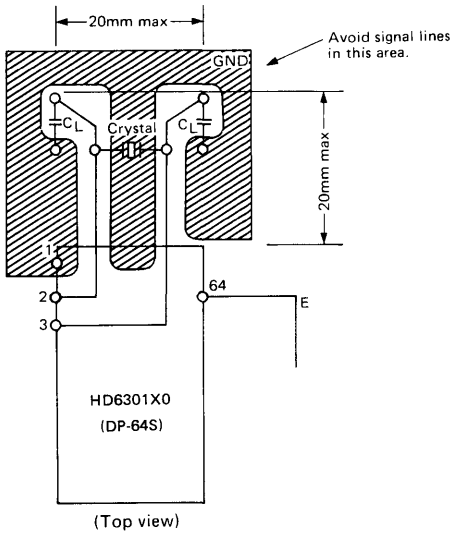


Figure 29 Example of Oscillation Circuits in Board Design

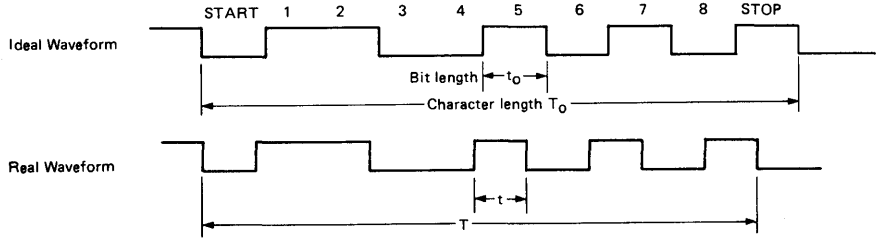
■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD6301X0 is shown in Table 20.

Note: SCI = Serial Communication Interface

Table 20

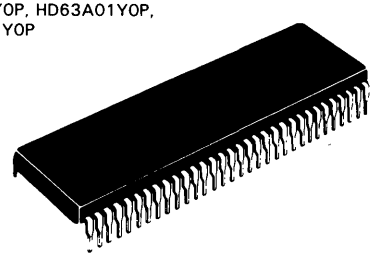
Bit distortion tolerance (t–to) /to	Character distortion tolerance (T–To) /To
±43.7%	±4.37%



HD6301Y0, HD63A01Y0, HD63B01Y0
CMOS MCU (Microcomputer Unit)

The HD6301Y0 is a CMOS 8-bit single-chip microcomputer unit which contains a CPU compatible with the CMOS 8-bit microcomputer HD6301V, 16k bytes of ROM, 256 bytes of RAM, 53 parallel I/O pins, Serial Communication Interface (SCI) and two timers.

HD6301Y0P, HD63A01Y0P, HD63B01Y0P

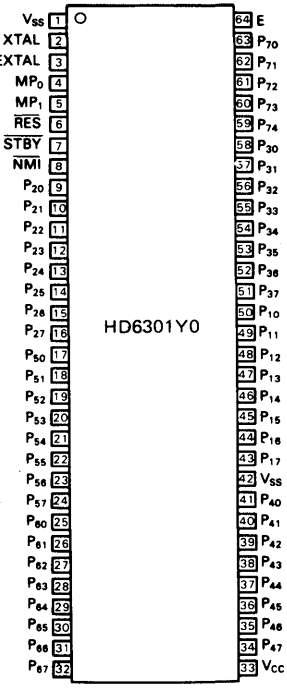


(DP-64S)

■ FEATURES

- Instruction Set Compatible with the HD6301V1
- 16k Bytes of ROM, 256 Bytes of RAM
- 53 Parallel I/O Pins
  (48 I/O Pins, 5 Output Pins)
- Parallel Handshake Interface (Port 6)
- Darlington Transistor Drive (Port 2, 6)
- 16-Bit Programmable Timer
  - Input Capture Register × 1
  - Free Running Counter × 1
  - Output Compare Register × 2
- 8-Bit Reloadable Timer
  - External Event Counter
  - Square Wave Generation
- Serial Communication Interface (SCI)
  - Asynchronous Mode (8 Transmit Formats, Hardware Parity)
  - Clocked Synchronous Mode
- Memory Ready
  - 3 Kinds of Memory Ready
- Halt
- Error Detection
  (Address Error, Op-code Error)
- Interrupt — External 3, Internal 7
- Operation Mode
  - Mode 1; Expanded Mode
    (Internal ROM Inhibited)
  - Mode 2; Expanded Mode
    (Internal ROM Valid)
  - Mode 3; Single Chip Mode
- Maximum 65K Bytes Address Space
- Low Power Dissipation Mode
  - Sleep Mode
  - Standby Mode (Hardware Standby, Software Standby)
- Minimum Instruction Execution Time — 0.5μs (f = 2MHz)
- Wide Range of Operation
  - V_{CC}=3 to 5.5V (f=0.1 to 0.5MHz)
  - V_{CC}=5V±10% { f=0.1 to 1.0MHz : HD6301Y0 / f=0.1 to 1.5MHz : HD63A01Y0 / f=0.1 to 2.0MHz : HD63B01Y0 }

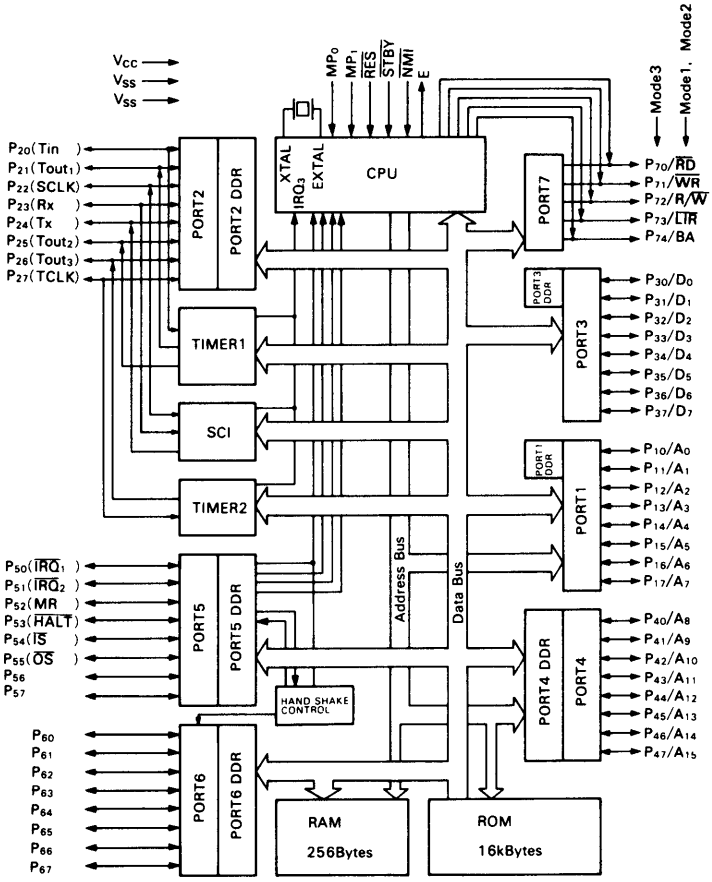
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BLOCK DIAGRAM**



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3~+7.0	V
Input Voltage	V_{in}	−0.3~V_{CC}+0.3	V
Operating Temperature	T_{opr}	0~+70	°C
Storage Temperature	T_{stg}	−55~+150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field.
But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out}: $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}−0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}×0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		−0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$, MP_0, MP_1	$\|I_{in}\|$	V_{in} = 0.5~V_{CC}−0.5V	–	–	1.0	μA
Three State Leakage Current	Ports 1, 2, 3, 4, 5, 6, 7	$\|I_{TSI}\|$	V_{in} = 0.5~V_{CC}−0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = −200μA	2.4	–	–	V
			I_{OH} = −10μA	V_{CC}−0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
Darlington Drive Current	Ports 2, 6	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Input Capacitance	All Inputs	C_{in}	V_{in} = 0V, f = 1MHz, Ta = 25°C	–	–	12.5	pF
Standby current	Non Operation	I_{STB}		–	3.0	15.0	μA
Current Dissipation*		I_{SLP}	Sleeping (f=1MHz**)	–	1.5	3.0	mA
			Sleeping (f=1.5MHz**)	–	2.3	4.5	mA
			Sleeping (f=2MHz**)	–	3.0	6.0	mA
		I_{CC}	Operating (f=1MHz**)	–	7.0	10.0	mA
			Operating (f=1.5MHz**)	–	10.5	15.0	mA
			Operating (f=2MHz**)	–	14.0	20.0	mA
RAM Standby Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC} − 1.0V, V_{IL} max = 0.8V (All output terminals are at no load.)

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at X MHz operation are decided according to the following formula:

typ. value (f = X MHz) = typ. value (f = 1MHz) × X
max. value (f = X MHz) = max. value (f = 1MHz) × X
(both the sleeping and operating)

● **AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, T_a = 0 ~ +70°C, unless otherwise noted.)**

BUS TIMING

Item		Symbol	Test Condition	HD6301Y0			HD63A01Y0			HD63B01Y0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time		t_{Er}		–	–	25	–	–	25	–	–	25	ns
Enable Fall Time		t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*		t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time	Write	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Set-up Time	Read	t_{DSR}		80	–	–	70	–	–	60	–	–	ns
Address, R/$\overline{W}$ Hold Time*		t_{AH}		80	–	–	50	–	–	40	–	–	ns
Data Hold Time	Write*	t_{HW}		80	–	–	50	–	–	40	–	–	ns
	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
$\overline{RD}$, $\overline{WR}$ Pulse Width*		PW_{RW}		450	–	–	300	–	–	220	–	–	ns
$\overline{RD}$, $\overline{WR}$ Delay Time		t_{RWD}		–	–	40	–	–	40	–	–	40	ns
$\overline{RD}$, $\overline{WR}$ Hold Time		t_{HRW}		–	–	20	–	–	20	–	–	20	ns
$\overline{LIR}$ Delay Time		t_{DLR}		–	–	200	–	–	160	–	–	120	ns
$\overline{LIR}$ Hold Time		t_{HLR}		10	–	–	10	–	–	10	–	–	ns
MR Set-up Time*		t_{SMR}	Fig. 2	400	–	–	280	–	–	230	–	–	ns
MR Hold Time*		t_{HMR}		–	–	100	–	–	70	–	–	50	ns
E Clock Pulse Width at MR		PW_{EMR}		–	–	9	–	–	9	–	–	9	μs
Processor Control Set-up Time		t_{PCS}	Fig. 3, 13, 14	200	–	–	200	–	–	200	–	–	ns
Processor Control Rise Time		t_{PCr}	Fig. 2, 3	–	–	100	–	–	100	–	–	100	ns
Processor Control Fall Time		t_{PCf}		–	–	100	–	–	100	–	–	100	ns
BA Delay Time		t_{BA}	Fig. 3	–	–	250	–	–	190	–	–	160	ns
Oscillator Stabilization Time		t_{RC}	Fig. 14	20	–	–	20	–	–	20	–	–	ms
Reset Pulse Width		PW_{RST}		3	–	–	3	–	–	3	–	–	t_{cyc}

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD6301Y0			HD63A01Y0			HD63B01Y0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set Up Time	Port 1, 2, 3, 4, 5, 6	t_{PDSU}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4, 5, 6	t_{PDH}		200	–	–	200	–	–	200	–	–	ns
Delay Time (From Enable Fall Edge to Peripheral Output)	Port 1, 2, 3, 4, 5, 6, 7	t_{PWD}	Fig. 6	–	–	300	–	–	300	–	–	300	ns
Input Strobe Pulse Width		t_{PWIS}	Fig. 10	200	–	–	200	–	–	200	–	–	ns
Input Data Hold Time	Port 6	t_{IH}		150	–	–	150	–	–	150	–	–	ns
Input Data Set-Up Time	Port 6	t_{IS}		100	–	–	100	–	–	100	–	–	ns
Output Strobe Delay Time		t_{OSD1}	Fig. 11	–	–	200	–	–	200	–	–	200	ns
		t_{OSD2}											

TIMER, SCI TIMING

Item		Symbol	Test Condition	HD6301Y0			HD63A01Y0			HD63B01Y0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 9	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Output)		t_{TOD}	Fig. 7, 8	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Async. Mode	t_{Scyc}	Fig. 9	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Clock Sync.		Fig. 4	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Clock Sync. Mode)		t_{TXD}	Fig. 4	–	–	220	–	–	220	–	–	220	ns
SCI Receive Data Set-up Time (Clock Sync. Mode)		t_{SRX}		260	–	–	260	–	–	260	–	–	ns
SCI Receive Data Hold Time (Clock Sync. Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 9	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}		200	–	–	200	–	–	200	–	–	ns
Timer 1·2, SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1·2, SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

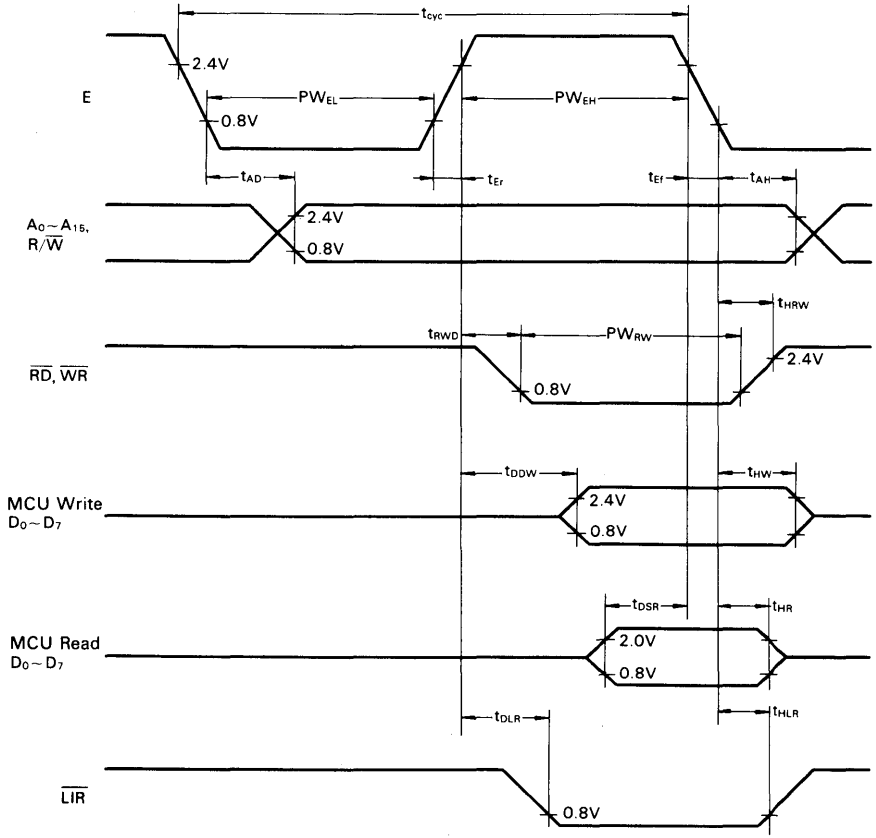


Figure 1 Mode 1, Mode 2 Bus Timing

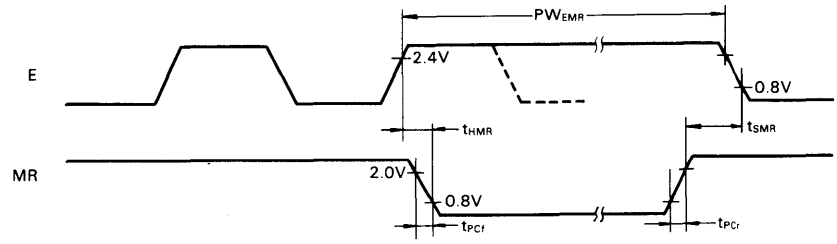


Figure 2 Memory Ready and E Clock Timing.

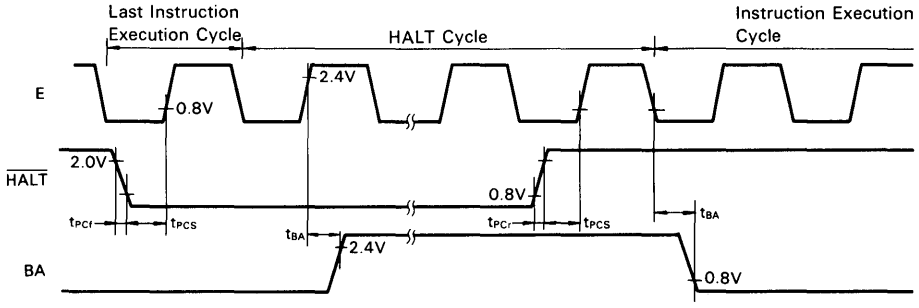
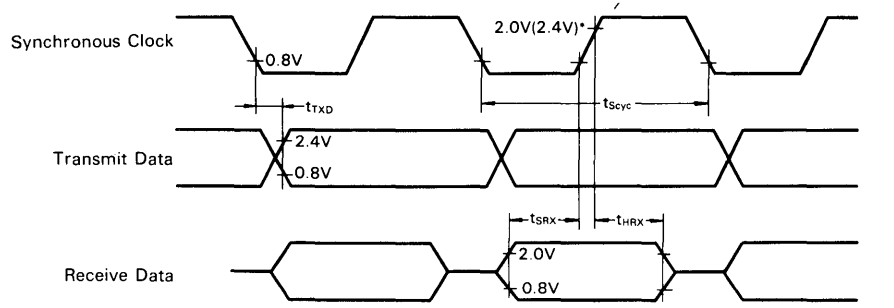


Figure 3 $\overline{\text{HALT}}$ and BA Timing



*2.0V is high level when clock input.
2.4V is high level when clock output.

Figure 4 SCI Clocked Synchronous Timing

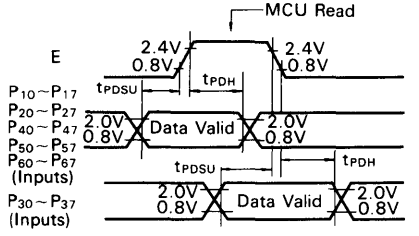


Figure 5 Port Data Set-up and Hold Times (MCU Read)

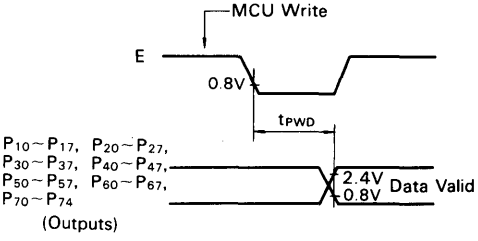


Figure 6 Port Data Delay Times (MCU Write)

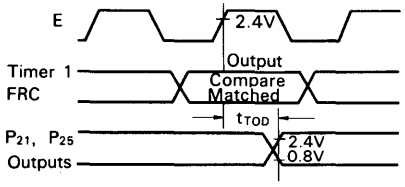
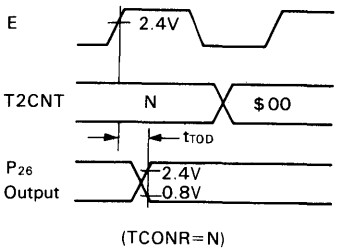


Figure 7 Timer 1 Output Timing



(TCONR=N)

Figure 8 Timer 2 Output Timing

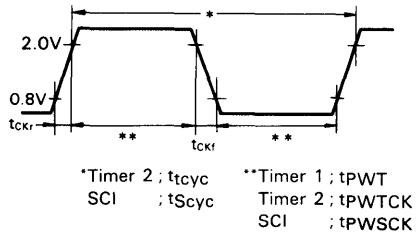


Figure 9 Timer 1·2, SCI Input Clock Timing

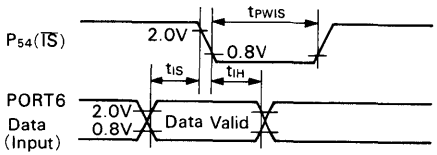


Figure 10 Port 6 Input Latch Timing

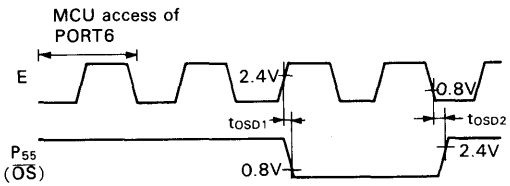
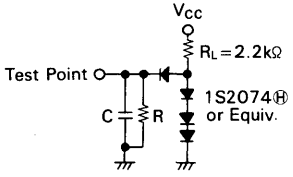


Figure 11 Output Strobe Timing



C=90pF for Port 1, Port 3, Port 4, E
=30pF for Port 2, Port 5, Port 6, Port 7
R=12kΩ for Port 1 ~ Port 7, E

Figure 12 Bus Timing Test Loads (TTL Load)

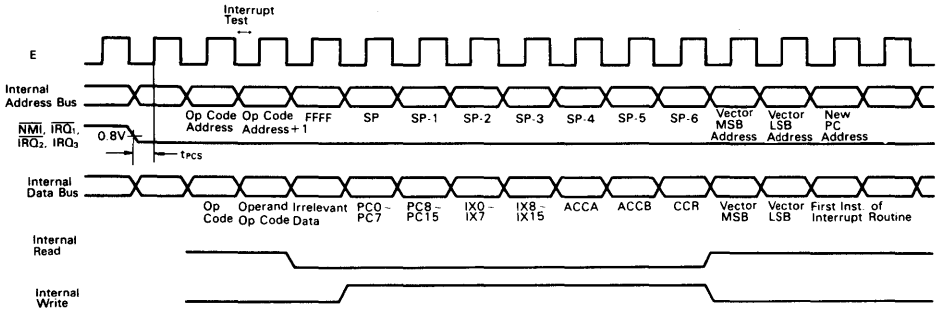


Figure 13 Interrupt Sequence

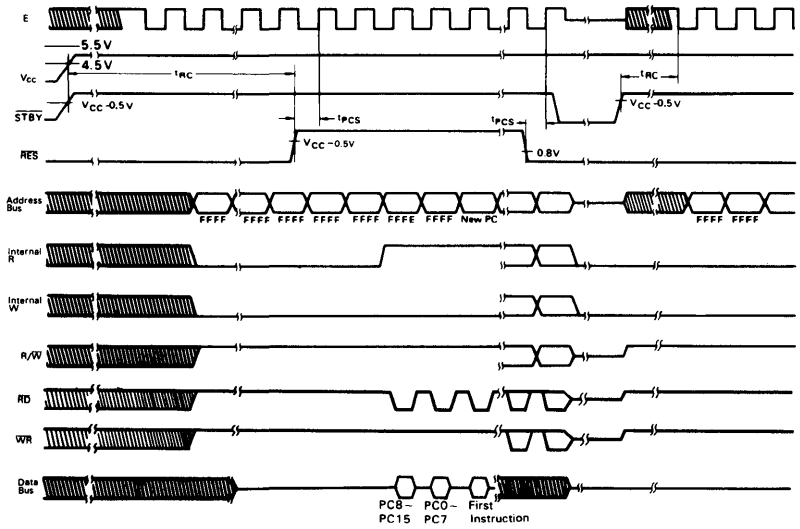


Figure 14 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

V_{CC} and V_{SS} provide power to the MCU with 5V±10% supply. In the case of low speed operation (fmax=500kHz), the MCU can operate with 3 to 5.5 volts. Two V_{SS} pins should be tied to ground.

• XTAL, EXTAL

These two pins interface with an AT-cut parallel resonant crystal. Divide-by-four circuit is on chip, so if 4MHz crystal oscillator is used, the system clock is 1MHz for example.

EXTAL pin can be drived by the external clock with 45% to 55% duty. The system clock which is one fourth frequency of the external clock is generated in the LSI. The external clock frequency should be less than four times of the maximum operating frequency. When using the external clock, XTAL pin should be open. Fig. 15 shows examples of connection circuit. The crystal and C_{L1}, C_{L2} should be mounted as close as possible to XTAL and EXTAL pins. Any line must not cross the line between the crystal oscillator and XTAL, EXTAL.

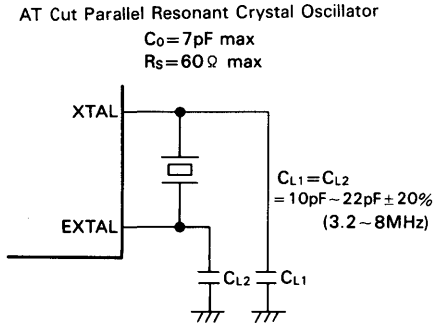


Figure 15 Connection Circuit

• STBY

This pin makes the MCU standby mode. In "Low" level, the oscillation stops and the internal clock is stabilized to make reset condition. To retain the contents of RAM at standby mode, "0" should be written into RAM enable bit (RAME). RAME is the bit 6 of the RAM/port 5 control register at $0014. RAM is disabled by this operation and its contents is sustained.

Refer to "LOW POWER DISSIPATION MODE" for the standby mode.

• Reset (RES)

This pin resets the MCU from power OFF state and provides a startup procedure. During power-on, RES pin must be held "Low" level for at least 20ms.

The CPU registers (accumulator, index register, stack pointer, condition code register except for interrupt mask bit), RAM and the data register of ports are not initialized during reset, so their contents are undefined in this procedure.

To reset the MCU during operation, RES should be held "Low" for at least 3 system-clock cycles. At the 3rd cycle during "Low" level, all the address buses become "High". When RES remains "Low", the address buses keep "High". If RES becomes "High", the MCU starts the next operation.

(1) Latch the value of the mode program pins; MP_0 and MP_1.
(2) Initialize each internal register (Refer to Table 6).
(3) Set the interrupt mask bit. For the CPU to recognize the maskable interrupts $\overline{IRQ_1}$, $\overline{IRQ_2}$ and IRQ_3, this bit should be cleared in advance.
(4) Put the contents (=start address) of the last two addresses ($FFFE, $FFFF) into the program counter and start the program from this address. (Refer to Table 1).

• Enable (E)

This pin provides a TTL-compatible system clock to external circuits. Its frequency is one fourth that of the crystal oscillator or external clock. This pin can drive one TTL load and 90pF capacitance.

• Non-Maskable Interrupt (NMI)

When the falling edge of the input signal is detected at this pin, the CPU begins non-maskable interrupt sequence internally. As well as the $\overline{IRQ}$ mentioned below, the instruction being executed at $\overline{NMI}$ signal detection will proceed to its compeletion. The interrupt mask bit of the condition code register doesn't affect non-maskable interrupt at all.

In response to an $\overline{NMI}$ interrupt, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack. Upon completion of this sequence, a vector is fetched from $FFFC and $FFFD to transfer their contents into the program counter and branch to the non-maskable interrupt service routine.

(Note) At reset start, the stack pointer should be initialized on an appropriate memory area and then the falling edge be input to $\overline{NMI}$ pin.

- **Interrupt Request ($\overline{IRQ_1}$, $\overline{IRQ_2}$)**

These are level-sensitive pins which request an internal interrupt sequence to the CPU. At interrupt request, the CPU will complete the current instruction before the acceptance of the request. Unless the interrupt mask in the condition code register is set, the CPU starts an interrupt sequence; if set, the interrupt request will be ignored. When the sequence starts, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack, then the CPU sets the interrupt mask bit and will not acknowledge the maskable request. During the last cycle, the CPU fetches vectors depicted in Table 1 and transfers their contents to the program counter and branches to the service routine.

The CPU uses the external interrupt pins ($\overline{IRQ_1}$ and $\overline{IRQ_2}$) also as port pins P_{50} and P_{51}, so it provides an enable bit to Bit 0 and 1 of the RAM port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for the details.

When one of the internal interrupts, ICI, OCI, TOI, CMI or SIO is generated, the CPU produces internal interrupt signal (IRQ_3). IRQ_3 functions just the same as $\overline{IRQ_1}$ or $\overline{IRQ_2}$ except for its vector address. Fig. 16 shows the block diagram of the interrupt circuit.

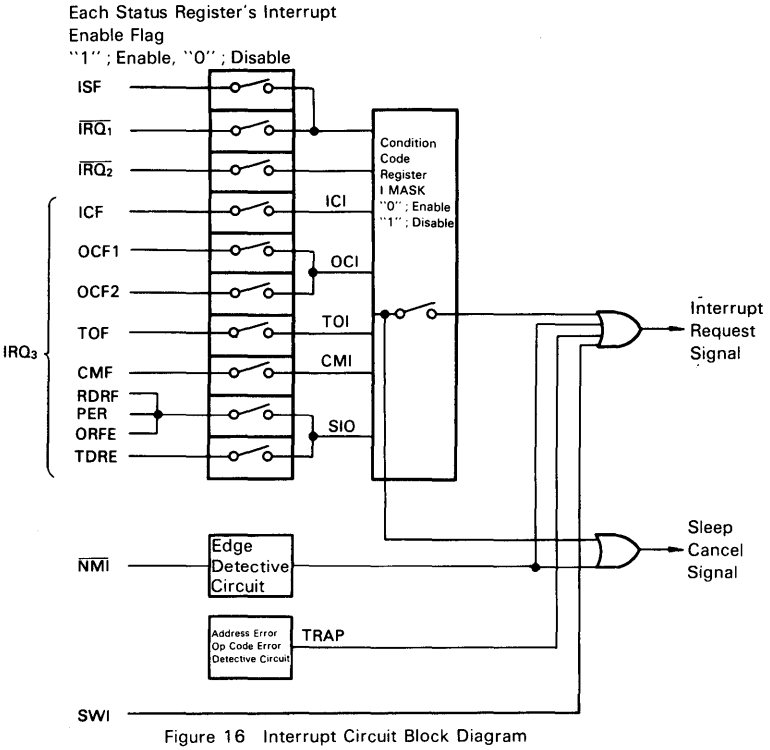


Figure 16 Interrupt Circuit Block Diagram

Table 1 Interrupt Vector Memory Map

Priority	Vector MSB	Vector LSB	Interrupt
Highest	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	$\overline{IRQ_1}$, ISF (port 6 Input Strobe)
	FFF6	FFF7	ICI (Timer 1 Input Capture)
	FFF4	FFF5	OCI (Timer 1 Output Compare 1, 2)
	FFF2	FFF3	TOI (Timer 1 Overflow)
	FFEC	FFED	CMI (Timer 2 Counter Match)
	FFEA	FFEB	$\overline{IRQ_2}$
Lowest	FFF0	FFF1	SIO (RDRF + ORFE + TDRE + PER)

- **Mode Program (MP_0, MP_1)**

These two pins decide the operation mode. Refer to "MODE SELECTION" for more details.

The following signal descriptions are applicable only for the expanded mode.

- **Read/Write (R/$\overline{W}$; P_{72})**

This signal, usually be in read state ("High"), shows whether the CPU is in read ("High") or write ("Low") state to the peripheral or memory devices. This can drive one TTL load and 30pF capacitance.

- **$\overline{RD}$, $\overline{WR}$ (P_{70}, P_{71})**

These signals show active low outputs when the CPU is reading/writing to the peripherals or memories. This enables the CPU easy to access the peripheral LSI with $\overline{RD}$ and $\overline{WR}$ input pins. These pins can drive one TTL load and 30pF capacitance.

- **Load Instruction Register ($\overline{LIR}$; P_{73})**

This signal shows the instruction opecode being on data bus (active low). this pin can drive one TTL load and 30pF capacitance.

- **Memory Ready (MR; P_{52})**

This is the input control signal which stretches the system clock's "High"period to access low-speed memories. HD6301Y0 can select three kinds of low-speed memory access method by RAM/Port 5 Control Register's MRE bit and AMRE bit. In the case that CPU accesses low-speed memories by the external MR signal (MRE="1", AMRE="0"), the system clock operates in normal sequence when this signal is in "High".

But this signal in "Low", the "High" period of the system clock will be stretched depending on its "Low" level duration in integral multiples of the cycle time. This allows the CPU to interface with low-speed memories (See Fig. 2). Up to 9μs can be stretched.

During internal address space access or nonvalid memory access, MR is prohibited internally to prevent decrease of operation speed. Even in the halt state, MR can also stretch "High" period of system clock to allow peripheral devices to access low-speed memories. Refer to "RAM/PORT 5 CONTROL REGISTER" for more details.

- **Halt ($\overline{HALT}$; P_{53})**

This is an input control signal to stop instruction execution and to release buses. When this signal switches to "Low", the CPU stops to enter into the halt state after having executed the present instruction. When entering into the halt state, it makes BA (P_{74}) "High" and also an address bus, data bus, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ high impedance. When an interrupt is generated in the halt state, the CPU uses the interrupt handler after the halt is cancelled.

(Note) Please don't switch the $\overline{HALT}$ signal to "Low" when the CPU executes the WAI instruction and is in the interrupt wait state to avoid the trouble of the CPU's operation after the halt is cancelled.

- **Bus Available (BA; P_{74})**

This is an output control signal which is normally "Low" but "High" when the CPU accepts $\overline{HALT}$ and releases the buses. The HD6800 and HD6802 make BA "High" and release the buses at WAI execution, while the HD6301Y0 doesn't make BA "High" under the same condition.

- **PORT**

The HD6301Y0 provides seven I/O ports. Port 1, 2, 3, 4, 5, and 6 are 8-bit I/O ports. Each port provides Data Direction Register(DDR). Port 1 and port 3 select the I/O state by the byte and port 2, 4, 5 and 6 the I/O state by the bit. Port 7 is a 5-bit output-only port. In the expanded mode (mode 1, mode 2), port 3 becomes data buses, port 1 and port 4 address buses and port 7 control signal pins.

Table 2 Port and Data Direction Register Address

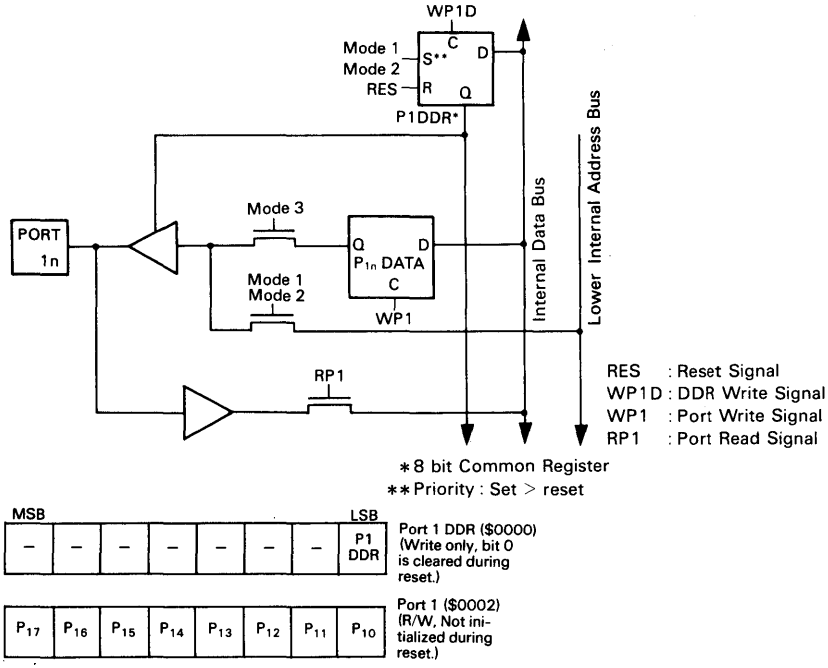
Port	Port Address	Data Direction Register
Port 1	$0002	$0000
Port 2	$0003	$0001
Port 3	$0006	$0004
Port 4	$0007	$0005
Port 5	$0015	$0020
Port 6	$0017	$0016
Port 7	$0018	

- **Port 1**

An 8-bit I/O port. The DDR of port 1 (P1DDR) controls the I/O state. It provides a bit which select the I/O state by the byte ("0" for input and "1" for output).

As it is cleared during reset, port 1 is an input port.

In the expanded mode (mode 1, mode 2), port 1 functions as a lower address buses (A_0 to A_7). Port 1 can drive one TTL load and 90pF capacitance.



- **Port 2**

An 8-bit I/O port. Port 2 DDR (P2DDR) controls the I/O state. This port provides DDR corresponding to each bit and can define input or output by the bit ("0" for input, "1" for output).

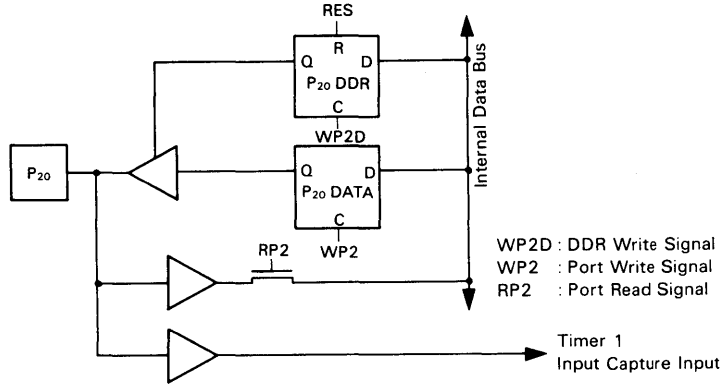
As Port 2 DDR is cleared during reset, it will be an input port.

Port 2 is also used as an I/O pin for timer 1, Timer 2 and the SCI. Pins for Timers and the SCI set or reset each DDR depending on their functions and become I/O pins. When port 2 functions as an I/O port after used as I/O pins of the timers or the SCI, the I/O direction of the pins remain as it is used as the I/O pin of timer and SCI.

Port 2 can drive one TTL load and 30pF capacitance. This port can produce 1mA when V_{out}=1.5V to drive directly the base of Darlington transistor.

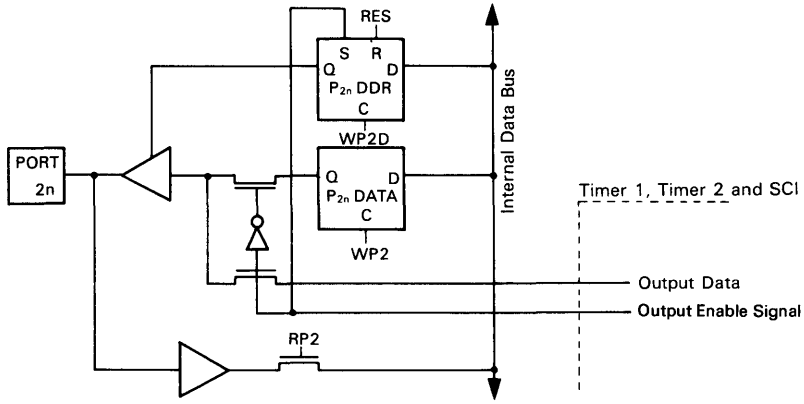
P_{20} (Tin)

P_{20} is also used as an external input pin for the input-capture. This pin is an I/O port which is an input or output as defined by the Data Direction Register (P_{20}DDR) ("0" for an input and "1" for an output). Then either a signal to or from P_{20} ("to" for an output port, "from" for an input port) is always input to the Timer 1 input capture.



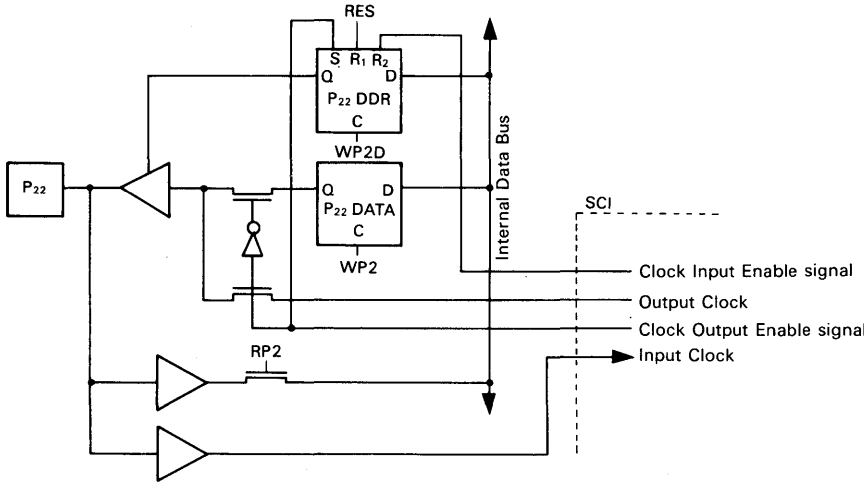
P_{21} (Tout 1), P_{24} (Tx), P_{25} (Tout 2), P_{26} (Tout 3)

These four pins can be also used as output pins for Timer 1, Timer 2 and a transmit output of the SCI. Timer 1, and the SCI have a register which enables output. By setting these registers, they automatically will be output pins of timer or the SCI.



P_{22} (SCLK)

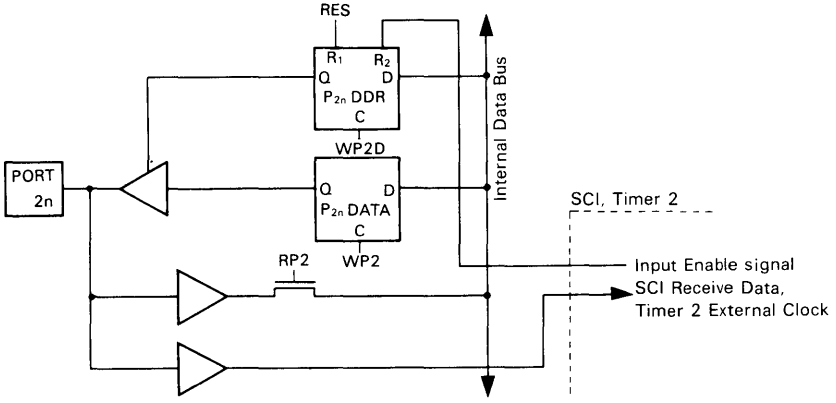
P_{22} is also used as a clock I/O pin for the SCI. It is selected as a clock input or output pin by the operating mode of the SCI. It is usable as an I/O port when the SCI has no clock input or output (as an output port if P_{22} DDR = 1, as an input port if P_{22} DDR = 0).

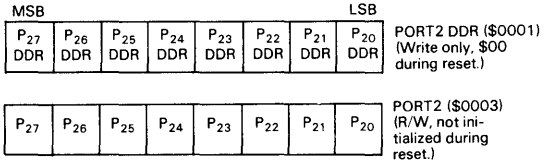


P_{23} (Rx), P_{27} (TCLK)

P_{23} and P_{27} are also used as received data input pins for the SCI and external clock input pins for Timer 2. The SCI and Timer 2 have registers which enable input. If the registers are set, the DDR (P_{23}DDR, P_{27}DDR) are cleared and P_{23} and P_{27} will be input pin for Rx and TCLK.

Since the SCI will be a clocked synchronous mode by an external clock-input during reset, the DDR of P_{22} is cleared automatically and P_{22} is an input port. Set the SCI to a mode where P_{22} is not used (CC0 or CC1 of the RMC Register is "0" or "1" respectively) and write "1" to the P_{22} DDR to make P_{22} an output port.



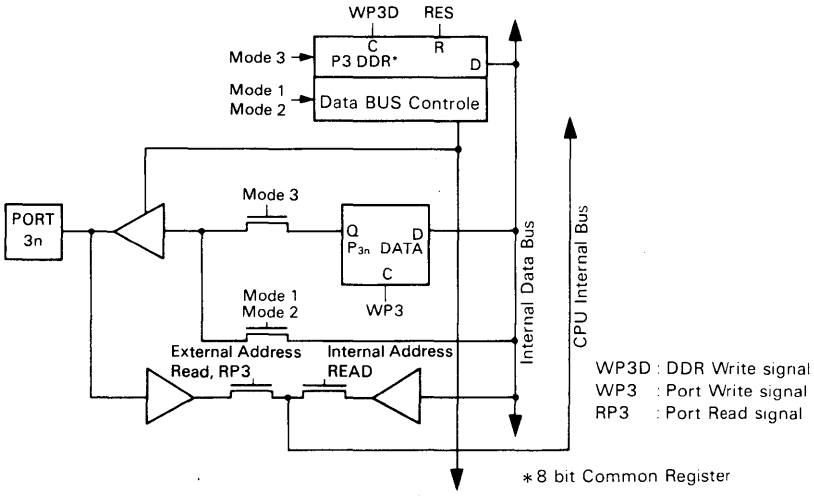


• Port 3

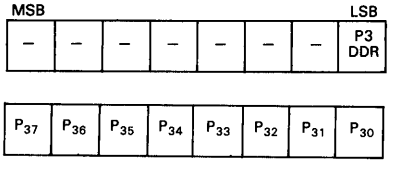
An 8-bit I/O port. The DDR of port 3 controls the I/O state. It provides only one bit which defines I/O state by the byte ("0" for input and "1" for output).

During reset, it is cleared and port 3 becomes an input port.

In the expanded modes (Mode 1, Mode 2), port 3 functions as data buses (D_0 to D_7). Port 3 can drive one TTL load and 90pF capacitance.



Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

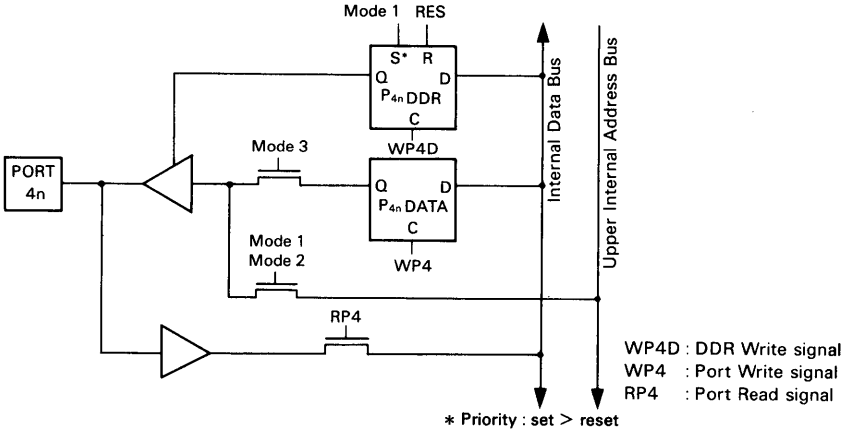


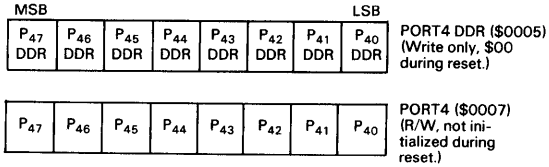
- **Port 4**

An 8-bit I/O port. The DDR of port 4 controls I/O state. Each bit of port 4 has a DDR which defines I/O state ("0" for input and "1" for output).

During reset, the DDR of port 4 is cleared and port 4 becomes an input port.

In the expanded modes (Mode 1, Mode 2), port 4 functions as address buses (A_8 to A_{15}). In Mode 1 (expanded mode with no internal ROM), the DDR is set automatically and outputs addresses. But in Mode 2 (expanded mode with internal ROM), the DDR is cleared and port 4 becomes an input port during reset. Set the DDR to "1" to output the upper addresses (A_8 to A_{15}) to the outside. If not all of the upper addresses have to be output, the pins which don't output addresses can be used as input pins. Port 4 can drive one TTL load and 90pF capacitance.





- **Port 5**

An 8-bit I/O port. The DDR of port 5 controls I/O state. Each bit of port 5 has a DDR which defines I/O state ("0" for input and "1" for output).

During reset, the DDR of port 5 is cleared and port 5 becomes an input port.

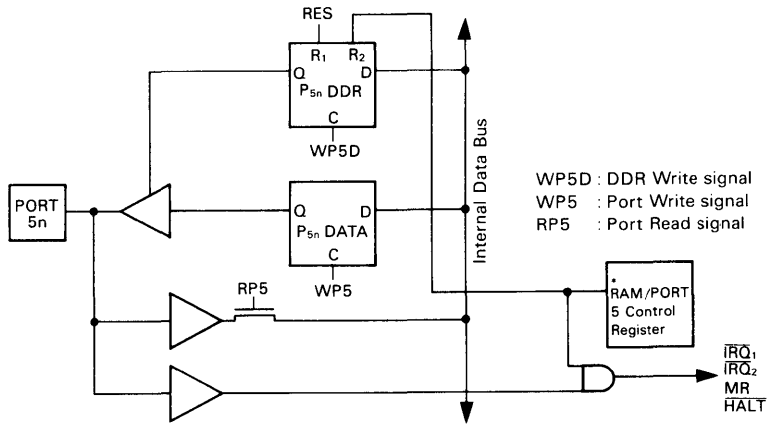
Port 5 is also usable as $\overline{IRQ_1}$, $\overline{IRQ_2}$, $\overline{HALT}$, MR and the strobed signal of port 6 for handshake ($\overline{IS}$, $\overline{OS}$). It is set to input or output automatically if it is used as these control signal pins (except P_{54}, IS). Since the DDR of port 5, as is port 2, is set or reset by the control signal, I/O directions of the I/O ports are retained after the control signal is disabled. Port 5 can drive one TTL load and 90pF capacitance.

P_{50} ($\overline{IRQ_1}$), P_{51} ($\overline{IRQ_2}$)

P_{50} and P_{51} are also usable as interrupt pins. The RAM/port 5 control registers of $\overline{IRQ_1}$ and $\overline{IRQ_2}$ have enable bits (IQ1E, IQ2E). When these bits are set to "1", P_{50} and P_{51} will automatically be interrupt input pins.

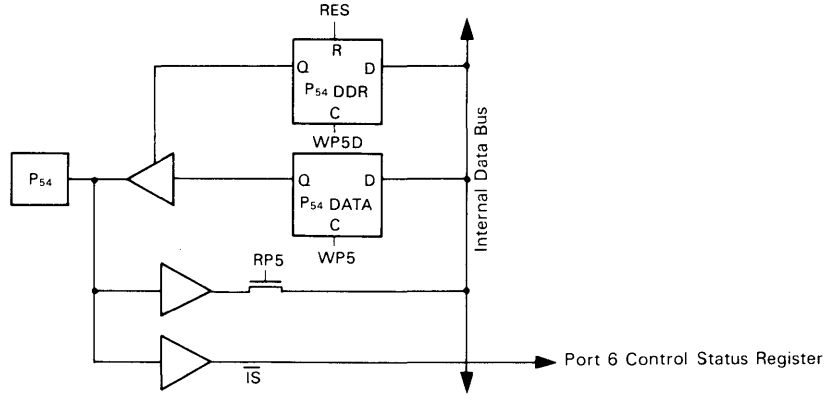
P_{52} (MR), P_{53} ($\overline{HALT}$)

P_{52} and P_{53} are also usable as MR and $\overline{HALT}$ inputs. MR and $\overline{HALT}$ have enable bits (MRE, HLTE) in the RAM/Port 5 Control Register as $\overline{IRQ_1}$ and $\overline{IRQ_2}$. In the single chip mode (Mode 3), P_{52} and P_{53} are usable as I/O ports regardless of the value of the enable bits. In the expanded mode (Mode 1 or Mode 2), since MRE is cleared during reset, P_{52} is usable as an I/O port. Since HLTE is set during reset, the DDR of P_{53} will be automatically reset to be a $\overline{HALT}$ input pin. HLTE of the RAM/Port 5 Control Register has to be cleared to use P_{53} as an I/O port.

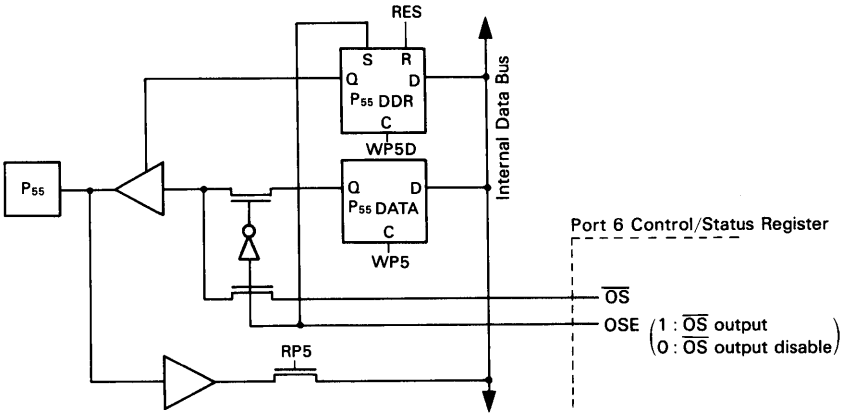


* Initializing value during reset;
IRQ1E = "0", IRQ2E = "0", MRE = "0"**, HLTE = "1"**
** P_{52} and P_{53} can be used as I O ports
in spite of the value of this register in Mode 3.

P_{54} ($\overline{IS}$)
P_{54} is also usable as the input strobe ($\overline{IS}$) for port 6 handshake interface. This pin, as is P_{20}, is always an I/O port. If P_{54} is used as an output port (set the DDR of P_{54} to "1"), an output signal from P_{54} will be the input to $\overline{IS}$.

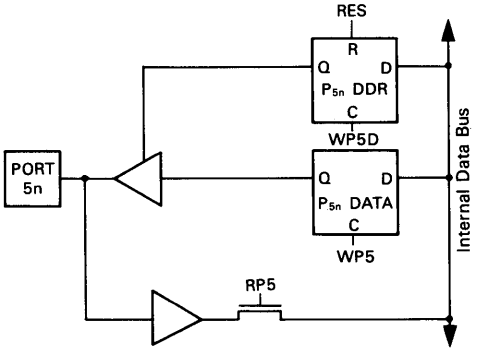


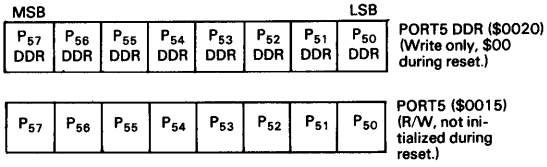
P_{55} ($\overline{OS}$)
P_{55} is also usable as the output strobe ($\overline{OS}$) for port 6 handshake interface. It will be an I/O port during reset, and an $\overline{OS}$ output pin by setting the $\overline{OS}$ enable register (OSE) of the port 6 Control Status Register (P6CSR).



P_{56}, P_{57}

P_{56} and P_{57} are I/O ports.



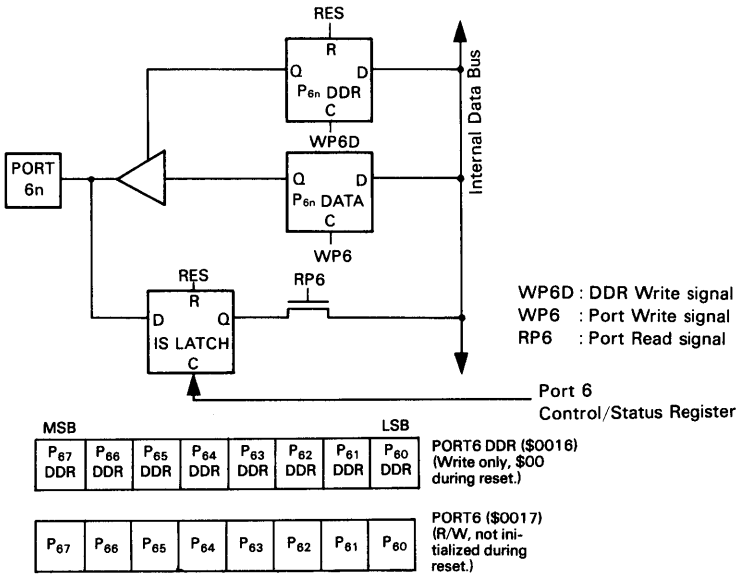


- **Port 6**

8-bit I/O port. Port 6 DDR controls I/O state. Each bit of port 6 has a DDR and designates input or output ("0" for input, "1" for output). During reset, Port 6 DDR is cleared and port 6 becomes an input port.

Port 6 controls parallel handshake interface besides functions as an I/O port. Therefore, it provides DDRs to control and IS LATCH to latch the input data.

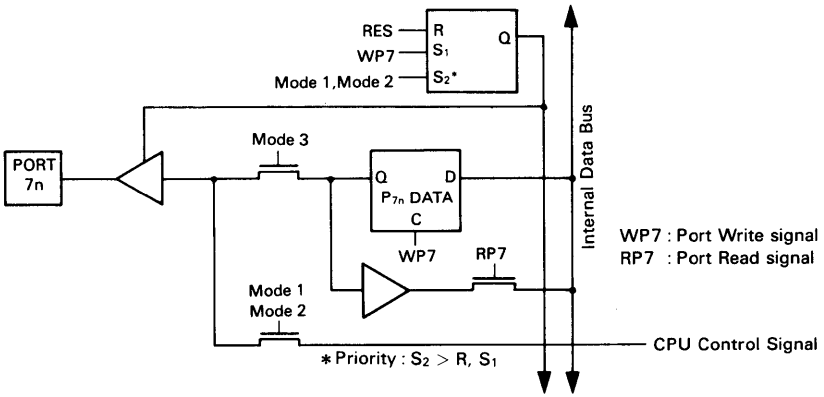
Port 6 can drive one TTL load and 30pF capacitance. It can drive directly the base of Darlington transistor as port 2.



- **Port 7**

A 5-bit output port. In single-chip mode (Mode 3), port 7 goes to a high impedance state during reset. By a write to Port 7, Port 7 goes to the output state from the high impedance state, and it outputs the written data. Once it becomes output state, Port 7 functions as an output port. CPU 7 can also read the Port 7 data register to execute bit manipulation instruction. In the expanded mode (Mode 1, Mode 2), Port 7 is an output pin for control signals ($\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA) from the CPU.

Port 7 can drive one TTL load and 30pF capacitance.



■ RAM/PORT 5 CONTROL REGISTER

The control register located at $0014 controls on-chip RAM and port 5.

RAM/Port 5 Control Register (RP5CR)

7	6	5	4	3	2	1	0	
STBY PWR	RAME	STBY FLAG	AMR E	HLTE	MRE	IRQ_2 E	IRQ_1 E	$0014

Bit 0, Bit 1 $\overline{IRQ_1}$, $\overline{IRQ_2}$ Enable Bit (IRQ_1E, IRQ_2E)

When using P_{50} and P_{51} as interrupt pins, write "1" in these bits.

When the bit is set to "1", the DDRs corresponding to P_{50} and P_{51} are cleared and become $\overline{IRQ_1}$ input pin and $\overline{IRQ_2}$ input pin. When IRQ_1E and IRQ_2E are set, P_{50} and P_{51} cannot be used as an output ports. When "0", the CPU doesn't accept an external interrupt or a sleep cancellation by the external interrupt. These bits are cleared during reset.

Bit 2 Memory Ready Enable Bit (MRE)

When using P_{52} as an input pin of the "memory ready" signal, write "1" in this bit. When set, P_{52} DDR is automatically cleared and becomes the MR input pin. In Mode 3, however, the "memory ready" is inhibited regardless of the bit. The bit is cleared during reset.

Bit 3 Halt Enable Bit (HLTE)

When using P_{53} as an input pin of the $\overline{HALT}$ signal, write "1" in this bit. When this bit is set, P_{53} DDR is automatically cleared and becomes the Halt input pin. If the bit is "0", the Halt function is inhibited and P_{53} is used as an I/O port. The bit is set to "1" during reset. However, in Mode 3, Halt function is inhibited regardless of the bit.

(Note) When using P_{52} and P_{53} as the input ports in mode 1 and 2, MRE and HLTE bit should be cleared just after the reset.

Bit 4 Auto Memory Ready Enable Bit (AMRE)

When the bit is set and the CPU accesses the external address, "memory ready" operates automatically and stretches the E clock's "High" duration for one system clock. When MRE bit of bit 2 is cleared and when the CPU accesses the external address space, the function operates. When MRE bit is set and then the CPU accesses the external address space with P_{52}(MR) pin in "low", "memory ready" operates automatically. In Mode 3, regardless of the bit value, the "auto memory ready" function is inhibited. (See Table 3 and Fig. 17.)

(Note) Since this bit is set to "1" during reset, clear the bit at the beginning of the program when auto memory ready doesn't have to operate.

Table 3 "Memory Ready" Function

MRE	AMRE	Function
0	0	"Memory ready" inhibited.
0	1	When the CPU accesses the external address, "High" duration of E clock automatically becomes one-cycle longer. This state is retained during reset.
1	0	"Memory ready" operates by P_{52} (MR) pin. The function is the same as that of the HD6301X0.
1	1	When the CPU accesses the external address space with the P_{52} (MR) pin in "low", the "auto memory ready" operates. This function is effective if it has both "high-speed memory" and "slow memory" outside. Input $\overline{CS}$ signal of "slow memory" to MR pin.

Bit 5 Standby Flag (STBY FLAG)

By clearing this flag, HD6301Y0 gets into the standby mode by software. This flag is set to "1" during reset, so the standby mode is canceled with $\overline{RES}$ pin in "low". The $\overline{RES}$ pin should be in "low" until oscillation becomes stable (min. 20ms.). If the $\overline{STBY}$ pin in is in "low", the standby mode can not be canceled with the $\overline{RES}$ pin in "low".

Bit 6 RAM Enable (RAME)

On-chip RAM can be disabled by this control bit. By resetting the MCU, "1" is set to this bit, and on-chip RAM is enabled. When this bit is cleared (=logic "0") on-chip RAM is invalid and the CPU can read data from external memory. This bit should be "0" before getting into the standby mode to protect on-chip RAM data.

Bit 7 Standby Power Bit (STBY PWR)

When V_{CC} is not provided in standby mode, this bit is cleared. This is a flag for read/write and can be read by software. If this bit is set before standyby mode, and remains set even after returning from standby mode, V_{CC} voltage is provided during standby mode and the on-chip RAM data is valid.

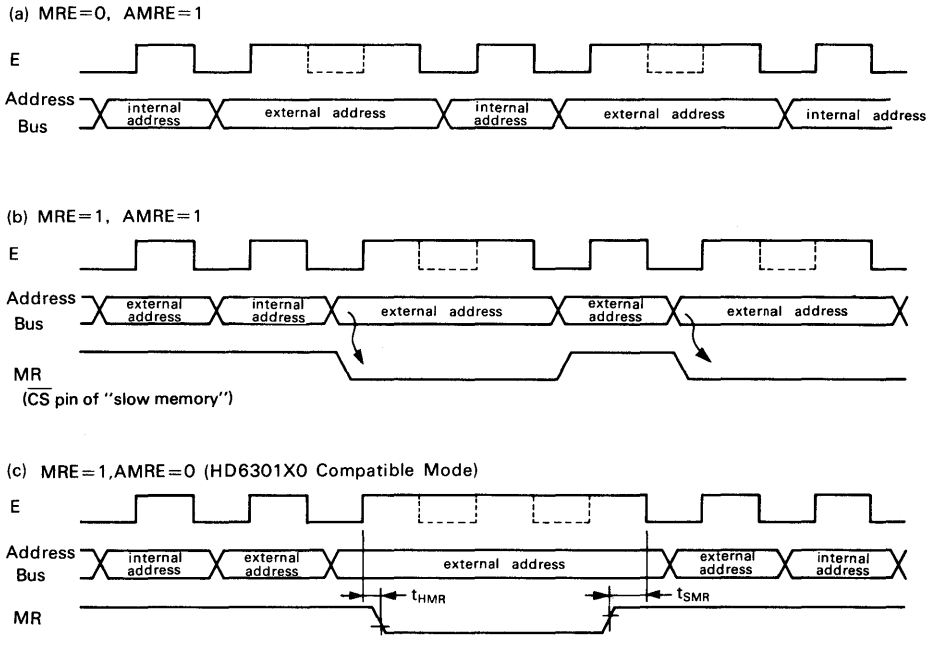


Figure 17 Memory Ready Timing

■ **Port 6 Control/Status Register**

This is the Control/Status Register for parallel handshake interface using Port 6. The functions are as follows;

1) Latches input data to Port 6 at the $\overline{IS}$ (P_{54}) falling edge.
2) Outputs a strobe signal $\overline{OS}$ (P_{55}) outward by reading or writing to port 6.
3) When IS FLAG is set at the $\overline{IS}$ falling edge, an interrupt occurs.

The following shows Port 6 Control/Status Register (P6CSR).

7	6	5	4	3	2	1	0	
IS* FLAG	IS IRQ$_1$ ENABLE	OSE	OSS	LATCH ENABLE	–	–	–	$0021

*Bit 7 is Read-Only bit

Bit 0
Bit 1 Not used.
Bit 2

Bit 3: Latch Enable

This register controls the input latch for Port 6 (ISLATCH). When this bit is set to "1", the input data to port 6 will be latched inward at the $\overline{IS}$ (P_{54}) falling edge. An input latch will be canceled by reading Port 6, which enables to latch the next data. If cleared, the input latch remains canceled and this bit functions as a usual I/O port. This bit is cleared during reset.

Bit 4: OSS Output Strobe Select

This register initiates an output strobe ($\overline{OS}$) from P_{55} by reading or writing to port 6. When cleared, $\overline{OS}$ occurs by reading Port 6. When set, $\overline{OS}$ occurs by writing to Port 6. This bit is cleared during reset.

Bit 5: OSE Output Strobe Enable

This register decides the enabling or disabling of the output strobe. When cleared, P_{55} functions as an I/O port. When set, P_{55} functions as an $\overline{OS}$ output pin. (P_{55} DDR is set by OSE.) This bit is cleared during reset.

Bit 6: IS IRQ$_1$ Enable Input Strobe Interrupt Enable

When set, an $\overline{IRQ_1}$ interrupt to the CPU occurs by setting IS FLAG of bit 7. When cleared, the interrupt does not occur. This bit is cleared during reset.

Bit 7: IS Flag Input Strobe Flag

This flag is set at the IS (P_{54}) falling edge. This flag is for read-only. When set, the flag is cleared by reading or writing to Port 6 after reading the Port 6 Control Status Register. This bit is cleared during reset.

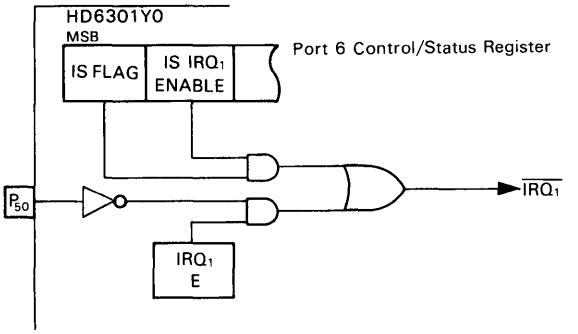


Figure 18 Input Strobe Interrupt block Diagram

■ MODE SELECTION

Mode program pins, MP_0 and MP_1 determine the operation mode of the HD6301Y0 as Table 4 shows.

• Mode 1 (Expanded Mode)

In this mode, port 3 is data bus and port 1 "Lower" address bus and port 4 "Upper" address bus to interface directly with the HMCS6800 buses. A control signal such as R/$\overline{W}$ is produced at port 7. In mode 1, on-chip ROM is disabled and 65k bytes of address space are externally expandable (refer to Fig. 19).

• Mode 2 (Expanded Mode)

This mode is also expandable as well as mode 1. But in this mode, on-chip ROM is enabled and the expandable address space is 48k bytes (refer to Fig. 20).

In Mode 2, port 4 is available as an input port during reset, and so the upper address is not output outwards. After reset starts, set the P4DDR corresponding to the external address output. By setting the DDR, the upper address is output. When a small external memory space is provided, the pin not required to output the address externally can be used as the input port

• Mode 3 (Single-chip Mode)

In this mode, all ports are available (refer to Fig. 21).

Table 4 Mode Selection

Mode	MP_1	MP_0	ROM	RAM	Interrupt Vector	Operation Mode
1	"L"	"H"	E	I*	E	Expanded Mode
2	"H"	"L"	I	I*	I	Expanded Mode
3	"H"	"H"	I	I	I	Single-chip Mode

"L" = Logic "0", "H" = Logic "1", I; Internal, E; External.
* The addressing RAM area can be external by clearing RAME bit $0014.

• Mode and Port

Table 5 shows MCU signals in each mode.

Table 5 MCU Signals in Each Mode

Port \ Mode	Mode 1	Mode 2	Mode 3
Port 1	Address Bus (A_0~A_7)	Address Bus (A_0~A_7)	I/O Port
Port 2	I/O Port	I/O Port	I/O Port
Port 3	Data Bus (D_0~D_7)	Data Bus (D_0~D_7)	I/O Port
Port 4	Address Bus (A_8~A_{15})	I/O Port or Address Bus (A_8~A_{15})	I/O Port
Port 5	I/O Port	I/O Port	I/O Port
Port 6	I/O Port	I/O Port	I/O Port
Port 7	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	Output Port

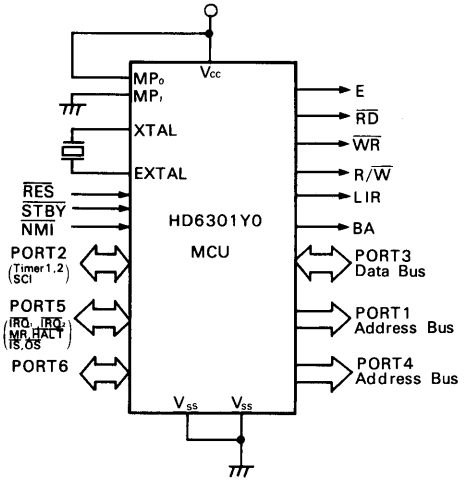


Figure 19 Mode 1

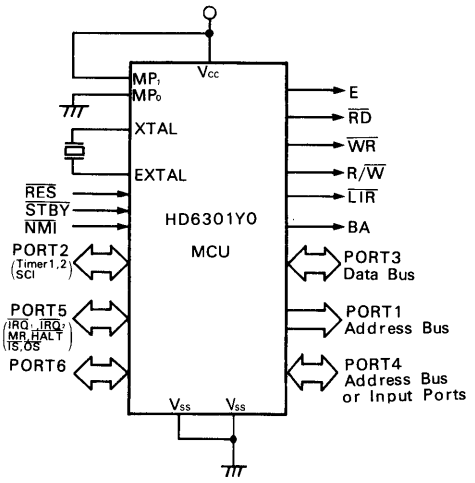


Figure 20 Mode 2

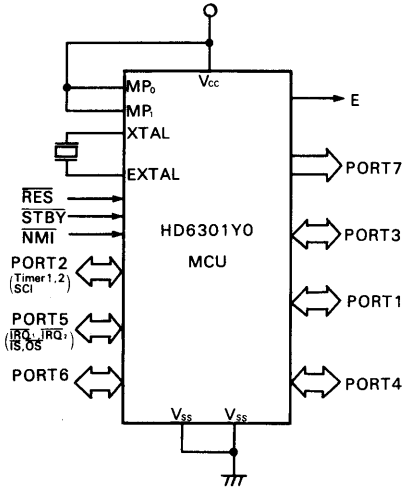


Figure 21 Mode 3

Table 6 Internal Register

Address	Register	Abbreviation	R/W**	Initialized value during reset***
00	Port 1 DDR (Data Direction Register)	P1DDR	W	$FE
01	Port 2 DDR	P2DDR	W	$00
02	Port 1	PORT1	R/W	indefinite
03	Port 2	PORT2	R/W	indefinite
04	Port 3 DDR	P3DDR	W	$FE
05	Port 4 DDR	P4DDR	W	$00
06	Port 3	PORT3	R/W	indefinite
07	Port 4	PORT4	R/W	indefinite
08	Timer Control/Status Register 1	TCSR1	R/W	$00
09	Free Running Counter (MSB)	FRCH	R/W	$00
0A	Free Running Counter (LSB)	FRCL	R/W	$00
0B	Output Compare Register 1 (MSB)	OCR1H	R/W	$FF
0C	Output Compare Register 1 (LSB)	OCR1L	R/W	$FF
0D	Input Capture Register (MSB)	ICRH	R	$00
0E	Input Capture Register (LSB)	ICRL	R	$00
0F	Timer Control/Status Register 2	TCSR2	R/W	$10
10	Rate/Mode Control Register	RMCR	R/W	$C0
11	Tx/Rx Control Status Register 1	TRCSR1	R/W	$20
12	Receive Data Register	RDR	R	$00
13	Transmit Data Register	TDR	W	indefinite
14	RAM/Port 5 Control Register	RP5CR	R/W	$F8 or $78
15	Port 5	PORT5	R/W	indefinite
16	Port 6 DDR	P6DDR	W	$00
17	Port 6	PORT6	R/W	indefinite
18	Port 7	PORT7	R/W	indefinite
19	Output Compare Register 2 (MSB)	OCR2H	R/W	$FF
1A	Output Compare Register 2 (LSB)	OCR2L	R/W	$FF
1B	Timer Control/Status Register 3	TCSR3	R/W	$20
1C	Time Constant Register	TCONR	W	$FF
1D	Timer 2 Up Counter	T2CNT	R/W	$00
1E	Tx/Rx Control Status Register 2	TRCSR2	R/W	$28
1F	Test Register*	TSTREG	—	—
20	PORT 5 DDR	P5DDR	W	$00
21	PORT 6 Control/Status Register	P6CSR	R/W	$07
22	— Reserved	—	—	—
23	—	—	—	—
24	—	—	—	—
25	—	—	—	—
26	—	—	—	—
27	—	—	—	—

* Register for test. Don't access this register.
** R: Read-only register, W: Write-only register, R/W: Read/Write register.
*** When empty bit is in the register, it is set to "1"

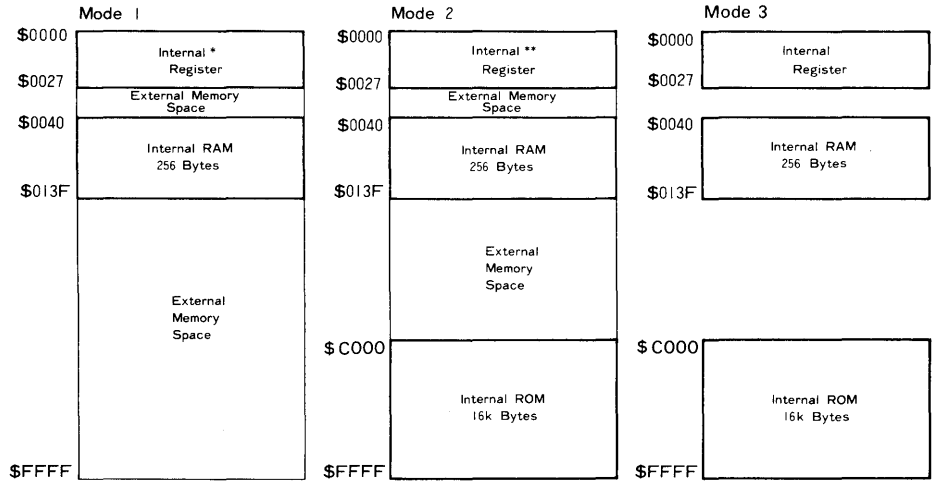


Figure 22 HD6301Y0 Memory Map

■ TIMER 1

The HD6301Y0 provides a 16-bit programmable timer which can simultaneously measure an input waveform and generate two independent output waveforms. The pulse widths of both input/output waveforms vary from microseconds to seconds.

Timer 1 is configured as follows (refer to Fig. 24).

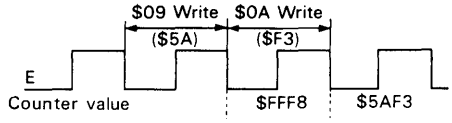
- Control/Status Register 1 (8 bit)
- Control/Status Register 2 (7 bit)
- Free Running Counter (16 bit)
- Output Compare Register 1 (16 bit)
- Output Compare Register 2 (16 bit)
- Input Capture Register (16 bit)

• Free-Running Counter (FRC)($0009:000A)

The key timer element is a 16-bit free-running counter driven and incremented by system clock. The counter value is readable by software without affecting the counter. The counter is cleared during reset.

When writing to the upper byte ($09), the CPU writes the preset value ($FFF8) into the counter (address $09, $0A) regardless of the write data value. But when writing to the lower byte ($0A) after the upper byte writing, the CPU writes not only the lower byte data into lower 8 bit, but also the upper byte data into higher 8 bit of the FRC.

The counter will be as follows when the CPU writes to it by double store instructions (STD, STX, etc.)



In the case of the CPU write ($5AF3) to the FRC

Figure 23 Counter Write Timing

• Output Compare Register (OCR) ($000B, $000C; OCR1) ($0019, $001A: OCR2)

The output compare register is a 16-bit read/write register which can control an output waveform. The data of OCR is always compared with the FRC.

When the data matches, output compare flag (OCF) in the timer control/status register (TCSR) is set. If an output enable bit (OE) in the TCSR2 is "1", an output level bit(OLVL) in the TCSR will be output to bit 1 (OCR 1) and bit 5 (OCR 2) of port 2. To control the output level again by the next compare, the value of OCR and OLVL should be changed. The OCR is set to $FFFF at reset. The compare function is inhibited for a cycle just after a write to the upper byte of the OCR or FRC. This is to begin the comparison after setting the 16 bit value in the register and to inhibit the compare function at this cycle, because the CPU writes the upper byte to the FRC, and at the next cycle the counter is set to $FFF8.

* For data write to the FRC or the OCR, 2-byte transfer instruction (such as STX, etc.) should be used.

• Input Capture Register (ICR) ($000D : 000E)

The input capture register is a 16-bit read-only register which stores the FRC's value when external input signal transition generates an input capture pulse. Such transition is controled by input edge bit (IEDG) in the TCSR1.

In order to input the external input signal to the edge detector, a bit of the DDR corresponding to bit 0 of port 2 should be cleared ("0"). When an input capture pulse occurs by external input signal transition at the next cycle of CPU's high-byte read of the ICR, the input capture pulse will be delayed by one cycle. In order to ensure the input capture operation, a CPU read of the ICR needs 2-byte transfer instruction. The input pulse width should be at least 2 system cycles. This register is cleared ($0000) during reset.

• Timer Control/Status Register 1 (TCSR1) ($0008)

The timer control/status register 1 is an 8-bit register. All bits are readable and the lower 5 bits are also writable. The upper 3 bits are

read-only which indicate the following timer status.

Bit 5 The counter value reached to $0000 as a result of counting-up (TOF).

Bit 6 A match has occurred between the FRC and the OCR 1 (OCF1).

Bit 7 Defined transition of the timer input signal causes the counter to transfer its data to the ICR (ICF).

The followings are the each bit descriptions.

Timer Control/Status Register 1

7	6	5	4	3	2	1	0	
ICF	OCF1	TOF	EICI	EOCI1	ETOI	IEDG	OLVL1	$0008

Bit 0 OLVL1 Output Level 1

OLVL1 is transferred to port 2, bit 1 when a match occurs between the counter and the OCR1. If bit 0 of the TCSR2 (OE1) is set to "1", OLVL1 will appear at bit 1 of port 2.

Bit 1 IEDG Input Edge

This bit determines which edge, rising or falling, of input signal of bit 0 of port 2 will trigger data transfer from the counter to the ICR. For this function, the DDR corresponding to port 2, bit 0 should be cleared beforehand.

IEDG=0, triggered on a falling edge ("High" to "Low")

IEDG=1, triggered on a rising edge ("Low" to "High")

Bit 2 ETOI Enable Timer Overflow Interrupt

When this bit is set, an internal interrupt (IRQ_3) by TOI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 3 EOCI1 Enable Output Compare Interrupt 1

When this bit is set, an internal interrupt (IRQ_3) by OCI1 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 EICI Enable Input Capture Interrupt

When this bit is set, an internal interrupt (IRQ_3) by ICI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag

This read-only bit is set when the counter increments from $FFFF by 1. Cleared when the counter's MSB byte ($0009) is read by the CPU after the TCSR1 read at TOF=1.

Bit 6 OCF1 Output Compare Flag 1

This read-only bit is set when a match occurs between the OCR1 and the FRC. Cleared when writing to the OCR1 ($000B or $000C) after the TCSR1 or TCSR2 read at OCF=1.

Bit 7 ICF Input Capture Flag

This read-only bit is set when an input signal of port 2, bit 0 makes a transition as defined by IEDG and the FRC is transferred to the ICR. Cleared when reading the upper byte ($000D) of the ICR after the TCSR1 or TCSR2 read at ICF=1.

- **Timer Control/Status Register 2 (TCSR2) ($000F)**

The timer control/status register 2 is a 7-bit register. All bits are readable and the lower 4 bits are also writable. But the upper 3 bits are read-only which indicate the following timer status.

Bit 5 A match has occurred between the FRC and the OCR2 (OCF2).

Bit 6 The same status flag as the OCF1 flag of the TCSR1, bit 6.

Bit 7 The same status flag as the ICF flag of the TCSR1, bit 7.

The followings are the each bit descriptions.

Timer Control/Status Register 2

7	6	5	4	3	2	1	0	
ICF	OCF1	OCF2	–	EOCI2	OLVL2	OE2	OE1	$000F

Bit 0 OE1 Output Enable 1

This bit enables the OLVL1 to appear at port 2, bit 1 when a match has occurred between the counter and the output compare register 1. When this bit is cleared, bit 1 of port 2 will be an I/O port. When set, it will be an output of OLVL1 automatically.

Bit 1 OE2 Output Enable 2

This bit enables the OLVL2 to appear at port 2, bit 5 when a match has occurred between the counter and the output compare register 2. When this bit is cleared, port 2, bit 5 will be an I/O port. When set, it will be an output of OLVL2 automatically.

Bit 2 OLVL2 Output Level 2

OLVL2 is transferred to port 2, bit 5 when a match has occurred between the counter and the OCR2. If bit 5 of the TCSR2 (OE2) is set to "1", OLVL2 will appear at port 2, bit 5.

Bit 3 EOCI2 Enable Output Compare Interrupt 2

When this bit is set, an internal interrupt (IRQ_3) by OCI2 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 Not used

Bit 5 OCF2 Output Compare Flag 2

This read-only bit is set when a match has occurred between the counter and the OCR2. Cleared when writing to the OCR2 ($0019 or $001A) after the TCSR2 read at OCF2=1.

Bit 6 OCF1 Output Compare Flag 1

Bit 7 ICF Input Capture Flag

OCF1 and ICF are dual addressed. If which register, TCSR1 or TCSR2, CPU reads, it can read OCF1 and ICF to bit 6 and bit 7.

Both the TCSR1 and TCSR2 will be cleared during reset.

(Note) If OE1 or OE2 is set to "1" before the first output compare match occurs after reset restart, bit 1 or bit 5 of port 2 will produce "0" respectively.

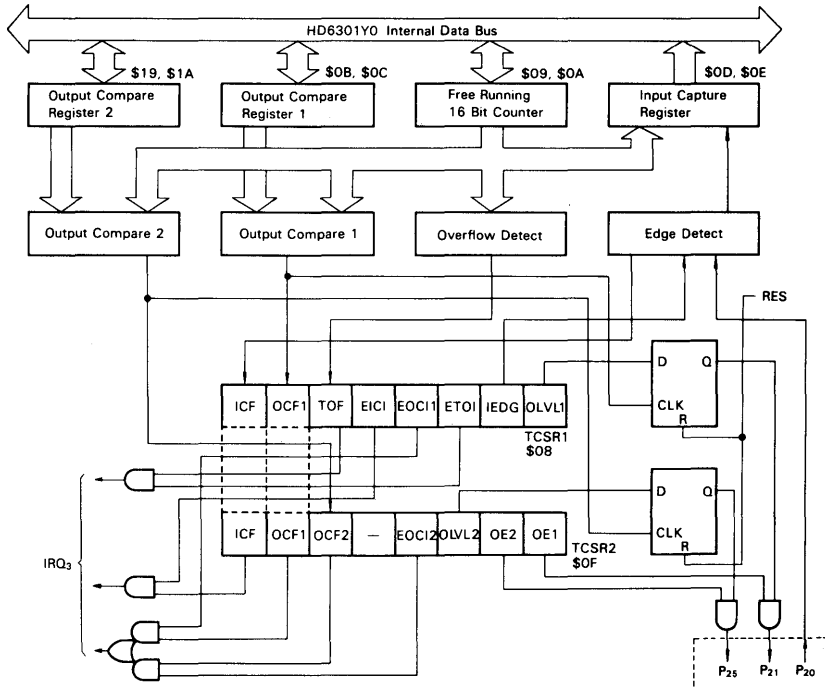


Figure 24 Timer 1 Block Diagram

■ TIMER 2

In addition to the timer 1, the HD6301Y0 provides an 8-bit reloadable timer, which is capable of counting the external event. The timer 2 contains a timer output, so the MCU can generate three independent waveforms. (Refer to Fig. 25.)

The timer 2 is configured as follows:

- Control/Status Register 3 (7 bits)
- 8-bit Up Counter
- Time Constant Register (8 bits)

• Timer 2 Up Counter (T2CNT) ($001D)

This is an 8-bit up counter which operates with the clock decided by CKS0 and CKS1 of the TCSR3. The CPU can read the value of the counter without affecting the counter. In addition, any value can be written to the counter by software even during counting.

The counter is cleared when a match occurs between the counter and the TCONR or during reset.

If the write operation is made by software to the counter at the cycle of counter clear, it does not reset the counter but put the write data to the counter.

• Time Constant Register (TCONR) ($001C)

The time constant register is an 8-bit write only register. It is always compared with the counter.

When a match has occurred, the counter match flag (CMF) of the timer control status register 3 (TCSR3) is set and the value selected by TOS0 and TOS1 of the TCSR3 will appear at port 2, bit 6. When CMF is set, the counter will be cleared simultaneously and then start counting from $00. This enables regular interrupts and waveform outputs without any software support. The TCONR is set to "$FF" during reset.

• Timer Control/Status Register 3 (TCSR3) ($001B)

The timer control/status register 3 is a 7-bit register. All bits are readable and 6 bits except for CMF can be written.

The followings are each pin descriptions.

Timer Control/Status Register 3

7	6	5	4	3	2	1	0	
CMF	ECMI	—	T2E	TOS1	TOS0	CKS1	CKS0	$001B

Bit 0 CKS0 Input Clock Select 0
Bit 1 CKS1 Input Clock Select 1

Input clock to the counter is selected as shown in Table 7 depending on these two bits. When an external clock is selected, bit 7 of port 2 will be a clock input automatically. Timer 2 detects the rising edge of the external clock and increments the counter. The external clock is countable up to half the frequency of the system clock.

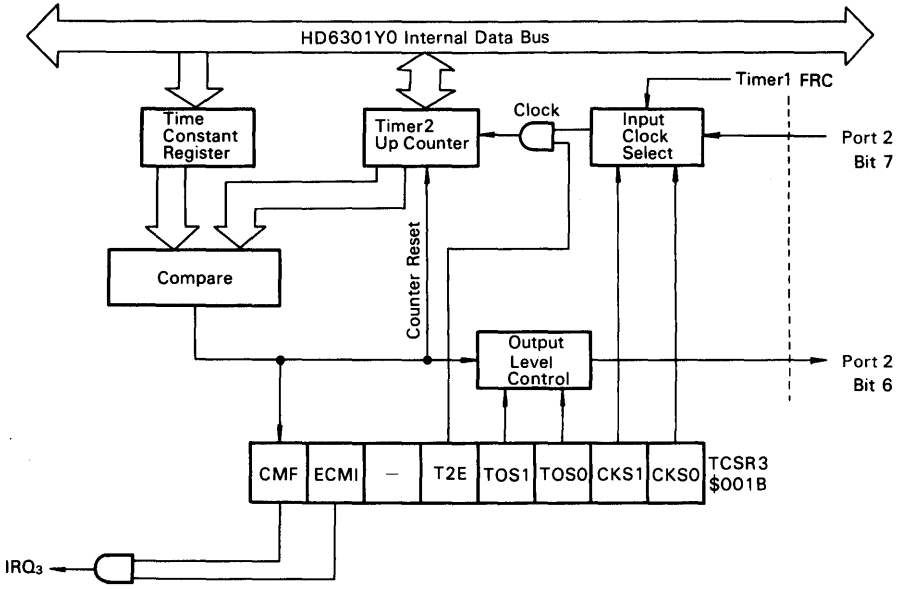


Figure 25 Timer 2 Block Diagram

Table 7 Input Clock Select

CKS1	CKS0	Input Clock to the Counter
0	0	E clock
0	1	E clock/8*
1	0	E clock/128*
1	1	External clock

* These clocks come from the FRC of the timer 1. If one of these clocks is selected as an input clock to the up counter, the CPU should not write to the FRC of the timer 1.

Bit 2 TOS0 Timer Output Select 0
Bit 3 TOS1 Timer Output Select 1

When a match occurs between the counter and the TCONR timer 2 outputs shown in Table 8 will appear at port 2, bit 6 depending on these two bits. When both TOS0 and TOS1 are "0", bit 6 of port 2 will be an I/O port.

Table 8 Timer 2 Output Select

TOS1	TOS0	Timer Output
0	0	Timer Output Inhibited
0	1	Toggle Output*
1	0	Output "0"
1	1	Output "1"

* When a match occurs between the counter and the TCONR, timer 2 output level is reversed. This leads to production of a square wave with 50% duty to the external without any software support.

Bit 4 T2E Timer 2 Enable Bit

When this bit is cleared, a clock input to the up counter is inhibited and the up counter stops. When set to "1", a clock selected by CKS1 and CKS0 (Table 7) is input to the up counter.

(Note) P_{26} outputs "0" when T2E bit cleared and timer 2 set in output enable condition by TOS1 or TOS0. It also outputs "0" when T2E bit set "1" and timer 2 set in output enable condition before the first counter match occurs.

Bit 5 Not Used.

Bit 6 ECMI Enable Counter Match Interrupt

When this bit is set, an internal interrupt (IRQ_3) by CMI is enabled. When cleared, the interrupt is inhibited.

Bit 7 CMF Counter Match Flag

This read-only bit is set when a match occurs between the up counter and the TCONR. Cleared by writing "0" at CMF=1 by software (unable to write "1" by software).

Each bit of the TCSR3 is cleared during reset.

■ SERIAL COMMUNICATION INTERFACE (SCI)

The Serial Communication Interface (SCI) in the HD6301Y0 contains the following two operating modes: asynchronous mode by the NRZ format, and clocked synchronous mode which transfers data synchronously with the clock. In the asynchronous mode, data length, parity bits and number of stop bits can be selected, and eight transfer formats are provided.

The SCI consists of the following registers as shown in Fig. 26 Block Diagram.

- Transmit/Receive Control Status Register 1 (TRCSR1)
- Rate/Mode Control Register (RMCR)
- Transmit/Receive Control Status Register 2 (TRCSR2)
- Receive Data Register (RDR)
- Recevie Shift Register
- Transmit Data Register (TDR)
- Transmit Shift Register

To operate the SCI, initialize the RMCR and TRCSR2, after selecting the desirable operating mode and transfer format. Next,

set the enable bit (TE or RE) of the TRCSR1. Operating mode and transfer format should be changed when the enable bit (TE, RE) is cleared. When setting the TE or RE again after changing the operating mode or transfer format, interval of more than a 1-bit cycle of the baud rate or bit rate is necessary. If a 1-bit cycle or more is not allowed, the SCI block may not be initialized.

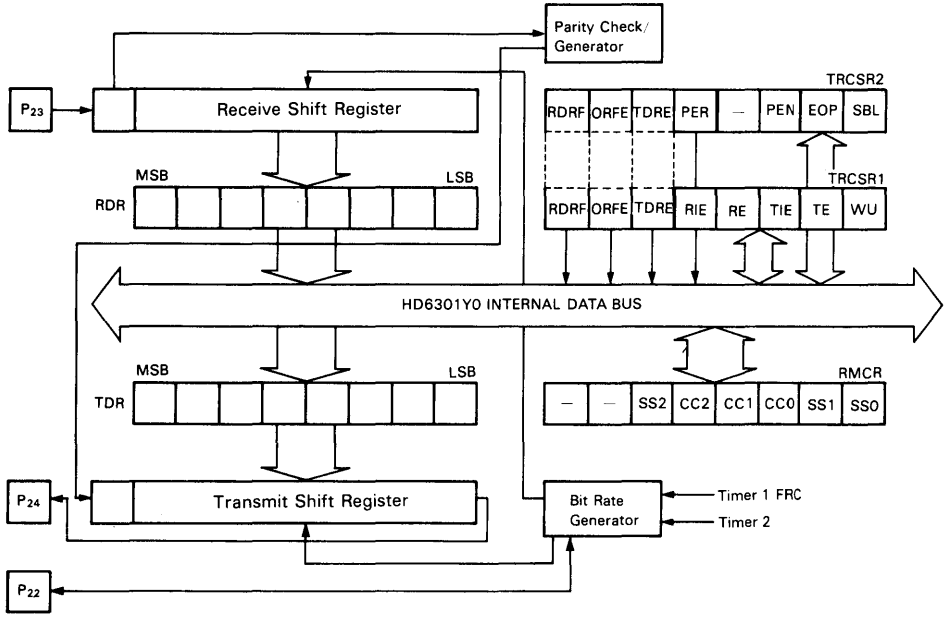


Figure 26 SCI Block Diagram

- **Asynchronous Mode**

Asynchronous mode contains 8 transfer formats as shown in Fig. 27.

Data transmission is enabled by setting TE bit of the TRCSR1, then port 2, bit 4 will unconditionally become a serial output independently of the corresponding DDR.

To transmit data, set the desirable transmit format with RMCR and TRCSR2. When the TE bit is set, the data can be transmitted after transmitting the one frame of preamble ("1").

The conditions at this stage are as follows.

1) If the TDR is empty (TDRE=1), consecutive 1's are produced to indicate the idle state.
2) If the TDR contains data (TDRE=0), data is sent to the Transmit Shift Register and data transmit starts.

During data transmit, a start bit of "0" is transmitted first. Then 7-bit or 8-bit data (starts from bit 0) is transmitted. With PEN=1, the parity bit, even or odd, selected by EOP bit is added, lastly the stop bit (1 bit or 2 bis) is sent.

When the TDR is "empty", hardware sets TDRE flag bit. If the CPU doesn't respond to the flag in proper timing (the TDRE is in set condition till the next normal data transfer starts from the transmit data register to the transmit sift register), "1" is transferred instead of the start bit "0" and continues to be transferred till data is provided to the data register. While the TDRE is "1", "0" is not transferred.

Data receive is possible by setting RE bit. This makes port 2, bit 3 a serial input. The operation mode of data receive is decided by the contents of the TRCSR2 and RMCR at first, and set RE bit of TRCSR1. The first "0" (space) synchronizes the receive bit flow. Each bit of the following data will be strobed in the middle. If a stop bit is not "1", a framing error assumed and ORFE is set.

When a framing error occurs, receive data is transferred to the Receive Data Register and the CPU can read the error-generating data. This makes it possible to detect a line break.

When PEN bit is set, the parity check is done. If the parity bit does not match the EOP bit, a parity error occurs and the PER bit is set, not the RDRF bit. Also, when the parity error occurs the receive data can be read just like in the case of the framing error.

The RDRF flag is set when the data is received without a framing error and a parity error.

If RDRF is still set when receiving the stop bit of the next data, ORFE is set to indicate the overrun generation. CPU can get the receive data by reading RDR. When 7 bit data format is selected, the 8th bit of RDR is "0".

When the CPU read the receive Data Register as a response to RDRF flag or ORFE flag after having read TRCSR, RDRF or ORFE is cleared.

(Note) Clock Source in Asynchronous Mode

If CC1:CC0=10, the internal bit rate clock is provided at P_{22} regardless of the values for TE or RE. Maximum clock rate is E÷16.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P_{22} at sixteen times (16×) the desired bit rate, but not greater than E.

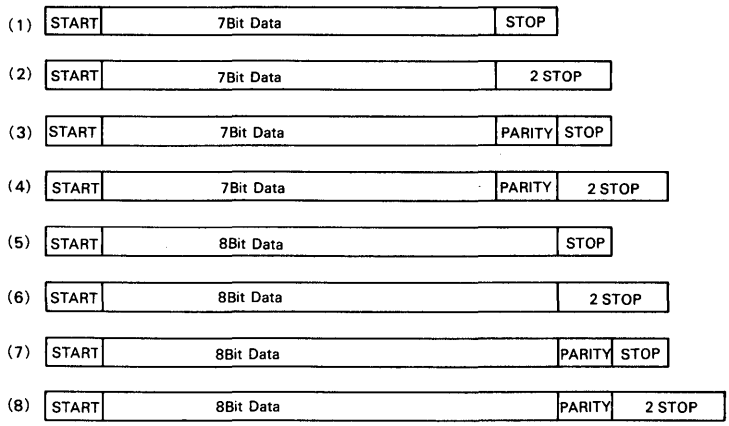


Figure 27 Asynchronous Mode Transfer Format.

- **Clocked Synchronous Mode**

In the clocked synchronous mode, data transmit is synchronized with the clock pulse. The HD6301Y0 SCI provides functionally independent transmitter and receiver which makes full duplex operation possible in the asynchronous mode. But in the clocked synchronous mode an SCI clock I/O pin is only P_{22}, so the simultaneous receive and transmit operation is not available. In this mode, TE and RE should not be in set condition ("1") simultaneously. Fig. 28 gives a synchronous clock and a data format in the clocked synchronous mode.

1) Data transmit

Data transmit is realized by setting TE bit in the TRCSR1. Port 2, bit 4 becomes an output unconditionally independent of the value of the corresponding DDR.

Both the RMCR and TRCSR should be set in the desirable operating condition for data transmit.

When an external clock input is selected and the TDRE flag is "0", data transmit is performed from port 2, bit 4, synchronizing with 8 clock pulses input from external to port 2, bit 2.

Data is transmitted from bit 0 and the TDRE is set when the Transmit Shift Register (TSR) is "empty". More than 9th clock pulse of external are ignored.

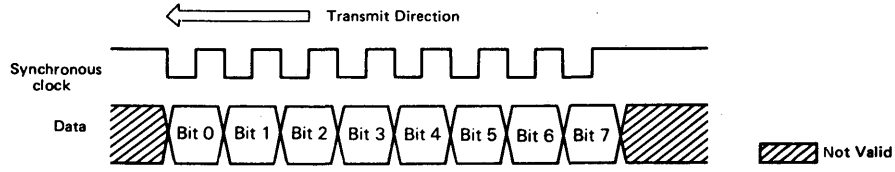
When data transmit is selected to the clock output, the MCU produces transmit data and synchronous clock at TDRE flag clear.

2) Data receive

Data receive is enabled by setting RE bit. Port 2, bit 3 will be a serial input. The operating mode of data receive is decided by the TRCSR1 and the RMCR.

If the external clock input is selected, 8 external clock pulses and the synchronized receive data are input to port 2, bit 2 and bit 3 respectively. The MCU put receive data into the receive data shift register by this clock and set the RDRF flag at the termination of 8 bit data receive. More than 9th clock pulse of external input are ignored. When RDRF is cleared, the MCU starts receiving the next data instantly. So, RDRF should be cleared with P_{22} "High".

When data receive is selected with the clock output, 8 synchronous clocks are output to the external by setting RE bit. So receive data should be input from external synchronously with this clock. When the first byte data is received, the RDRF flag is set. After the second byte, receive operation is performed by sending the synchronous clock to the external after clearing the RDRF bit.



- Transmit data is produced from a falling edge of a synchronous clock to the next falling edge.
- Receive data is latched at the rising edge.

Figure 28 Clocked Synchronous Mode Format

- **Transmit/Receive Control Status Register (TRCSR1) ($0011)**

The TRCSR1 is composed of 8 bits which are all readable. Bits 0 to 4 are also writable. This register is initialized to $20 during reset. Each bit functions are as follows.

Transmit/Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU Wake-up

In a typical multi-processor configuration, the software protocol provides the destination address at the first byte of the message. In order to make uninterested MCU ignore the remaining message, a wake-up function is available. By this, uninterested MCU can inhibit all further receive processing till the next message starts.

Then wake-up function is triggered by consecutive 1's with 1 frame length. The software protocol should provide the idle time between messages.

By setting this bit, the MCU stops data receive till the next message. The receive of consecutive "1" with one frame length wakes up and clears this bit by hardware and then the MCU restarts receive operation. However, the RE flag should be already set before setting this bit. In the clocked synchronous mode WU is not available, so this bit should not be set.

Bit 1 TE Transmit Enable

When this bit is set, transmit data will appear at port 2, bit 4 after one frame preamble in asynchronous mode, while in clocked synchronous mode it appears immediately. This is executed regardless of the value of the corresponding DDR. When TE is cleared, the serial I/O doesn't affect port 2, bit 4.

Bit 2 TIE Transmit Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when TDRE (bit 5) is set. When cleared, the interrupt is inhibited.

Bit 3 RE Receive Enable

When set, a signal is input to the receiver from port 2, bit 3 regardless of the value of the DDR. When RE is cleared, the serial I/O doesn't afffect port 2, bit 3.

Bit 4 RIE Receive Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when RDRF (bit 7) or ORFE (bit 6) is set. When cleared, the interrupt is inhibited.

Bit 5 TDRE Transmit Data Register Empty

TDRE is set by hardware when the TDR is transferred to the Transmit Shift Register in the asynchronous mode, while in clocked synchronous mode when the TDSR is "empty". This bit is cleared by reading the TRCSR1 or TRCSR2 and writing new transmit data to the TDR when TDRE=1. TDRE is set to "1" during reset.

(Note) TDRE should be cleared in the transmittable state after the TE set.

Bit 6 ORFE Overrun Framing Error

ORFE is set by hardware when an overrun or a framing error is generated (during data-receive only). An overrun error occurs when new receive data is ready to be transferred to the RDR during RDRF still being set. A framing error occurs when a stop bit is "0". But in clocked synchronous mode, this bit is not affected. This bit is cleared by reading the TRCSR1 or TRCSR2, and the RDR, when RDRF=1. ORFE is cleared during reset.

Bit 7 RDRF Receive Data Register Full

RDRF is set by hardware when data is received normally and transferred from the Receive Shift Register (RSR) to the RDR. This bit is cleared by reading TRCSR1 or TRCSR2, and the RDR, when RDRF=1. This bit is cleared during reset.

- **Transmit Rate/Mode Control Register (RMCR)**

The RMCR controls the following serial I/O:

- Baud Rate
- Data Format
- Clock source
- Port 2, Bit 2 Function
- Operation Mode

All bits are readable/writable. Bit 0 to 5 of the RMCR are cleared during reset.

Transfer Rate/Mode Control Register

7	6	5	4	3	2	1	0	
–	–	SS2	CC2	CC1	CC0	SS1	SS0	$0010

Bit 0 SS0
Bit 1 SS1
Bit 5 SS2
} Speed Select

These bits control the baud rate used for the SCI. Table 9 lists the available baud rates. The timer 1 FRC (SS2=0) and the timer 2 up counter (SS2=1) provide the internal clock to the SCI. When selecting the timer 2 as a baud rate clock source, it functions as a baud rate generator. The timer 2 generates the baud rate listed in Table 10 depending on the value of the TCONR.

(Note) When operating the SCI with internal clock, do not perform write operation to the timer/counter which is the clock source of the SCI.

Bit 2 CC0
Bit 3 CC1
Bit 4 CC2
} Clock Control/Format Select*

These bits control the data format and the clock source (refer to Table 11).

* CC0, CC1 and CC2 are cleared during reset and the MCU goes to the clocked synchronous mode of the external clock operation. Then the MCU automatically set port 2, bit 2 into the clock input state. When using port 2, bit 2 as an output port, the DDR of port 2 should be set to "1" and CC1 and CC0 to "0" and "1" respectively.

Bit 6 Not Used.
Bit 7 Not Used

- **Transmit/Receive Control Status Register 2 (TRCSR2)**

The TRCSR2 is a 7-bit register which can select a data format in the asynchronous mode. The upper 3 bits are the same address as the TRCSR1. Therefore, the RDRF, ORFE and TDRE can be read by either the TRCSR1 or TRCSR2. Bits 0 to 2 of the TRCSR2 are used for read/write. Bits 4 to 7 are used only for read.

Transmit/Receive Control Status Register 2

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	PER	–	PEN	EOP	SBL	$001E

Bit 0 SBL Stop Bit Length

This bit selects the stop bit length in the asynchronous mode. If this bit is "0", the stop bit is 1-bit. If "1", the stop bit is 2-bit. This bit is cleared during reset.

Bit 1 EOP Even/Odd Parity

This bit selects the parity generated and checked when the PEN is "1". If this bit is "0", the parity is even. If "1", it is odd. This bit is cleared during reset.

Bit 2 PEN Parity Enable

This bit decides whether the parity bit should be generated and checked in the asynchronous mode or not. If this bit is "0", the parity bit is neither generated nor checked. If "1", it is generated and checked. This bit is cleared during reset.

The 3 bits above do not affect the SCI opertion in the clocked synchronous mode.

Bit 3 Not Used

Table 9 SCI Bit Times and Transfer Rates

(1) Asynchronous Mode

			XTAL	2.4576MHz	4.0MHz	4.9152MHz
SS2	SS1	SS0	E	614.4kHz	1.0MHz	1.2288MHz
0	0	0	E÷16	26μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

* When SS2 is "1", Timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32\,(N+1)} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

(2) Clocked Synchronous Mode *

			XTAL	4.0MHz	6.0MHz	8.0MHz
SS2	SS1	SS0	E	1.0MHz	1.5MHz	2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

* Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC ~ 1/2 system clock.

** The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate } (\mu s/\text{bit}) = \frac{4\,(N+1)}{f} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

Table 10 Baud Rate and Time Constant Register Example

Baud Rate (Baud) \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

* E/8 clock is input to the timer 2 up counter and E clock otherwise.

Table 11 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock Source	Port 2, Bit 2	Port 2, Bit 3	Port 2, Bit 4
0	0	0	8-bit data	Clocked Synchronous	External	Input	When the TRCSR1, RE bit is "1", bit 3 is used as a serial input.	When the TRCSR1, TE bit is "1", bit 4 is used as a serial output.
0	0	1	8-bit data	Asynchronous	Internal	Not Used**		
0	1	0	8-bit data	Asynchronous	Internal	Output*		
0	1	1	8-bit data	Asynchronous	External	Input		
1	0	0	8-bit data	Clocked Synchronous	Internal	Output		
1	0	1	7-bit data	Asynchronous	Internal	Not Used**		
1	1	0	7-bit data	Asynchronous	Internal	Output*		
1	1	1	7-bit data	Asynchronous	External	Input		

* Clock output regardless of the TRCSR1, bit RE and TE.
** Not used for the SCI.

Bit 4 PER Parity Error
This bit is set when the PEN is "1" and a parity error occurs. It is cleared by reading the RDR after reading the TRCSR2, when PER = 1.

Bit 5 TDRE
Transmit Data Register Empty

Bit 6 ORFE
Overrun/Framing Error

Bit 7 RDRF
Receive Data Register Full

* Each flag of the TDRE, ORFE, and RDRF can be read from either the TRCSR1 or TRCSR2.

■ TIMER, SCI STATUS FLAG

Table 12 shows the set and reset conditions of each status flag in the timer 1, timer 2 and SCI.

Table 12 Timer 1, Timer 2 and SCI Status Flag

		Set Condition	Clear Condition
P6CSR	IS FLAG	Falling edge input to P_{54} ($\overline{IS}$)	1. Read the P6CSR then read or write the PORT6, when IS FLAG = 1 2. $\overline{RES}$ = 0
	ICF	FRC → ICR by Rising or Falling edge input to P_{20}. (Selecting with the IEDG bit)	1. Read the TCSR1 or TCSR2 then ICRH, when ICF = 1 2. $\overline{RES}$ = 0
Timer 1	OCF1	OCR1 = FRC	1. Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1 = 1 2. $\overline{RES}$ = 0
	OCF2	OCR2 = FRC	1. Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2 = 1 2. $\overline{RES}$ = 0
	TOF	FRC = $FFFF + 1 cycle	1. Read the TCSR1 then FRCH, when TOF = 1 2. $\overline{RES}$ = 0
Timer 2	CMF	T2CNT = TCONR	1. Write "0" to CMF, when CMF = 1 2. $\overline{RES}$ = 0
SCI	RDRF	Receive Shift Register → RDR	1. Read the TRCSR1 or TRCSR2 then RDR, when RDRF = 1 2. $\overline{RES}$ = 0
	ORFE	1. Framing Error (Asynchronous Mode) Stop Bit = 0 2. Overrun Error (Asynchronous Mode) Receive Shift Register → RDR when RDRF = 1	1. Read the TRCSR1 or TRCSR2 then RDR, when ORFE = 1 2. $\overline{RES}$ = 0
	TDRE	1. Asynchronous Mode TDR → Transmit Shift Register 2. Clocked Synchronous Mode Transmit Shift Register is "empty" 3. $\overline{RES}$ = 0	Read the TRCSR1 or TRCSR2 then write to the TDR, when TDRE = 1 Note) TDRE should be reset after the TE set.
	PER	Parity when PEN = 1	1. Read the TRCSR2 then RDR, when PER = 1 2. $\overline{RES}$ = 0

(Note) → ; Transfer = ; equal

ICRH; Upper byte of ICR
OCR1H; Upper byte of OCR1
OCR2H; Upper byte of OCR2

OCR1L; Lower byte of OCR1
OCR2L; Lower byte of OCR2
FRCH; Upper byte of FRC

■ LOW POWER DISSIPATION MODE

The HD6301Y0 provides two low power dissipation modes; sleep and standby.

• Sleep Mode

The MCU goes to the sleep mode by SLP instruction execution. In the sleep mode, the CPU stops its operation, while the registers' contents are retained. In this mode, the peripherals except the CPU such as timers, SCI, etc. continue their functions. The power dissipation of sleep-condition is one fifth that of operating condition.

The MCU returns from this mode by an interrupt, $\overline{RES}$ or $\overline{STBY}$; it goes to the reset state by $\overline{RES}$ and the standby mode by $\overline{STBY}$. When the CPU acknowledges an interrupt request, it cancels the sleep mode, returns to the operation mode and branches to the interrupt routine. When the CPU masks this interrupt, it cancels the sleep mode and executes the next instruction. However, for example, if the timer 1 or 2 prohibits a timer interrupt, the CPU doesn't cancel the sleep mode because of no interrupt request.

This sleep mode is effective to reduce the power dissipation for a system with no need of the HD6301Y0's consecutive operation.

• Standby Mode

The MCU goes to the standby mode with the $\overline{STBY}$ "Low" or by clearing the STBY flag. In this mode, the HD6301Y0 stops all the clocks and goes to the reset state. In this mode, the power dissipation is reduced to several μA. During standby, all pins, except the power supply (V_{CC}, V_{SS}), the $\overline{STBY}$, $\overline{RES}$ and XTAL (which outputs "0"), go to the high impedance state. In this mode, power (V_{CC}) is supplied to the HD6301Y0, and the contents of RAM is retained. The MCU returns from this mode during reset. When the MCU goes to the standby mode with $\overline{STBY}$ "Low", it will restart at the timing shown in Fig. 29(a). When the MCU goes to the standby mode by clearing the STBY flag, it will restart only by keeping the $\overline{RES}$ "Low" for longer than the oscillating stabilization time. (Fig. 29(b))

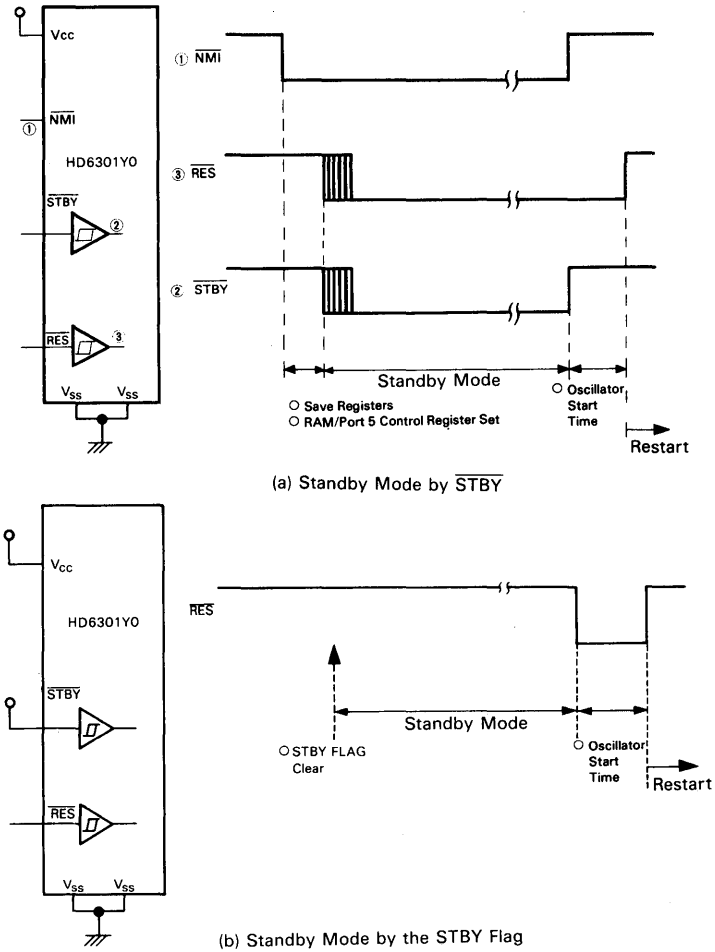


Figure 29 Standby Mode Timing

■ TRAP FUNCTION

The CPU generates an interrupt with the highest priority (TRAP) when fetching an undefined instruction or an instruction from non-memory space. The TRAP prevents the system-burst caused by noise or a program error.

• Op Code Error

When fetching an undefined op code, the CPU saves registers as well as a normal interrupt and branches to the TRAP ($FFEE, $FFEF). This has the priority next to reset.

• Address Error

When an instruction fetch is made excluding internal ROM, RAM and external memory area, the MCU generates an interrupt as well as an op code error. But on the system with no memory in its external memory area, this function is not applicable if an instruction fetch is made from the external non-memory area. Table 13 provides addresses where an address error occurs to each mode.

This function is available only for an instruction fetch and is not applicable to the access of normal data read/write.

Table 13 Addresses Applicable to Address Errors

Mode	1	2	3
Address	$0000	$0000	$0000
	⌇	⌇	⌇
	$0027	$0027	$003F
			$0140
			⌇
			$BFFF

(Note) The TRAP interrupt provides a retry function differently from other interrupts. This is a program flow return to the address where the TRAP occurs when a sequence returns to a main routine from the TRAP interrupt routine by RTI. The retry can prevent the system burst caused by noise, etc.
However, if another TRAP occurs, the program repeats the TRAP interrupt forever, so the consideration is necessary in programming.

■ INSTRUCTION SET

The HD6301Y0 provides object code upward compatible with the HD6801 to utilize all instruction set of the HMCS6800. It also reduces the execution times of key instructions for throughput improvement.

Bit manipulation instruction, change instruction of the index register and accumulator and sleep instruction are also added.

The followings are explained here.

- CPU Programming Model (refer to Fig. 30)
- Addressing Mode
- Accumulator and Memory Manipulation Instruction (refer to Table 14)
- New Instruction
- Index Register and Stack Manipulation Instruction (refer to Table 15)
- Jump and Branch Instruction (refer to Table 16)
- Condition Code Register Manipulation (refer to Table 17)
- Op Code Map (refer to Table 18)

• Programming Model

Fig. 30 depicts the HD6301Y0 programming model. The double accumulator D consists of accumulator A and B, so when using the accumulator D, the contents of A and B are destroyed.

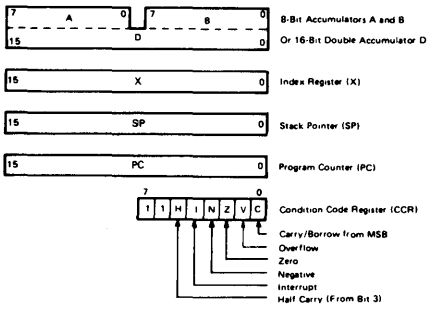


Figure 30 CPU Programming Model

• CPU Addressing Mode

The HD6301Y0 provides 7 addressing modes. The addressing mode is determined by an instruction type and code. Tables 14 through 18 show addressing modes of each instrution with the execution times counted by the machine cycle.

When the clock frequency is 4MHz, the machine cycle time becomes microseconds directly.

Accumulator (ACCX) Addressing

Only an accumulator is addressed and the accumulator A or B is selected. This is a one-byte instruction.

Immediate Addressing

This addressing locates a data in the second byte of an instruction. However, LDS and LDX locate a data in the second and third byte exceptionally. This addressing is a 2 or 3-byte instruction.

Direct Addressing

In this addressing mode, the second byte of an instruction shows the address where a data is stored. 256 bytes ($0 through $255) can be addressed directly. Execution times can be reduced by storing data in this area so it is recommended to make it RAM for users' data area in configurating a system. This is a 2-byte instruction, while 3 byte with regard to AIM, OIM, EIM and TIM.

Extended Addressing

In this mode, the second byte shows the upper 8 bit of the data stored address and the third byte the lower 8 bit. This indicates the absolute address of 3 byte instruction in the memory.

Indexed Addressing

The second byte of an instruction and the lower 8 bit of the index register are added in this mode. As for AIM, OIM, EIM and TIM, the third byte of an instruction and the lower 8 bits of the index register are added.

This carry is added to the upper 8 bit of the index register and the result is used for addressing the memory. The modified address is retained in the temporary address register, so the contents of the index register doesn't change. This is a 2-byte instruction except AIM, OIM, EIM and TIM (3-byte instruction).

Implied Addressing

An instruction itself specifies the address. This is, the instruction addresses a stack pointer, index register, etc. This is a one-byte instruction.

Relative Addressing

The second byte of an instruction and the lower 8 bits of the program counter are added. The carry or borrow is added to the upper 8 bit. So addressing from −126 to +129 byte of the current instruction is enabled. This is a 2-byte instruction.

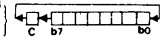
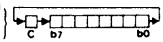
(Note) CLI, SEI Instructions and Interrupt Operation
When accepting the IRQ at a preset timing with the CLI and SEI instructions, more than 2 cycles are necessary between the CLI and SEI instructions. For example, the following program (a) (b) don't accept the IRQ

but (c) accepts it.

.	.	.
.	.	.
.	.	.
.	.	CLI
CLI	CLI	NOP
SEI	NOP	NOP
.	SEI	SEI
.	.	.
.	.	.
.	.	.
.	.	.
(a)	(b)	(c)

The same thing can be said to the TAP instruction instead of the CLI and SEI instructions.

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M→ A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M→ B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M→ B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M→ A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M→ B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3				M, A, B 	•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				M, A, B 	•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

(Note) Condition Code Register will be explained in Note of Table 17.

(continued)

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M, A, B	•	•	↕	↕	⑥	↕
	ASLA													48	1	1		•	•	↕	↕	⑥	↕
	ASLB													58	1	1		•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	ACC A/ ACC B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M, A, B	•	•	↕	↕	⑥	↕
	ASRA													47	1	1		•	•	↕	↕	⑥	↕
	ASRB													57	1	1		•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M, A, B	•	•	R	↕	⑥	↕
	LSRA													44	1	1		•	•	R	↕	⑥	↕
	LSRB													54	1	1		•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	ACC A/ ACC B	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

(Note) Condition Code Register will be explained in Note of Table 17.

- **Additional Instruction**

In addition to the HD6801 instruction set, the HD6301Y0 prepares the following new instructions.

AIM ..... (M)·(IMM) → (M)
Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM ..... (M)+(IMM) → (M)
Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM ..... (M)⊕(IMM) → (M)
Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM ..... (M)·(IMM)
Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These are the 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX ..... (ACCD)↔(IX)
Exchanges the contents of accumulator and the index register.

SLP
Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 15 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD··IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 17.

Table 16 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——		⑧		——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

(Note) * WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 17.

Table 17 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	———	⑩	———			
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MCU Cycles
M_{SP} Contents of memory location pointed by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
• Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3 to bit 4
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from/to bit 7
R Reset Always
S Set Always
↕ Set if true after test or clear
• Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕ C = 1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 18 OP-Code Map

OP CODE HI	LO	0000 0	0001 1	0010 2	0011 3	ACC A 0100 4	ACC B 0101 5	IND 0110 6	EXT / DIR* 0111 7	ACCA or SP IMM 1000 8	DIR 1001 9	IND 1010 A	EXT 1011 B	ACCB or X IMM 1100 C	DIR 1101 D	IND 1110 E	EXT 1111 F	LO
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD			ADDD					3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX			LDD					C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

■ CPU OPERATION

• CPU Instruction Flow

When operating, the CPU fetches an instrution from a memory and executes the required function. This sequence starts with $\overline{RES}$ cancel and repeats itself limitlessly if not affected by a special instruction or a control signal. SWI, RTI, WAI and SLP instructions change this operation, while $\overline{NMI}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, IRQ_3, $\overline{HALT}$ and $\overline{STBY}$ control it. Fig. 31 gives the CPU mode transition and Fig. 32 the CPU system flow chart. Table 19 shows CPU operating states and port states.

• Operation at Each Instruction Cycle

Table 20 shows the operation at each instruction cycle. By the pipeline control of the HD6301Y0, MULT, PUL, DAA and XGDX instructions, etc. prefetch the next instruction. So attention is necessary to the counting of the instruction cycles because it is different from the usual one—from op code fetch to the next instruction op code.

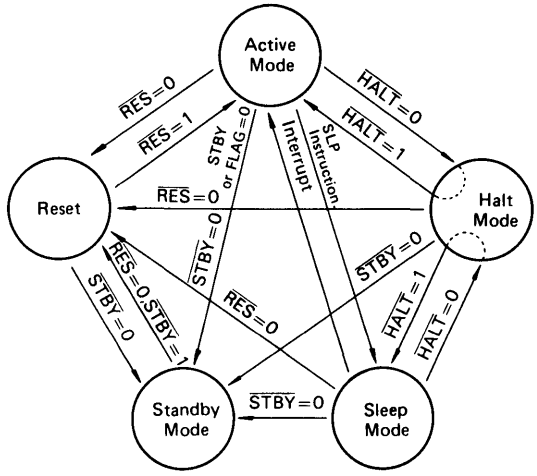


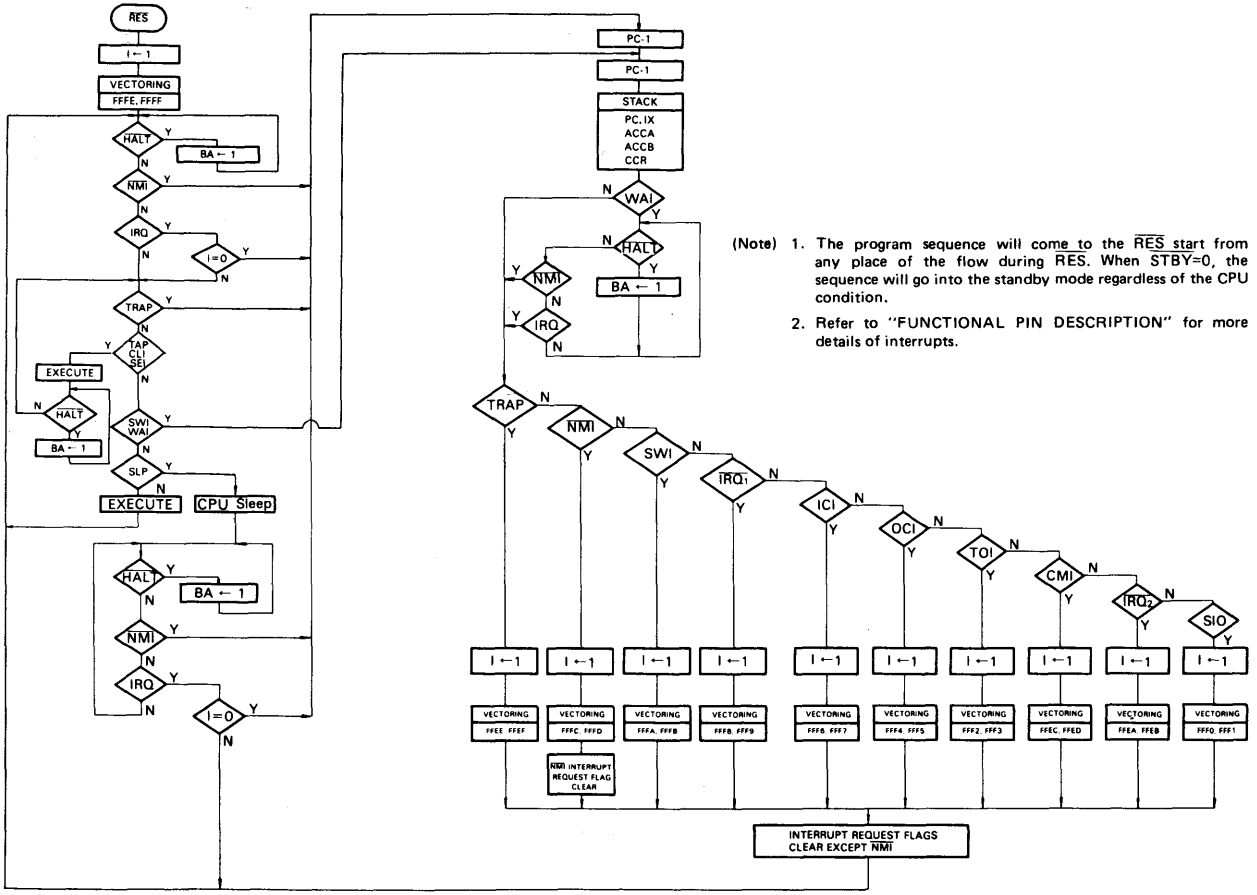
Figure 31 CPU Operation Mode Transition

Table 19 CPU Operation State and Port State

Port	Mode	Reset	STBY****	HALT***	Sleep
Port 1 (A0 to A7)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 2	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 3 (D0 to D7)	Mode 1, 2	T	T	T	T
	Mode 3				Keep
Port 4 (A8 to A15)	Mode 1	H	T	T	H
	Mode 2				*****
	Mode 3	T			Keep
Port 5	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 6	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 7	Mode 1, 2	*	T	**	*
	Mode 3	T			Keep

H; High, L; Low, T; High Impedance
Keep; The output port is retained, and the input port goes to the high impedance state.

* $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$ = H, BA = L
** $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ = T, $\overline{LIR}$, BA = H
*** HALT is unacceptable in mode 3.
**** E pin goes to high impedance state.
***** Address output pin = H
Input port = T



(Note) 1. The program sequence will come to the RES start from any place of the flow during RES. When STBY=0, the sequence will go into the standby mode regardless of the CPU condition.

2. Refer to "FUNCTIONAL PIN DESCRIPTION" for more details of interrupts.

Figure 32 HD6301Y0 System Flow Chart

Table 20 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMMEDIATE								
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	2	1	Op Code Address+1	1	0	1	1	Operand Data
		2	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	3	1	Op Code Address+1	1	0	1	1	Operand Data (MSB)
		2	Op Code Address+2	1	0	1	1	Operand Data (LSB)
		3	Op Code Address+3	1	0	1	0	Next Op Code
DIRECT								
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	3	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
		2	Address of Operand	1	0	1	1	Operand Data
		3	Op Code Address+2	1	0	1	0	Next Op Code
STA	3	1	Op Code Address+1	1	0	1	1	Destination Address
		2	Destination Address	0	1	0	1	Accumulator Data
		3	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	4	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
		2	Address of Operand	1	0	1	1	Operand Data (MSB)
		3	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		4	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	4	1	Op Code Address+1	1	0	1	1	Destination Address (LSB)
		2	Destination Address	0	1	0	1	Register Data (MSB)
		3	Destination Address+1	0	1	0	1	Register Data (LSB)
		4	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Jump Address (LSB)
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM	4	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
AIM EIM OIM	6	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
INDEXED								
JMP	3	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Accumulator Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data (MSB)
		4	IX+Offset+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Register Data (MSB)
		4	IX+Offset+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	IX+Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX+Offset	0	1	0	1	New Operand Data
		6	Op Code Address+2	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	IX+Offset	0	1	0	1	00
		5	Op Code Address+2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX+Offset	0	1	0	1	New Operand Data
		7	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
EXTEND								
JMP	3	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	0	1	0	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Accumulator Data
		4	Op Code Address+1	1	0	1	0	Next Op Code
PULX	4	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	0	1	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Index Register (LSB)
		4	Stack Pointer−1	0	1	0	1	Index Register (MSB)
		5	Op Code Address+1	1	0	1	0	Next Op Code
RTS	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Return Address (MSB)
		4	Stack Pointer+2	1	0	1	1	Return Address (LSB)
		5	Return Address	1	0	1	0	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	FFFF	1	1	1	1	Restart Address (LSB)
		7	FFFF	1	1	1	1	Restart Address (LSB)

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
WAI	9	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
RTI	10	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Conditional Code Register
		4	Stack Pointer+2	1	0	1	1	Accumulator B
		5	Stack Pointer+3	1	0	1	1	Accumulator A
		6	Stack Pointer+4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer+5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer+6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer+7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		Sleep						
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address+1	1	0	1	0	Next Op Code
RELATIVE								
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	0	1	1	Branch Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Branch Address ····· Test="1" Op Code Address+1 ··· Test="0"	1	0	1	0	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Branch Address	1	0	1	0	First Op Code of Subroutine

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 33, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD6301Y0 as possible.

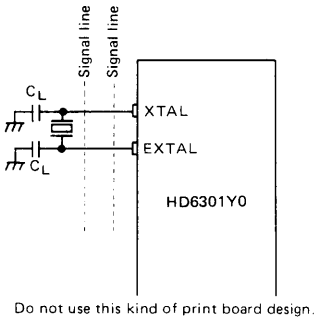


Figure 33 Precaution to the boad design of oscillation circuit

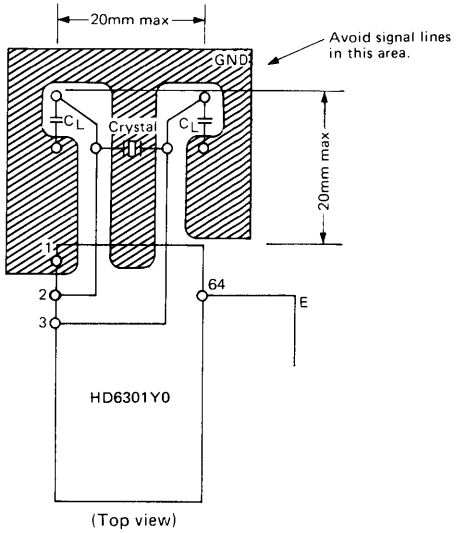


Figure 34 Example of Oscillation Circuits in Board Design

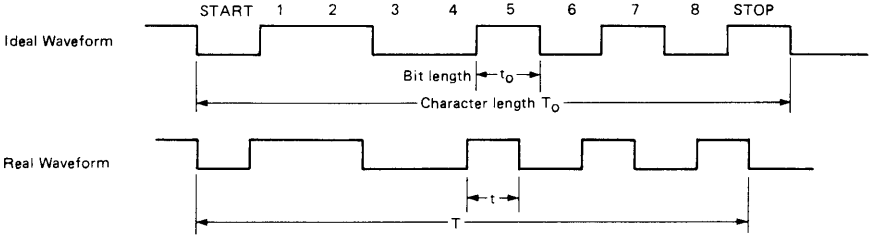
■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD6301Y0 is shown in Table 21.

Note: SCI = Serial Communication Interface.

Table 21

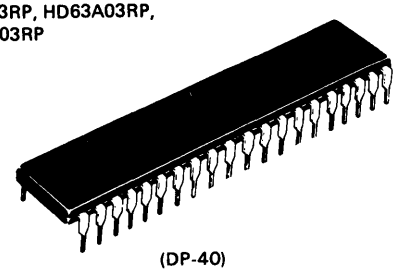
Bit distortion tolerance (t−to) /to	Character distortion tolerance (T−To) /To
±43.7%	±4.37%



HD6303R, HD63A03R, HD63B03R
CMOS MPU (Micro Processing Unit)

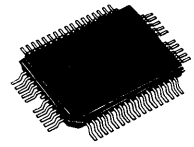
The HD6303R is an 8-bit CMOS micro processing unit which has the completely compatible instruction set with the HD6301V1. 128 bytes RAM, Serial Communication Interface (SCI), parallel I/O ports and multi function timer are incorporated in the HD6303R. It is bus compatible with HMCS6800 and can be expanded up to 65k bytes. Like the HMCS6800 family, I/O level is TTL compatible with +5.0V single power supply. As the HD6303R is CMOS MPU, power dissipation is extremely low. And also HD6303R has Sleep Mode and Stand-by Mode as lower power dissipation mode. Therefore, flexible low power consumption application is possible.

HD6303RP, HD63A03RP, HD63B03RP



(DP-40)

HD6303RF, HD63A03RF, HD63B03RF



(FP-54)

HD6303RCG, HD63A03RCG, HD63B03RCG

(CG-40)

■ FEATURES

- Object Code Upward Compatible with the HD6800, HD6801, HD6802
- Multiplexed Bus ($D_0/A_0 \sim D_7/A_7 A_8 \sim A_{15}$), Non Multiplexed Bus ($D_0 \sim D_7$, $A_0 \sim A_{15}$)
- Abundant On-Chip Functions Compatible with the HD6301V1; 128 Bytes RAM, 13 Parallel I/O Lines, 16-bit Timer, Serial Communication Interface (SCI)
- Low Power Consumption Mode; Sleep Mode, Stand-By Mode
- Minimum Instruction Execution Time
  1μs (f=1MHz), 0.67μs (f=1.5MHz), 0.5μs (f=2.0MHz)
- Bit Manipulation, Bit Test Instruction
- Error Detecting Function; Address Trap, Op Code Trap
- Up to 65k Bytes Address Space
- Wide Operation Range
  V_{CC} = 3 to 6V (f = 0.1 ~ 0.5 MHz)
  f = 0.1 to 2.0 MHz (V_{CC} = 5V ± 10%)

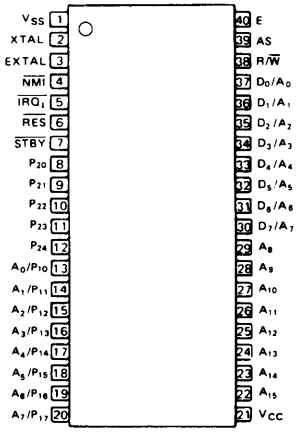
■ TYPE OF PRODUCTS

Type No.	Bus Timing
HD6303R	1.0 MHz
HD63A03R	1.5 MHz
HD63B03R	2.0 MHz

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

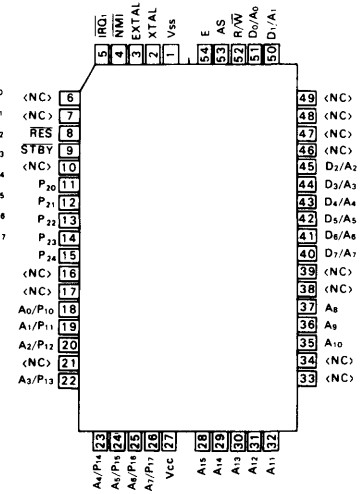
■ PIN ARRANGEMENT

- HD6303RP, HD63A03RP, HD63B03RP



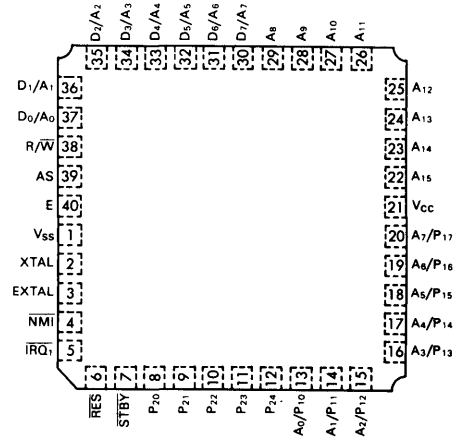
(Top View)

- HD6303RF, HD63A03RF, HD63B03RF



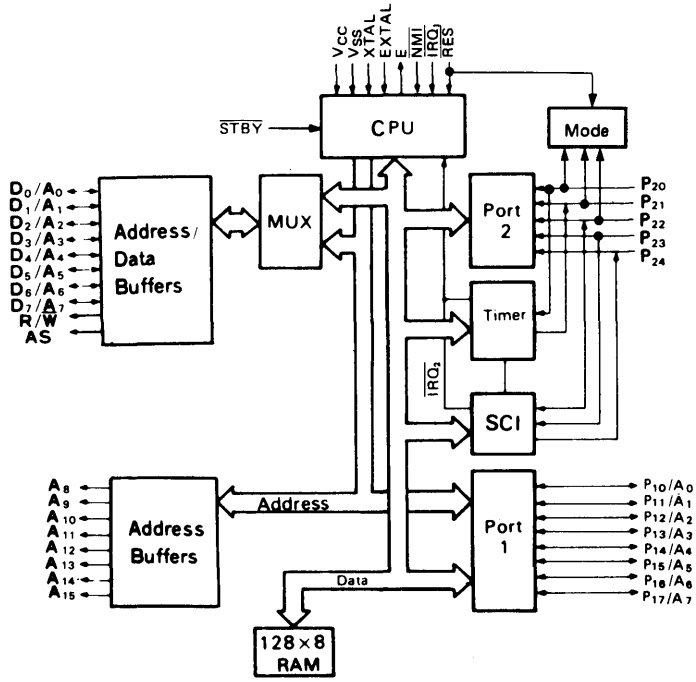
(Top View)

- HD6303RCG, HD63A03RCG, HD63B03RCG



(Top View)

■ BLOCK DIAGRAM



Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend $V_{in}, V_{out} : V_{SS} \leqq (V_{in} \text{ or } V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}x0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$, $\overline{STBY}$	$\|I_{in}\|$	V_{in} = 0.5~V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	P_{10}~P_{17}, P_{20}~P_{24}, D_0~D_7, A_8~A_{15}	$\|I_{TSI}\|$	V_{in} = 0.5~V_{CC}-0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input Capacitance	All Inputs	C_{in}	V_{in} = 0V, f = 1.0MHz, Ta = 25°C	–	–	12.5	pF
Standby Current	Non Operation	I_{CC}		–	2.0	15.0	μA
Current Dissipation*		I_{CC}	Operating (f=1MHz**)	–	6.0	10.0	mA
			Sleeping (f=1MHz**)	–	1.0	2.0	
RAM Stand-By Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at f = x MHz operation are decided according to the following formula;

typ. value (f = xMHz) = typ. value (f = 1MHz) x x
max. value (f = xMHz) = max. value (f = 1MHz) x x
(both the sleeping and operating)

- **AC CHARACTERISTICS** (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

BUS TIMING

Item		Symbol	Test Condition	HD6303R			HD63A03R			HD63B03R			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}		1	–	10	0.666	–	10	0.5	–	10	μs
Address Strobe Pulse Width * "High"		PW_{ASH}		220	–	–	150	–	–	110	–	–	ns
Address Strobe Rise Time		t_{ASr}		–	–	20	–	–	20	–	–	20	ns
Address Strobe Fall Time		t_{ASf}		–	–	20	–	–	20	–	–	20	ns
Address Strobe Delay Time *		t_{ASD}		60	–	–	40	–	–	20	–	–	ns
Enable Rise Time		t_{Er}		–	–	20	–	–	20	–	–	20	ns
Enable Fall Time		t_{Ef}		–	–	20	–	–	20	–	–	20	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address Strobe to Enable Delay* Time		t_{ASED}		60	–	–	40	–	–	20	–	–	ns
Address Delay Time		t_{AD1}		–	–	250	–	–	190	–	–	160	ns
		t_{AD2}	Fig. 1	–	–	250	–	–	190	–	–	160	ns
Address Delay Time for Latch*		t_{ADL}	Fig. 2	–	–	250	–	–	190	–	–	160	ns
Data Set-up Time	Write	t_{DSW}		230	–	–	150	–	–	100	–	–	ns
	Read	t_{DSR}		80	–	–	60	–	–	50	–	–	ns
Data Hold Time	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	20	–	–	ns
Address Set-up Time for Latch *		t_{ASL}		60	–	–	40	–	–	20	–	–	ns
Address Hold Time for Latch		t_{AHL}		30	–	–	20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	20	–	–	ns
$A_0 \sim A_7$ Set-up Time Before E*		t_{ASM}		200	–	–	110	–	–	60	–	–	ns
Peripheral Read Access Time	Non-Multiplexed Bus *	(t_{ACCN})		–	–	650	–	–	395	–	–	270	ns
	Multiplexed Bus*	(t_{ACCM})		–	–	650	–	–	395	–	–	270	ns
Oscillator stabilization Time		t_{RC}	Fig. 8	20	–	–	20	–	–	20	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 9	200	–	–	200	–	–	200	–	–	ns

*These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD6303R			HD63A03R			HD63B03R			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set-up Time	Port 1, 2	t_{PDSU}	Fig. 3	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2	t_{PDH}	Fig. 3	200	–	–	200	–	–	200	–	–	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*	t_{PWD}	Fig. 4	–	–	300	–	–	300	–	–	300	ns

* Except P_{21}

TIMER, SCI TIMING

Item	Symbol	Test Condition	HD6303R			HD63A03R			HD63B03R			Unit
			min	typ	max	min	typ	max	min	typ	max	
Timer Input Pulse Width	t_{PWT}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 5	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	t_{Scyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING

Item	Symbol	Test Condition	HD6303R			HD63A03R			HD63B03R			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{RES}$ "Low" Pulse Width	PW_{RSTL}	Fig. 6	3	–	–	3	–	–	3	–	–	t_{cyc}
Mode Programming Set-up Time	t_{MPS}		2	–	–	2	–	–	2	–	–	t_{cyc}
Mode Programming Hold Time	t_{MPH}		150	–	–	150	–	–	150	–	–	ns

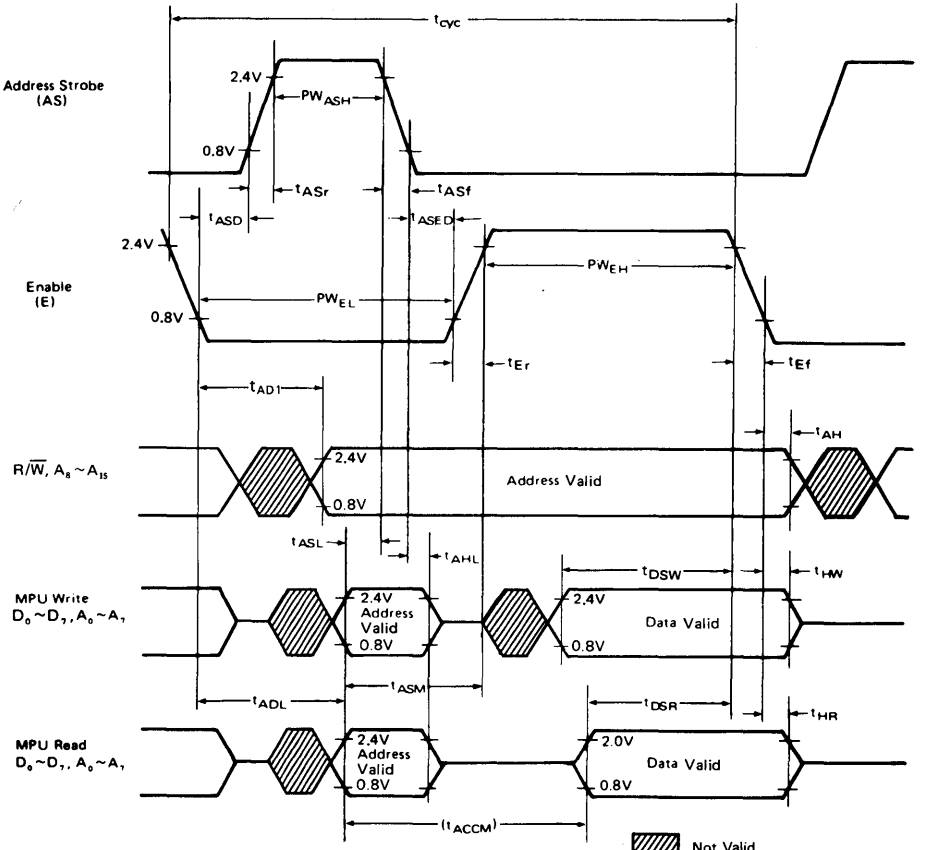


Figure 1 Multiplexed Bus Timing

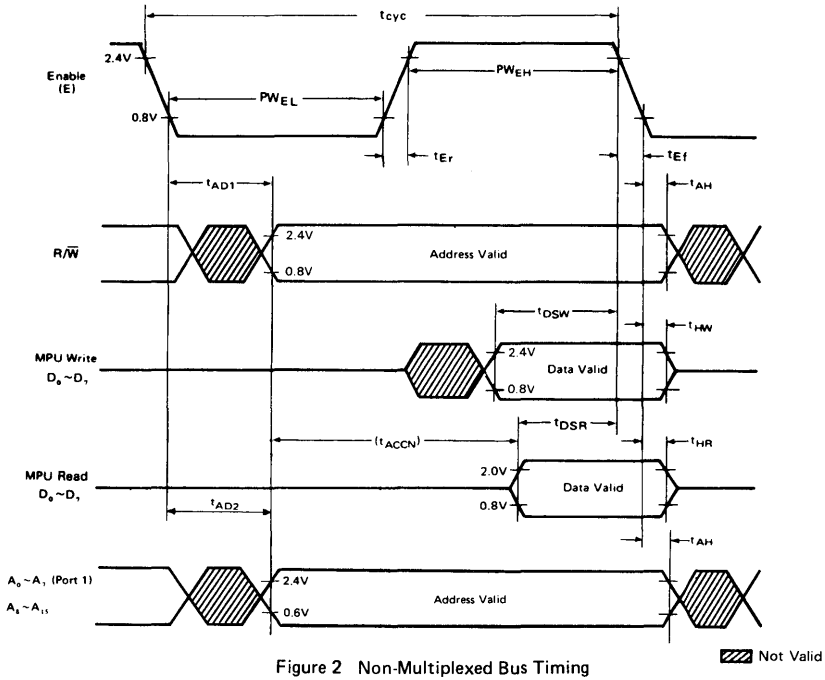


Figure 2 Non-Multiplexed Bus Timing

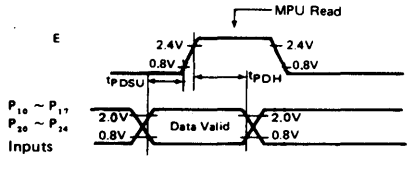
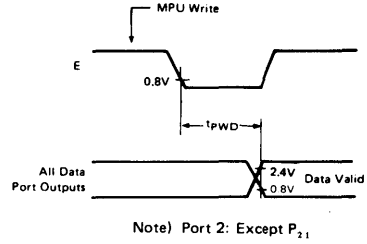


Figure 3 Port Data Set-up and Hold Times (MPU Read)



Note) Port 2: Except P_{21}

Figure 4 Port Data Delay Times (MPU Write)

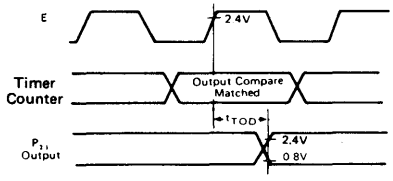


Figure 5 Timer Output Timing

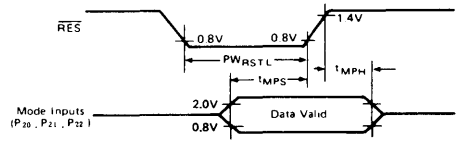


Figure 6 Mode Programming Timing

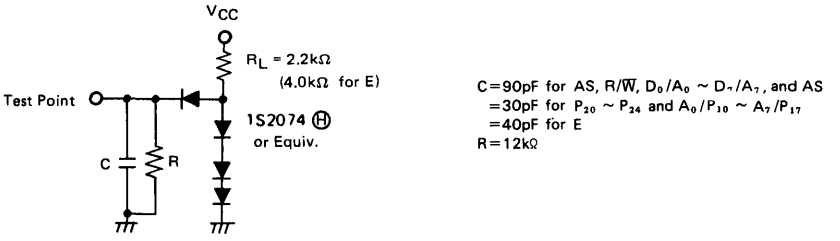


Figure 7 Bus Timing Test Loads (TTL Load)

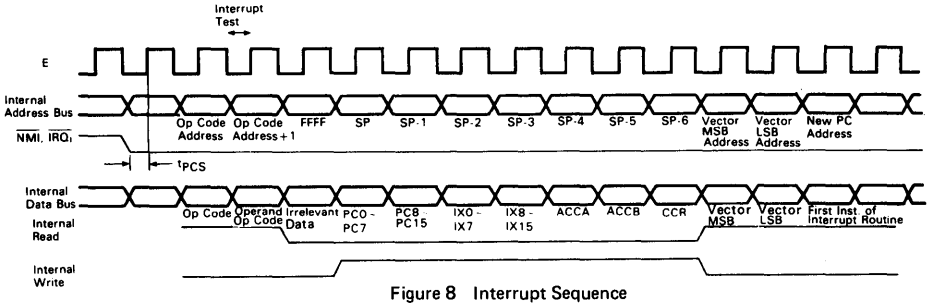


Figure 8 Interrupt Sequence

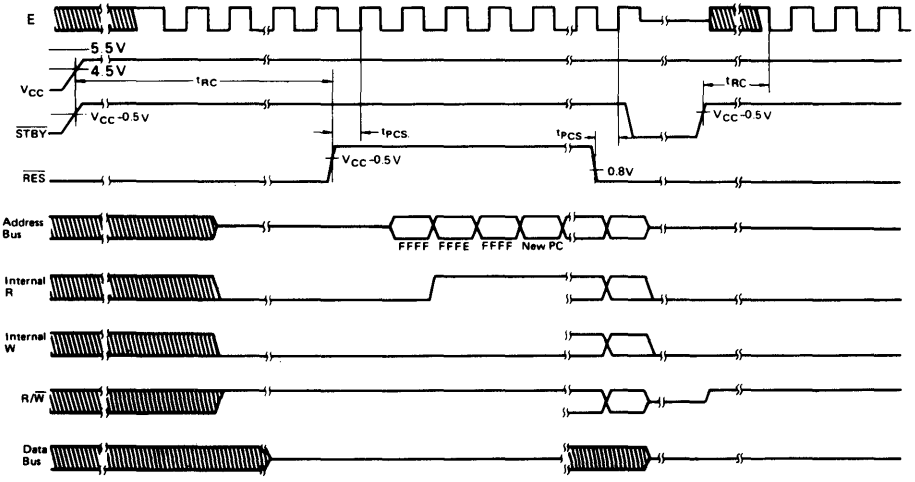


Figure 9 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

These two pins are used for power supply and GND. Recommended power supply voltage is 5V ± 10%. 3 to 6V can be used for low speed operation (100 ~ 500 kHz).

• XTAL, EXTAL

These two pins are connected with parallel resonant fundamental crystal, AT cut. For instance, in order to obtain the system clock 1MHz, a 4MHz resonant fundamental crystal is used because the devide-by-4 circuitry is included. An example of the crystal interface is shown in Fig. 10. EXTAL accepts an external clock input of duty 45% to 55% to drive. For external clock, XTAL pin should be open. The crystal and capacitors should be mounted as close as possible to the pins.

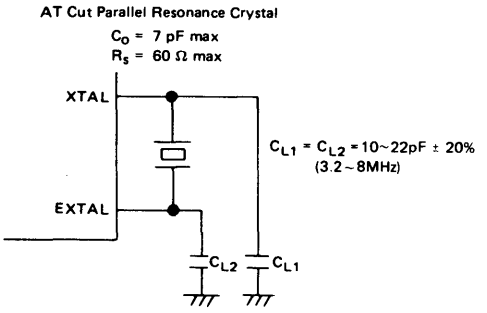


Figure 10 Crystal Interface

- **Standby ($\overline{STBY}$)**

This pin is used to place the MPU in the standby mode. If this goes to "Low" level, the oscillation stops, the internal clock is tied to V_{SS} or V_{CC} and the MPU is reset. In order to retain information in RAM during standby, write "0" into RAM enable bit (RAME). RAME is bit 6 of the RAM Control Register at address $0014. This disables the RAM, so the contents of RAM is guaranteed. For details of the standby mode, see the Standby section.

- **Reset ($\overline{RES}$)**

This input is used to reset the MPU. $\overline{RES}$ must be held "Low" for at least 20ms when the power starts up. It should be noted that, before clock generator stabilize, the internal state and I/O ports are uncertain, because MPU can not be reset without clock. To reset the MPU during system operation, it must be held "Low" for at least 3 system clock cycles. From the third cycle, all address buses become "high-impedance" and it continues while $\overline{RES}$ is "Low". If $\overline{RES}$ goes to "High", CPU does the following.

(1) I/O Port 2 bits 2,1,0 are latched into bits PC2, PC1, PC0 of program control register.
(2) The contents of the two Start Addresses, $FFFE, $FFFF are brought to the program counter, from which program starts (see Table 1).
(3) The interrupt mask bit is set. In order to have the CPU recognize the maskable interrupts $\overline{IRQ_1}$ and $\overline{IRQ_2}$, clear it before those are used.

- **Enable (E)**

This output pin supplies system clock. Output is a single-phase, TTL compatible and 1/4 the crystal oscillation frequency. It will drive two LS TTL load and 40pF capacitance.

- **Non Maskable Interrupt ($\overline{NMI}$)**

When the falling edge of the input signal of this pin is recognized, NMI sequence starts. The current instruction is continued to complete, even if $\overline{NMI}$ signal is detected. Interrupt mask bit in Condition Code Register has no effect on NMI detection. In response to NMI interrupt, the information of Program Counter, Index Register, Accumulators, and Condition Code Register are stored on the stack. On completion of this sequence, vectoring address $FFFC and $FFFD are generated to load the contents to the program counter. Then the CPU branch to a non maskable interrupt service routine.

- **Interrupt Request ($\overline{IRQ_1}$)**

This level sensitive input requests a maskable interrupt sequence. When $\overline{IRQ_1}$ goes to "Low", the CPU waits until it completes the current instruction that is being executed. Then, if the interrupt mask bit in Condition Code Register is not set, CPU begins interrupt sequence; otherwise, interrupt request is neglected.

Once the sequence has started, the information of Program Counter, Index Register, Accumulator, Condition Code Register are stored on the stack. Then the CPU sets the interrupt mask bit so that no further maskable interrupts may be responded.

Table 1 Interrupt Vectoring memory map

	Vector		Interrupt
	MSB	LSB	
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$ (or $\overline{IS3}$)
	FFF6	FFF7	ICF (Timer Input Capture)
	FFF4	FFF5	OCF (Timer Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

At the end of the cycle, the CPU generates 16 bit vectoring addresses indicating memory addresses $FFF8 and $FFF9, and loads the contents to the Program Counter, then branch to an interrupt service routine.

The Internal Interrupt will generate signal ($\overline{IRQ_2}$) which is quite the same as $\overline{IRQ_1}$ except that it will use the vector address $FFF0 to $FFF7.

When $\overline{IRQ_1}$ and $\overline{IRQ_2}$ are generated at the same time, the former precedes the latter. Interrupt Mask Bit in the condition code register, if being set, will keep the both interrupts off.

On occurrence of Address error or Op-code error, TRAP interrupt is invoked. This interrupt has priority next to $\overline{RES}$. Regardless of the interrupt Mask Bit condition, the CPU will start an interrupt sequence. The vector for this interrupt will be $FFEE, $FFEF.

- **Read/Write (R/$\overline{W}$)**

This TTL compatible output signals peripheral and memory devices whether CPU is in Read ("High"), or in Write ("Low"). The normal stand-by state is Read ("High"). Its output will drive one TTL load and 90pF capacitance.

- **Address Strobe (AS)**

In the multiplexed mode, address strobe signal appears at this pin. It is used to latch the lower 8 bits addresses multiplexed with data at $D_0/A_0 \sim D_7/A_7$. The 8-bit latch is controlled by address strobe as shown in Figure 15. Thereby, $D_0/A_0 \sim D_7/A_7$ can become data bus during E pulse. The timing chart of this signal is shown in Figure 1.

Address Strobe (AS) is sent out even if the internal address is accessed.

■ PORTS

There are two I/O ports on HD6303R MPU (one 8-bit ports and one 5-bit port). Each port has an independent write-only data direction register to program individual I/O pins for input or output.*

When the bit of associated Data Direction Register is "1", I/O pin is programmed for output, if "0", then programmed for

an input.

There are two ports : Port 1, Port 2. Addresses of each port and associated Data Direction Register are shown in Table 2.

* Only one exception is bit 1 of Port 2 which becomes either a data input or a timer output. It cannot be used as an output port.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001

- **I/O Port 1**

This is an 8-bit port, each bit being defined individually as input or outputs by associated Data Direction Register. The 8-bit output buffers have three-state capability, maintaining in high impedance state when they are used for input. In order to be read accurately, the voltage on the input lines must be more than 2.0V for logic "1" and less than 0.8V for logic "0".

These are TTL compatible. After the MPU has been reset, all I/O lines are configured as inputs in Multiplexed mode. In Non Multiplexed mode, Port 1 will be output line for lower order address lines ($A_0 \sim A_7$), which can drive one TTL load and 30 pF capacitance.

- **I/O Port 2**

This port has five lines, whose I/O direction depends on its data direction register. The 5-bit output buffers have three-state capability, going high impedance state when used as inputs. In order to be read accurately, the voltage on the input lines must be more than 2.0V for logic "1" and less than 0.8V for logic "0". After the MPU has been reset, I/O lines are configured as inputs. These pins on Port 2 ($P_{20} \sim P_{22}$ of the chip) are used to program the mode of operation during reset. The values of these three pins during reset are latched into the upper 3 bits (bit 7, 6 and 5) of Port 2 Data Register which is explained in the MODE SELECTION section.

In all modes, Port 2 can be configured as I/O lines. This port also provides access to the Serial I/O and the Timer. However, note that bit 1 (P_{21}) is the only pin restricted to data input or Timer output.

- **BUS**
- **$D_0/A_0 \sim D_7/A_7$**

This TTL compatible three-state buffer can drive one TTL load and 90 pF capacitance.

Non Multiplexed Mode

In this mode, these pins become only data bus ($D_0 \sim D_7$).

Multiplexed Mode

These pins becomes both the data bus ($D_0 \sim D_7$) and lower bits of the address bus ($A_0 \sim A_7$). An address strobe output is "High" when the address is on the pins.

- **$A_8 \sim A_{15}$**

Each line is TTL compatible and can drive one TTL load and 90 pF capacitance. After reset, these pins become output for upper order address lines ($A_8 \sim A_{15}$).

- **MODE SELECTION**

The operation mode after the reset must be determined by the user wiring the P_{20}, P_{21}, and P_{22} externally. These three pins are lower order bits; I/O 0, I/O 1, I/O 2 of Port 2. They are latched into the control bits PC0, PC1, PC2 of I/O Port 2 register when $\overline{RES}$ goes "High". I/O Port 2 Register is shown below.

Port 2 DATA REGISTER

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

An example of external hardware used for Mode Selection is shown in Figure 11. The HD14053B is used to separate the peripheral device from the MPU during reset. It is necessary if the data may conflict between peripheral device and Mode generation circuit.

No mode can be changed through software because the bits 5, 6, and 7 of Port 2 Data Register are read-only. The mode selection of the HD6303R is shown in Table 3.

The HD6303R operates in two basic modes: (1) Multiplexed Mode, (2) Non Multiplexed Mode.

- **Multiplexed Mode**

The data bus and the lower order address bus are multiplexed in the $D_0/A_0 \sim D_7/A_7$ and can be separated by the Address Strobe.

Port 2 is configured for 5 parallel I/O or Serial I/O, or Timer, or any combination thereof. Port 1 is configured for 8 parallel I/O.

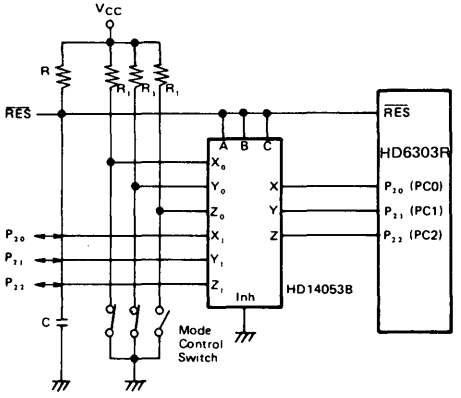
- **Non Multiplexed Mode**

In this mode, the HD6303R can directly address HMCS6800 peripherals with no address latch. $D_0/A_0 \sim D_7/A_7$ become a data bus and Port 1 becomes $A_0 \sim A_7$ address bus.

In this mode, the HD6303R is expandable up to 65k bytes with no address latch.

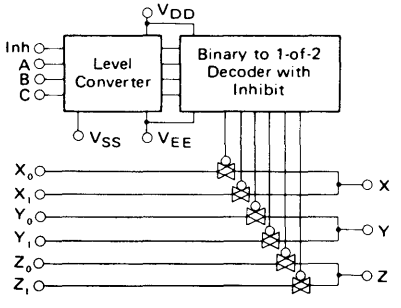
- **Lower Order Address Bus Latch**

Because the data bus is multiplexed with the lower order address bus in $D_0/A_0 \sim D_7/A_7$ in the multiplexed mode, address bits must be latched. It requires the 74LS373 Transparent octal D-type to latch the LSB. Latch connection of the HD6303R is shown in Figure 15.



Note 1) Figure of Multiplexed Mode
2) RC≈Reset Constant
3) R_1 = 10kΩ

Figure 11 Recommended Circuit for Mode Selection



Truth Table

Control Input				On Switch
Inhibit	Select			
	C	B	A	HD14053B
0	0	0	0	Z_0 Y_0 X_0
0	0	0	1	Z_0 Y_0 X_1
0	0	1	0	Z_0 Y_1 X_0
0	0	1	1	Z_0 Y_1 X_1
0	1	0	0	Z_1 Y_0 X_0
0	1	0	1	Z_1 Y_0 X_1
0	1	1	0	Z_1 Y_1 X_0
0	1	1	1	Z_1 Y_1 X_1
1	X	X	X	

Figure 12 HD14053B Multiplexers/De-Multiplexers

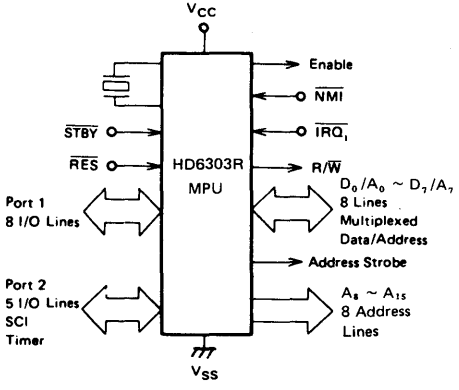


Figure 13 HD6303R MPU Multiplexed Mode

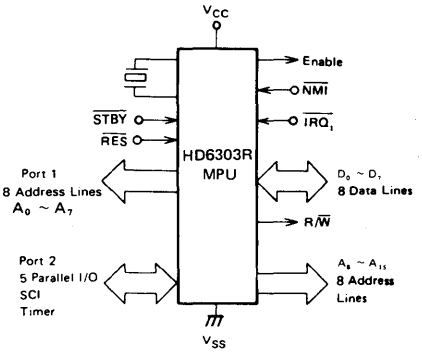
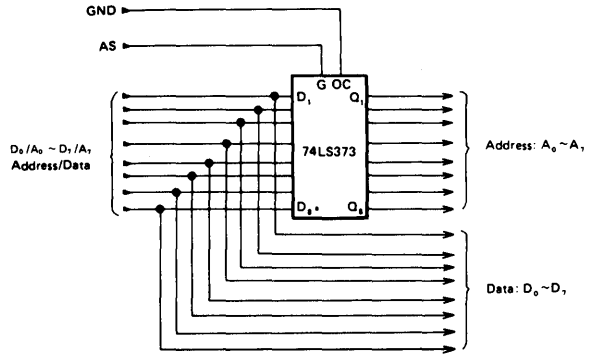


Figure 14 HD6303R MPU Non Multiplexed Mode



Function Table

Output Control (OC)	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 15 Latch Connection

Table 3 Mode Selection

Operating Mode	P_{20}	P_{21}	P_{22}
Multiplexed Mode	L	H	L
	L	L	H
Non Multiplexed Mode	H	L	L

L: logic "0"
H: logic "1"

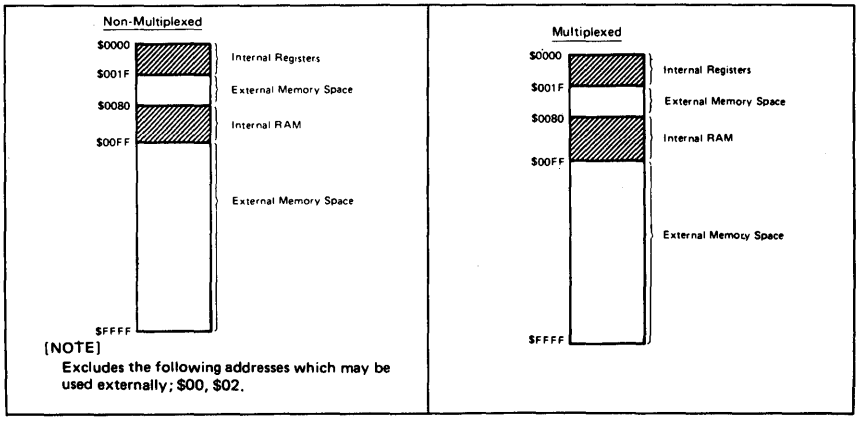
■ MEMORY MAP

The MPU can provide up to 65k byte address space. Figure 16 shows a memory map for each operating mode. The first 32 locations of each map are for the CPU's internal register only, as shown in Table 4.

Table 4 Internal Register Area

Register	Address
Port 1 Data Direction Register**	00*
Port 2 Data Direction Register**	01
Port 1 Data Register	02*
Port 2 Data Register	03
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External address in Non Multiplexed Mode
** 1 = Output, 0 = Input



[NOTE]
Excludes the following addresses which may be used externally; $00, $02.

Figure 16 HD6303R Memory Maps

■ PROGRAMMABLE TIMER

The HD6303R contains 16-bit programmable timer which may measure input waveform. In addition to that it can generate an output waveform by itself. For both input and output waveform, the pulse width may vary from a few microseconds to several seconds.

The timer hardware consists of

- an 8-bit control and status register
- a 16-bit free running counter
- a 16-bit output compare register
- a 16-bit input capture register

A block diagram of the timer is shown in Figure 17.

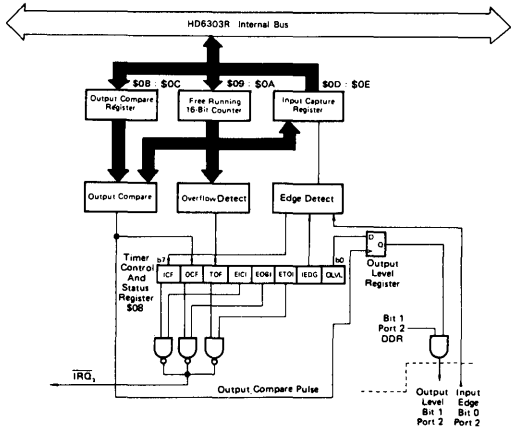


Figure 17 Programmable Timer Block Diagram

• Free Running Counter ($0009: $000A)

The key element in the programmable timer is a 16-bit free running counter, that is driven by an E (Enable) clock to increment its values. The counter value will be read out by the CPU software at any time with no effects on the counter. Reset will clear the counter.

When the MSB of this counter is read, the LSB is stored in temporary latch. The data is fetched from this latch by the subsequent read of LSB. Thus consistent double byte data can be read from the counter.

When the CPU writes arbitrary data to the MSB ($09), the value of $FFF8 is being pre-set to the counter ($09, $0A) regardless of the write data value. Then the CPU writes arbitrary data to the LSB ($0A), the data is set to the "Low" byte of the counter, at the same time, the data preceedingly written in the MSB ($09) is set to "High" byte of the counter.

When the data is written to this counter, a double byte store instruction (ex. STD) must be used. If only the MSB of counter is written, the counter is set to $FFF8.

The counter value written to the counter using the double byte store instruction is shown in Figure 18.

To write to the counter can disturb serial operations, so it should be inhibited during using the SCI. If external clock mode is used for SCI, this will not disturb serial operations.

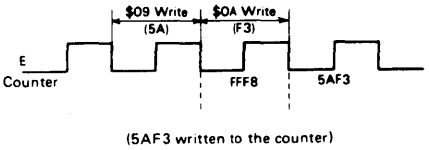


Figure 18 Counter Write Timing

• Output Compare Register ($000B:$000C)

This is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly being compared with current value of the free running counter.

When the contents match with the value of the free running counter, a flag (OCF) in the timer control/status register (TCSR) is set and the current value of an output level Bit (OLVL) in the TCSR is transferred to Port 2 bit 1. When bit 1 of the Port 2 data direction register is "1" (output), the OLVL value will appear on the bit 1 of Port 2. Then, the value of Output Compare Register and Output level bit may be changed for the next compare.

The output compare register is set to $FFFF during reset.

The compare function is inhibited at the cycle of writing to the high byte of the output compare register and at the cycle just after that to ensure valid compare. It is also inhibited in same manner at writing to the free running counter.

In order to write a data to Output Compare Register, a double byte store instruction (ex.STD) must be used.

• Input Capture Register ($000D: $000E)

The input capture register is a 16-bit read-only register used to hold the current value of free running counter captured when the proper transition of an external input signal occurs.

The input transition change required to trigger the counter transfer is controlled by the input edge bit (IEDG).

To allow the external input signal to go in the edge detect unit, the bit of the Data Direction Register corresponding to bit 0 of Port 2 must have been cleared (to zero).

To insure input capture in all cases, the width of an input pulse requires at least 2 Enable cycles.

• Timer Control/Status Register (TCSR) ($0008)

This is an 8-bit register. All 8-bits are readable and the lower 5 bits may be written. The upper 3 bits are read-only, indicating the timer status information as is shown below.

(1) A proper transition has been detected on the input pin (ICF).
(2) A match has been found between the value in the free running counter and the output compare register (OCF).
(3) When counting up to $0000 (TOF).

Each flag has an individual enable bit in TCSR which determines whether or not an interrupt request may occur ($\overline{IRQ_2}$). If the I-bit in Condition Code Register has been cleared, a prior vectored address occurs corresponding to each flag. A description of each bit is as follows.

Timer Control / Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

Bit 0 OLVL (Output Level); When a match is found in the value between the counter and the output com-

pare register, this bit is transferred to the Port 2 bit 1. If the DDR corresponding to Port 2 bit 1 is set "1", the value will appear on the output pin of Port 2 bit 1.

Bit 1 IEDG (Input Edge): This bit control which transition of an input of Port 2 bit 0 will trigger the data transfer from the counter to the input capture register. The DDR corresponding to Port 2 bit 0 must be clear in advance of using this function. When IEDG = 0, trigger takes place on a negative edge ("High"-to-"Low" transition). When IEDG = 1, trigger takes place on a positive edge ("Low"-to-"High" transition).

Bit 2 ETOI (Enable Timer Overflow Interrupt); When set, this bit enables TOF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 3 EOCI (Enable Output Compare Interrupt); When set, this bit enables OCF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 4 EICI (Enable Input Capture Interrupt); When set, this bit enables ICF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 5 TOF (Timer Over Flow Flag); This read-only bit is set at the transition of $FFFF to $0000 of the counter. It is cleared by CPU read of TCSR (with TOF set) followed by a CPU read of the counter ($0009).

Bit 6 OCF (Output Compare Flag); This read-only bit is set when a match is found in the value between the output compare register and the counter. It is cleared by a read of TCSR (with OCF set) followed by a CPU write to the output compare register ($000B or $000C).

Bit 7 ICF (Input Capture Flag); The read-only bit is set by a proper transition on the input, and is cleared by a read of TCSR (with ICF set) followed by a CPU read of Input Capture Register ($000D).

Reset will clear each bit of Timer Control and Status Register.

■ SERIAL COMMUNICATION INTERFACE

The **HD6303R** contains a full-duplex asynchronous Serial Communication Interface (SCI). SCI may select the several kinds of the data rate. It consists of a transmitter and a receiver which operate independently but with the same data format and the same data rate. Both of transmitter and receiver communicate with the CPU via the data bus and with the outside world through Port 2 bit 2, 3 and 4. Description of hardware, software and register is as follows.

• Wake-Up Feature

In typical multiprocessor applications the software protocol will usually have the designated address at the initial byte of the message. The purpose of Wake-Up feature is to have the non-selected MPU neglect the remainder of the message. Thus the non-selected MPU can inhibit the all further interrupt process until the next message begins.

Wake-Up feature is re-enabled by a ten consecutive "1"s which indicates an idle transmit line. Therefore software protocol must put an idle period between the messages and must prevent it within the message.

With this hardware feature, the non-selected MPU is re-enabled (or "waked-up") by the next message.

• Programmable Options

The HD6303R has the following programmable features.

- data format; standard mark/space (NRZ)
- clock source; external or internal
- baud rate; one of 4 rates per given E clock frequency or 1/8 of external clock
- wake-up feature; enabled or disabled
- interrupt requests; enabled or masked individually for transmitter and receiver
- clock output; internal clock enabled or disabled to Port 2 bit 2
- Port 2 (bits 3, 4); dedicated or not dedicated to serial I/O individually

• Serial Communication Hardware

The serial communications hardware is controlled by 4 registers as shown in Figure 19. The registers include:

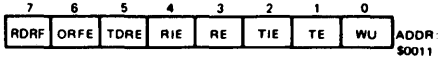
- an 8-bit control/status register
- a 4-bit rate/mode control register (write-only)
- an 8-bit read-only receive data register
- an 8-bit write-only transmit data register

Besides these 4 registers, Serial I/O utilizes Port 2 bit 3 (input) and bit 4 (output). Port 2 bit 2 can be used when an option is selected for the internal-clock-out or the external-clock-in.

• Transmit/Receive Control Status Register (TRCSR)

TRCS Register consists of 8 bits which all may be read while only bits 0 to 4 may be written. The register is initialized to $20 on $\overline{RES}$. The bits of the TRCS Register are explained below.

Transmit / Receive Control Status Register



Bit 0 WU (Wake Up); Set by software and cleared by hardware on receipt of ten consecutive "1"s. While this bit is "1", RDRF and ORFE flags are not set even if data are received or errors are detected. Therefore received data are ignored. It should be noted that RE flag must have already been set in advance of WU flag's set.

Bit 1 TE (Transmit Enable); This bit enables transmitter. When this bit is set, bit 4 of Port 2 DDR is also forced to be set. It remains set even if TE is cleared. Preamble of ten consecutive "1"s is transmitted just after this bit is set, and then transmitter becomes ready to send data. If this bit is cleared, the transmitter is disabled and serial I/O affects nothing on Port 2 bit 4.

Bit 2 TIE (Transmit Interrupt Enable); When this bit is set, TDRE (bit 5) causes an $\overline{IRQ_2}$ interrupt. When cleared, TDRE interrupt is masked.

Bit 3 RE (Receive Enable); When set, Port 2 bit 3 can be used as an input of receive regardless of DDR value for this bit. When cleared, the receiver is disabled.

Bit 4 RIE (Receive Interrupt Enable); When this bit is set, RDRF (bit 7) or ORFE (bit 6) cause an $\overline{IRQ_2}$ interrupt. When cleared, this interrupt is masked.

Bit 5 TDRE (Transmit Data Register Empty); When the data is transferred from the Transmit Data Register to Output Shift Register, this bit is set by hardware. The bit is cleared by reading the status register followed by writing the next new data into the Transmit Data Register. TDRE is initialized to 1 by $\overline{RES}$.

Bit 6 ORFE (Over Run Framing Error); When overrun or framing error occurs (receive only), this bit is set by hardware. Over Run Error occurs if the attempt is made to transfer the new byte to the receive data register while the RDRF is "1". Framing Error occurs when the bit counter is not synchronized with the boundary of the byte in the receiving bit stream. When Framing Error is detected, RDRF is not set. Therefore Framing Error can be distinguished from Overrun Error. That is, if ORFE is "1" and RDRF is "1", Overrun Error is detected. Otherwise Framing Error occurs. The bit is cleared by reading the status register followed by reading the receive data register, or by $\overline{RES}$.

Bit 7 RDRF (Receive Data Register Full); This bit is set by hardware when the data is transferred from the receive shift register to the receive data register. It is cleared by reading the status register followed by reading the receive data register, or by $\overline{RES}$.

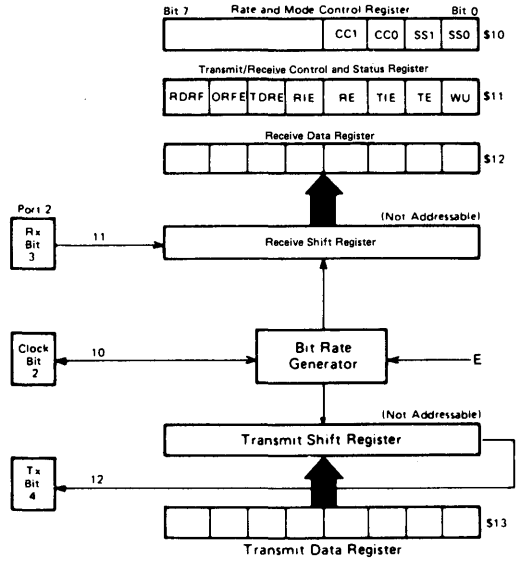
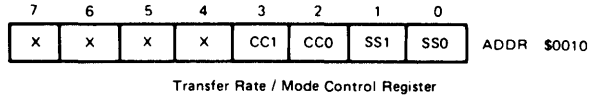


Figure 19 Serial I/O Register



Transfer Rate / Mode Control Register

Table 5 SCI Bit Times and Transfer Rates

		XTAL	2.4576 MHz	4.0 MHz	4.9152MHz
SS1	SS0	E	614.4 kHz	1.0 MHz	1.2288MHz
0	0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800Baud
0	1	E ÷ 128	208μs/4,800 Baud	128 μs/7812.5 Baud	104.2μs/ 9,600Baud
1	0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud	833.3μs/ 1,200Baud
1	1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.333ms/ 300Baud

Table 6 SCI Format and Clock Source Control

CC1	CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0	0	–	–	–	–	–
0	1	NRZ	Internal	Not Used***	••	••
1	0	NRZ	Internal	Output*	••	••
1	1	NRZ	External	Input	••	••

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS.
Bit 4 is used for serial output if TE = "1" in TRCS.
*** This pin can be used as I/O port.

- **Transfer Rate/Mode Control Register (RMCR)**

The register controls the following serial I/O functions:
- Bauds rate
- data format
- clock source
- Port 2 bit 2 feature

It is 4-bit write-only register, cleared by $\overline{RES}$. The 4 bits are considered as a pair of 2-bit fields. The lower 2 bits control the bit rate of internal clock while the upper 2 bits control the format and the clock select logic.

Bit 0 SS0, Bit 1 SS1 } Speed Select

These bits select the Baud rate for the internal clock. The rates selectable are function of E clock frequency of the CPU. Table 5 lists the available Baud Rates.

Bit2 CC0, Bit 3 CC1 } Clock Control/Format Select

They control the data format and the clock select logic. Table 6 defines the bit field.

- **Internally Generated Clock**

If the user wish to use externally an internal clock of the serial I/O, the following requirements should be noted.
- CC1, CC0 must be set to "10".
- The maximum clock rate must be E/16.
- The clock rate is equal to the bit rate.
- The values of RE and TE have no effect.

- **Externally Generated Clock**

If the user wish to supply an external clock to the Serial I/O, the following requirements should be noted.
- The CC1, CC0 must be set to "11" (See Table 6).
- The external clock must be set to 8 times of the desired baud rate.
- The maximum external clock frequency is E/2 clock.

- **Serial Operations**

The serial I/O hardware must be initialized by the software before operation. The sequence will be normally as follows.
- Writing the desired operation control bits of the Rate and Mode Control Register.
- Writing the desired operation control bits of the TRCS register.

If Port 2 bit 3, 4 are used for serial I/O, TE, RE bits may be kept set. When TE, RE bit are cleared during SCI operation, and subsequently set again, it should be noted that TE, RE must be kept "0" for at least one bit time of the current baud rate. If TE, RE are set again within one bit time, there may be the case where the initializing of internal function for transmitter and receiver does not take place correctly.

- **Transmit Operation**

Data transmission is enabled by the TE bit in the TRCS register. When set, the output of the transmit shift register is connected with Port 2 bit 4 which is unconditionally configured as an output.

After $\overline{RES}$, the user should initialize both the RMC register and the TRCS register for desired operation. Setting the TE bit causes a transmission of ten-bit preamble of "1"s. Following the preamble, internal synchronization is established and the transmitter is ready to operate. Then either of the following states exists.

(1) If the transmit data register is empty (TDRE = 1), the consecutive "1"s are transmitted indicating an idle states.
(2) If the data has been loaded into the Transmit Data Register (TDRE = 0), it is transferred to the output shift register and data transmission begins.

During the data transfer, the start bit ("0") is first transferred. Next the 8-bit data (beginning at bit 0) and finally the stop bit ("1"). When the contents of the Transmit Data Register is transferred to the output shift register, the hardware sets the TDRE flag bit: If the CPU fails to respond to the flag within the proper time, TDRE is kept set and then a continuous string of 1's is sent until the data is supplied to the data register.

- **Receive Operation**

The receive operation is enabled by the RE bit. The serial input is connected with Port 2 bit 3. The receiver operation is determined by the contents of the TRCS and RMC register. The received bit stream is synchronized by the first "0" (start bit). During 10-bit time, the data is strobed approximately at the center of each bit. If the tenth bit is not "1" (stop bit), the system assumes a framing error and the ORFE is set.

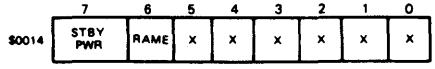
If the tenth bit is "1", the data is transferred to the receive data register, and the RDRF flag is set. If the tenth bit of the next data is received and still RDRF is preserved set, then ORFE is set indicating that an overrun error has occurred.

After the CPU read of the status register as a response to RDRF flag or ORFE flag, followed by the CPU read of the receive data register, RDRF or ORFE will be cleared.

■ RAM CONTROL REGISTER

The register assigned to the address $0014 gives a status information about standby RAM.

RAM Control Register



Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.

Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 RAM Enable (RAME).

Using this control bit, the user can disable the RAM. RAM Enable bit is set on the positive edge of $\overline{RES}$ and RAM is enabled. The program can write "1" or "0". If RAME is cleared, the RAM address becomes external address and the CPU may read the data from the outside memory.

Bit 7 Standby Power Bit (STBY PWR)

This bit can be read or written by the user program. It is cleared when the V_{CC} voltage is removed. Normally this bit is set by the program before going into stand-by mode. When the CPU recovers from stand-by mode, this bit should be checked. If it is "1", the data of the RAM is retained during stand-by and it is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD6303R has an upward object code compatible with the HD6801 to utilize all instruction sets of the HMCS6800. The execution time of the key instruction is reduced to increase the system through-put. In addition, the bit operation instruction, the exchange instruction between the index and the accumulator, the sleep instruction are added. This section describes:

- CPU programming model (See Fig. 20)
- Addressing modes
- Accumulator and memory manipulation instructions (See Table 7)
- New instructions
- Index register and stack manipulation instructions (See Table 8)
- Jump and branch instructions (See Table 9)
- Condition code register manipulation instructions (See Table 10)
- Op-code map (See Table 11)
- Cycle-by-cycle operation (See Table 12)

• CPU Programming Model

The programming model for the HD6303R is shown in Figure 20. The double accumulator is physically the same as the accumulator A concatenated with the accumulator B, so that the contents of A and B is changed with executing operation of an accumulator D.

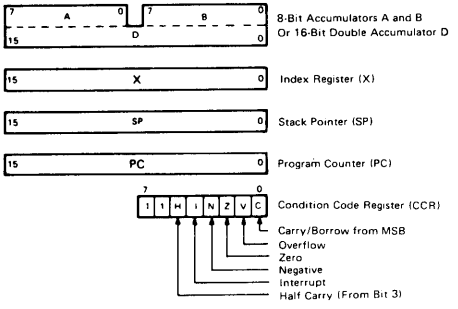


Figure 20 CPU Programming Model

• CPU Addressing Modes

The HD6303R has seven address modes which depend on both of the instruction type and the code. The address mode for every instruction is shown along with execution time given in terms of machine cycles (Table 7 to 11). When the clock frequency is 4 MHz, the machine cycle will be microseconds.

Accumulator (ACCX) Addressing

Only the accumulator (A or B) is addressed. Either accumulator A or B is specified by one-byte instructions.

Immediate Addressing

In this mode, the operand is stored in the second byte of the instruction except that the operand in LDS and LDX, etc are stored in the second and the third byte. These are two or three-byte instructions.

Direct Addressing

In this mode, the second byte of instruction indicates the address where the operand is stored. Direct addressing allows the user to directly address the lowest 256 bytes in the machine; locations zero through 255. Improved execution times are achieved by storing data in these locations. For system configuration, it is recommended that these locations should be RAM and be utilized preferably for user's data realm. These are two-byte instructions except the AIM, OIM, EIM and TIM which have three-byte.

Extended Addressing

In this mode, the second byte indicates the upper 8 bit addresses where the operand is stored, while the third byte indicates the lower 8 bits. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the Index Register. For each of AIM, OIM, EIM and TIM instructions, the contents of the third byte are added to the lower 8 bits in the Index Register. In addition, the resulting "carry" is added to the upper 8 bits in the Index Register. The result is used for addressing memory. Because the modified address is held in the Temporary Address Register, there is no change to the Index Register. These are two-byte instructions but AIM, OIM, EIM, TIM have three-byte.

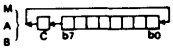
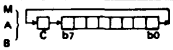
Implied Addressing

In this mode, the instruction itself gives the address; stack pointer, index register, etc. These are 1-byte instructions.

Relative Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the program counter. The resulting carry or borrow is added to the upper 8 bits. This helps the user to address the data within a range of −126 to +129 bytes of the current execution instruction. These are two-byte instructions.

Table 7 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M→ A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M→ B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B→ A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C→ A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C→ B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M→ A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M→ B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M→ B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M→ A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M→ B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

Note) Condition Code Register will be explained in Note of Table 10.

(to be continued)

Table 7 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes														Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑤	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑤	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑤	↕
Double Shift Left, Arithmetic	ASLD													05	1	1		•	•	↕	↕	⑤	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑤	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑤	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑤	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑤	↕
	LSRA													44	1	1	A	•	•	R	↕	⑤	↕
	LSRB													54	1	1	B	•	•	R	↕	⑤	↕
Double Shift Right Logical	LSRD													04	1	1		•	•	R	↕	⑤	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

Note) Condition Code Register will be explained in Note of Table 10.

- **New Instructions**

In addition to the HD6801 Instruction Set, the HD6303R has the following new instructions:

AIM----(M) · (IMM) → (M)
Evaluates the AND of the immediate data and the memory, places the result in the memory.

OIM----(M) + (IMM) → (M)
Evaluates the OR of the immediate data and the memory, places the result in the memory.

EIM----(M) ⊕ (IMM) → (M)
Evaluates the EOR of the immediate data and the memory, places the result in the memory.

TIM----(M) · (IMM)
Evaluates the AND of the immediate data and the memory, changes the flag of associated condition code register

Each instruction has three bytes; the first is op-code, the second is immediate data, the third is address modifier.

XGDX--(ACCD) ↔ (IX)
Exchanges the contents of accumulator and the index register.

SLP----The MPU is brought to the sleep mode. For sleep mode, see the "sleep mode" section.

Table 8 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes													Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD↔IX	•	•	•	•	•	•

Note) Condition Code Register will be explained in Note of Table 10.

Table 9 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes													Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED			5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		———		⑧		———	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑧	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

Note) *WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 10.

Table 10 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	——— ⑩ ———					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

[NOTE 1] Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 9? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕C=1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1 of ACCB?

[NOTE 2] CLI instruction and interrupt.
If interrupt mask-bit is set (I="1") and interrupt is requested ($\overline{IRQ_1}$ = "0" or $\overline{IRQ_2}$ = "0"), and then CLI instruction is executed, the CPU responds as follows.

① The next instruction of CLI is one-machine cycle instruction.
Subsequent two instructions are executed before the interrupt is responded.
That is, the next and the next of the next instruction are executed.
② The next instruction of CLI is two-machine cycle (or more) instruction.
Only the next instruction is executed and then the CPU jump to the interrupt routine.
Even if TAP instruction is used, instead of CLI, the same thing occurs.

Table 11 OP-Code Map

OP CODE HI \ LO		0 0000	1 0001	2 0010	3 0011	ACC A 4 0100	ACC B 5 0101	IND 6 0110	EXT / DIR 7 0111	ACCA or SP IMM 8 1000	DIR 9 1001	IND A 1010	EXT B 1011	ACCB or X IMM C 1100	DIR D 1101	IND E 1110	EXT F 1111	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE ▨
* Only for instructions of AIM, OIM, EIM, TIM

- **Instruction Execution Cycles**

In the HMCS6800 series, the execution cycle of each instruction is the number of cycles between the start of the current instruction fetch and just before the start of the subsequent instruction fetch.

The HD6303R uses a mechanism of the pipeline control for the instruction fetch and the subsequent instruction fetch is performed during the current instruction being executed.

Therefore, the method to count instruction cycles used in the HMCS6800 series cannot be applied to the instruction cycles such as MULT, PULL, DAA and XGDX in the HD6303R.

Table 12 provides the information about the relationship among each data on the Address Bus, Data Bus, and R/$\overline{W}$ status in cycle-by-cycle basis during the execution of each instruction.

Table 12 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMMEDIATE					
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	2	1 2	Op Code Address+1 Op Code Address+2	1 1	Operand Data Next Op Code
ADDD CPX LDD LDS LDX SUBD	3	1 2 3	Op Code Address+1 Op Code Address+2 Op Code Address+3	1 1 1	Operand Data (MSB) Operand Data (LSB) Next Op Code
DIRECT					
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	3	1 2 3	Op Code Address+1 Address of Operand Op Code Address+2	1 1 1	Address of Operand (LSB) Operand Data Next Op Code
STA	3	1 2 3	Op Code Address+1 Destination Address Op Code Address+2	1 0 1	Destination Address Accumulator Data Next Op Code
ADDD CPX LDD LDS LDX SUBD	4	1 2 3 4	Op Code Address+1 Address of Operand Address of Operand+1 Op Code Address+2	1 1 1 1	Address of Operand (LSB) Operand Data (MSB) Operand Data (LSB) Next Op Code
STD STS STX	4	1 2 3 4	Op Code Address+1 Destination Address Destination Address+1 Op Code Address+2	1 0 0 1	Destination Address (LSB) Register Data (MSB) Register Data (LSB) Next Op Code
JSR	5	1 2 3 4 5	Op Code Address+1 FFFF Stack Pointer Stack Pointer−1 Jump Address	1 1 0 0 1	Jump Address (LSB) Restart Address (LSB) Return Address (LSB) Return Address (MSB) First Subroutine Op Code
TIM	4	1 2 3 4	Op Code Address+1 Op Code Address+2 Address of Operand Op Code Address+3	1 1 1 1	Immediate Data Address of Operand (LSB) Operand Data Next Op Code
AIM EIM OIM	6	1 2 3 4 5 6	Op Code Address+1 Op Code Address+2 Address of Operand FFFF Address of Operand Op Code Address+3	1 1 1 1 0 1	Immediate Data Address of Operand (LSB) Operand Data Restart Address (LSB) New Operand Data Next Op Code

— Continued —

Table 12 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
INDEXED					
JMP	3	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Jump Address	1	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	Op Code Address+2	1	Next Op Code
STA	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Accumulator Data
		4	Op Code Address+2	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data (MSB)
		4	IX+Offset+1	1	Operand Data (LSB)
		5	Op Code Address+2	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Register Data (MSB)
		4	IX+Offset+1	0	Register Data (LSB)
		5	Op Code Address+2	1	Next Op Code
JSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	IX+Offset	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	IX+Offset	0	New Operand Data
		6	Op Code Address+2	1	Next Op Code
TIM	5	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	IX+Offset	0	00
		5	Op Code Address+2	1	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	FFFF	1	Restart Address (LSB)
		6	IX+Offset	0	New Operand Data
		7	Op Code Address+3	1	Next Op Code

— Continued —

Table 12 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
EXTEND					
JMP	3	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	Jump Address	1	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Op Code Address+3	1	Next Op Code
STA	4	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Accumulator Data
		4	Op Code Address+3	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data (MSB)
		4	Address of Operand+1	1	Operand Data (LSB)
		5	Op Code Address+3	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Register Data (MSB)
		4	Destination Address+1	0	Register Data (LSB)
		5	Op Code Address+3	1	Next Op Code
JSR	6	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	Stack Pointer	0	Return Address (LSB)
		5	Stack Pointer−1	0	Return Address (MSB)
		6	Jump Address	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	Address of Operand	0	New Operand Data
		6	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Address of Operand	0	00
		5	Op Code Address+3	1	Next Op Code

— Continued —

Table 12 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Accumulator Data
		4	Op Code Address+1	1	Next Op Code
PULX	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Index Register (LSB)
		4	Stack Pointer−1	0	Index Register (MSB)
		5	Op Code Address+1	1	Next Op Code
RTS	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Return Address (MSB)
		4	Stack Pointer+2	1	Return Address (LSB)
		5	Return Address	1	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	FFFF	1	Restart Address (LSB)
		5	FFFF	1	Restart Address (LSB)
		6	FFFF	1	Restart Address (LSB)
		7	FFFF	1	Restart Address (LSB)

— Continued —

Table 12 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
WAI	9	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	Index Register (LSB)
		6	Stack Pointer−3	0	Index Register (MSB)
		7	Stack Pointer−4	0	Accumulator A
		8	Stack Pointer−5	0	Accumulator B
		9	Stack Pointer−6	0	Conditional Code Register
RTI	10	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer +1	1	Conditional Code Register
		4	Stack Pointer+2	1	Accumulator B
		5	Stack Pointer+3	1	Accumulator A
		6	Stack Pointer+4	1	Index Register (MSB)
		7	Stack Pointer+5	1	Index Register (LSB)
		8	Stack Pointer+6	1	Return Address (MSB)
		9	Stack Pointer+7	1	Return Address (LSB)
		10	Return Address	1	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer − 1	0	Return Address (MSB)
		5	Stack Pointer − 2	0	Index Register (LSB)
		6	Stack Pointer − 3	0	Index Register (MSB)
		7	Stack Pointer − 4	0	Accumulator A
		8	Stack Pointer − 5	0	Accumulator B
		9	Stack Pointer − 6	0	Conditional Code Register
		10	Vector Address FFFA	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		↑	FFFF		High Impedance-Non MPX Mode
		Sleep	↓		Address Bus -MPX Mode
		↓			↓
		3	FFFF		Restart Address (LSB)
		4	Op Code Address+1		Next Op Code

— Continued —

Table 12 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
RELATIVE					
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	Branch Offset
		2	FFFF	1	Restart Address (LSB)
		3	{ Branch Address·····Test="1" { Op Code Address+1···Test="0"	1	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Branch Address	1	First Op Code of Subroutine

■ LOW POWER CONSUMPTION MODE

The HD6303R has two low power consumption modes; sleep and standby mode.

• Sleep Mode

On execution of SLP instruction, the MPU is brought to the sleep mode. In the sleep mode, the CPU stops its operation, but the contents of the registers in the CPU are retained. In this mode, the peripherals of CPU will remain active. So the operations such as transmit and receive of the SCI data and counter may keep in operation. In this mode, the power consumption is reduced to about 1/6 the value of a normal operation.

The escape from this mode can be done by interrupt, $\overline{RES}$, $\overline{STBY}$. The $\overline{RES}$ resets the MPU and the $\overline{STBY}$ brings it into the standby mode (This will be mentioned later). When interrupt is requested to the CPU and accepted, the sleep mode is released, then the CPU is brought in the operation mode and jumps to the interrupt routine. When the CPU has masked the interrupt, after recovering from the sleep mode, the next instruction of SLP starts to execute. However, in such a case that the timer interrupt is inhibited on the timer side, the sleep mode cannot be released due to the absence of the interrupt request to the CPU.

This sleep mode is available to reduce an average power consumption in the applications of the HD6303R which may not be always running.

• Standby Mode

Bringing $\overline{STBY}$ "Low", the CPU becomes reset and all clocks of the HD6303R become inactive. It goes into the standby mode. This mode remarkably reduces the power consumptions of the HD6303R.

In the standby mode, if the HD6303R is continuously supplied with power, the contents of RAM is retained. The standby mode should escape by the reset start. The following is the typical application of this mode.

First, $\overline{NMI}$ routine stacks the CPU's internal information and the contents of SP in RAM, disables RAME bit of RAM control register, sets the standby bit, and then goes into the standby mode. If the standby bit keeps set on reset start, it means that the power has been kept during stand-by mode and the contents of RAM is normally guaranteed. The system recovery may be possible by returning SP and bringing into the condition before the standby mode has started. The timing relation for each line in this application is shown in Figure 21.

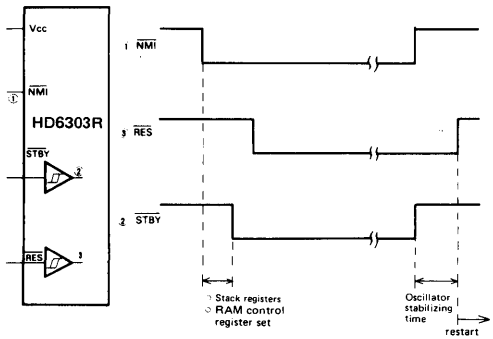


Figure 21 Standby Mode Timing

■ ERROR PROCESSING

When the HD6303R fetches an undefined instruction or fetches an instruction from unusable memory area, it generates the highest priority internal interrupt, that may protect from system upset due to noise or a program error.

• Op-Code Error

Fetching an undefined op-code, the HD6303R will stack the CPU register as in the case of a normal interrupt and vector to the TRAP ($FFEE, $FFEF), that has a second highest priority ($\overline{RES}$ is the highest).

• Address Error

When an instruction is fetched from other than a resident RAM, or an external memory area, the CPU starts the same interrupt as op-code error. In the case which the instruction is fetched from external memory area and that area is not usable, the address error can not be detected.

The address which cause address error are shown in Table 13.

This feature is applicable only to the instruction fetch, not to normal read/write of data accessing.

Transitions among the active mode, sleep mode, standby mode and reset are shown in Figure 22.

Figures 23, 24 show a system configuration.

The system flow chart of HD6303R is shown in Figure 25.

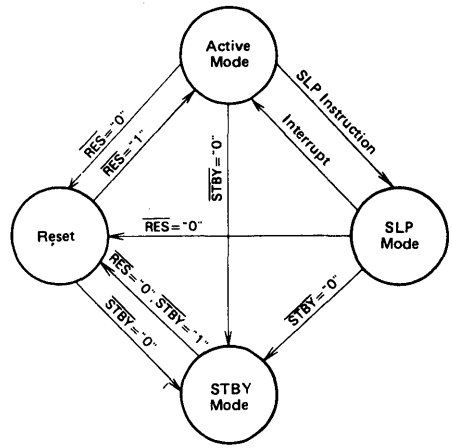


Figure 22 Transitions among Active Mode, Standby Mode, Sleep Mode, and Reset

Table 13 Address Error

Address Error
$0000 ~ $001F

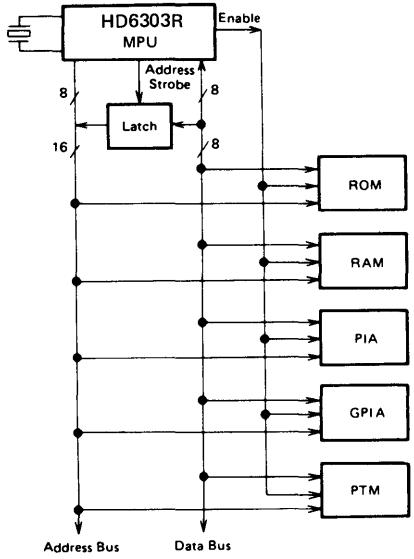


Figure 23 HD6303R MPU Multiplexed Mode

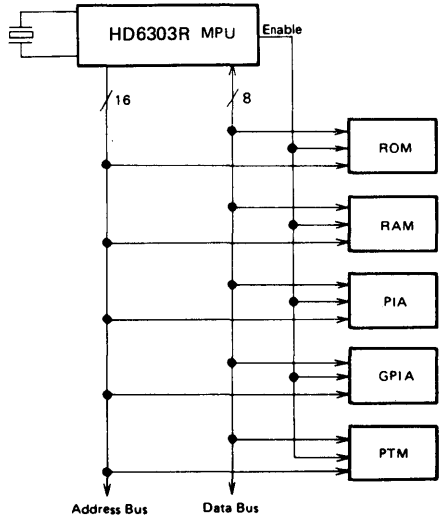


Figure 24 HD6303R MPU Non-Multiplexed Mode

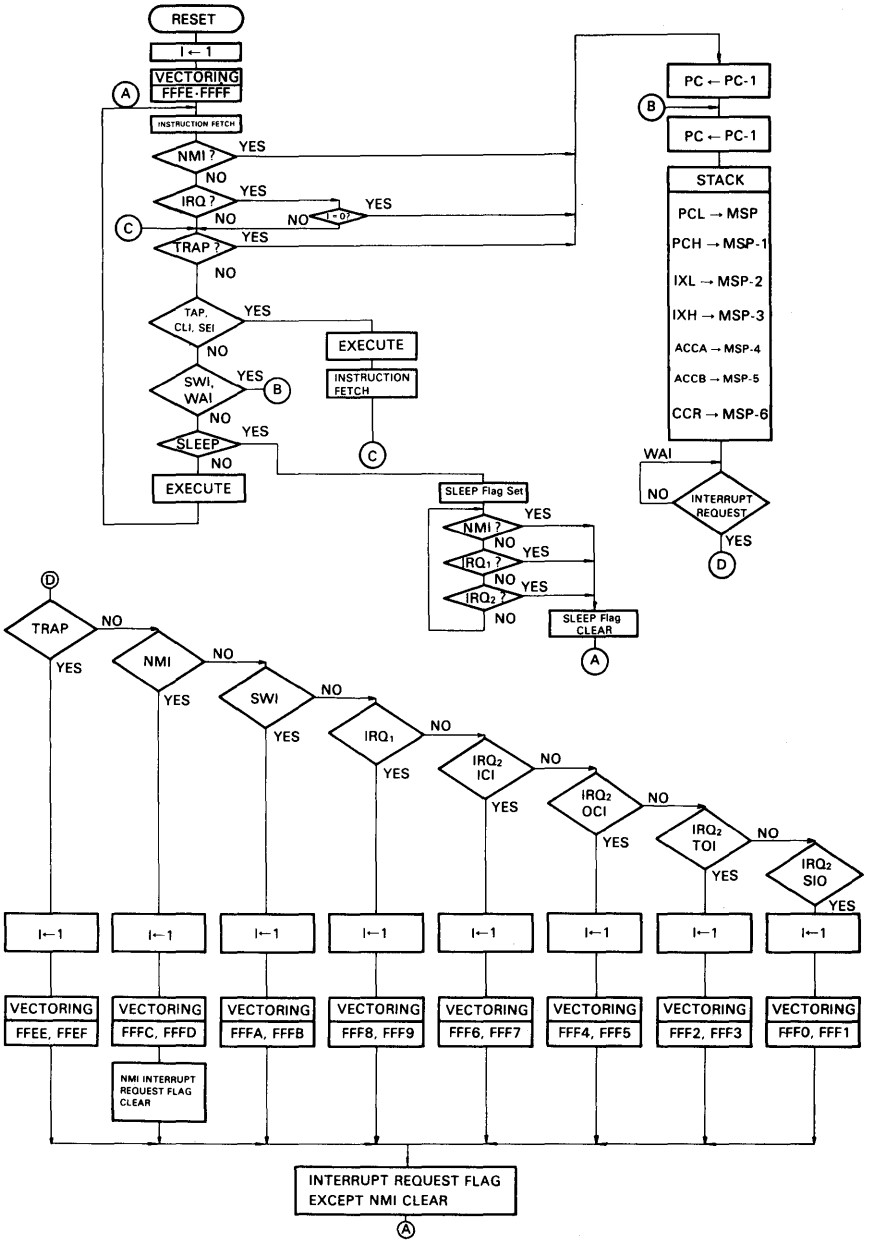


Figure 25 HD6303R System Flow Chart

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 26, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD6303R as possible.

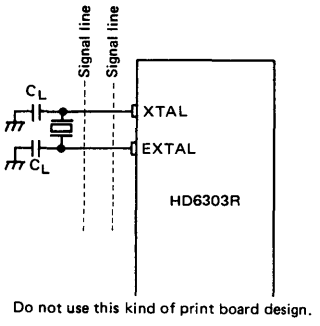


Figure 26 Precaution to the boad design of oscillation circuit

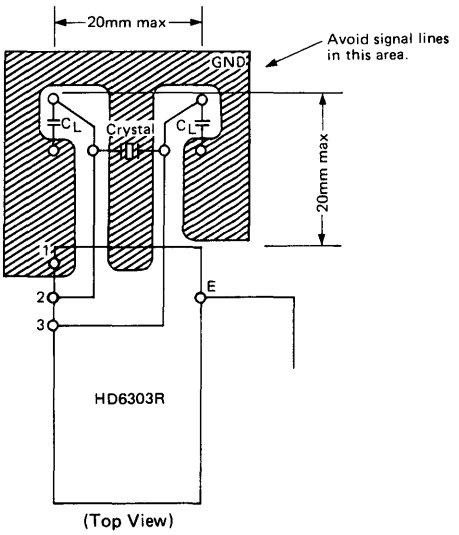


Fig. 27 Example of Oscillation Circuits in Board Design

■ PIN CONDITIONS AT SLEEP AND STANDBY STATE

• Sleep State

The conditions of power supply pins, clock pins, input pins and E clock pin are the same as those of operation. Refer to Table 14 for the other pin conditions.

• Standby State

Only power supply pins and $\overline{STBY}$ are active. As for the clock pin EXTAL, its input is fixed internally so the MPU is not influenced by the pin conditions. XTAL is in "1" output. All the other pins are in high impedance.

Table 14 Pin Condition in Sleep State

Pin \ Mode		Non Multiplexed Mode	Multiplexed Mode
$P_{20} \sim P_{24}$	Function	I/O Port	I/O Port
	Condition	Keep the condition just before sleep	←
$A_0/P_{10} \sim A_7/P_{17}$	Function	Address Bus ($A_0 \sim A_7$)	I/O Port
	Condition	Output "1"	Keep the condition just before sleep
$A_8 \sim A_{15}$	Function	Address Bus ($A_8 \sim A_{15}$)	Address Bus ($A_8 \sim A_{15}$)
	Condition	Output "1"	←
$D_0/A_0 \sim D_7/A_7$	Function	Data Bus ($D_0 \sim D_7$)	$\overline{E}$: Address Bus ($A_0 \sim A_7$), E: Data Bus
	Condition	High Impedance	$\overline{E}$: Output "1", E: High Impedance
R/$\overline{W}$	Function	R/$\overline{W}$ Signal	R/$\overline{W}$ Signal
	Condition	Output "1"	←
AS		–	Output AS

Table 15 Pin Condition during RESET

Pin \ Mode	Non-Multiplexed Mode	Multiplexed Mode
$P_{20} \sim P_{24}$	High Impedance	←
$A_0/P_{10} \sim A_7/P_{17}$	High Impedance	←
$A_8 \sim A_{15}$	High Impedance	←
$D_0/A_0 \sim D_7/A_7$	High Impedance	$\overline{E}$: "1" Output E : "1" Output (Note) (High Impedance)
R/$\overline{W}$	"1" Output	←
AS	$\overline{E}$: "1" Output E : High Impedance	←

(Note) In the multiplexed mode, the data bus is set to "1" output state during E = "1" and it causes the conflict with the output of external memory. Following 1 and 2 should be done to avoid the conflict;
(1) Construct the system that disables the external memory during reset.
(2) Add 4.7 kΩ pull-down resistance to the AS pin to make AS pin "0" level during E = "1". This operation makes the data bus high impedance state.

■ DIFFERENCE BETWEEN HD6303 AND HD6303R

The HD6303R is an upgraded version of the HD6303. The difference between HD6303 and HD6303R is shown in Table 16.

Table 16 Difference between HD6303 and HD6303R

Item	HD6303	HD6303R
Operating Mode	Mode 2: Not defined	Mode 2: Multiplexed Mode (Equivalent to Mode 4)
Electrical Characteristics	The electrical characteristics of 2MHz version (B version) are not specified.	Some characteristics are improved. The 2MHz version is guaranteed.
Timer	Has problem in output compare function. (Can be avoided by software.)	The problem is solved.

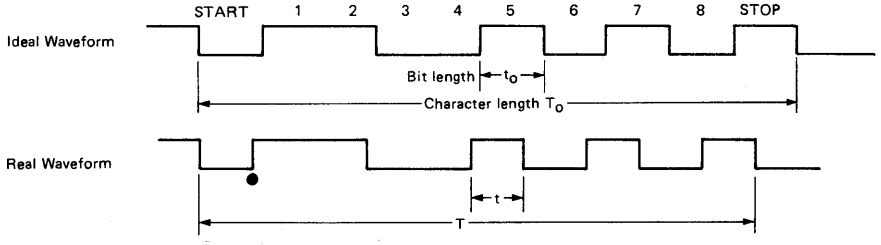
■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD6303R is shown in Table 17.

Note: SCI = Serial Communication Interface

Table 17

	Bit distortion tolerance (t−to) /to	Character distortion tolerance (T−To) /To
HD6303R	±37.5%	+3.75% -2.5%



■ APPLICATION NOTE FOR HIGH SPEED SYSTEM DESIGN USING THE HD6303R

This note describes the solutions of the potential problem caused by noise generation in the system using the HD6303R.

The CMOS ICs and LSIs featured by low power consumption and high noise immunity are generally considered to be enough with simply designed power source and the GND line.

But this does not apply to the applications configured of high speed system or of high speed parts. Such high speed system may have a chance to work incorrectly because of the noise

by the transient current generated during switching. One of example is a system in which the HD6303R directly accesses high speed memory such as the HM6264. The noise generation owing to the over current (Sometimes it may be several hundreds mA for peak level.) during switching may cause data write error.

This noise problem may be observed only at the Expanded Mode (Mode 1, 2, 4, 5 and 6) of the HD6303R.

Assuming the HD6303R is used as CPU in a system.

I. Noise Occurrence

If the HD6303R is connected to high speed RAM, a write error may occur. As shown in Fig. 28, the noise is generated in address bus during write cycle and data is written into an unexpected address from the HD6303R. This phenomenon causes random failures in systems whose data bus load capacitance exceeds the specification value (90 pF max.) and/or the impedance of the GND line is high.

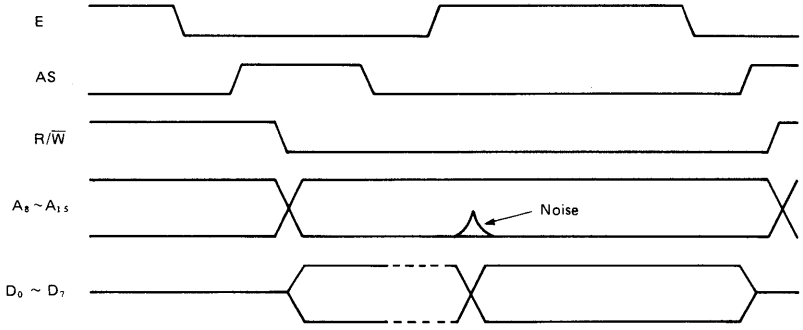


Fig. 28 Noise Occurrence in address bus during write cycle

If the data bus $D_0 \sim D_7$ changes from "FF" to "00", extremely large transient current flows through the GND line. Then the noise is generated on the LSI's V_{SS} pins proportioning to the transient current and to the impedance [Zg] of the GND line.

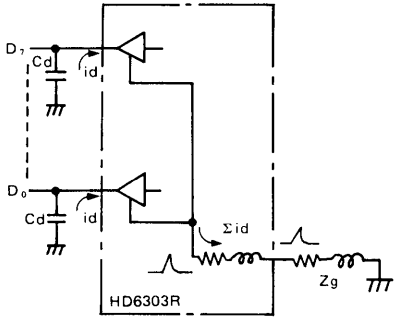
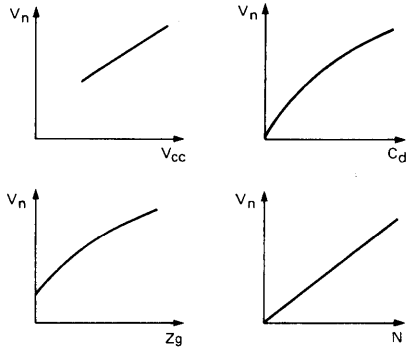


Fig. 29 Noise Source

This noise level, V_n, appears on all output pins on the LSI including the address bus.

Fig. 30 shows the dependency of the noise voltage on the each parameter.



Vn: Noise Voltage Zg: GND Impedance
Cd: Data bus load capacitance
N: Number of data bus lines switching from H to L

Fig. 30 Dependency of the noise voltage on each parameter

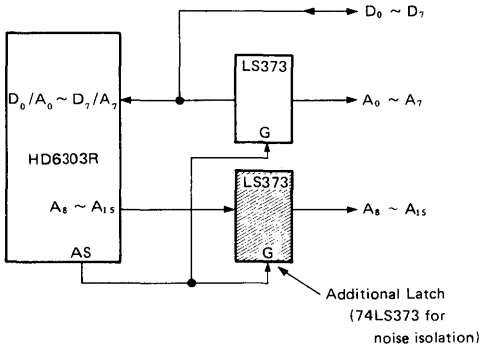
II. Noise Protection

To avoid the noise on the address bus during the system operation mentioned before, there are two solutions as follows:

The one method is to isolate the HD6303R from peripheral devices so that peripherals are not affected by the noise. The other is to reduce noise level to the extent of not affecting peripherals using analog method.

1. Noise Isolation

Addresses should be latched at the negative edge of the AS signal or at the positive edge of the E signal. The 74LS373 is often used in this case.



2. Noise Reduction

As the noise level depends on each parameter such Cd, V_{CC}, Zg, the noise level can be reduced to the allowable level by controlling those analog parameters.

(a) Transient Current Reduction

(1) Reduce the data bus load capacitance. If large load capacitance is expected, a bus buffer should be inserted.

(2) Lower the power supply voltage V_{CC} within specification.

(3) Increase a time constant at transient state by inserting a resistor (100 ~ 200Ω) to Data Buses in series to keep noise level down.
Table 18 shows the relationship between a series resistor and noise level or a resistor and DC/AC characteristics.

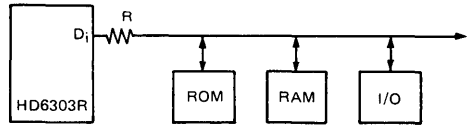


Table 18.

Item		Resistor	No	100Ω	200Ω
Noise Voltage Level			See Fig. 31		
DC Characteristics		I_{OL}	1.6 mA	1.6 mA	1.0 mA
AC Characteristics	f = 1 MHz		No change		
	f = 1.5 MHz	t_{ADL}	190 ns	190 ns	210 ns
		t_{ACCM}	395 ns	395 ns	375 ns
	f = 2 MHz	t_{ADL}	160 ns	180 ns	200 ns
		t_{ASL}	20 ns	20 ns	0 ns
		t_{ACCM}	270 ns	250 ns	230 ns

Fig. 31 shows an example of the dependency of the noise voltage on the load capacitance of the data bus.*

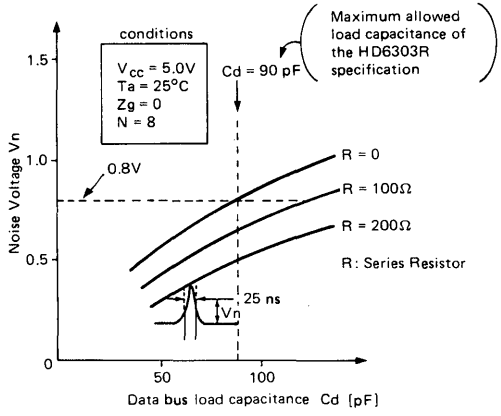


Fig. 31

*Note: The value of series resistor should be carefully selected because it heavily depends on each parameter of actual application system.

Fig. 32 shows the typical wave form of the noise.

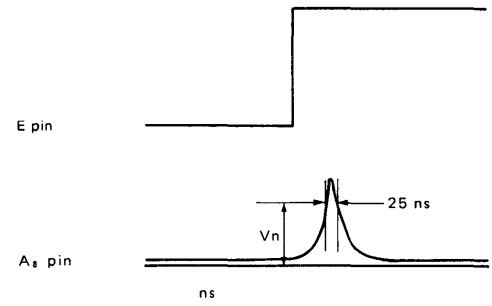


Fig. 32

(b) Reduction of GND line impedance
(1) Widen the GND line width on the PC board.
(2) Place the HD6303R close by power source.
(3) Insert a bypass capacitor between the V_{CC} line and the GND of the HD6303R. A tantalum capacitor (about 0.1μF) is effective on the reduction.

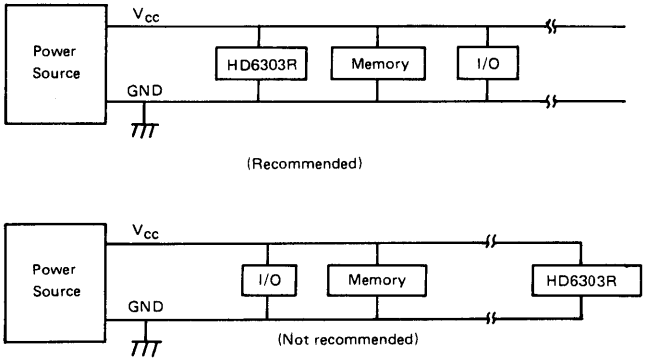
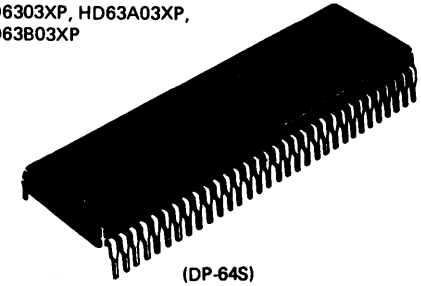


Fig. 33 Layout of the HD6303R on the PC board

HD6303X, HD63A03X, HD63B03X
CMOS MPU (Micro Processing Unit)

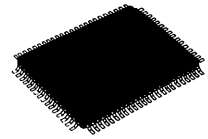
The HD6303X is a CMOS 8-bit micro processing unit (MPU) which includes a CPU compatible with the HD6301V1, 192 bytes of RAM, 24 parallel I/O pins, a Serial Communication Interface (SCI) and two timers on chip.

HD6303XP, HD63A03XP, HD63B03XP



(DP-64S)

HD6303XF, HD63A03XF, HD63B03XF



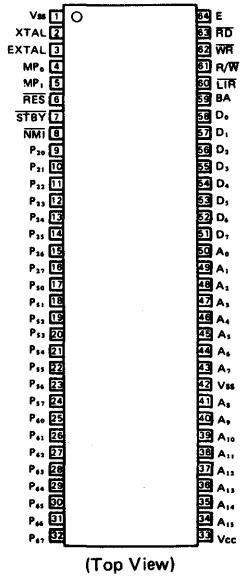
(FP-80)

■ **FEATURES**

- Instruction Set Compatible with the HD6301V1
- 192 Bytes of RAM
- 24 Parallel I/O Pins
  16 I/O Pins-Port 2, 6
  8 Input Pins-Port 5
- Darlington Transistor Drive (Port 2, 6)
- 16-Bit Programmable Timer
  Input Capture Register x 1
  Free Running Counter x 1
  Output Compare Register x 2
- 8-Bit Reloadable Timer
  External Event Counter Square Wave Generation
- Serial Communication Interface
- Memory Ready
- Halt
- Error-Detection (Address Trap, Op-Code Trap)
- Interrupts . . . 3 External, 7 Internal
- Up to 65k Bytes Address Space
- Low Power Dissipation Mode
  Sleep Mode
  Standby Mode
- Minimum Instruction Execution Time -0.5μs (f = 2.0 MHz)
- Wide Range of Operation
  V_{CC} = 3 ~ 6V (f = 0.1 ~ 0.5 MHz).
  V_{CC} = 5V±10% { f = 0.1 ~ 1.0 MHz; HD6303X; f = 0.1 ~ 1.5 MHz; HD63A03X; f = 0.1 ~ 2.0 MHz; HD63B03X

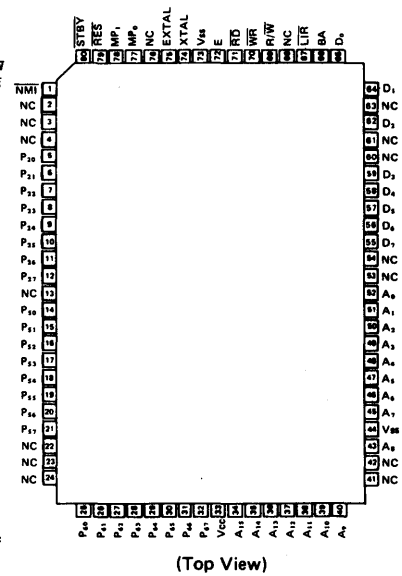
■ **PIN ARRANGEMENT**

- **HD6303XP, HD63A03XP, HD63B03XP**



(Top View)

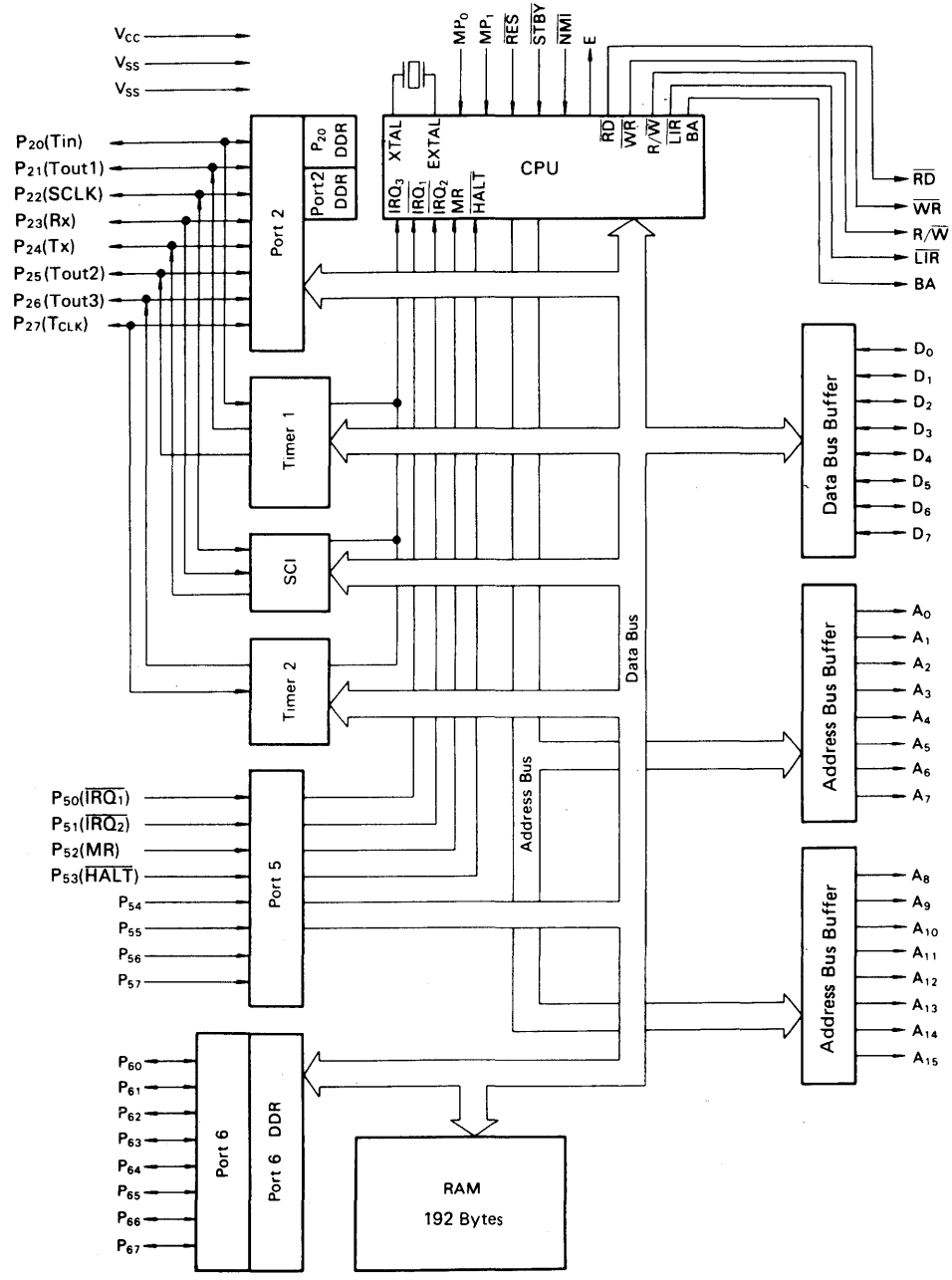
- **HD6303XF, HD63A03XF, HD63B03XF**



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BLOCK DIAGRAM**



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out}: $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}x0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$, MP_0, MP_1, Port 5	$\|I_{in}\|$	V_{in} = 0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	A_0~A_{15}, D_0~D_7, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, Port 2, Port 6	$\|I_{TSI}\|$	V_{in} = 0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
Darlington Drive Current	Ports 2, 6	$-I_{OH}$	Vout = 1.5V	1.0	–	10.0	mA
Input Capacitance	All Inputs	C_{in}	V_{in} = 0V, f = 1MHz, Ta = 25°C	–	–	12.5	pF
Standby Current	Non Operation	I_{STB}		–	3.0	15.0	μA
Current Dissipation*		I_{SLP}	Sleeping (f = 1MHz**)	–	1.5	3.0	mA
			Sleeping (f = 1.5MHz**)	–	2.3	4.5	mA
			Sleeping (f = 2MHz**)	–	3.0	6.0	mA
		I_{CC}	Operating (f = 1MHz**)	–	7.0	10.0	mA
			Operating (f = 1.5MHz**)	–	10.5	15.0	mA
			Operating (f = 2MHz**)	–	14.0	20.0	mA
RAM Standby Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V , All output terminals are at no load.

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at x MHz operation are decided according to the following formula;

typ. value (f = x MHz) = typ. value (f = 1MHz) x x
max. value (f = x MHz) = max. value (f = 1MHz) x x
(both the sleeping and operating)

- **AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, T_a = 0 ~ +70°C, unless otherwise noted.)**

BUS TIMING

Item		Symbol	Test Condition	HD6303X			HD63A03X			HD63B03X			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time		t_{Er}		–	–	25	–	–	25	–	–	25	ns
Enable Fall Time		t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*		t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time	Write	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Set-up Time	Read	t_{DSR}		80	–	–	70	–	–	70	–	–	ns
Address, R/$\overline{W}$ Hold Time*		t_{AH}		80	–	–	50	–	–	35	–	–	ns
Data Hold Time	Write*	t_{HW}		80	–	–	50	–	–	40	–	–	ns
	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
$\overline{RD}$, $\overline{WR}$ Pulse Width*		PW_{RW}		450	–	–	300	–	–	220	–	–	ns
$\overline{RD}$, $\overline{WR}$ Delay Time		t_{RWD}		–	–	40	–	–	40	–	–	40	ns
$\overline{RD}$, $\overline{WR}$ Hold Time		t_{HRW}		–	–	30	–	–	30	–	–	25	ns
$\overline{LIR}$ Delay Time		t_{DLR}		–	–	200	–	–	160	–	–	120	ns
$\overline{LIR}$ Hold Time		t_{HLR}		10	–	–	10	–	–	10	–	–	ns
MR Set-up Time*		t_{SMR}	Fig. 2	400	–	–	280	–	–	230	–	–	ns
MR Hold Time*		t_{HMR}		–	–	90	–	–	40	–	–	0	ns
E Clock Pulse Width at MR		PW_{EMR}		–	–	9	–	–	9	–	–	9	μs
Processor Control Set-up Time		t_{PCS}	Fig. 3, 10, 11	200	–	–	200	–	–	200	–	–	ns
Processor Control Rise Time		t_{PCr}	Fig. 2, 3	–	–	100	–	–	100	–	–	100	ns
Processor Control Fall Time		t_{PCf}		–	–	100	–	–	100	–	–	100	ns
BA Delay Time		t_{BA}	Fig. 3	–	–	250	–	–	190	–	–	160	ns
Oscillator Stabilization Time		t_{RC}	Fig. 11	20	–	–	20	–	–	20	–	–	ms
Reset Pulse Width		PW_{RST}		3	–	–	3	–	–	3	–	–	t_{cyc}

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD6303X			HD63A03X			HD63B03X			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set-up Time	Ports 2, 5, 6	t_{PDSU}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Ports 2, 5, 6	t_{PDH}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Delay Time (Enable Negative Transition to Peripheral Data Valid)	Ports 2, 6	t_{PWD}	Fig. 6	–	–	300	–	–	300	–	–	300	ns

TIMER, SCI TIMING

Item		Symbol	Test Condition	HD6303X			HD63A03X			HD63B03X			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Output)		t_{TOD}	Fig. 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Async. Mode	t_{Scyc}	Fig. 8	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Clock Sync.		Fig. 4, 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Clock Sync. Mode)		t_{TXD}	Fig. 4	–	–	200	–	–	200	–	–	200	ns
SCI Receive Data Set-up Time (Clock Sync. Mode)		t_{SRX}		290	–	–	290	–	–	290	–	–	ns
SCI Receive Data Hold Time (Clock Sync. Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 8	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}		200	–	–	200	–	–	200	–	–	ns
Timer 1•2, SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1•2, SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

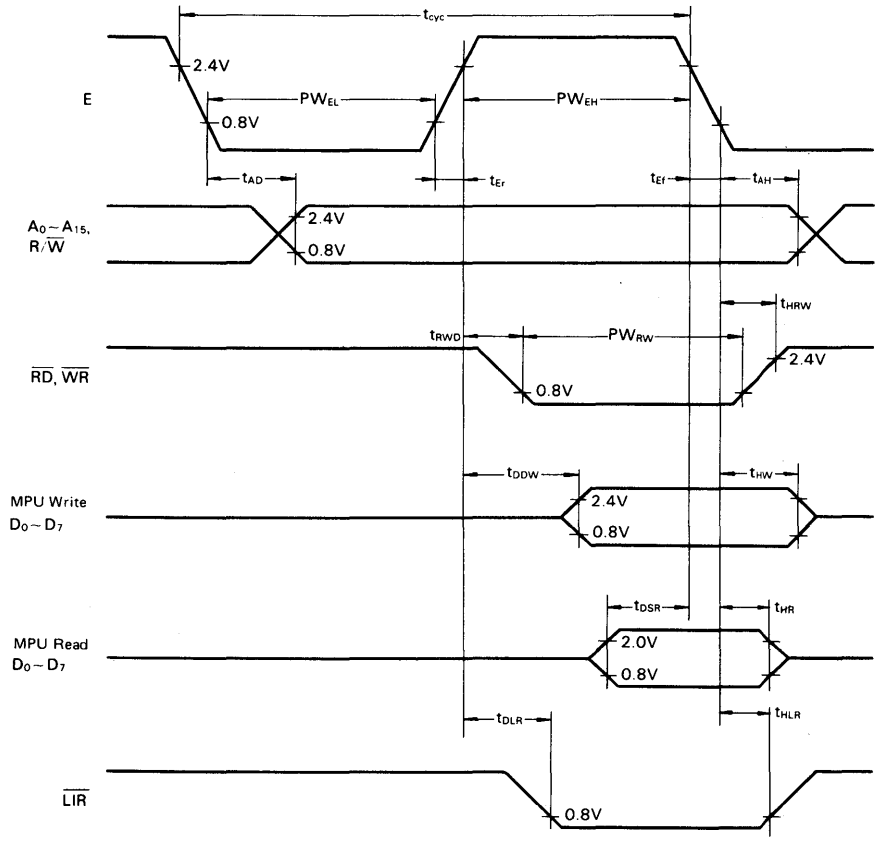


Figure 1 Bus Timing

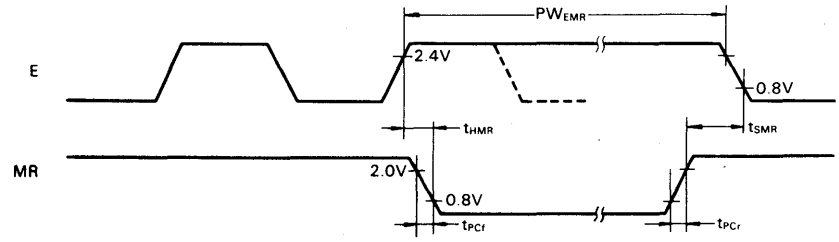


Figure 2 Memory Ready and E Clock Timing

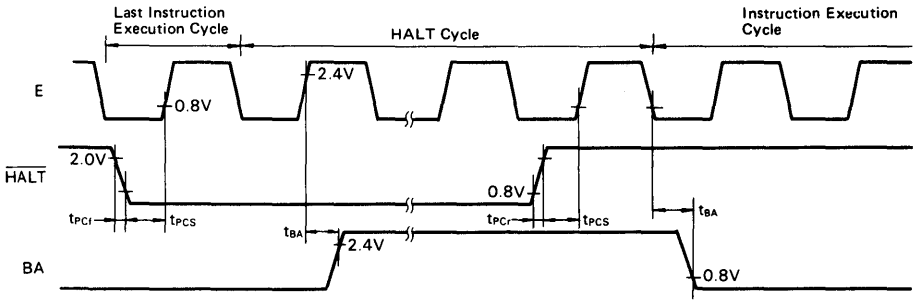


Figure 3 $\overline{\text{HALT}}$ and BA Timing

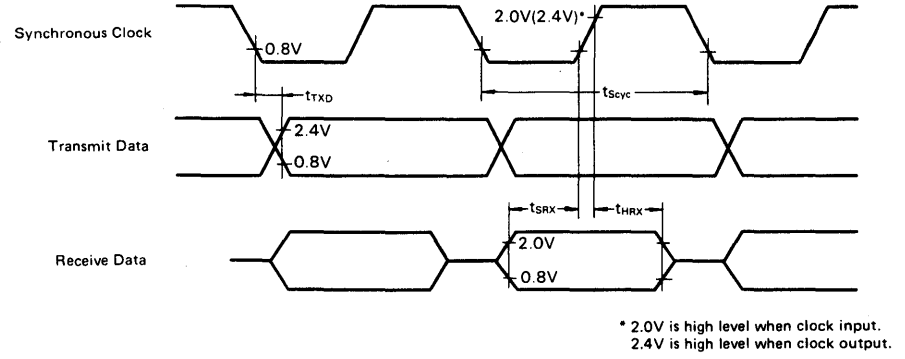


Figure 4 SCI Clocked Synchronous Timing

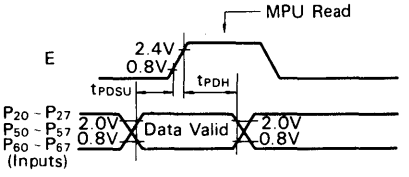


Figure 5 Port Data Set-up and Hold Times (MPU Read)

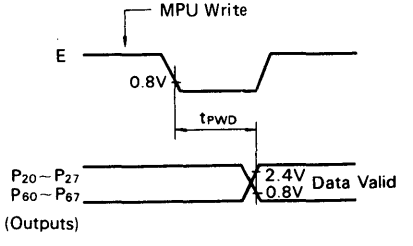
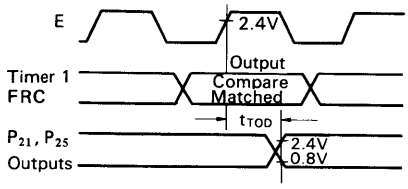
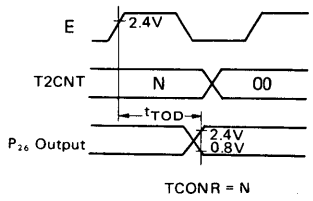


Figure 6 Port Data Delay Times (MPU Write)

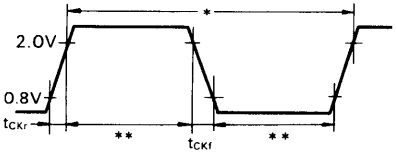


(a) Timer 1 Output Timing



(b) Timer 2 Output Timing

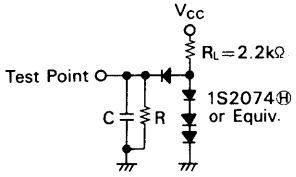
Figure 7 Timer Output Timing



* Timer 2; t_{TCYC} ; SCI ; t_{SCYC}

** Timer 1; t_{PWT} ; Timer 2; t_{PWTCK} ; SCI ; t_{PWSCK}

Figure 8 Timer 1·2, SCI Input Clock Timing



C=90pF for D_0~D_7, A_0~A_{15}, E
=30pF for Port 2, Port 6, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, BA, $\overline{LIR}$
R=12kΩ

Figure 9 Bus Timing Test Loads (TTL Load)

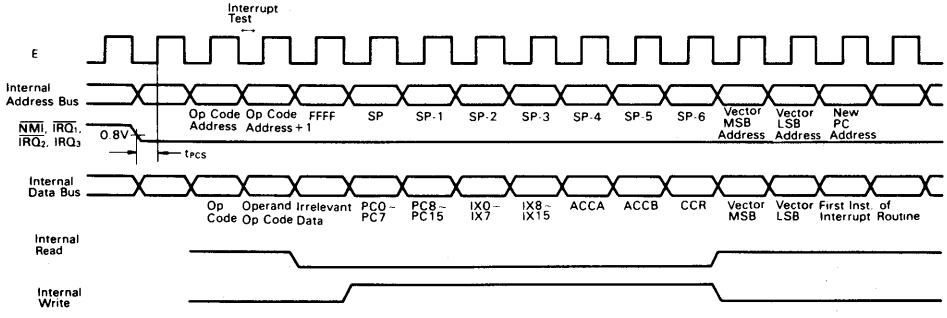


Figure 10 Interrupt Sequence

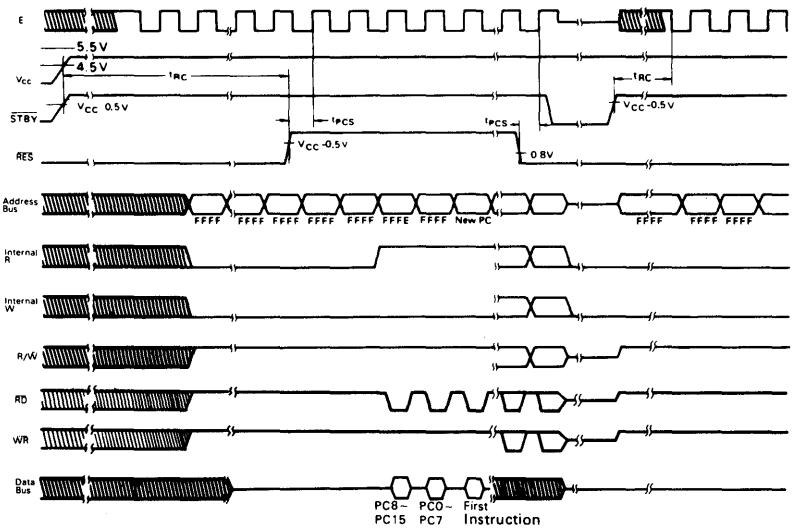


Figure 11 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

V_{CC} and V_{SS} provide power to the MPU with 5V±10% supply. In the case of low speed operation (fmax = 500kHz), the MPU can operate with three through six volts. Two V_{SS} pins should be tied to ground.

• XTAL, EXTAL

These two pins interface with an AT-cut parallel resonant crystal. Divide-by-four circuit is on chip, so if 4MHz crystal oscillator is used, the system clock is 1MHz for example.

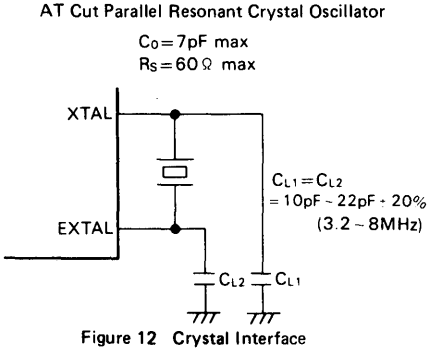


Figure 12 Crystal Interface

EXTAL pin can be drived by the external clock of 45 to 55% duty, and one fourth frequency of the external clock is produced in the LSI. The external clock frequency should be less than four times of the maximum operable frequency. When using the external clock, XTAL pin should be open. Fig. 12 shows an example of the crystal interface. The crystal and C_{L1}, C_{L2} should be mounted as close as possible to XTAL and EXTAL pins. Any line must not cross the line between the crystal oscillator and XTAL, EXTAL.

• $\overline{STBY}$

This pin makes the MPU standby mode. In "Low" level, the oscillation stops and the internal clock is stabilized to make reset condition. To retain the contents of RAM at standby mode, "0" should be written into RAM enable bit (RAME). RAME is the bit 6 of the RAM/port 5 control register at $0014. RAM is disabled by this operation and its contents is sustained.

Refer to "LOW POWER DISSIPATION MODE" for the standby mode.

• Reset ($\overline{RES}$)

This pin resets the MPU from power OFF state and provides a startup procedure. During power-on, $\overline{RES}$ pin must be held "Low" level for at least 20ms.

The CPU registers (accumulator, index register, stack pointer, condition code register except for interrupt mask bit), RAM and the data register of a port are not initialized during reset, so their contents are unknown in this procedure.

To reset the MPU during operation, $\overline{RES}$ should be held "Low" for at least 3 system-clock cycles. At the 3rd cycle during "Low" level, all the address buses become "High". When $\overline{RES}$ remains "Low", the address buses keep "High". If $\overline{RES}$ becomes "High", the MPU starts the next operation.

(1) Latch the value of the mode program pins; MP_0 and MP_1.
(2) Initialize each internal register (Refer to Table 3).
(3) Set the interrupt mask bit. For the CPU to recognize the maskable interrupts $\overline{IRQ_1}$, $\overline{IRQ_2}$ and IRQ_3, this bit should be cleared in advance.
(4) Put the contents (= start address) of the last two addresses ($FFFE, $FFFF) into the program counter and start the program from this address. (Refer to Table 1).

*The MPU is usable to accept a reset input until the clock

becomes normal oscillation after power on (max. 20ms). During this transient time, the MPU and I/O pins are undefined. Please be aware of this for system designing.

- **Enable (E)**

This pin provides a TTL-compatible system clock to external circuits. Its frequency is one fourth that of the crystal oscillator or external clock. This pin can drive one TTL load and 90pF capacitance.

- **Non-Maskable Interrupt ($\overline{NMI}$)**

When the falling edge of the input signal is detected at this pin, the CPU begins non-maskable interrupt sequence internally. As well as the IRQ mentioned below, the instruction being executed at $\overline{NMI}$ signal detection will proceed to its completion. The interrupt mask bit of the condition code register doesn't affect non-maskable interrupt at all.

When starting the acknowledge to the $\overline{NMI}$, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack. Upon completion of this sequence, a vector is fetched from $FFFC and $FFFD to transfer their contents into the program counter and branch to the non-maskable interrupt service routine.

(Note) After reset start, the stack pointer should be initialized on an appropreate memory area and then the falling edge should be input to $\overline{NMI}$ pin.

- **Interrupt Request ($\overline{IRQ_1}$, $\overline{IRQ_2}$)**

These are level-sensitive pins which request an internal interrupt sequence to the CPU. At interrupt request, the CPU will complete the current instruction before its request acknowledgement. Unless the interrupt mask in the condition code register is set, the CPU starts an interrupt sequence; if set, the interrupt request will be ignored. When the sequence starts, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack, then the CPU sets the interrupt mask bit and will not acknowledge the maskable request. During the last cycle, the CPU fetches vectors depicted in Table 1 and transfers their contents to the program counter and branches to the service routine.

The CPU uses the external interrupt pins, $\overline{IRQ_1}$ and $\overline{IRQ_2}$, also as port pins P_{50} and P_{51}, so it provides an enable bit to Bit 0 and 1 of the RAM port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for the details.

When one of the internal interrupts, ICI, OCI, TOI, CMI or SIO is generated, the CPU produces internal interrupt signal (IRQ_3). IRQ_3 functions just the same as $\overline{IRQ_1}$ or $\overline{IRQ_2}$ except for its vector address. Fig. 13 shows the block diagram of the interrupt circuit.

Table 1 Interrupt Vector Memory Map

Priority	Vector		Interrupt
	MSB	LSB	
Highest	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	$\overline{IRQ_1}$
	FFF6	FFF7	ICI (Timer 1 Input Capture)
	FFF4	FFF5	OCI (Timer 1 Output Compare 1, 2)
	FFF2	FFF3	TOI (Timer 1 Overflow)
	FFEC	FFED	CMI (Timer 2 Counter Match)
	FFEA	FFEB	$\overline{IRQ_2}$
Lowest	FFF0	FFF1	SIO (RDRF+ORFE+TDRE)

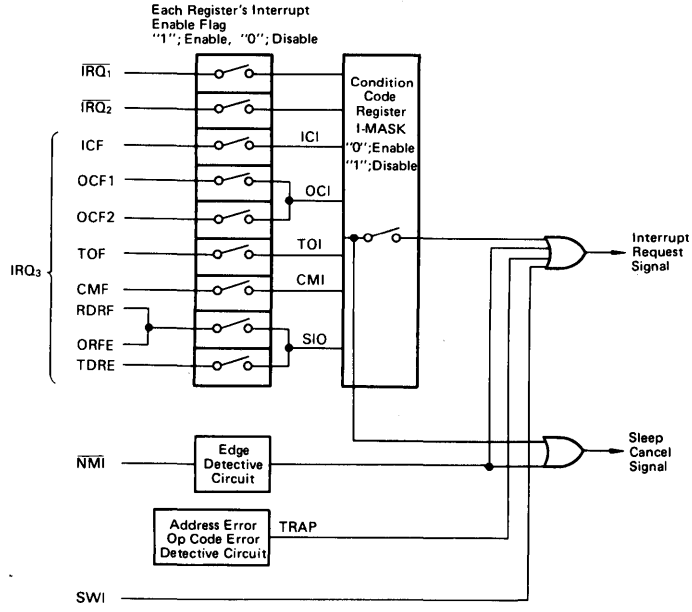


Figure 13 Interrupt Circuit Block Diagram

- **Mode Program (MP_0, MP_1)**

To operate MPU, MP_0 pin should be connected to "High" level and MP_1 should be connected to "Low" level (refer to Fig. 15).

- **Read/Write (R/$\overline{W}$)**

This signal, usually be in read state ("High"), shows whether the CPU is in read ("High") or write ("Low") state to the peripheral or memory devices. This can drive one TTL load and 30pF capacitance.

- **$\overline{RD}$, $\overline{WR}$**

These signals show active low outputs when the CPU is reading/writing to the peripherals or memories. This enables the CPU easy to access the peripheral LSI with $\overline{RD}$ and $\overline{WR}$ input pins. These pins can drive one TTL load and 30pF capacitance.

- **Load Instruction Register ($\overline{LIR}$)**

This signal shows the instruction opecode being on data bus (active low). This pin can drive one TTL load and 30pF capacitance.

- **Memory Ready (MR; P_{52})**

This is the input control signal which stretches the system clock's "High" period to access low-speed memories. During this signal is in "High", the system clock operates in normal sequence. But this signal in "Low", the "High" period of the system clock will be stretched depending on its "Low" level duration in integral multiples of the cycle time. This allows the CPU to interface with low-speed memories (see Fig. 2). Up to 9 μs can be stretched.

During internal address space access or nonvalid memory access, MR is prohibited internally to prevent decrease of operation speed. Even in the halt state, MR can also stretch "High" period of system clock to allow peripheral devices to access low-speed memories. As this signal is used also as P_{52}, an enable bit is provided at bit 2 of the RAM/port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for more details.

- **Halt ($\overline{HALT}$; P_{53})**

This is an input control signal to stop instruction execution and to release buses. When this signal switches to "Low", the CPU stops to enter into the halt state after having executed the present instruction. When entering into the halt state, it makes BA (P_{74}) "High" and also an address bus, data bus, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ high impedance. When an interrupt is generated in the halt state, the CPU uses the interrupt handler after the halt is cancelled.

(Note) Please don't switch the $\overline{HALT}$ signal to "Low" when the CPU executes the WAI instruction and is in the interrupt wait state to avoid the trouble of the CPU's operation after the halt is cancelled.

- **Bus Available (BA)**

This is an output control signal which is normally "Low" but "High" when the CPU accepts $\overline{HALT}$ and releases the buses. The HD6800 and HD6802 make BA "High" and release the buses at WAI execution, while the HD6303X doesn't make BA "High" under the same condition. But if the $\overline{HALT}$ becomes "Low" when the CPU is in the interrupt wait state after having executed the WAI, the CPU makes BA "High" and releases the buses. And when the $\overline{HALT}$ becomes "High", the CPU returns to the interrupt wait state.

■ PORT

The HD6303X provides three I/O ports. Table 2 gives the address of ports and the data direction register and Fig. 14 the block diagrams of each port.

Table 2 Port and Data Direction Register Address

Port	Port Address	Data Direction Register
Port 2	$0003	$0001
Port 5	$0015	–
Port 6	$0017	$0016

• Port 2

An 8-bit input/output port. The data direction register (DDR) of port 2 controls the I/O state. It provides two bits; bit 0 decides the I/O direction of P_{20} and bit 1 the I/O direction of P_{21} to P_{27} ("0" for input, "1" for output).

Port 2 is also used as an I/O pin for the timers and the SCI. When used as an I/O pin for the timers and the SCI, port 2 except P_{20} automatically becomes an input or an output depending on their functions regardless of the data direction register's value.

Port 2 Data Direction Register

7	6	5	4	3	2	1	0	
–	–	–	–	–	–	DDR 1~7	DDR 0	$0001

A reset clears the DDR of port 2 and configures port 2 as an input port. This port can drive one TTL and 30pF capacitance. In addition, it can produce 1mA current when V_{out} = 1.5V to drive directly the base of Darlington transistors.

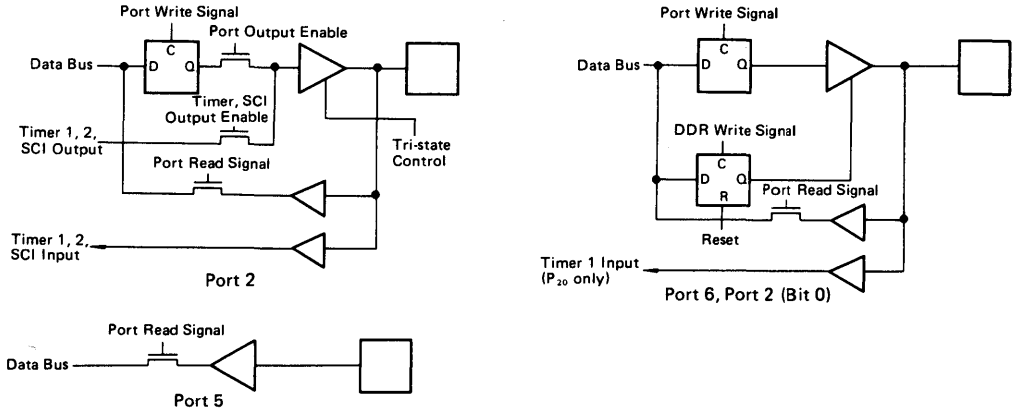


Figure 14 Port Block Diagram

• Port 5

An 8-bit port for input only. The lower four bits are also usable as input pins for interrupt, MR and $\overline{HALT}$.

• Port 6

An 8-bit I/O port. This port provides an 8-bit DDR corresponding to each bit and can specify input or output by the bit ("0" for input, "1" for output). This port can drive one TTL load and 30pF capacitance. A reset clears the DDR of port 6. In addition, it can produce 1mA current when V_{out} = 1.5V to drive directly the base of Darlington transistors.

■ BUS

• D_0~D_7

These pins are data bus and can drive one TTL load and 90pF capacitance respectively.

• A_0~A_{15}

These pins are address bus and can drive one TTL load and 90pF capacitance respectively.

■ RAM/PORT 5 CONTROL REGISTER

The control register located at $0014 controls on-chip RAM and port 5.

RAM/Port 5 Control Register

7	6	5	4	3	2	1	0	
STBY PWR	RAME	–	–	HLTE	MRE	IRQ_2 E	IRQ_1 E	$0014

Bit 0, Bit 1 $\overline{IRQ_1}$, $\overline{IRQ_2}$ Enable Bit (IRQ_1E, IRQ_2E)

When using P_{50} and P_{51} as interrupt pins, write "1" in these bits. When "0", the CPU doesn't accept an external interrupt or a sleep cancellation by the external interrupt. These bits become "0" during reset.

Bit 2 Memory Ready Enable Bit (MRE)

When using P_{52} as an input for Memory Ready signal, write "1" in this bit. When "0", the memory ready function is prohibited and P_{52} can be used as I/O port. This bit becomes "1" during reset.

Bit 3 Halt Enable bit (HLTE)

When using P_{53} as an input for Halt signal, write "1" in this

bit. When "0", the halt function is prohibited and P_{53} can be used as I/O port. This bit becomes "1" during reset.

(Note) When using P_{52} and P_{53} as the input ports in mode 1 and 2, MRE and HLTE bit should be cleared just after the reset.
Notice that memory ready and halt function is enable till MRE and HLTE bit is cleared.

Bit 4, Bit 5 Not Used.

Bit 6 RAM Enable (RAME)

On-chip RAM can be disabled by this control bit. By resetting the MPU, "1" is set to this bit, and on-chip RAM is enabled. This bit can be written "1" or "0" by software. When RAM is in disable condition (= logic "0"), on-chip RAM is invalid and the CPU can read data from external memory. This bit should be "0" before getting into the standby mode to protect on-chip RAM data.

Bit 7 Standby Power Bit (STBY PWR)

When V_{CC} is not provided in standby mode, this bit is cleared. This is a flag for both read/write by software. If this bit is set before standby mode, and remains set even after returning from standby mode, V_{CC} voltage is provided during standby mode and the on-chip RAM data is valid.

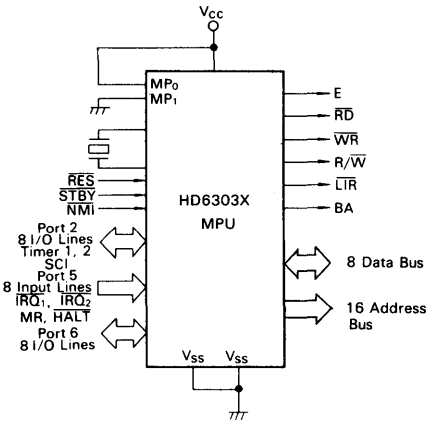


Figure 15 Operation Mode

■ MEMORY MAP

The MPU can address up to 65k bytes. Fig. 16 gives memory map of HD6303X. 32 internal registers use addresses from "00" as shown in Table 3.

Table 3 Internal Register

Address	Registers	R/W***	Initialize at RESET
00	–	–	–
01	Port 2 Data Direction Register	W	$FC
02*	–	–	–
03	Port 2	R/W	Undefined
04*	–	–	–
05	–	–	–
06*	–	–	–
07*	–	–	–
08	Timer Control/Status Register 1	R/W	$00
09	Free Running Counter ("High")	R/W	$00
0A	Free Running Counter ("Low")	R/W	$00
0B	Output Compare Register 1 ("High")	R/W	$FF
0C	Output Compare Register 1 ("Low")	R/W	$FF
0D	Input Capture Register ("High")	R	$00
0E	Input Capture Register ("Low")	R	$00
0F	Timer Control/Status Register 2	R/W	$10
10	Rate, Mode Control Register	R/W	$00
11	Tx/Rx Control Status Register	R/W	$20
12	Receive Data Register	R	$00
13	Transmit Data Register	W	$00
14	RAM/Port 5 Control Register	R/W	$7C or $FC
15	Port 5	R	–
16	Port 6 Data Direction Register	W	$00

(continued)

Table 3 Internal Register

Address	Registers	R/W***	Initialize at RESET
17	Port 6	R/W	Undefined
18*	–	–	–
19	Output Compare Register 2 ("High")	R/W	$FF
1A	Output Compare Register 2 ("Low")	R/W	$FF
1B	Timer Control/Status Register 3	R/W	$20
1C	Time Constant Register	W	$FF
1D	Timer 2 Up Counter	R/W	$00
1E	–	–	–
1F**	Test Register	–	–

* External Address.
** Test Register. Do not access to this register.
*** R : Read Only Register
W : Write Only Register
R/W : Read/Write Register

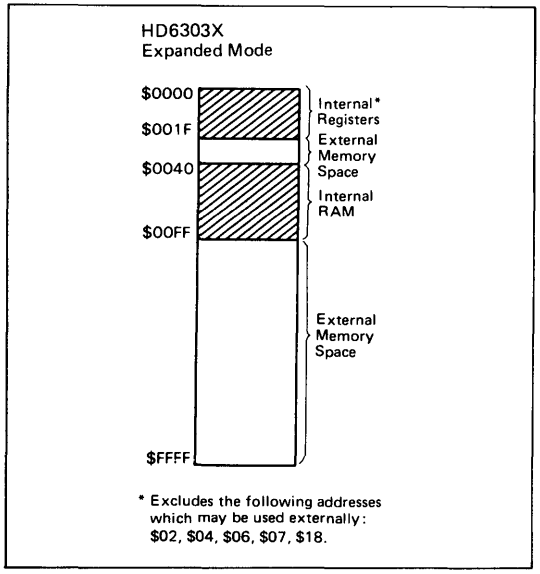


Figure 16 HD6303X Memory Map

■ TIMER 1

The HD6303X provides a 16-bit programmable timer which can simultaneously measure an input waveform and generate two independent output waveforms. The pulse widths of both input/output waveforms vary from microseconds to seconds.

Timer 1 is configurated as follows (refer to Fig. 18).

- Control/Status Register 1 (8 bit)
- Control/Status Register 2 (7 bit)
- Free Running Counter (16 bit)
- Output Compare Register 1 (16 bit)
- Output Compare Register 2 (16 bit)
- Input Capture Register (16 bit)

● Free-Running Counter (FRC) ($0009 : 000A)

The key timer element is a 16-bit free-running counter driven and incremented by system clock. The counter value is readable by software without affecting the counter. The counter is cleared by reset.

When writing to the upper byte ($09), the CPU writes the preset value ($FFF8) into the counter (address $09, $0A) regardless of the write data value. But when writing to the lower byte ($0A) after the upper byte writing, the CPU writes not only the lower byte data into lower 8 bit, but also the upper byte data into higher 8 bit of the FRC.

The counter will be as follows when the CPU writes to it by double store instructions (STD, STX etc.).

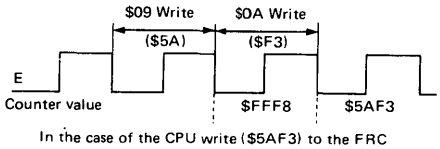


Figure 17 Counter Write Timing

● Output Compare Register (OCR) ($000B, $000C; OCR1) ($0019, $001A ; OCR2)

The output compare register is a 16-bit read/write register which can control an output waveform. The data of OCR is always compared with the FRC.

When the data matches, output compare flag (OCF) in the timer control/status register (TCSR) is set. If an output enable bit (OE) in the TCSR2 is "1", an output level bit (OLVL) in the TCSR will be output to bit 1 (Tout 1) and bit 5 (Tout 2) of port 2. To control the output level again by the next compare, the value of OCR and OLVL should be changed. The OCR is set to $FFFF at reset. The compare function is inhibited for a cycle just after a write to the OCR or to the upper byte of the FRC. This is to begin the comparison after setting the 16-bit value valid in the register and to inhibit the compare function at this cycle, because the CPU writes the upper byte to the FRC, and at the next cycle the counter is set to $FFF8.

* For data write to the FRC or the OCR, 2-byte transfer instruction (such as STX etc.) should be used.

● Input Capture Register (ICR) ($000D : 000E)

The input capture register is a 16-bit read only register which stores the FRC's value when external input signal transition

generates an input capture pulse. Such transition is controlled by input edge bit (IEDG) in the TCSR1.

In order to input the external input signal to the edge detecter, a bit of the DDR corresponding to bit 0 of port 2 should be cleared ("0"). When an input capture pulse occurs by the external input signal transition at the next cycle of CPU's high-byte read of the ICR, the input capture pulse will be delayed by one cycle. In order to ensure the input capture operation, a CPU read of the ICR needs 2-byte transfer instruction. The input pulse width should be at least 2 system cycles. This register is cleared ($0000) during reset.

- **Timer Control/Status Register 1 (TCSR1) ($0008)**

The timer control/status register 1 is an 8-bit register. All bits are readable and the lower 5 bits are also writable. The upper 3 bits are read only which indicate the following timer status.

Bit 5 The counter value reached to $0000 as a result of counting-up (TOF).

Bit 6 A match has occured between the FRC and the OCR 1 (OCF1).

Bit 7 Defined transition of the timer input signal causes the counter to transfer its data to the ICR (ICF).

The followings are each bit descriptions.

Timer Control/Status Register 1

7	6	5	4	3	2	1	0	
ICF	OCF1	TOF	EICI	EOCI1	ETOI	IEDG	OLVL1	$0008

Bit 0 OLVL1 Output Level 1
OLVL1 is transferred to port 2, bit 1 when a match occurs between the counter and the OCR1. If bit 0 of the TCSR2 (OE1) is set to "1", OLVL1 will appear at bit 1 of port 2.

Bit 1 IEDG Input Edge
This bit determines which edge, rising or falling, of input signal of port 2, bit 0 will trigger data transfer from the counter to the ICR. For this function, the DDR corresponding to port 2, bit 0 should be cleared beforehand.
IEDG=0, triggered on a falling edge ("High" to "Low")
IEDG=1, triggered on a rising edge ("Low" to "High")

Bit 2 ETOI Enable Timer Overflow Interrupt
When this bit is set, an internal interrupt (IRQ_3) by TOI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 3 EOCI1 Enable Output Compare Interrupt 1
When this bit is set, an internal interrupt (IRQ_3) by OCI1 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 EICI Enable Input Capture Interrupt
When this bit is set, an internal interrupt (IRQ_3) by ICI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag
This read-only bit is set when the counter increments from $FFFF by 1. Cleared when the counter's upper byte ($0009) is ready by the CPU after the TCSR1 read.

Bit 6 OCF1 Output Compare Flag 1
This read-only bit is set when a match occurs between the OCR1 and the FRC. Cleared when writing to the OCR1 ($000B or $000C) after the TCSR1 or TCSR2 read.

Bit 7 ICF Input Capture Flag
This read-only bit is set when an input signal of port 2, bit 0 makes a transition as defined by IEDG and the FRC is transferred to the ICR. Cleared when reading the upper byte ($000D) of the ICR following the TCSR1 or TCSR2 read.

- **Timer Control/Status Register 2 (TCSR2) ($000F)**

The timer control/status register 2 is a 7-bit register. All bits are readable and the lower 4 bits are also writable. But the upper 3 bits are read-only which indicate the following timer status.

Bit 5 A match has occured between the FRC and the OCR2 (OCF2).

Bit 6 The same status flag as the OCF1 flag of the TCSR1, bit 6.

Bit 7 The same status flag as the ICF flag of the TCSR1, bit 7.

The followings are the each bit descriptions.

Timer Control/Status Register 2

7	6	5	4	3	2	1	0	
ICF	OCF1	OCF2	–	EOCI2	OLVL2	OE2	OE1	$000F

Bit 0 OE1 Output Enable 1
This bit enables the OLVL1 to appear at port 2, bit 1 when a match has occurred between the counter and the output compare register 1. When this bit is cleared, bit 1 of port 2 will be an I/O port. When set, it will be an output of OLVL1 automatically.

Bit 1 OE2 Output Enable 2
This bit enables the OLVL2 to appear at port 2, bit 5 when a match has occurred between the counter and the output compare register 2. When this bit is cleared, port 2, bit 5 will be an I/O port. When set, it will be an output of OLVL2 automatically.

Bit 2 OLVL2 Output Level 2
OLVL2 is transferred to port 2, bit 5 when a match has occurred between the counter and the OCR2. If bit 5 of the TCSR2 (OE2) is set to "1", OLVL2 will appear at port 2, bit 5.

Bit 3 EOCI2 Enable Output Compare Interrupt 2
When this bit is set, an internal interrupt (IRQ_3) by OCI2 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 Not Used

Bit 5 OCF2 Output Compare Flag 2
This read-only bit is set when a match has occurred between the counter and the OCR2. Cleared when writing to the OCR2 ($0019 or $001A) after the TCSR2 read.

Bit 6 OCF1 Output Compare Flag 1

Bit 7 ICF Input Capture Flag
OCF1 and ICF addresses are partially decoded. The CPU read of the TCSR1/TCSR2 makes it possible to read OCF1 and ICF into bit 6 and bit 7.

Both the TCSR1 and TCSR2 will be cleared during reset.

(Note) If OE1 or OE2 is set to "1" before the first output compare match occurs after reset restart, bit 1 or bit 5 of port 2 will produce "0" respectively.

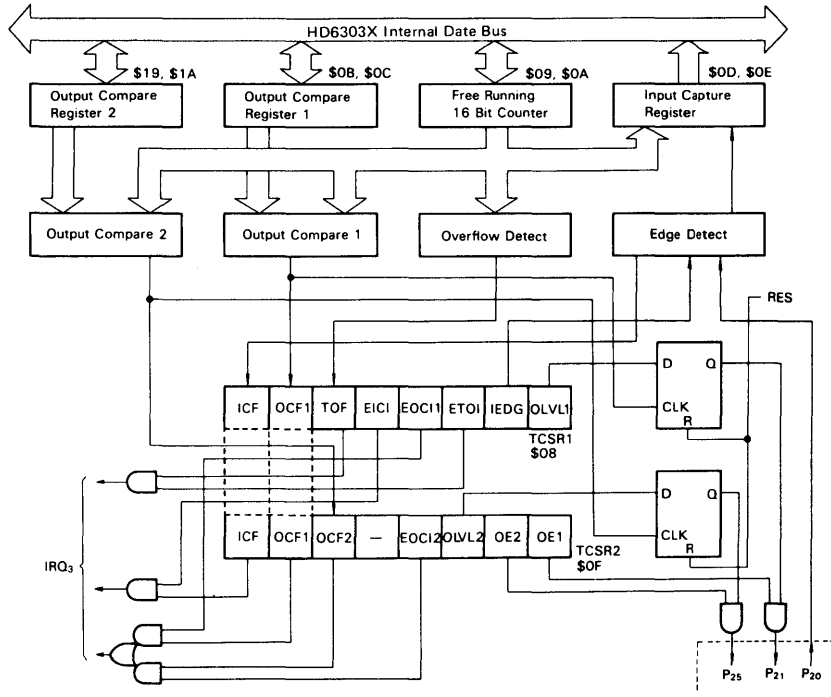


Figure 18 Timer 1 Block Diagram

■ TIMER 2

In addition to the timer 1, the HD6303X provides an 8-bit reloadable timer, which is capable of counting the external event. This timer 2 contains a timer output, so the MPU can generate three independent waveforms (refer to Fig. 19).

The timer 2 is configured as follows:

Control/Status Register 3 (7 bit)
8-bit Up Counter
Time Constant Register (8 bit)

• Timer 2 Up Counter (T2CNT) ($001D)

This is an 8-bit up counter which operates with the clock decided by CKS0 and CKS1 of the TCSR3. The CPU can read the value of the counter without affecting the counter. In addition, any value can be written to the counter by software even during counting.

The counter is cleared when a match occurs between the counter and the TCONR or during reset.

If a write operation is made by software to the counter at the cycle of counter clear, it does not reset the counter but put the write data to the counter.

• Time Constant Register (TCONR) ($001C)

The time constant register is an 8-bit write only register. It is always compared with the counter.

When a match has occurred, counter match flag (CMF) of the timer control status register 3 (TCSR3) is set and the value selected by TOS0 and TOS1 of the TCSR3 will appear at port 2, bit 6. When CMF is set, the counter will be cleared simultaneously and then start counting from $00. This enables regular interrupts and waveform outputs without any software support. The TCONR is set to "$FF" during reset.

• Timer Control/Status Register 3 (TCSR3) ($001B)

The timer control/status register 3 is a 7-bit register. All bits are readable and 6 bits except for CMF can be written.

The followings are each pin descriptions.

Timer Control/Status Register 3

7	6	5	4	3	2	1	0	
CMF	ECMI	—	T2E	TOS1	TOS0	CKS1	CKS0	$001B

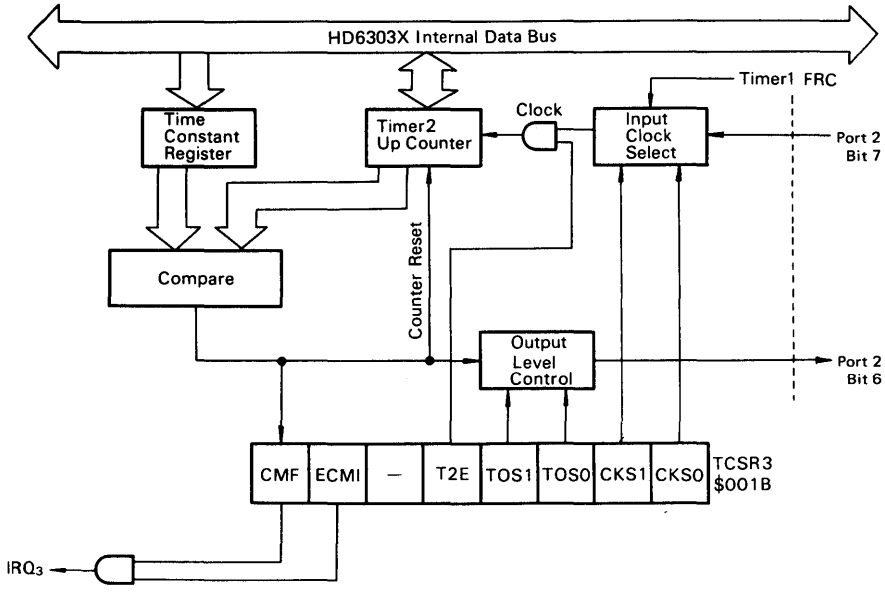


Figure 19 Timer 2 Block Diagram

Bit 0 CKS0 Input Clock Select 0
Bit 1 CKS1 Input Clock Select 1

Input clock to the counter is selected as shown in Table 4 depending on these two bits. When an external clock is selected, bit 7 of port 2 will be a clock input automatically. Timer 2 detects the rising edge of the external clock and increments the counter. The external clock is countable up to half the frequency of the system clock.

Table 4 Input Clock Select

CKS1	CKS0	Input Clock to the Counter
0	0	E clock
0	1	E clock/8*
1	0	E clock/128*
1	1	External clock

* These clocks come from the FRC of the timer 1. If one of these clocks is selected as an input clock to the up counter, the CPU should not write to the FRC of the timer 1.

Bit 2 TOS0 Timer Output Select 0
Bit 3 TOS1 Timer Output Select 1

When a match occurs between the counter and the TCONR timer 2 outputs shown in Table 5 will appear at port 2, bit 6 depending on these two bits. When both TOS0 and TOS1 are "0", bit 6 of port 2 will be an I/O port.

Table 5 Timer 2 Output Select

TOS1	TOS0	Timer Output
0	0	Timer Output Inhibited
0	1	Toggle Output*
1	0	Output "0"
1	1	Output "1"

* When a match occurs between the counter and the TCONR, timer 2 output level is reversed. This leads to production of a square wave with 50% duty to the external without any software support.

Bit 4 T2E Timer 2 Enable Bit

When this bit is cleared, a clock input to the up counter is prohibited and the up counter stops. When set to "1", a clock selected by CKS1 and CKS0 (Table 4) is input to the up counter.

(Note) P_{26} outputs "0" when T2E bit cleared and timer 2 set in output enable condition by TOS1 or TOS0. It also outputs "0" when T2E bit set "1" and timer 2 set in output enable condition before the first counter match occurs.

Bit 5 Not Used
Bit 6 ECMI Enable Counter Match Interrupt

When this bit is set, an internal interrupt (IRQ_3) by CMI is enabled. When cleared, the interrupt is inhibited.

Bit 7 CMF Counter Match Flag

This read-only bit is set when a match occurs between the up counter and the TCONR. Cleared by writing "0" by software write (unable to write "1" by software).

Each bit of the TCSR3 is cleared during reset.

■ SERIAL COMMUNICATION INTERFACE (SCI)

The HD6303X SCI contains two operation modes; one is an asynchronous mode by the NRZ format and the other is a clocked synchronous mode which transfers data synchronizing with the serial clock.

The SCI consists of the following registers as shown in Fig. 20 Block Diagram:

- Control/Status Register (TRCSR)
- Rate/Mode Control Register (RMCR)
- Receive Data Register (RDR)
- Receive Data Shift Register (RDSR)
- Transmit Data Register (TDR)
- Transmit Data Shift Register (TDSR)

The serial I/O hardware requires an initialization by software for operation. The procedure is usually as follows:

1) Write a desirable operation mode into each corresponding control bit of the RMCR.
2) Write a desirable operation mode into each corresponding control bit of the TRCSR.

When using bit 3 and 4 of port 2 for serial I/O only, there is no problem even if TE and RE bit are set. But when setting the baud rate and operation mode, TE and RE should be "0". When clearing TE and RE bit and setting them again, more than 1 bit cycle of the current baud rate is necessary. If set in less than 1 bit cycle, there may be a case that the internal transmit/receive initialization fails.

• Asynchronous Mode

An asynchronous mode contains the following two data formats:

1 Start Bit + 8 Bit Data + 1 Stop Bit
1 Start Bit + 9 Bit Data + 1 Stop Bit

In addition, if the 9th bit is set to "1" when making 9 bit data format, the format of

1 Start bit + 8 Bit Data + 2 Stop Bit

is also transferred.

Data transmission is enabled by setting TE bit of the TRCSR, then port 2, bit 4 will become a serial output independently of the corresponding DDR.

For data transmit, both the RMCR and TRCSR should be set under the desirable operating conditions. When TE bit is set during this process, 10 bit preamble will be sent in 8-bit data format and 11 bit in 9-bit data format. When the preamble is produced, the internal synchronization will become stable and the transmitter is ready to act.

The conditions at this stage are as follows.

1) If the TDR is empty (TDRE=1), consecutive 1's are produced to indicate the idle state.
2) If the TDR contains data (TDRE=0), data is sent to the transmit data shift register and data transmit starts.

During data transmit, a start bit of "0" is transmitted first. Then 8-bit or 9-bit data (starts from bit 0) and a stop bit "1" are transmitted.

When the TDR is "empty", hardware sets TDRE flag bit. If the CPU doesn't respond to the flag in proper timing (the TDRE is in set condition till the next normal data transfer starts from the transmit data register to the transmit sift register), "1" is transferred instead of the start bit "0" and continues to be transferred till data is provided to the data register. While the TDRE is "1", "0" is not transferred.

Data receive is possible by setting RE bit. This makes port 2, bit 3 be a serial input. The operation mode of data receive is decided by the contents of the TRCSR and RMCR. The first "0" (space) synchronizes the receive bit flow. Each bit of the following data will be strobed in the middle. If a stop bit is not "1", a framing error assumed and ORFE is set.

When a framing error occurs, receive data is transferred to the receive data register and the CPU can read error-generating data. This makes it possible to detect a line break.

If the stop bit is "1", data is transferred to the receive data register and an interrupt flag RDRF is set. If RDRF is still set when receiving the stop bit of the next data, ORFE is set to indicate overrun generation.

When the CPU read the receive data register as a response to RDRF flag or ORFE flag after having read TRCS, RDRF or ORFE is cleared.

(Note) Clock Source in Asynchronous Mode

If CC1 : CC0 = 10, the internal bit rate clock is provided at P_{22} regardless of the values for TE or RE. Maximum clock rate is E ÷ 16.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P_{22} at sixteen times (16×) the desired bit rate, but not greater than E.

• Clocked Synchronous Mode

In the clocked synchronous mode, data transmit is synchronized with the clock pulse. The HD6303X SCI provides functionally independent transmitter and receiver which makes full duplex operation possible in the asynchronous mode. But in the clocked synchronous mode an SCI clock I/O pin is only P_{22}, so the simultaneous receive and transmit operation is not available. In this mode, TE and RE should not be in set condition ("1") simultaneously. Fig. 21 gives a synchronous clock and a data format in the clocked synchronous mode.

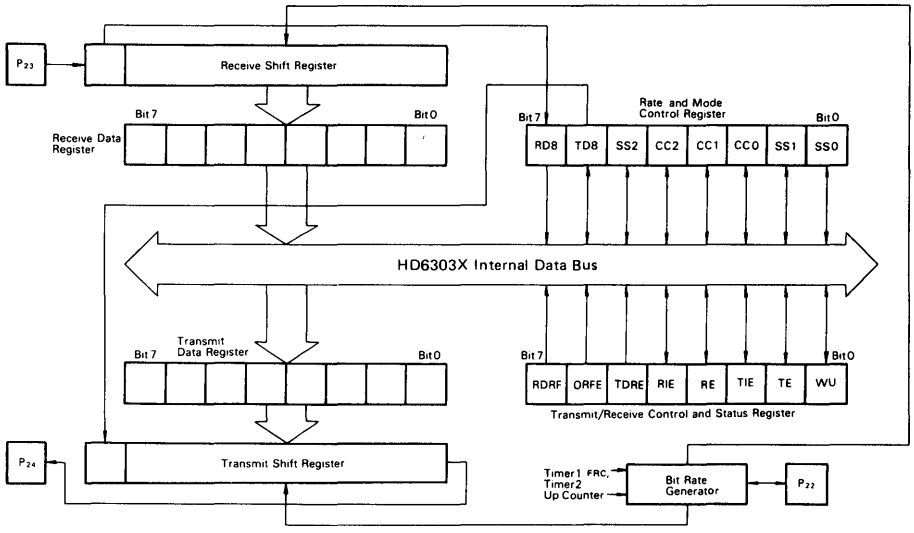


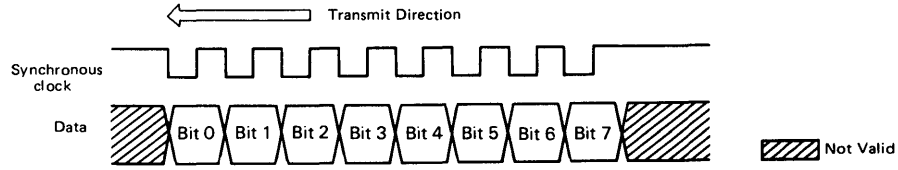
Figure 20 Serial Communication Interface Block Diagram

Data transmit is realized by setting TE bit in the TRCSR. Port 2, bit 4 becomes an output unconditionally independent of the value of the corresponding DDR.

Both the RMCR and TRCSR should be set in the desirable operating condition for data transmit.

When an external clock input is selected, data transmit is performed under the TDRE flag "0" from port 2, bit 4, synchronizing with 8 clock pulses input from external to port 2, bit 2.

Data is transmitted from bit 0 and the TDRE is set when the transmit data shift register is "empty". More than 9th clock pulse of external are ignored.



- Transmit data is output from a falling edge of a synchronous clock to the next falling edge.
- Receive data is latched at the rising edge.

Figure 21 Clocked Synchronous Mode Format

When data transmit is selected to the clock output, the MPU produces transmit data and synchronous clock at TDRE flag clear.

Data receive is enabled by setting RE bit. Port 2, bit 3 will be a serial input. The operating mode of data receive is decided by the TRCSR and the RMCR.

If the external clock input is selected, RE bit should be set when P_{22} is "High". Then 8 external clock pulses and the synchronized receive data are input to port 2, bit 2 and bit 3 respectively. The MPU put receive data into the receive data shift register by this clock and set the RDRF flag at the termination of 8 bit data receive. More than 9th clock pulse of external input are ignored. When RDRF is cleared by reading the receive data register, the MPU starts receiving the next data. So RDRF should be cleared with P_{22} "High"

When data receive is selected to the clock output, 8 synchronous clocks are output to the external by setting RE bit. So receive data should be input from external, synchronously with this clock. When the first byte data is received, the RDRF flag is set. After the second byte, receive operation is performed and output the synchronous clock to the external by clearing the RDRF bit.

- **Transmit/Receive Control Status Register (TRCSR) ($0011)**

The TRCSR is composed of 8 bits which are all readable. Bits 0 to 4 are also writable. This register is initialized to $20 during reset. Each bit functions as follows.

Transmit/Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU Wake-up

In a typical multi-processor configuration, the software protocol provides the destination address at the first byte of the message. In order to make uninterested MPU ignore the remaining message, a wake-up function is available. By this, uninterested MPU can inhibit all further receive processing till the next message starts.

Then wake-up function is triggered by consecutive 1's with 1 frame length (10 bits for 8-bit data, 11 for 9-bit). The software protocol should provide the idle time between messages.

By setting this bit, the MPU stops data receive till the next message. The receive of consecutive "1" with one frame length wakes up and clears this bit and then the MPU restarts receive operation. However, the RE flag should be already set before setting this bit. In the clocked synchronous mode WU is not available, so this bit should not be set.

Bit 1 TE Transmit Enable

When this bit is set, transmit data will appear at port 2, bit 4 after one frame preamble in asynchronous mode, while in clocked synchronous mode it appears immediately. This is executed regardless of the value of the corresponding DDR. When TE is cleared, the serial I/O doesn't affect port 2, bit 4.

Bit 2 TIE Transmit Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when TDRE (bit 5) is set. When cleared, the interrupt is inhibited.

Bit 3 RE Receive Enable

When set, a signal is input to the receiver from port 2, bit 3 regardless of the value of the DDR. When RE is cleared, the serial I/O doesn't affect port 2, bit 3.

Bit 4 RIE Receive Interrupt Enable

When this bit is set, an internal interrupt, IRQ_3 is enabled when RDRF (bit 7) or ORFE (bit 6) is set. When cleared, the interrupt is inhibited.

Bit 5 TDRE Transmit Data Register Empty

TDRE is set when the TDR is transferred to the transmit data shift register in the asynchronous mode, while in clocked synchronous mode when the TDSR is "empty". This bit is reset by reading the TRCSR and writing new transmit data to the transmit data register. TDRE is set to "1" during reset.

(Note) TDRE should be cleared in the transmittable state after the TE set.

Bit 6 ORFE Overrun Framing Error

ORFE is set by hardware when an overrun or a framing error is generated (during data receive only). An overrun error occurs when new receive data is ready to be transferred to the RDR during RDRF still being set. A framing error occurs when a stop bit is "0". But in clocked synchronous mode, this bit is not affected. This bit is cleared when reading the TRCSR, then the RDR, or during reset.

Bit 7 RDRF Receive Data Register Full

RDRF is set by hardware when the RDSR is transferred to the RDR. Cleared when reading the TRCSR, then the RDR, or during reset.

(Note) When a few bits are set between bit 5 to bit 7 in the TRCSR, a read of the TRCSR is sufficient for clearing those bits. It is not necessary to read the TRCSR every-time to clear each bit.

- **Transmit Rate/Mode Control Register (RMCR)**

The RMCR controls the following serial I/O:

- Baud Rate
- Clock Source
- Data Format
- Port 2, Bit 2 Function

In addition, if 9-bit data format is set in the asynchronous mode, the 9th bit is put in this register. All bits are readable and writable except bit 7 (read only). This register is set to $0C during reset.

Transfer Rate/Mode Control Register

7	6	5	4	3	2	1	0	
RD8	TD8	SS2	CC2	CC1	CC0	SS1	SS0	$0010

Bit 0	SS0	Speed Select
Bit 1	SS1	
Bit 5	SS2	

These bits control the baud rate used for the SCI. Table 6 lists the available baud rates. The timer 1 FRC (SS2=0) and the timer 2 up counter (SS2=1) provide the internal clock to the SCI. When selecting the timer 2 as a baud rate source, it functions as a baud rate generator. The timer 2 generates the baud rate listed in Table 7 depending on the value of the TCONR.

(Note) When operating the SCI with internal clock, do not perform write operation to the timer/counter which is the clock source of the SCI.

Bit 2	CC0	Clock Control/Format Select*
Bit 3	CC1	
Bit 4	CC2	

These bits control the data format and the clock source (refer to Table 8).

* CC0, CC1 and CC2 are cleared during reset and the MPU goes to the clocked synchronous mode of the external clock operation. Then the MPU sets port 2, bit 2 into the clock input state. When using port 2, bit 2 as an output port, the DDR of port 2 should be set to "1" and CC1 and CC0 to "0" and "1" respectively.

Table 6 SCI Bit Times and Transfer Rates

(1) Asynchronous Mode

SS2	SS1	SS0	XTAL / E	2.4576MHz / 614.4kHz	4.0MHz / 1.0MHz	4.9152MHz / 1.2288MHz
0	0	0	E÷16	26μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

* When SS2 is "1", Timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32\,(N+1)} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

(2) Clocked Synchronous Mode *

SS2	SS1	SS0	XTAL / E	4.0MHz / 1.0MHz	6.0MHz / 1.5MHz	8.0MHz / 2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

* Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC ~ 1/2 system clock.

** The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate } (\mu s/\text{bit}) = \frac{4\,(N+1)}{f} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

Table 7 Baud Rate and Time Constant Register Example

Baud Rate (Baud) \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

* E/8 clock is input to the timer 2 up counter and E clock otherwise.

Table 8 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock Source	Port 2, Bit 2	Port 2, Bit 3	Port 2, Bit 4
0	0	0	8-bit data	Clocked Synchronous	External	Input	When the TRCSR, RE bit is "1", bit 3 is used as a serial input.	When the TRCSR, TE bit is "1", bit 4 is used as a serial output.
0	0	1	8-bit data	Asynchronous	Internal	Not Used**		
0	1	0	8-bit data	Asynchronous	Internal	Output*		
0	1	1	8-bit data	Asynchronous	External	Input		
1	0	0	8-bit data	Clocked Synchronous	Internal	Output		
1	0	1	9-bit data	Asynchronous	Internal	Not Used**		
1	1	0	9-bit data	Asynchronous	Internal	Output*		
1	1	1	9-bit data	Asynchronous	External	Input		

* Clock output regardless of the TRCSR, bit RE and TE.
** Not used for the SCI.

Bit 6 TD8 Transmit Data Bit 8

When selecting 9-bit data format in the asynchronous mode, this bit is transmitted as the 9th data. In transmitting 9-bit data, write the 9th data into this bit then write data to the receive data register.

Bit 7 RD8 Receive Data Bit 8

When selecting 9-bit data format in the asynchronous mode, this bit stores the 9th bit data. In receiving 9-bit data, read this bit then the receive data register.

■ TIMER, SCI STATUS FLAG

Table 9 shows the set and reset conditions of each status flag in the timer 1, timer 2 and SCI.

As for Timer 1 and Timer 2 status flag, if the set and reset condition occur simultaneously, the set condition is prior to the reset condition. But in case of SCI control status flag, the reset condition has priority. Especially as for OCF1 and OCF2 of Timer 1, the set signal is generated periodically whenever FRC matches OCR after the set, and which can cause the unclear of the flag. To clear surely, the method is necessary to avoid the occurence of the set signal between TCSR Read and OCR write. For example, match the OCR value to FRC first, and next read TCSR, and then write OCR at once.

Table 9 Timer 1, Timer 2 and SCI Status Flag

		Set Condition	Reset Condition
Timer 1	ICF	FRC → ICR by edge input to P_{20}.	1. Read the TCSR1 or TCSR2 then ICRH, when ICF=1 2. $\overline{RES}$=0
	OCF1	OCR1=FRC	1. Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1=1 2. $\overline{RES}$=0
	OCF2	OCR2=FRC	1. Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2=1 2. $\overline{RES}$=0
	TOF	FRC=$FFFF+1 cycle	1. Read the TCSR1 then FRCH, when TOF=1 2. $\overline{RES}$=0
Timer 2	CMF	T2CNT=TCONR	1. Write "0" to CMF, when CMF=1 2. $\overline{RES}$=0
SCI	RDRF	Receive Shift Register → RDR	1. Read the TRCSR then RDR, when RDRF=1 2. $\overline{RES}$=0
	ORFE	1. Framing Error (Asynchronous Mode) Stop Bit = 0 2. Overrun Error (Asynchronous Mode) Receive Shift Register → RDR when RDRF=1	1. Read the TRCSR then RDR, when ORFE=1 2. $\overline{RES}$=0
	TDRE	1. Asynchronous Mode TDR → Transmit Shift Register 2. Clocked Synchronous Mode Transmit Shift Register is "empty" 3. $\overline{RES}$=0	Read the TRCSR then write to the TDR, when TDRE=1 (Note) TDRE should be reset after the TE set.

(Note) 1. →; transfer
2. For example; "ICRH" means High byte of ICR.

■ LOW POWER DISSIPATION MODE

The HD6303X provides two low power dissipation modes; sleep and standby.

• Sleep Mode

The MPU goes to the sleep mode by SLP instruction execution. In the sleep mode, the CPU stops its operation, while the registers' contents are retained. In this mode, the peripherals except the CPU such as timers, SCI etc. continue their functions. The power dissipation of sleep-condition is one fifth that of operating condition.

The MPU returns from this mode by an interrupt, $\overline{RES}$ or $\overline{STBY}$; it goes to the reset state by $\overline{RES}$ and the standby mode by $\overline{STBY}$. When the CPU acknowledges an interrupt request, it cancels the sleep mode, returns to the operation mode and branches to the interrupt routine. When the CPU masks this interrupt, it cancels the sleep mode and executes the next instruction. However, for example if the timer 1 or 2 prohibits a timer interrupt, the CPU doesn't cancel the sleep mode because of no interrupt request.

This sleep mode is effective to reduce the power dissipation for a system with no need of the HD6303X's consecutive operation.

• Standby Mode

The HD6303X stops all the clocks and goes to the reset state with $\overline{STBY}$ "Low". In this mode, the power dissipation is reduced conspicuously. All pins except for the power supply, the $\overline{STBY}$ and XTAL are detached from the MPU internally and go to the high impedance state.

In this mode the power is supplied to the HD6303X, so the contents of RAM is retained. The MPU returns from this mode during reset. The followings are typical usage of this mode.

Save the CPU information and SP contents on RAM by $\overline{NMI}$. Then disable the RAME bit of the RAM control register and set the STBY PWR bit to go to the standby mode. If the STBY PWR bit is still set at reset start, that indicates the power is supplied to the MPU and RAM contents are retained properly. So system can restore itself by returning their pre-standby informations to the SP and the CPU. Fig. 22 depicts the timing at each pin with this example.

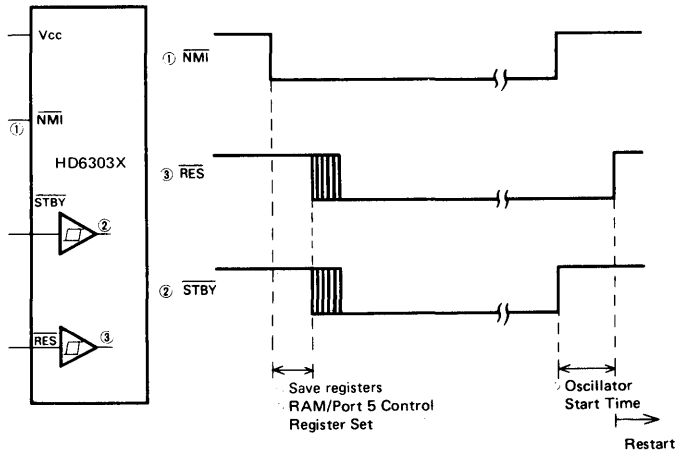


Figure 22 Standby Mode Timing

■ TRAP FUNCTION

The CPU generates an interrupt with the highest priority (TRAP) when fetching an undefined instruction or an instruction from non-memory space. The TRAP prevents the system-burst caused by noise or a program error.

• Op Code Error

When fetching an undefined op code, the CPU saves CPU registers as well as a normal interrupt and branches to the TRAP ($FFEE, $FFEF). This has the priority next to reset.

• Address Error

When an instruction fetch is made from internal register ($0000~$001F), the MPU generates an interrupt as well as an op code error. But on the system with no memory in its external memory area, this function is not applicable if an instruction fetch is made from the external non-memory area.

This function is available only for an instruction fetch and is not applicable to the access of normal data read/write.

(Note) The TRAP interrupt provides a retry function differently from other interrupts. This is a program flow return to the address where the TRAP occurs when a sequence returns to a main routine from the TRAP interrupt routine by RTI. The retry can prevent the system burst caused by noise etc.
However, if another TRAP occurs, the program repeats the TRAP interrupt forever, so the consideration is necessary in programming.

■ INSTRUCTION SET

The HD6303X provides object code upward compatible with the HD6801 to utilize all instruction set of the HMCS6800. It also reduces the execution times of key instruc-

tions for throughput improvement.

Bit manipulation instruction, change instruction of the index register and accumulator and sleep instruction are also added.

The followings are explained here.

- CPU Programming Model (refer to Fig. 23)
- Addressing Mode
- Accumulator and Memory Manipulation Instruction (refer to Table 10)
- New Instruction
- Index Register and Stack Manipulation Instruction (refer to Table 11)
- Jump and Branch Instruction (refer to Table 12)
- Condition Code Register Manipulation (refer to Table 13)
- Op Code Map (refer to Table 14)

• Programming Model

Fig. 23 depicts the HD6303X programming model. The double accumulator D consists of accumulator A and B, so when using the accumulator D, the contents of A and B are destroyed.

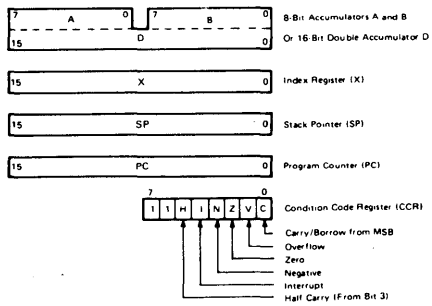


Figure 23 CPU Programming Model

• CPU Addressing Mode

The HD6303X provides 7 addressing modes. The addressing mode is decided by an instruction type and code. Table 10 through 14 show addressing modes of each instruction with the execution times counted by the machine cycle.

When the clock frequency is 4 MHz, the machine cycle time becomes microseconds directly.

Accumulator (ACCX) Addressing

Only an accumulator is addressed and the accumulator A or B is selected. This is a one-byte instruction.

Immediate Addressing

This addressing locates a data in the second byte of an instruction. However, LDS and LDX locate a data in the second and third byte exceptionally. This addressing is a 2 or 3-byte instruction.

Direct Addressing

In this addressing mode, the second byte of an instruction shows the address where a data is stored. 256 bytes ($0 through $255) can be addressed directly. Execution times can be reduced by storing data in this area so it is recommended to make it RAM for users' data area in configurating a system. This is a 2-byte instruction, while 3-byte with regard to AIM, OIM, EIM and TIM.

Extended Addressing

In this mode, the second byte shows the upper 8 bit of the data stored address and the third byte the lower 8 bit. This indicates the absolute address of 3-byte instruction in the memory.

Indexed Addressing

The second byte of an instruction and the lower 8 bit of the index register are added in this mode. As for AIM, OIM, EIM and TIM, the third byte of an instruction and the lower 8 bits of the index register are added.

This carry is added to the upper 8 bit of the index register and the result is used for addressing the memory. The modified address is retained in the temporary address register, so the contents of the index register doesn't change. This is a 2-byte instruction except AIM, OIM, EIM and TIM (3-byte instruction).

Implied Addressing

An instruction itself specifies the address. That is, the instruction addresses a stack pointer, index register etc. This is a one-byte instruction.

Relative Addressing

The second byte of an instruction and the lower 8 bits of the program counter are added. The carry or borrow is added to the upper 8 bit. So addressing from −126 to +129 byte of the current instruction is enabled. This is a 2-byte instruction.

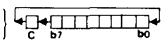
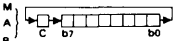
(Note) CLI, SEI Instructions and Interrupt Operation

When accepting the IRQ at a preset timing with CLI and SEI instructions, more than 2 cycles are necessary between the CLI and SEI instructions. For example, the following program (a) (b) don't accept the IRQ but (c) accepts it.

.	.	.
.	.	.
.	.	.
.	.	CLI
CLI	CLI	NOP
SEI	NOP	NOP
.	SEI	SEI
.	..	.
.	.	.
.	.	.
.	.	.
(a)	(b)	(c)

The same thing can be said to the TAP instruction instead of the CLI and SEI instructions.

Table 10 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	④
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M→ B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 →M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3				M, A, B 	•	•	↕	↕	⑮	↕
	ROLA													49	1	1		•	•	↕	↕	⑮	↕
	ROLB													59	1	1		•	•	↕	↕	⑮	↕
Rotate Right	ROR							66	6	2	76	6	3				M, A, B 	•	•	↕	↕	⑮	↕
	RORA													46	1	1		•	•	↕	↕	⑮	↕
	RORB													56	1	1		•	•	↕	↕	⑮	↕

(Note) Condition Code Register will be explained in Note of Table 13.

(continued)

Table 10 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M ← 0	•	•	↕	↕	⑥	↕
	ASLA													48	1	1	A ← 0	•	•	↕	↕	⑥	↕
	ASLB													58	1	1	B ← 0	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	C ← ACC A/ ACC B ← 0 (A7 A0 B7 B0)	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M →	•	•	↕	↕	⑥	↕
	ASRA													47	1	1	A →	•	•	↕	↕	⑥	↕
	ASRB													57	1	1	B →	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M → 0	•	•	R	↕	⑥	↕
	LSRA													44	1	1	A → 0	•	•	R	↕	⑥	↕
	LSRB													54	1	1	B → 0	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	0 → ACC A/ ACC B → C (A7 A0 B7 B0)	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

(Note) Condition Code Register will be explained in Note of Table 13.

- **Additional Instruction**

In addition to the HD6801 instruction set, the HD6303X prepares the following new instructions.

AIM . . . . . (M)·(IMM) → (M)

Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM . . . . . (M) + (IMM) → (M)

Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM . . . . . (M) ⊕ (IMM) → (M)

Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM . . . . . (M) · (IMM)

Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These area 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX . . . . . (ACCD) ↔ (IX)

Exchanges the contents of accumulator and the index register.

SLP

Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 11 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X − M:M + 1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M + 1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD·IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 13.

Table 12 Jump, Branch Instructions

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——		⑧		——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

(Note) * WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 13.

Table 13 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	———	⑩	———			
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MCU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
- Arithmetic Minus
• Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3 to bit 4
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from/to bit 7
R Reset Always
S Set Always
↕ Set if true after test or clear
• Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕ C = 1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 14 OP-Code Map

OP CODE HI	LO	ACC A	ACC B	IND	EXT / DIR*	ACCA or SP IMM	ACCA or SP DIR	ACCA or SP IND	ACCA or SP EXT	ACCB or X IMM	ACCB or X DIR	ACCB or X IND	ACCB or X EXT	HI
0000	0		SBA	BRA	TSX	NEG				SUB				0
0001	1	NOP	CBA	BRN	INS	AIM				CMP				1
0010	2			BHI	PULA	OIM				SBC				2
0011	3			BLS	PULB	COM				SUBD			ADDD	3
0100	4	LSRD		BCC	DES	LSR				AND				4
0101	5	ASLD		BCS	TXS	EIM				BIT				5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA				6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA		STA	7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR				8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC				9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA				A
1011	B	SEV	ABA	BMI	RTI	TIM				ADD				B
1100	C	CLC		BGE	PSHX	INC				CPX			LDD	C
1101	D	SEC		BLT	MUL	TST				BSR	JSR		STD	D
1110	E	CLI		BGT	WAI	JMP				LDS			LDX	E
1111	F	SEI		BLE	SWI	CLR					STS		STX	F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

■ CPU OPERATION

• CPU Instruction Flow

When operating, the CPU fetches an instruction from a memory and executes the required function. This sequence starts with $\overline{RES}$ cancel and repeats itself limitlessly if not affected by a special instruction or a control signal. SWI, RTI, WAI and SLP instructions change this operation, while $\overline{NMI}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, IRQ_3, $\overline{HALT}$ and $\overline{STBY}$ control it. Fig. 24 gives the CPU mode transition and Fig. 25 the CPU system flow chart. Table 15 shows CPU operating states and port states.

• Operation at Each Instruction Cycle

Table 16 shows the operation at each instruction cycle. By the pipeline control of the HD6303X, MULT, PUL, DAA and XGDX instructions etc. prefetch the next instruction. So attention is necessary to the counting of the instruction cycles because it is different from the usual one ------ op code fetch to the next instruction op code.

Table 15 CPU Operation State and Port State

Port	Reset	STBY***	HALT	Sleep
$A_0 \sim A_7$	H	T	T	H
Port 2	T	T	Keep	Keep
$D_0 \sim D_7$	T	T	T	T
$A_8 \sim A_{15}$	H	T	T	H
Port 5	T	T	T	T
Port 6	T	T	Keep	Keep
Control Signal	*	T	**	*

H ; High, L ; Low, T ; High Impedance
* $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$ = H, BA = L
** $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ = T, $\overline{LIR}$, BA = H
*** E pin goes to high impedance state.

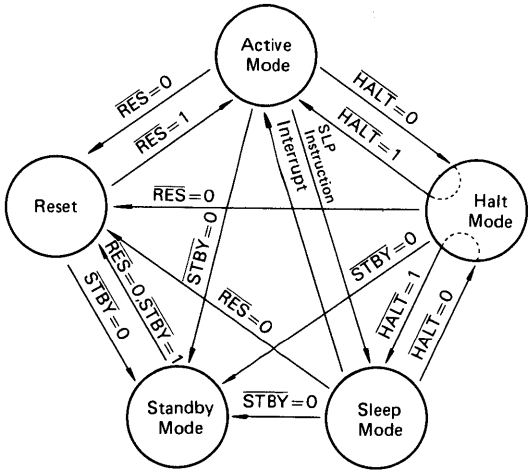
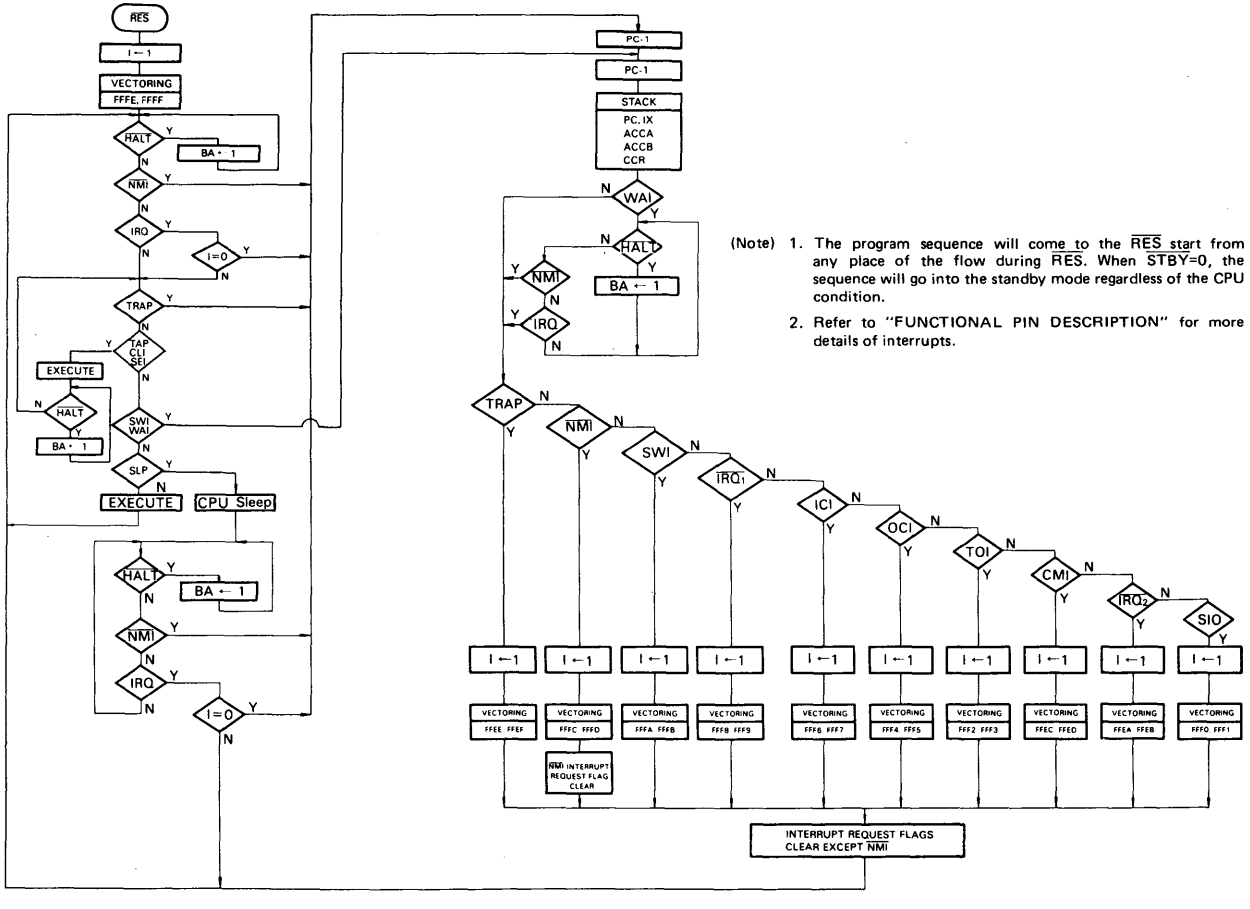


Figure 24 CPU Operation Mode Transition



(Note) 1. The program sequence will come to the RES start from any place of the flow during RES. When STBY=0, the sequence will go into the standby mode regardless of the CPU condition.

2. Refer to "FUNCTIONAL PIN DESCRIPTION" for more details of interrupts.

Figure 25 HD6303X System Flow Chart

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 16 Cycle-by-Cycle Operation

Address Mode & Instructions		Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMMEDIATE									
ADC	ADD	2	1	Op Code Address+1	1	0	1	1	Operand Data
AND	BIT		2	Op Code Address+2	1	0	1	0	Next Op Code
CMP	EOR								
LDA	ORA								
SBC	SUB								
ADDD	CPX	3	1	Op Code Address+1	1	0	1	1	Operand Data (MSB)
LDD	LDS		2	Op Code Address+2	1	0	1	1	Operand Data (LSB)
LDX	SUBD		3	Op Code Address+3	1	0	1	0	Next Op Code
DIRECT									
ADC	ADD	3	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
AND	BIT		2	Address of Operand	1	0	1	1	Operand Data
CMP	EOR		3	Op Code Address+2	1	0	1	0	Next Op Code
LDA	ORA								
SBC	SUB								
STA		3	1	Op Code Address+1	1	0	1	1	Destination Address
			2	Destination Address	0	1	0	1	Accumulator Data
			3	Op Code Address+2	1	0	1	0	Next Op Code
ADDD	CPX	4	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
LDD	LDS		2	Address of Operand	1	0	1	1	Operand Data (MSB)
LDX	SUBD		3	Address of Operand+1	1	0	1	1	Operand Data (LSB)
			4	Op Code Address+2	1	0	1	0	Next Op Code
STD	STS	4	1	Op Code Address+1	1	0	1	1	Destination Address (LSB)
STX			2	Destination Address	0	1	0	1	Register Data (MSB)
			3	Destination Address+1	0	1	0	1	Register Data (LSB)
			4	Op Code Address+2	1	0	1	0	Next Op Code
JSR		5	1	Op Code Address+1	1	0	1	1	Jump Address (LSB)
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer	0	1	0	1	Return Address (LSB)
			4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
			5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM		4	1	Op Code Address+1	1	0	1	1	Immediate Data
			2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
			3	Address of Operand	1	0	1	1	Operand Data
			4	Op Code Address+3	1	0	1	0	Next Op Code
AIM	EIM	6	1	Op Code Address+1	1	0	1	1	Immediate Data
OIM			2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
			3	Address of Operand	1	0	1	1	Operand Data
			4	FFFF	1	1	1	1	Restart Address (LSB)
			5	Address of Operand	0	1	0	1	New Operand Data
			6	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
INDEXED								
JMP	3	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Accumulator Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data (MSB)
		4	IX+Offset+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Register Data (MSB)
		4	IX+Offset+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	IX+Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX+Offset	0	1	0	1	New Operand Data
		6	Op Code Address+2	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	IX+Offset	0	1	0	1	00
		5	Op Code Address+2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX+Offset	0	1	0	1	New Operand Data
		7	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
EXTEND								
JMP	3	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	0	1	0	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Accumulator Data
		4	Op Code Address+1	1	0	1	0	Next Op Code
PULX	4	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	0	1	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Index Register (LSB)
		4	Stack Pointer−1	0	1	0	1	Index Register (MSB)
		5	Op Code Address+1	1	0	1	0	Next Op Code
RTS	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Return Address (MSB)
		4	Stack Pointer+2	1	0	1	1	Return Address (LSB)
		5	Return Address	1	0	1	0	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	FFFF	1	1	1	1	Restart Address (LSB)
		7	FFFF	1	1	1	1	Restart Address (LSB)

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
WAI		1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
	9	5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
RTI		1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Conditional Code Register
		4	Stack Pointer+2	1	0	1	1	Accumulator B
		5	Stack Pointer+3	1	0	1	1	Accumulator A
	10	6	Stack Pointer+4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer+5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer+6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer+7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI		1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
	12	6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP		1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
	4	Sleep						
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address+1	1	0	1	0	Next Op Code
RELATIVE								
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	0	1	1	Branch Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Branch Address ····Test="1" Op Code Address+1 ···Test="0"	1	0	1	0	First Op Code of Branch Routine Next Op Code
BSR		1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
	5	3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Branch Address	1	0	1	0	First Op Code of Subroutine

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 26, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD6303X as possible.

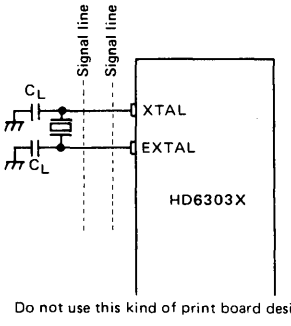


Figure 26 Precaution to the boad design of oscillation circuit

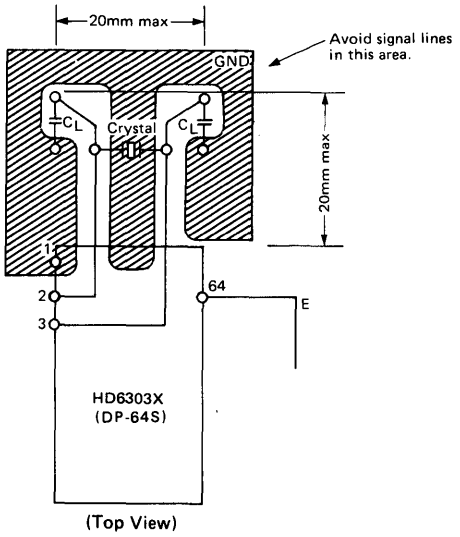


Figure 27 Example of Oscillation Circuits in Board Design

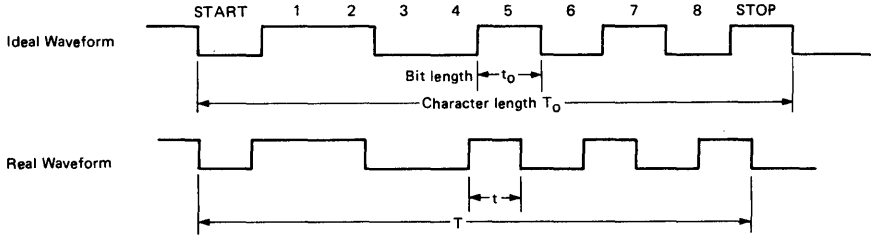
■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD6303X is shown in Table 17.

Note: SCI = Serial Communication Interface

Table 17

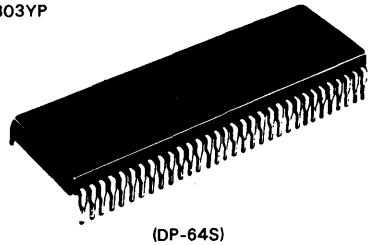
	Bit distortion tolerance (t–to) /to	Character distortion tolerance (T–To) /To
HD6303X	±43.7%	±4.37%



HD6303Y, HD63A03Y, HD63B03Y
CMOS MPU (Micro Processing Unit)

The HD6303Y is a CMOS 8-bit single-chip microprocessing unit which contains a CPU compatible with the CMOS 8-bit microcomputer HD6301V, 256 bytes of RAM, 24 parallel I/O pins, Serial Communication Interface (SCI) and two timers.

HD6303YP, HD63A03YP, HD63B03YP



(DP-64S)

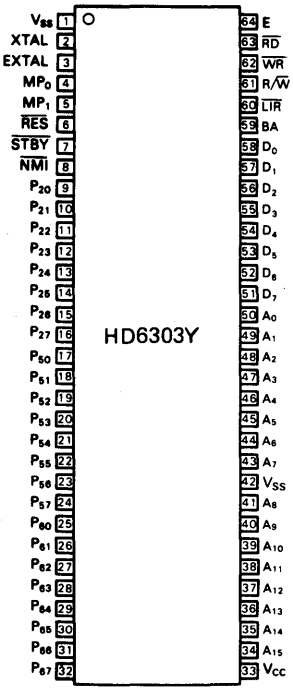
■ **FEATURES**

- Instruction Set Compatible with the HD6301V1
- 256 Bytes of RAM
- 24 Parallel I/O Pins
- Parallel Handshake Interface (Port 6)
- Darlington Transistor Drive (Port 2, 6)
- 16-Bit Programmable Timer
  - Input Capture Register × 1
  - Free Running Counter × 1
  - Output Compare Register × 2
- 8-Bit Reloadable Timer
  - External Event Counter
  - Square Wave Generation
- Serial Communication Interface (SCI)
  - Asynchronous Mode (8 Transmit Formats, Hardware Parity)
  - Clocked Synchronous Mode
- Memory Ready
  - 3 Kinds of Memory Ready
- Halt
- Error Detection
  - (Address Error, Op-code Error)
- Interrupt – External 3, Internal 7
- Maximum 65k Bytes Address Space
- Low Power Dissipation Mode
  - Sleep Mode
  - Standby Mode (Hardware Standby, Software Standby)
- Minimum Instruction Execution Time – 0.5μs (f = 2MHz)
- Wide Range of Operation

  V_{CC}=3 to 5.5V (f=0.1 to 0.5MHz)

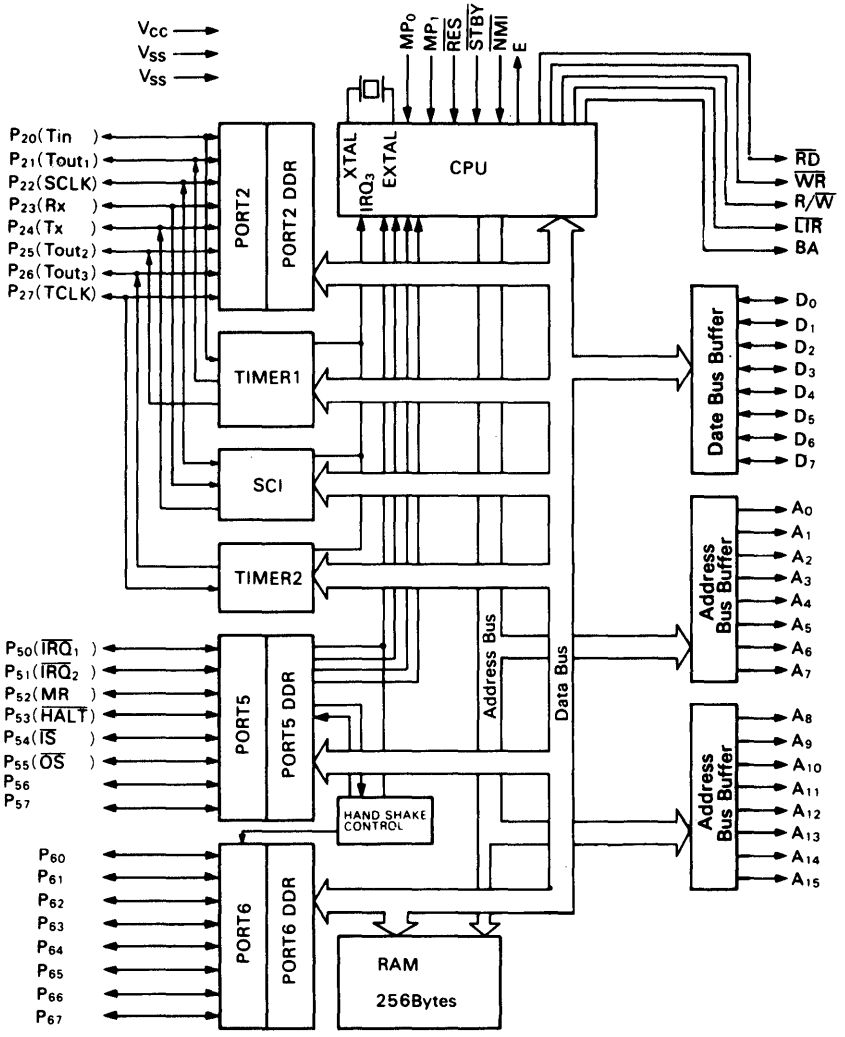
  V_{CC}=5V±10% { f=0.1 to 1.0MHz : HD6303Y; f=0.1 to 1.5MHz : HD63A03Y; f=0.1 to 2.0MHz : HD63B03Y }

■ **PIN ARRANGEMENT**



(Top View)

■ **BLOCK DIAGRAM**



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3~+7.0	V
Input Voltage	V_{in}	−0.3~V_{CC}+0.3	V
Operating Temperature	T_{opr}	0~+70	°C
Storage Temperature	T_{stg}	−55~+150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field.
But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out}: $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}−0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}×0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		−0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$, MP_0, MP_1	$\|I_{in}\|$	V_{in} = 0.5~V_{CC}−0.5V	–	–	1.0	μA
Three State Leakage Current	A_0~A_{15}, D_0~D_7, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, Ports 2, 5, 6	$\|I_{TSI}\|$	V_{in} = 0.5~V_{CC}−0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = −200μA	2.4	–	–	V
			I_{OH} = −10μA	V_{CC}−0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.4	V
Darlington Drive Current	Ports 2, 6	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Input Capacitance	All Inputs	C_{in}	V_{in} = 0V, f = 1MHz, Ta = 25°C	–	–	12.5	pF
Standby Current	Non Operation	I_{STB}		–	3.0	15.0	μA
Current Dissipation*		I_{SLP}	Sleeping (f=1MHz**)	–	1.5	3.0	mA
			Sleeping (f=1.5MHz**)	–	2.3	4.5	mA
			Sleeping (f=2MHz**)	–	3.0	6.0	mA
		I_{CC}	Operating (f=1MHz**)	–	7.0	10.0	mA
			Operating (f=1.5MHz**)	–	10.5	15.0	mA
			Operating (f=2MHz**)	–	14.0	20.0	mA
RAM Standby Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC} − 1.0V, V_{IL} max = 0.8V (All output terminals are at no load.)

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at X MHz operation are decided according to the following formula:

typ. value (f = X MHz) = typ. value (f = 1MHz) × X

max. value (f = X MHz) = max. value (f = 1MHz) × X

(both the sleeping and operating)

• AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, T_a = 0~+70°C, unless otherwise noted.)

BUS TIMING

Item		Symbol	Test Condition	HD6303Y			HD63A03Y			HD63B03Y			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time		t_{Er}		–	–	25	–	–	25	–	–	25	ns
Enable Fall Time		t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*		t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time	Write	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Set-up Time	Read	t_{DSR}		80	–	–	70	–	–	60	–	–	ns
Address, R/$\overline{W}$ Hold Time*		t_{AH}		80	–	–	50	–	–	40	–	–	ns
Data Hold Time	Write*	t_{HW}		70	–	–	50	–	–	40	–	–	ns
	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
$\overline{RD}$, $\overline{WR}$ Pulse Width*		PW_{RW}		450	–	–	300	–	–	220	–	–	ns
$\overline{RD}$, $\overline{WR}$ Delay Time		t_{RWD}		–	–	40	–	–	40	–	–	40	ns
$\overline{RD}$, $\overline{WR}$ Hold Time		t_{HRW}		–	–	20	–	–	20	–	–	20	ns
$\overline{LIR}$ Delay Time		t_{DLR}		–	–	200	–	–	160	–	–	120	ns
$\overline{LIR}$ Hold Time		t_{HLR}		10	–	–	10	–	–	10	–	–	ns
MR Set-up Time*		t_{SMR}	Fig. 2	400	–	–	280	–	–	230	–	–	ns
MR Hold Time*		t_{HMR}		–	–	100	–	–	70	–	–	50	ns
E Clock Pulse Width at MR		PW_{EMR}		–	–	9	–	–	9	–	–	9	μs
Processor Control Set-up Time		t_{PCS}	Fig. 3, 13, 14	200	–	–	200	–	–	200	–	–	ns
Processor Control Rise Time		t_{PCr}	Fig. 2, 3	–	–	100	–	–	100	–	–	100	ns
Processor Control Fall Time		t_{PCf}		–	–	100	–	–	100	–	–	100	ns
BA Delay Time		t_{BA}	Fig. 3	–	–	250	–	–	190	–	–	160	ns
Oscillator Stabilization Time		t_{RC}	Fig. 14	20	–	–	20	–	–	20	–	–	ms
Reset Pulse Width		PW_{RST}		3	–	–	3	–	–	3	–	–	t_{cyc}

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

Peripheral Port Timing

Item		Symbol	Test Condition	HD6303Y			HD63A03Y			HD63B03Y			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set Up Time	Port 2, 5, 6	t_{PDSU}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Port 2, 5, 6	t_{PDH}		200	–	–	200	–	–	200	–	–	ns
Delay Time (From Enable Fall Edge to Peripheral Output)	Port 2, 5, 6	t_{PWD}	Fig. 6	–	–	300	–	–	300	–	–	300	ns
Input Strobe Pulse Width		t_{PWIS}	Fig. 10	200	–	–	200	–	–	200	–	–	ns
Input Data Hold Time	Port 6	t_{IH}		150	–	–	150	–	–	150	–	–	ns
Input Data Set-Up Time	Port 6	t_{IS}		100	–	–	100	–	–	100	–	–	ns
Output Strobe Delay Time		t_{OSD1}	Fig. 11	–	–	200	–	–	200	–	–	200	ns
		t_{OSD2}											

TIMER, SCI TIMING

Item		Symbol	Test Condition	HD6303Y			HD63A03Y			HD63B03Y			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 9	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Output)		t_{TOD}	Fig. 7, 8	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Async. Mode	t_{Scyc}	Fig. 9	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Clock Sync.		Fig. 4	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Clock Sync. Mode)		t_{TXD}	Fig. 4	–	–	220	–	–	220	–	–	220	ns
SCI Receive Data Set-up Time (Clock Sync. Mode)		t_{SRX}		260	–	–	260	–	–	260	–	–	ns
SCI Receive Data Hold Time (Clock Sync. Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 9	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}		200	–	–	200	–	–	200	–	–	ns
Timer 1·2, SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1·2, SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

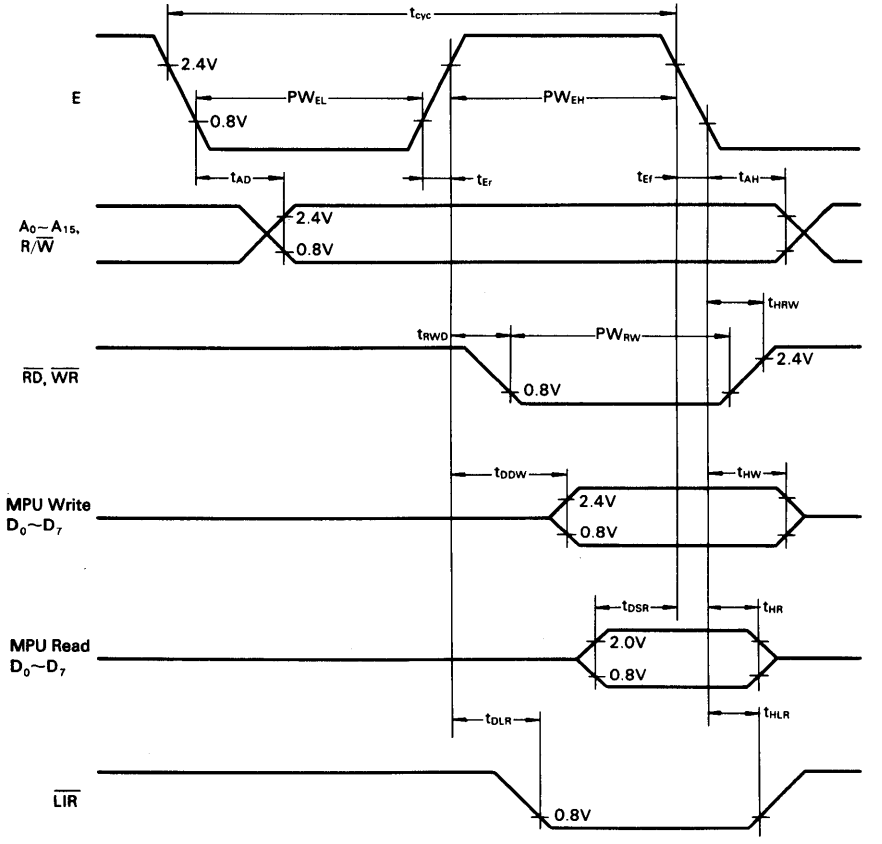


Figure 1 Bus Timing

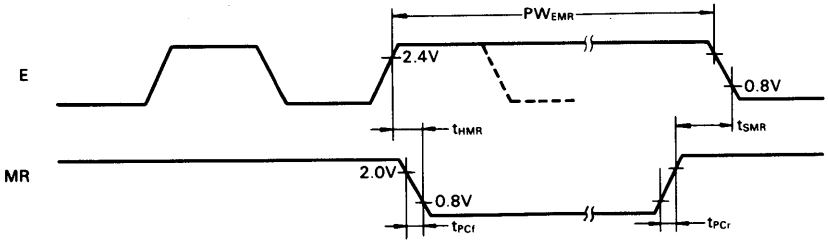


Figure 2 Memory Ready and E Clock Timing

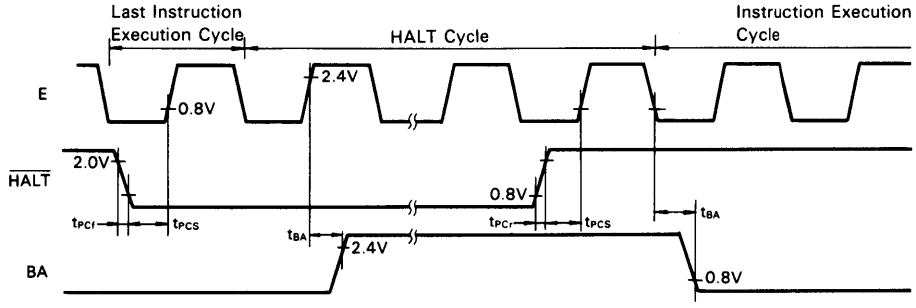


Figure 3 $\overline{\text{HALT}}$ and BA Timing

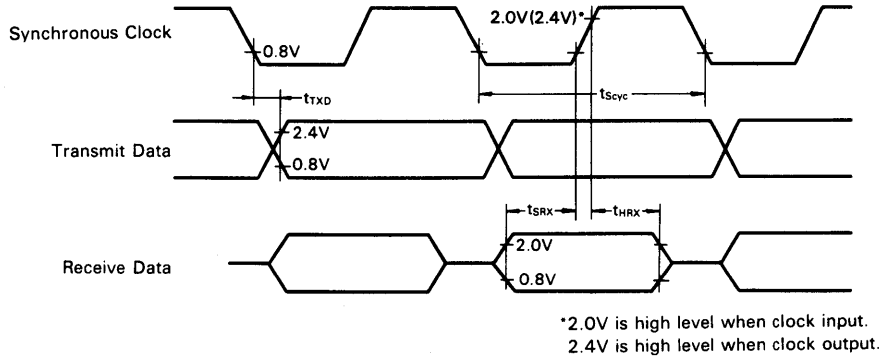


Figure 4 SCI Clocked Synchronous Timing

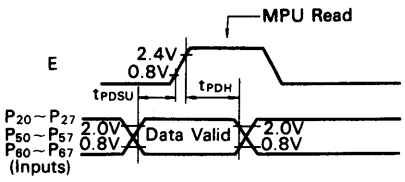


Figure 5 Port Data Set-up and Hold Times (MPU Read)

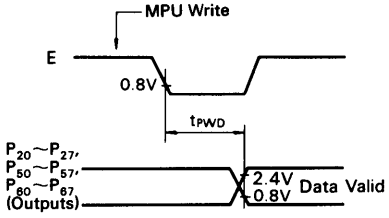


Figure 6 Port Data Delay Times (MPU Write)

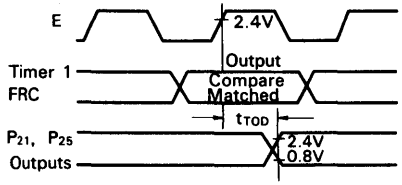


Figure 7 Timer 1 Output Timing

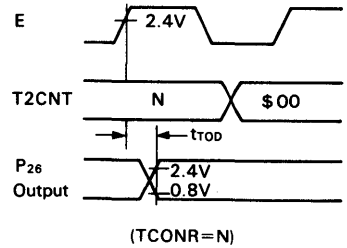


Figure 8 Timer 2 Output Timing

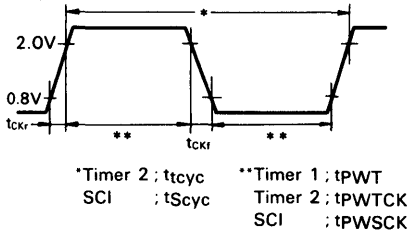


Figure 9 Timer 1·2, SCI Input Clock Timing

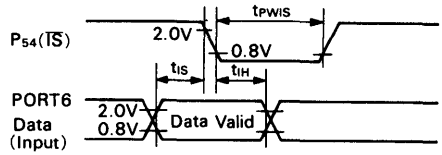


Figure 10 Port 6 Input Latch Timing

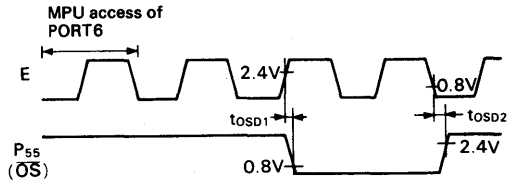


Figure 11 Output Strobe Timing

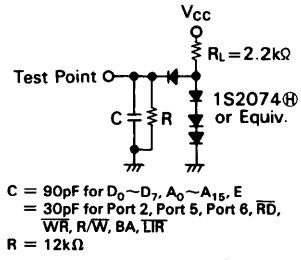


Figure 12 Bus Timing Test Loads (TTL Load)

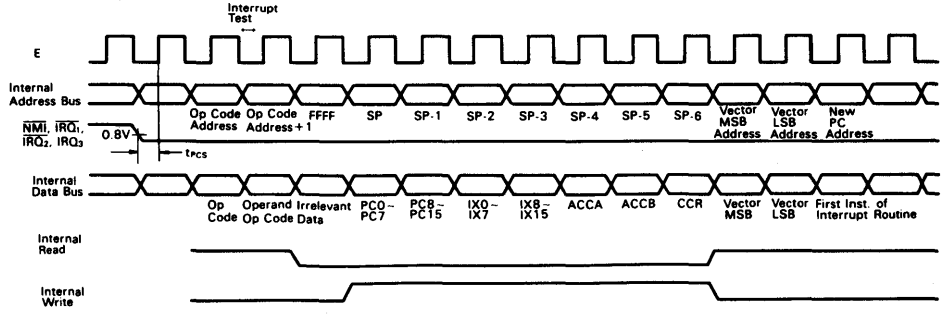


Figure 13 Interrupt Sequence

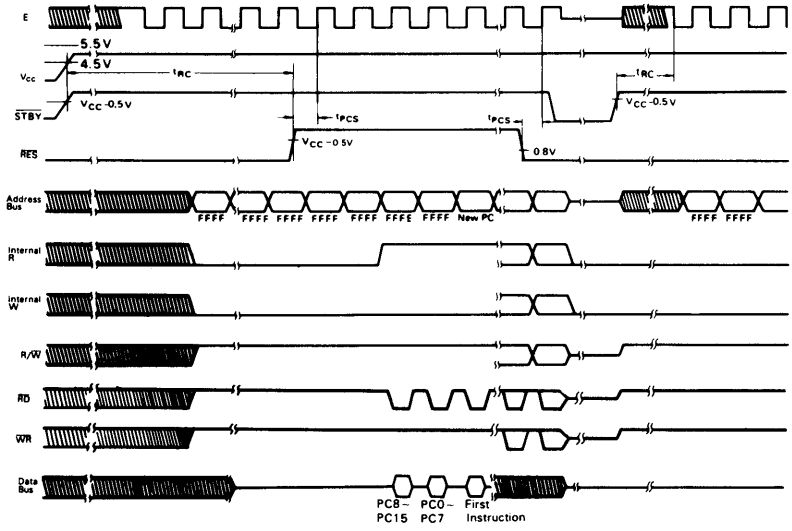


Figure 14 Reset Timing

■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

V_{CC} and V_{SS} provide power to the MPU with 5V±10% supply. In the case of low speed operation (fmax=500kHz), the MPU can operate with 3 to 5.5 volts. Two V_{SS} pins should be tied to ground.

• XTAL, EXTAL

These two pins interface with an AT-cut parallel resonant crystal. Divide-by-four circuit is on chip, so if 4MHz crystal oscillator is used, the system clock is 1MHz for example.

EXTAL pin can be drived by the external clock with 45% to 55% duty. The system clock which is one fourth frequency of the external clock is generated in the LSI. The external clock frequency should be less than four times of the maximum operating frequency. When using the external clock, XTAL pin should be open. Fig. 15 shows examples of connection circuit. The crystal and C_{L1}, C_{L2} should be mounted as close as possible to XTAL and EXTAL pins. Any line must not cross the line between the crystal oscillator and XTAL, EXTAL.

AT Cut Parallel Resonant Crystal Oscillator

C_O=7pF max
R_S=60Ω max

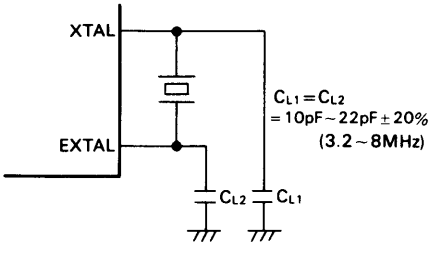


Figure 15 Connection Circuit

• STBY

This pin makes the MPU standby mode. In "Low" level, the oscillation stops and the internal clock is stabilized to make reset condition. To retain the contents of RAM at standby mode, "0" should be written into RAM enable bit (RAME). RAME is the bit 6 of the RAM/port 5 control register at $0014. RAM is disabled by this operation and its contents is sustained.

Refer to "LOW POWER DISSIPATION MODE" for the standby mode.

• Reset (RES)

This pin resets the MPU from power OFF state and provides a startup procedure. During power-on, RES pin must be held "Low" level for at least 20ms.

The CPU registers (accumulator, index register, stack pointer, condition code register except for interrupt mask bit), RAM and the data register of ports are not initialized during reset, so their contents are undefined in this procedure.

To reset the MPU during operation, RES should be held "Low" for at least 3 system-clock cycles. At the 3rd cycle during "Low" level, all the address buses become "High". When RES remains "Low", the address buses keep "High". If RES becomes "High", the MPU starts the next operation.

(1) Latch the value of the mode program pins; MP_0 and MP_1.
(2) Initialize each internal register (Refer to Table 4).
(3) Set the interrupt mask bit. For the CPU to recognize the maskable interrupts $\overline{IRQ_1}$, $\overline{IRQ_2}$ and IRQ_3, this bit should be cleared in advance.
(4) Put the contents (=start address) of the last two addresses ($FFFE, $FFFF) into the program counter and start the program from this address. (Refer to Table 1).

• Enable (E)

This pin provides a TTL-compatible system clock to external circuits. Its frequency is one fourth that of the crystal oscillator or external clock. This pin can drive one TTL load and 90pF capacitance.

• Non-Maskable Interrupt (NMI)

When the falling edge of the input signal is detected at this pin, the CPU begins non-maskable interrupt sequence internally. As

well as the $\overline{IRQ}$ mentioned below, the instruction being executed at $\overline{NMI}$ signal detection will proceed to its compeletion. The interrupt mask bit of the condition code register doesn't affect non-maskable interrupt at all.

In response to an $\overline{NMI}$ interrupt, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack. Upon completion of this sequence, a vector is fetched from $FFFC and $FFFD to transfer their contents into the program counter and branch to the non-maskable interrupt service routine.

(Note) At reset start, the stack pointer should be initialized on an appropriate memory area and then the falling edge be input to $\overline{NMI}$ pin.

- **Interrupt Request ($\overline{IRQ_1}$, $\overline{IRQ_2}$)**

These are level-sensitive pins which request an internal interrupt sequence to the CPU. At interrupt request, the CPU will complete the current instruction before the acceptance of the request. Unless the interrupt mask in the condition code register is set, the CPU starts an interrupt sequence; if set, the interrupt request will be ignored. When the sequence starts, the contents of the program counter, index register, accumulators and condition code register will be saved onto the stack, then the CPU sets the interrupt mask bit and will not acknowledge the maskable request. During the last cycle, the CPU fetches vectors depicted in Table 1 and transfers their contents to the program counter and branches to the service routine.

The CPU uses the external interrupt pins ($\overline{IRQ_1}$ and $\overline{IRQ_2}$) also as port pins P_{50} and P_{51}, so it provides an enable bit to Bit 0 and 1 of the RAM port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for the details.

When one of the internal interrupts, ICI, OCI, TOI, CMI or SIO is generated, the CPU produces internal interrupt signal (IRQ_3). IRQ_3 functions just the same as $\overline{IRQ_1}$ or $\overline{IRQ_2}$ except for its vector address. Fig. 16 shows the block diagram of the interrupt circuit.

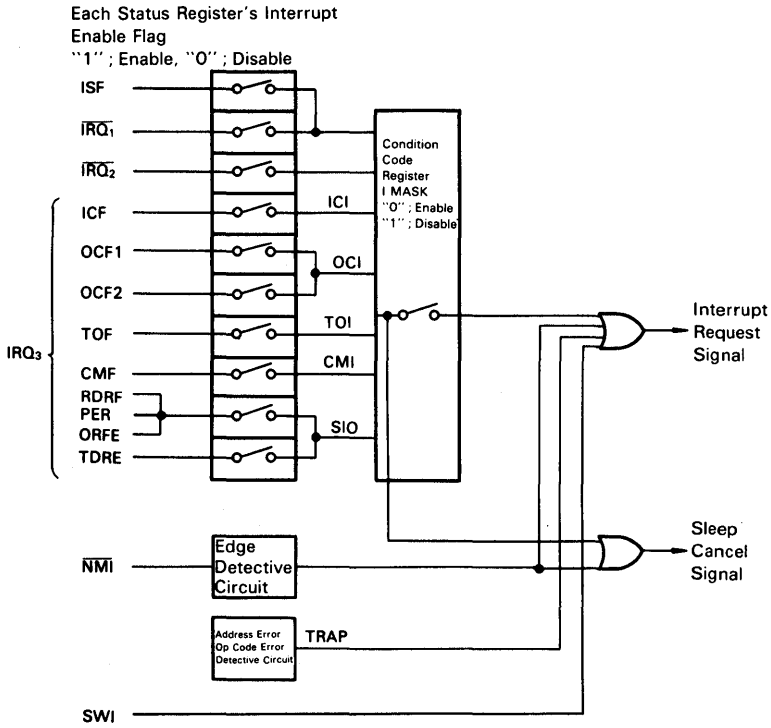


Figure 16 Interrupt Circuit Block Diagram

Table 1 Interrupt Vector Memory Map

Priority	Vector		Interrupt
	MSB	LSB	
Highest	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	$\overline{IRQ_1}$, ISF (port 6 Input Strobe)
	FFF6	FFF7	ICI (Timer 1 Input Capture)
	FFF4	FFF5	OCI (Timer 1 Output Compare 1, 2)
	FFF2	FFF3	TOI (Timer 1 Overflow)
	FFEC	FFED	CMI (Timer 2 Counter Match)
	FFEA	FFEB	$\overline{IRQ_2}$
Lowest	FFF0	FFF1	SIO (RDRF+ORFE+TDRE+PER)

- **Mode Program (MP_0, MP_1)**
  Set MP_0 "High" and MP_1 "Low".

- **Read/Write (R/$\overline{W}$)**
  This signal, usually be in read state ("High"), shows whether the CPU is in read ("High") or write ("Low") state to the peripheral or memory devices. This can drive one TTL load and 30pF capacitance.

- **$\overline{RD}$, $\overline{WR}$**
  These signals show active low outputs when the CPU is reading/writing to the peripherals or memories. This enables the CPU easy to access the peripheral LSI with $\overline{RD}$ and $\overline{WR}$ input pins. These pins can drive one TTL load and 30pF capacitance.

- **Load Instruction Register ($\overline{LIR}$)**
  This signal shows the instruction opecode being on data bus (active low). This pin can drive one TTL load and 30pF capacitance.

- **Memory Ready (MR; P_{52})**
  This is the input control signal which stretches the system clock's "High" period to access low-speed memories. HD6303Y can select three kinds of low-speed memory access method by RAM/Port 5 Control Register's MRE bit and AMRE bit. In the case that CPU accesses low-speed memories by the external MR signal (MRE="1", AMRE="0"), the system clock operates in normal sequence when this signal is in "High".
  But this signal in "Low", the "High" period of the system clock will be stretched depending on its "Low" level duration in integral multiples of the cycle time. This allows the CPU to interface with low-speed memories (See Fig. 2). Up to 9μs can be stretched.
  During internal address space access or nonvalid memory access, MR is prohibited internally to prevent decrease of operation speed. Even in the halt state, MR can also stretch "High" period of system clock to allow peripheral devices to access low-speed memories. Refer to "RAM/PORT 5 CONTROL REGISTER" for more details.

- **Halt ($\overline{HALT}$; P_{53})**
  This is an input control signal to stop instruction execution and to release buses. When this signal switches to "Low", the CPU stops to enter into the halt state after having executed the present instruction. When entering into the halt state, it makes BA "High" and also an address bus, data bus, $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ high impedance. When an interrupt is generated in the halt state, the CPU uses the interrupt handler after the halt is cancelled. When halted during the sleep state, the CPU keeps the sleep state, while BA is "High" and releases the buses. Then the CPU returns to the previous sleep state when the $\overline{HALT}$ signal becomes "High".
  (Note) Please don't switch the $\overline{HALT}$ signal to "Low" when the CPU executes the WAI instruction and is in the interrupt wait state to avoid the trouble of the CPU's operation after the halt is cancelled.

- **Bus Available (BA)**
  This is an output control signal which is normally "Low" but "High" when the CPU accepts $\overline{HALT}$ and releases the buses. The HD6800 and HD6802 make BA "High" and release the buses at WAI execution, while the HD6303Y doesn't make BA "High" under the same condition.

■ **PORT**

The HD6303Y provides three 8-bit I/O ports. Each port provides Data Direction Register (DDR) which controls the I/O state by the bit.

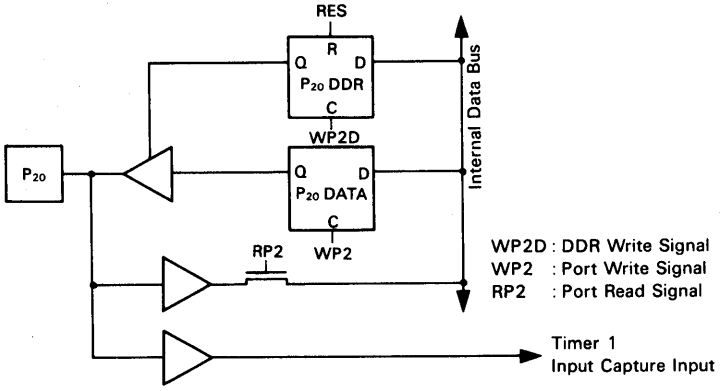
Table 2 Port and Data Direction Register Address

Port	Port Address	Data Direction Register
Port 2	$0003	$0001
Port 5	$0015	$0020
Port 6	$0017	$0016

- **Port 2**
  An 8-bit I/O port. Port 2 DDR (P2DDR) controls the I/O state. This port provides DDR corresponding to each bit and can define input or output by the bit ("0" for input, "1" for output).
  As Port 2 DDR is cleared during reset, it will be an input port.
  Port 2 is also used as an I/O pin for timer 1, Timer 2 and the SCI. Pins for Timers and the SCI set or reset each DDR depending on their functions and become I/O pins. When port 2 functions as an I/O port after used as I/O pins of the timers or the SCI, the I/O direction of the pins remain as it is used as the I/O pin of timer and SCI.
  Port 2 can drive one TTL load and 30pF capacitance. This port can produce 1mA when V_{out}=1.5V to drive directly the base of Darlington transistor.

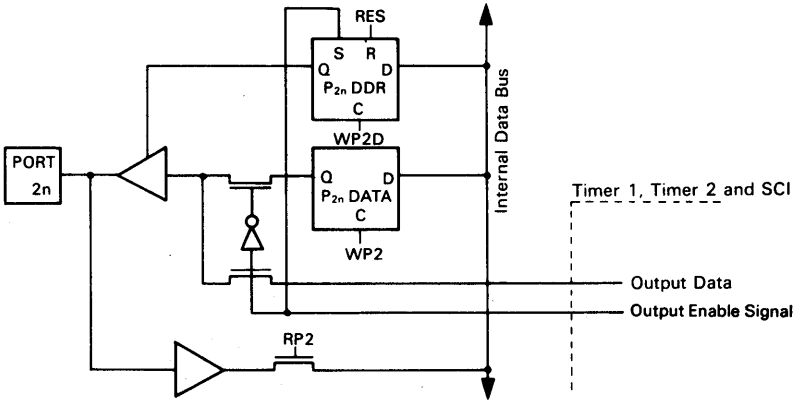
P_{20} (Tin)

P_{20} is also used as an external input pin for the input-capture. This pin is an I/O port which is an input or output as defined by the Data Direction Register (P_{20}DDR) ("0" for an input and "1" for an output). Then either a signal to or from P_{20} ("to" for an output port, "from" for an input port) is always input to the Timer 1 input capture.



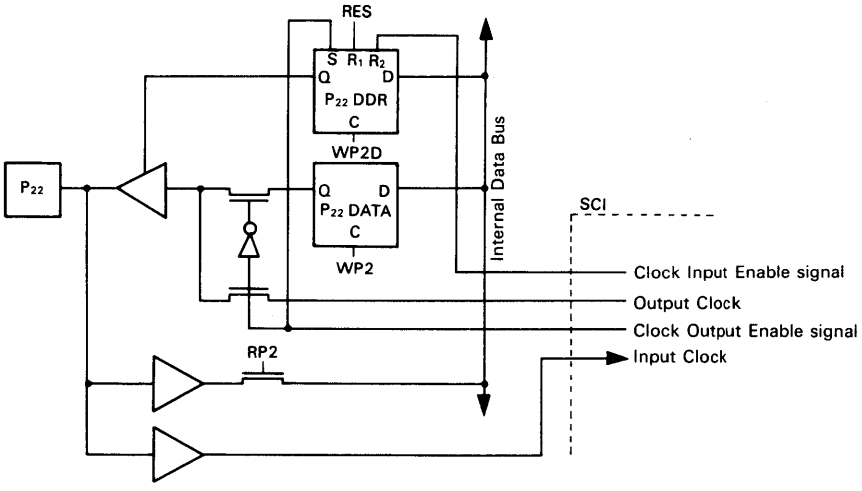
P_{21} (Tout 1), P_{24} (Tx), P_{25} (Tout 2), P_{26} (Tout 3)

These four pins can be also used as output pins for Timer 1, Timer 2 and a transmit output of the SCI. Timer 1, and the SCI have a register which enables output. By setting these registers, they automatically will be output pins of timer or the SCI.



P_{22} (SCLK)

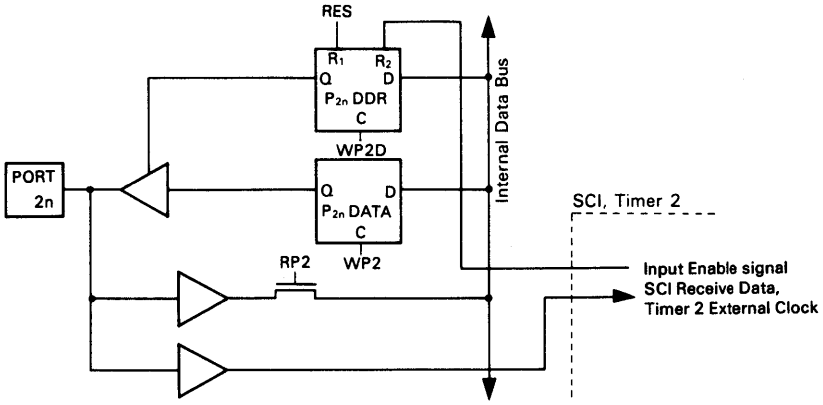
P_{22} is also used as a clock I/O pin for the SCI. It is selected as a clock input or output pin by the operating mode of the SCI. It is usable as an I/O port when the SCI has no clock input or output (as an output port if P_{22} DDR=1, as an input port if P_{22} DDR=0).



P_{23} (Rx), P_{27} (TCLK)

P_{23} and P_{27} are also used as received data input pins for the SCI and external clock input pins for Timer 2. The SCI and Timer 2 have registers which enable input. If the registers are set, the DDR (P_{23} DDR, P_{27} DDR) are cleared and P_{23} and P_{27} will be input pins for Rx and TCLK.

Since the SCI will be a clocked synchronous mode by an external clock-input during reset, the DDR of P_{22} is cleared automatically and P_{22} is an input port. Set the SCI to a mode where P_{22} is not used (CC0 or CC1 of the RMC Register is "0" or "1" respectively) and write "1" to the P_{22} DDR to make P_{22} an output port.



MSB LSB

P_{27} DDR	P_{26} DDR	P_{25} DDR	P_{24} DDR	P_{23} DDR	P_{22} DDR	P_{21} DDR	P_{20} DDR

PORT2 DDR ($0001) (Write only, $00 during reset.)

P_{27}	P_{26}	P_{25}	P_{24}	P_{23}	P_{22}	P_{21}	P_{20}

PORT2 ($0003) (R/W, not initialized during reset.)

• Port 5

An 8-bit I/O port. The DDR of port 5 controls I/O state. Each bit of port 5 has a DDR which defines I/O state ("0" for input and "1" for output).

During reset, the DDR of port 5 is cleared and port 5 becomes an input port.

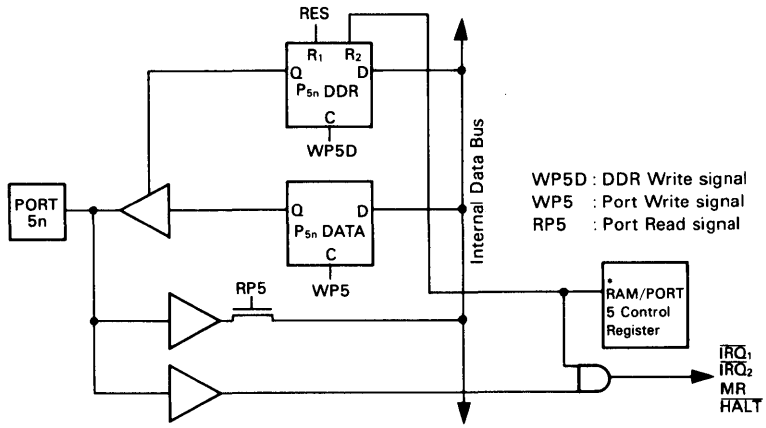
Port 5 is also usable as $\overline{IRQ_1}$, $\overline{IRQ_2}$, $\overline{HALT}$, MR and the strobed signal of port 6 for handshake ($\overline{IS}$, $\overline{OS}$). It is set to input or output automatically if it is used as these control signal pins (except P_{54}, $\overline{IS}$). Since the DDR of port 5, as is port 2, is set or reset by the control signal, I/O directions of the I/O ports are retained after the control signal is disabled. Port 5 can drive one TTL load and 90pF capacitance.

P_{50} ($\overline{IRQ_1}$), P_{51} ($\overline{IRQ_2}$)

P_{50} and P_{51} are also usable as interrupt pins. The RAM/port 5 control registers of $\overline{IRQ_1}$ and $\overline{IRQ_2}$ have enable bits (IQ1E, IQ2E). When these bits are set to "1", P_{50} and P_{51} will automatically be interrupt input pins.

P_{52} (MR), P_{53} ($\overline{HALT}$)

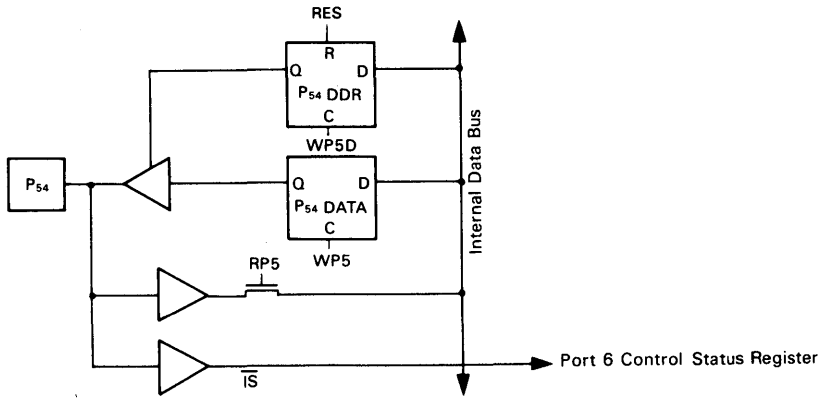
P_{52} and P_{53} are also usable as MR and $\overline{HALT}$ inputs. MR and $\overline{HALT}$ have enable bits (MRE, HLTE) in the RAM/Port 5 Control Register as $\overline{IRQ_1}$ and $\overline{IRQ_2}$. Since MRE is cleared during reset, P_{52} is usable as an I/O port, and HLTE is set during reset, the DDR of P_{53} will be automatically reset to be a $\overline{HALT}$ input pin. HLTE of the RAM/Port 5 Control Register has to be cleared to use P_{53} as an I/O port.



* Initializing value during reset;
IRQ1E = "0", IRQ2E = "0", MRE = "0", HLTE = "1"

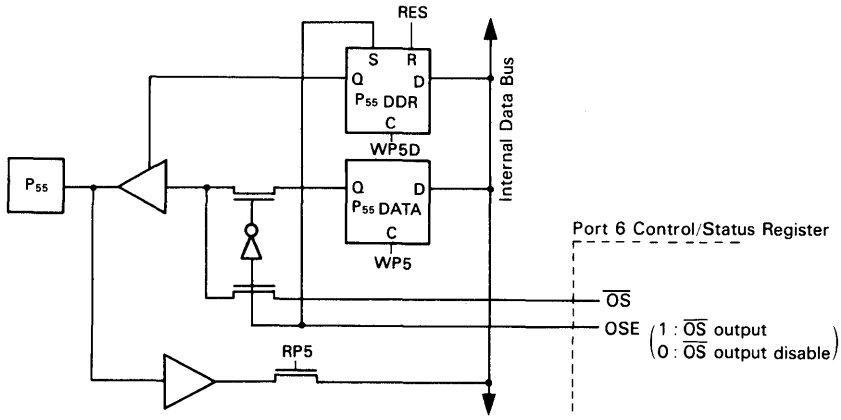
P_{54} ($\overline{IS}$)

P_{54} is also usable as the input strobe ($\overline{IS}$) for port 6 handshake interface. This pin, as is P_{20}, is always an I/O port. If P_{54} is used as an output port (set the DDR of P_{54} to "1"), an output signal from P_{54} will be the input to $\overline{IS}$.



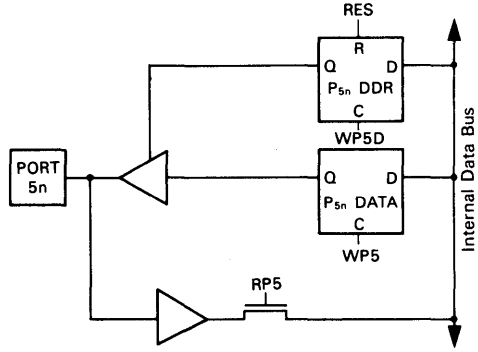
P_{55} ($\overline{OS}$)

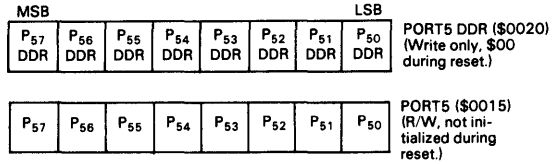
P_{55} is also usable as the output strobe ($\overline{OS}$) for port 6 handshake interface. It will be an I/O port during reset, and an $\overline{OS}$ output pin by setting the $\overline{OS}$ enable register (OSE) of the port 6 Control Status Register (P6CSR).



P_{56}, P_{57}

P_{56} and P_{57} are I/O ports.



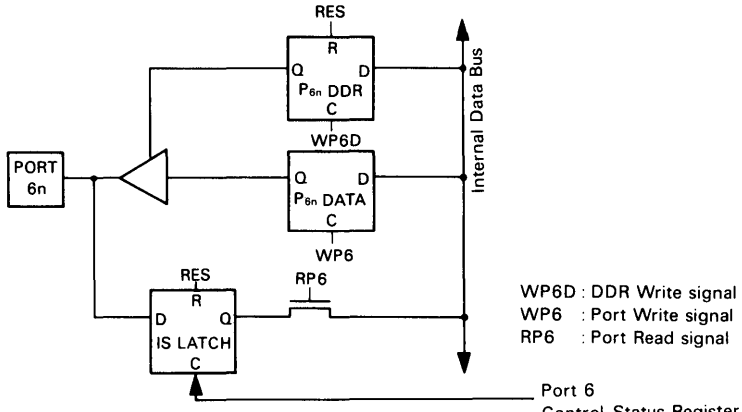


- **Port 6**

8-bit I/O port. Port 6 DDR controls I/O state. Each bit of port 6 has a DDR and designates input or output ("0" for input, "1" for output). During reset, Port 6 DDR is cleared and port 6 becomes an input port.

Port 6 controls parallel handshake interface besides functions as an I/O port. Therefore, it provides DDRs to control and IS LATCH to latch the input data.

Port 6 can drive one TTL load and 30pF capacitance. It can drive directly the base of Darlington transistor as port 2.



MSB LSB

P_{67} DDR	P_{66} DDR	P_{65} DDR	P_{64} DDR	P_{63} DDR	P_{62} DDR	P_{61} DDR	P_{60} DDR

PORT6 DDR ($0016) (Write only, $00 during reset.)

P_{67}	P_{66}	P_{65}	P_{64}	P_{63}	P_{62}	P_{61}	P_{60}

PORT6 ($0017) (R/W, not initialized during reset.)

■ BUS

- **Address Bus ($A_0 \sim A_{15}$)**

Address Bus ($A_0 \sim A_{15}$) is used for addressing the memory and peripheral LSI.

This bus can interface with the bus of HMCS 6800 and drive one TTL load and 90pF capacitance.

- **Data Bus ($D_0 \sim D_7$)**

8-bit parallel data bus for data transmit between the memory or peripheral LSI. This bus can drive one TTL load and 90pF capacitance.

■ RAM/PORT 5 CONTROL REGISTER

The control register located at $0014 controls on-chip RAM and port 5.

RAM/Port 5 Control Register (RP5CR)

7	6	5	4	3	2	1	0	
STBY PWR	RAME	STBY FLAG	AMR E	HLTE	MRE	IRQ_2 E	IRQ_1 E	$0014

Bit 0, Bit 1 $\overline{IRQ_1}$, $\overline{IRQ_2}$ Enable Bit (IRQ_1E, IRQ_2E)

When using P_{50} and P_{51} as interrupt pins, write "1" in these bits. When the bit is set to "1", the DDRs corresponding to P_{50} and P_{51} are cleared and become $\overline{IRQ_1}$ input pin and $\overline{IRQ_2}$ input pin. When IRQ_1E and IRQ_2E are set, P_{50} and P_{51} cannot be used as an output ports. When "0", the CPU doesn't accept an external interrupt or a sleep cancellation by the external interrupt. These bits are cleared during reset.

Bit 2 Memory Ready Enable Bit (MRE)

When using P_{52} as an input pin of the "memory ready" signal, write "1" in this bit. When set, P_{52} DDR is automatically cleared and becomes the MR input pin. The bit is cleared during reset.

Bit 3 Halt Enable Bit (HLTE)

When using P_{53} as an input pin of the $\overline{HALT}$ signal, write "1" in this bit. When this bit is set, P_{53} DDR is automatically cleared and becomes the Halt input pin. If the bit is "0", the Halt function is inhibited and P_{53} is used as an I/O port. The bit is set to "1" during reset.

Bit 4 Auto Memory Ready Enable Bit (AMRE)

When the bit is set and the CPU accesses the external address, "memory ready" operates automatically and stretches the E clock's "High" duration for one system clock. When MRE bit of bit 2 is cleared and when the CPU accesses the external address space, the function operates. When MRE bit is set and then the CPU accesses the external address space with P_{52}(MR) pin in "low", "memory ready" operates automatically. This bit is set to "1" during reset.

Table 3 "Memory Ready" Function

MRE	AMRE	Function
0	0	"Memory ready" inhibited.
0	1	When the CPU accesses the external address, "High" duration of E clock automatically becomes one-cycle longer. This state is retained during reset.
1	0	"Memory ready" operates by P_{52} (MR) pin. The function is the same as that of the HD6301X0.
1	1	When the CPU accesses the external address space with the P_{52} (MR) pin in "low", the "auto memory ready" operates. This function is effective if it has both "high-speed memory" and "slow memory" outside. Input $\overline{CS}$ signal of "slow memory" to MR pin.

Bit 5 Standby Flag (STBY FLAG)

By clearing this flag, HD6303Y gets into the standby mode by software. This flag is set to "1" during reset, so the standby mode is canceled with $\overline{RES}$ pin in "low". The $\overline{RES}$ pin should be in "low" until oscillation becomes stable (min. 20ms.). If the $\overline{STBY}$ pin in is in "low", the standby mode can not be canceled with the $\overline{RES}$ pin in "low".

Bit 6 RAM Enable (RAME)

On-chip RAM can be disabled by this control bit. By resetting the MPU, "1" is set to this bit, and on-chip RAM is enabled. When this bit is cleared (=logic "0") on-chip RAM is invalid and the CPU can read data from external memory. This bit should be "0" before getting into the standby mode to protect on-chip RAM data.

Bit 7 Standby Power Bit (STBY PWR)

When V_{CC} is not provided in standby mode, this bit is cleared. This is a flag for read/write and can be read by software. If this bit is set before standyby mode, and remains set even after returning from standby mode, V_{CC} voltage is provided during standby mode and the on-chip RAM data is valid.

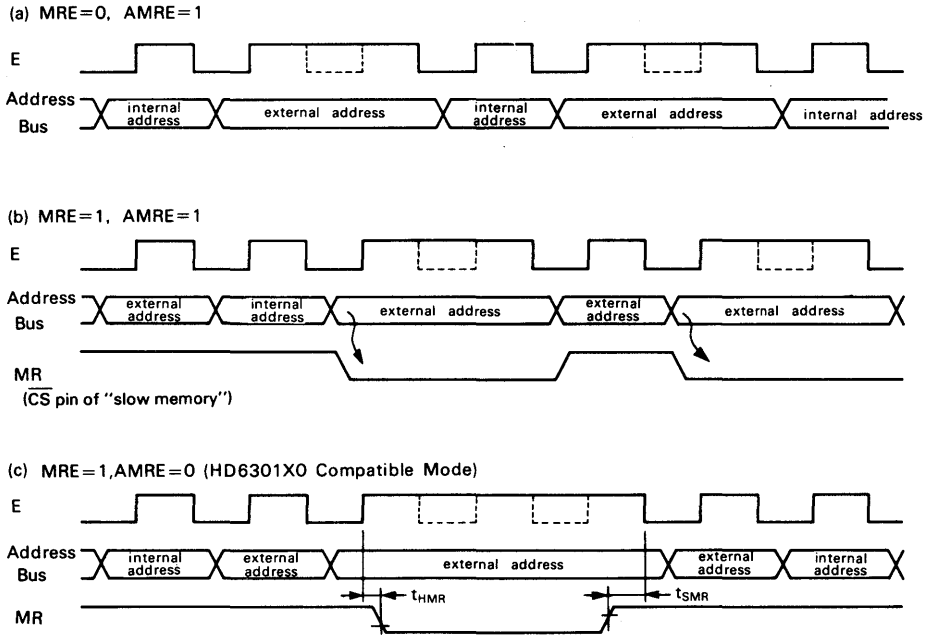


Figure 17 Memory Ready Timing

■ **Port 6 Control/Status Register**

This is the Control/Status Register for parallel handshake interface using Port 6. The functions are as follows;

1) Latches input data to Port 6 at the $\overline{IS}$ (P_{54}) falling edge.
2) Outputs a strobe signal $\overline{OS}$ (P_{55}) outward by reading or writing to port 6.
3) When IS FLAG is set at the $\overline{IS}$ falling edge, an interrupt occurs.

The following shows Port 6 Control/Status Register (P6CSR).

7	6	5	4	3	2	1	0	
IS* FLAG	IS IRQ1 ENABLE	OSE	OSS	LATCH ENABLE	–	–	–	$0021

*Bit 7 is Read-Only bit

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Bit 0
Bit 1 Not used.
Bit 2

Bit 3: Latch Enable
This register controls the input latch for Port 6 (ISLATCH). When this bit is set to "1", the input data to port 6 will be latched inward at the $\overline{IS}$ (P_{54}) falling edge. An input latch will be canceled by reading Port 6, which enables to latch the next data. If cleared, the input latch remains canceled and this bit functions as a usual input port. This bit is cleared during reset.

Bit 4: OSS Output Strobe Select
This register initiates an output strobe ($\overline{OS}$) from P_{55} by reading or writing to port 6. When cleared, $\overline{OS}$ occurs by reading Port 6. When set, $\overline{OS}$ occurs by writing to Port 6. This bit is cleared during reset.

Bit 5: OSE Output Strobe Enable
This register decides the enabling or disabling of the output strobe. When cleared, P_{55} functions as an I/O port. When set, P_{55} functions as an $\overline{OS}$ output pin. (P_{55} DDR is set by OSE.) This bit is cleared during reset.

Bit 6: IS $\overline{IRQ_1}$ Enable Input Strobe Interrupt Enable
When set, an $\overline{IRQ_1}$ interrupt to the CPU occurs by setting IS FLAG of bit 7. When cleared, the interrupt does not occur. This bit is cleared during reset.

Bit 7: IS Flag Input Strobe Flag
This flag is set at the IS (P_{54}) falling edge. This flag is for read-only. When set, the flag is cleared by reading or writing to Port 6 after reading the Port 6 Control Status Register. This bit is cleared during reset.

■ MEMORY MAP

The MPU can address up to 65k bytes. Memory map is shown in Fig. 20. 40 addresses ($0000 ~ $0027 except $00, $02, $04, $05, $06, $07, $18) are the internal registers as shown in Table 4.

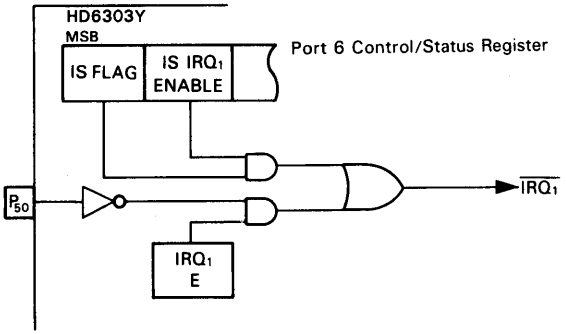


Figure 18 Input Strobe Interrupt block Diagram

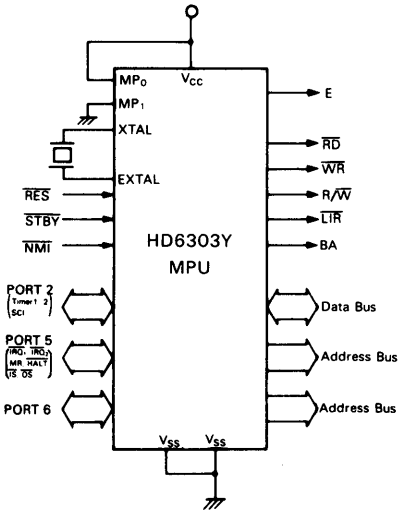


Figure 19 HD6303Y Operating Function

Table 4 Internal Register

Address	Register	Abbreviation	R/W**	Initialized value during reset***
00*	Port 1 DDR (Data Direction Register)	P1DDR	W	$FE
01	Port 2 DDR	P2DDR	W	$00
02*	Port 1	PORT1	R/W	indefinite
03	Port 2	PORT2	R/W	indefinite
04*	Port 3 DDR	P3DDR	W	$FE
05*	Port 4 DDR	P4DDR	W	$00
06*	Port 3	PORT3	R/W	indefinite
07*	Port 4	PORT4	R/W	indefinite
08	Timer Control/Status Register 1	TCSR1	R/W	$00
09	Free Running Counter (MSB)	FRCH	R/W	$00
0A	Free Running Counter (LSB)	FRCL	R/W	$00
0B	Output Compare Register 1 (MSB)	OCR1H	R/W	$FF
0C	Output Compare Register 1 (LSB)	OCR1L	R/W	$FF
0D	Input Capture Register (MSB)	ICRH	R	$00
0E	Input Capture Register (LSB)	ICRL	R	$00
0F	Timer Control/Status Register 2	TCSR2	R/W	$10
10	Rate/Mode Control Register	RMCR	R/W	$C0
11	Tx/Rx Control Status Register 1	TRCSR1	R/W	$20
12	Receive Data Register	RDR	R	$00
13	Transmit Data Register	TDR	W	indefinite
14	RAM/Port 5 Control Register	RP5CR	R/W	$F8 or $78
15	Port 5	PORT5	R/W	indefinite
16	Port 6 DDR	P6DDR	W	$00
17	Port 6	PORT6	R/W	indefinite
18	Port 7	PORT7	R/W	indefinite
19	Output Compare Register 2 (MSB)	OCR2H	R/W	$FF
1A	Output Compare Register 2 (LSB)	OCR2L	R/W	$FF
1B	Timer Control/Status Register 3	TCSR3	R/W	$20
1C	Time Constant Register	TCONR	W	$FF
1D	Timer 2 Up Counter	T2CNT	R/W	$00
1E	Tx/Rx Control Status Register 2	TRCSR2	R/W	$28
1F****	Test Register*	TSTREG	—	—
20	PORT 5 DDR	P5DDR	W	$00
21	PORT 6 Control/Status Register	P6CSR	R/W	$07
22	— Reserved	—	—	—
23	—	—	—	—
24	—	—	—	—
25	—	—	—	—
26	—	—	—	—
27	—	—	—	—

* External address.
** R: Read-only register, W: Write-only register, R/W: Read/Write register.
*** When empty bit is in the register, it is set to "1".
**** Register for test. Don't access this register.

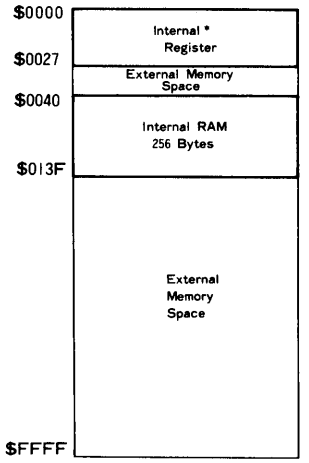


Figure 20 HD6303Y Memory Map

■ TIMER 1

The HD6303Y provides a 16-bit programmable timer which can simultaneously measure an input waveform and generate two independent output waveforms. The pulse widths of both input/output waveforms vary from microseconds to seconds.

Timer 1 is configured as follows (refer to Fig. 22).

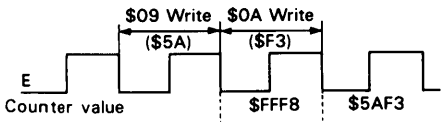
- Control/Status Register 1 (8 bit)
- Control/Status Register 2 (7 bit)
- Free Running Counter (16 bit)
- Output Compare Register 1 (16 bit)
- Output Compare Register 2 (16 bit)
- Input Capture Register (16 bit)

• Free-Running Counter (FRC)($0009:000A)

The key timer element is a 16-bit free-running counter driven and incremented by system clock. The counter value is readable by software without affecting the counter. The counter is cleared during reset.

When writing to the upper byte ($09), the CPU writes the preset value ($FFF8) into the counter (address $09, $0A) regardless of the write data value. But when writing to the lower byte ($0A) after the upper byte writing, the CPU writes not only lower byte data into lower 8 bit, but also upper byte data into higher 8 bit of the FRC.

The counter will be as follows when the CPU writes to it by double store instructions (STD, STX, etc.)



In the case of the CPU write ($5AF3) to the FRC

Figure 21 Counter Write Timing

• Output Compare Register (OCR) ($000B, $000C; OCR1) ($0019, $001A: OCR2)

The output compare register is a 16-bit read/write register which can control an output waveform. The data of OCR is always compared with the FRC.

When the data matches, output compare flag (OCF) in the timer control/status register (TCSR) is set. If an output enable bit (OE) in the TCSR2 is "1", an output level bit(OLVL) in the TCSR will be output to bit 1 (OCR 1) and bit 5 (OCR 2) of port 2. To control the output level again by the next compare, the value of OCR and OLVL should be changed. The OCR is set to $FFFF at reset. The compare function is inhibited for a cycle just after a write to the upper byte of the OCR or FRC. This is to set the 16-bit value valid in the counter register for compare. In addition, it is because counter is to set $FFF8 at the next cycle of the CPU's upper byte write to the FRC.

* For data write to the FRC or the OCR, 2-byte transfer instruction (such as STX, etc.) should be used.

• Input Capture Register (ICR) ($000D : 000E)

The input capture register is a 16-bit read-only register which stores the FRC's value when external input signal transition generates an input capture pulse. Such transition is controled by input edge bit (IEDG) in the TCSR1.

In order to input the external input signal to the edge detector, a bit of the DDR corresponding to bit 0 of port 2 should be cleared ("0"). When an input capture pulse occurs by external input signal transition at the next cycle of CPU's high-byte read of the ICR, the input capture pulse will be delayed by one cycle. In order to ensure the input capture operation, a CPU read of the ICR needs 2-byte transfer instruction. The input pulse width should be at least 2 system cycles. This register is cleared ($0000) during reset.

• Timer Control/Status Register 1 (TCSR1) ($0008)

The timer control/status register 1 is an 8-bit register. All bits are readable and the lower 5 bits are also writable: The upper 3 bits are read-only which indicate the following timer status.

Bit 5 The counter value reached to $0000 as a result of counting-up (TOF).

Bit 6 A match has occurred between the FRC and the OCR 1 (OCF1).

Bit 7 Defined transition of the timer input signal causes the counter to transfer its data to the ICR (ICF).

The followings are the each bit descriptions.

Timer Control/Status Register 1

7	6	5	4	3	2	1	0	
ICF	OCF1	TOF	EICI	EOCI1	ETOI	IEDG	OLVL1	$0008

Bit 0 OLVL1 Output Level 1

OLVL1 is transferred to port 2, bit 1 when a match occurs between the counter and the OCR1. If bit 0 of the TCSR2 (OE1), is set to "1", OLVL1 will appear at bit 1 of port 2.

Bit 1 IEDG Input Edge

This bit determines which edge, rising or falling, of input signal of bit 0 of port 2 will trigger data transfer from the counter to the ICR. For this function, the DDR corresponding to port 2, bit 0 should be cleared beforehand.

IEDG=0, triggered on a falling edge ("High" to "Low")

IEDG=1, triggered on a rising edge ("Low" to "High")

Bit 2 ETOI Enable Timer Overflow Interrupt

When this bit is set, an internal interrupt (IRQ_3) by TOI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 3 EOCI1 Enable Output Compare Interrupt 1

When this bit is set, an internal interrupt (IRQ_3) by OCI1 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 EICI Enable Input Capture Interrupt

When this bit is set, an internal interrupt (IRQ_3) by ICI interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag

This read-only bit is set when the counter increments from $FFFF by 1. Cleared when the counter's MSB byte ($0009) is read by the CPU after the TCSR1 read at TOF=1.

Bit 6 OCF1 Output Compare Flag 1

This read-only bit is set when a match occurs between the OCR1 and the FRC. Cleared when writing to the OCR1 ($000B or $000C) after the TCSR1 or TCSR2 read at OCF=1.

Bit 7 ICF Input Capture Flag

This read-only bit is set when an input signal of port 2, bit 0 makes a transition as defined by IEDG and the FRC is transferred to the ICR. Cleared when reading the upper byte ($000D) of the ICR after the TCSR1 or TCSR2 read at ICF=1.

- **Timer Control/Status Register 2 (TCSR2) ($000F)**

The timer control/status register 2 is a 7-bit register. All bits are readable and the lower 4 bits are also writable. But the upper 3 bits are read-only which indicate the following timer status.

Bit 5 A match has occurred between the FRC and the OCR2 (OCF2).

Bit 6

Timer Control/Status Register 2

7	6	5	4	3	2	1	0	
ICF	OCF1	OCF2	–	EOCI2	OLVL2	OE2	OE1	$000F

Bit 7 The same status flag as the ICF flag of the TCSR1, bit 7.

The followings are the each bit descriptions.

Bit 0 OE1 Output Enable 1

This bit enables the OLVL1 to appear at port 2, bit 1 when a match has occurred between the counter and the output compare register 1. When this bit is cleared, bit 1 of port 2 will be an I/O port. When set, it will be an output of OLVL1 automatically.

Bit 1 OE2 Output Enable 2

This bit enables the OLVL2 to appear at port 2, bit 5 when a match has occurred between the counter and the output compare register 2. When this bit is cleared, port 2, bit 5 will be an I/O port. When set, it will be an output of OLVL2 automatically.

Bit 2 OLVL2 Output Level 2

OLVL2 is transferred to port 2, bit 5 when a match has occurred between the counter and the OCR2. If bit 5 of the TCSR2 (OE2), is set to "1", OLVL2 will appear at port 2, bit 5.

Bit 3 EOCI2 Enable Output Compare Interrupt 2

When this bit is set, an internal interrupt (IRQ_3) by OCI2 interrupt is enabled. When cleared, the interrupt is inhibited.

Bit 4 Not used

Bit 5 OCF2 Output Compare Flag 2

This read-only bit is set when a match has occurred between the counter and the OCR2. Cleared when writing to the OCR2 ($0019 or $001A) after the TCSR2 read at OCF2=1.

Bit 6 OCF1 Output Compare Flag 1

Bit 7 ICF Input Capture Flag

OCF1 and ICF are dual addressed. If which register, TCSR1 or TCSR2, CPU reads, it can read OCF1 and ICF to bit 6 and bit 7.

Both the TCSR1 and TCSR2 will be cleared during reset.

(Note) If OE1 or OE2 is set to "1" before the first output compare match occurs after reset restart, bit 1 or bit 5 of port 2 will produce "0" respectively.

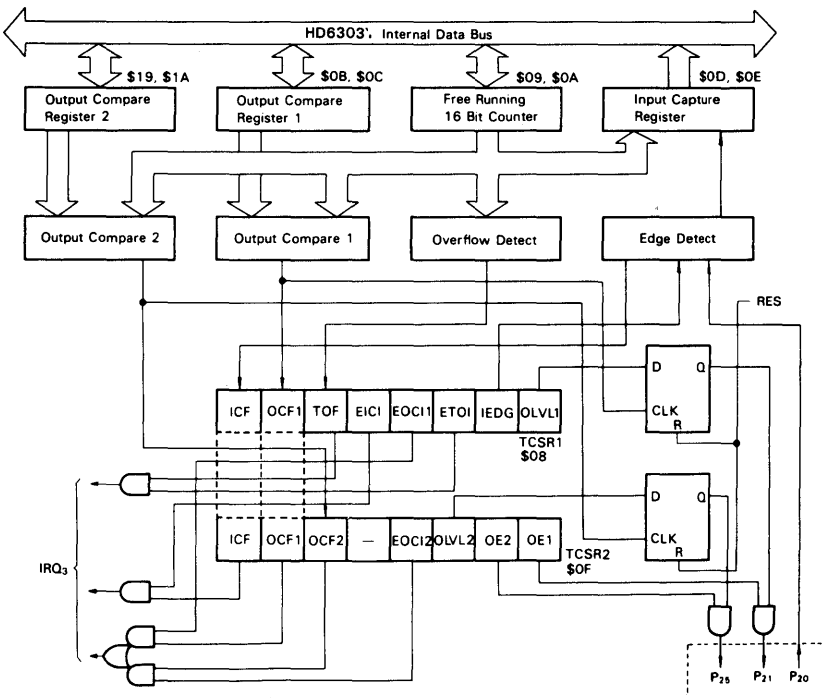


Figure 22 Timer 1 Block Diagram

■ TIMER 2

In addition to the timer 1, the HD6303Y provides an 8-bit reloadable timer, which is capable of counting the external event. The timer 2 contains a timer output, so the MPU can generate three independent waveforms. (Refer to Fig. 23.)

The timer 2 is configured as follows:

- Control/Status Register 3 (7 bits)
- 8-bit Up Counter
- Time Constant Register (8 bits)

• Timer 2 Up Counter (T2CNT) ($001D)

This is an 8-bit up counter which operates with the clock decided by CKS0 and CKS1 of the TCSR3. The CPU can read the value of the counter without affecting the counter. In addition, any value can be written to the counter by software even during counting.

The counter is cleared when a match occurs between the counter and the TCONR or during reset.

If the write operation is made by software to the counter at the cycle of counter clear, it does not reset the counter but put the write data to the counter.

• Time Constant Register (TCONR) ($001C)

The time constant register is an 8-bit write only register. The data of register is always compared with the counter.

When a match has occurred, the counter match flag (CMF) of the timer control status register 3 (TCSR3) is set and the value selected by TOS0 and TOS1 of the TCSR3 will appear at port 2, bit 6. When CMF is set, the counter will be cleared simultaneously and then start counting from $00. This enables regular interrupts and waveform outputs without any software support. The TCONR is set to "$FF" during reset.

• Timer Control/Status Register 3 (TCSR3) ($001B)

The timer control/status register 3 is a 7-bit register. All bits are readable and 6 bits except for CMF can be written.

The followings are each pin descriptions.

Timer Control/Status Register 3

7	6	5	4	3	2	1	0	
CMF	ECMI	–	T2E	TOS1	TOS0	CKS1	CKS0	$001B

Bit 0 CKS0 Input Clock Select 0
Bit 1 CKS1 Input Clock Select 1

Input clock to the counter is selected as shown in Table 5 depending on these two bits. When an external clock is selected, bit 7 of port 2 will be a clock input automatically. Timer 2 detects the rising edge of the external clock and increments the counter. The external clock is countable up to half the frequency of the system clock.

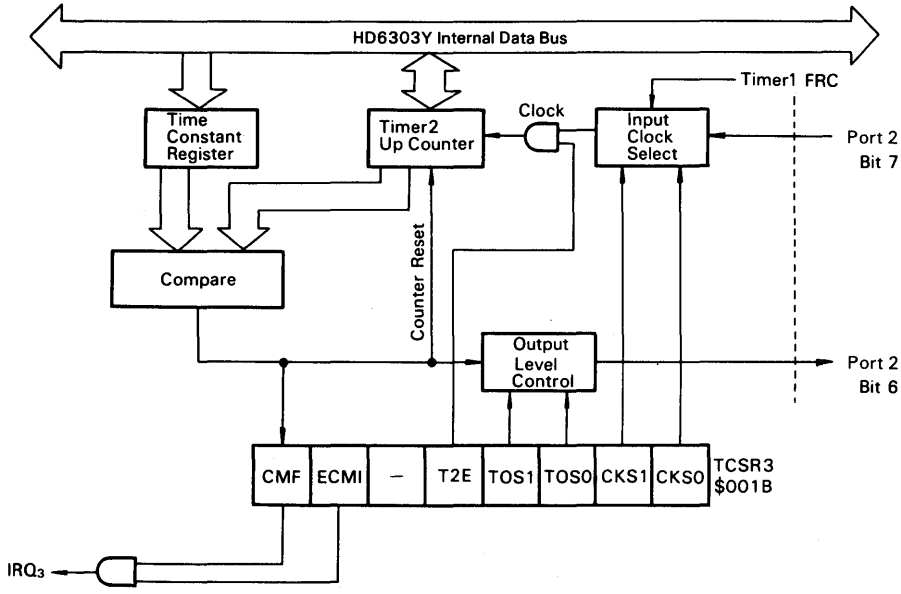


Figure 23 Timer 2 Block Diagram

Table 5 Input Clock Select

CKS1	CKS0	Input Clock to the Counter
0	0	E clock
0	1	E clock/8*
1	0	E clock/128*
1	1	External clock

* These clocks come from the FRC of the timer 1. If one of these clocks is selected as an input clock to the up counter, the CPU should not write to the FRC of the timer 1.

Bit 2 TOS0 Timer Output Select 0
Bit 3 TOS1 Timer Output Select 1

When a match occurs between the counter and the TCONR timer 2 outputs shown in Table 6 will appear at port 2, bit 6 depending on these two bits. When both TOS0 and TOS1 are "0", bit 6 of port 2 will be an I/O port.

Table 6 Timer 2 Output Select

TOS1	TOS0	Timer Output
0	0	Timer Output Inhibited
0	1	Toggle Output*
1	0	Output "0"
1	1	Output "1"

* When a match occurs between the counter and the TCONR, timer 2 output level is reversed. This leads to production of a square wave with 50% duty to the external without any software support.

Bit 4 T2E Timer 2 Enable Bit

When this bit is cleared, a clock input to the up counter is inhibited and the up counter stops. When set to "1", a clock selected by CKS1 and CKS0 (Table 5) is input to the up counter.

(Note) P_{26} outputs "0" when T2E bit cleared and timer 2 set in output enable condition by TOS1 or TOS0. It also outputs "0" when T2E bit set "1" and timer 2 set in output enable condition before the first counter match occurs.

Bit 5 Not Used.

Bit 6 ECMI Enable Counter Match Interrupt

When this bit is set, an internal interrupt (IRQ_3) by CMI is enabled. When cleared, the interrupt is inhibited.

Bit 7 CMF Counter Match Flag

This read-only bit is set when a match occurs between the up counter and the TCONR. Cleared by writing "0" at CMF=1 by software (unable to write "1" by software).

Each bit of the TCSR3 is cleared during reset.

■ SERIAL COMMUNICATION INTERFACE (SCI)

The Serial Communication Interface (SCI) in the HD6303Y contains the following two operating modes: asynchronous mode by the NRZ format, and clocked synchronous mode which transfers data synchronously with the clock. In the asynchronous mode, data length, parity bits and number of stop bits can be selected, and eight transfer formats are provided.

The SCI consists of the following registers as shown in Fig. 24 Block Diagram.

- Transmit/Receive Control Status Register 1 (TRCSR1)
- Rate/Mode Control Register (RMCR)
- Transmit/Receive Control Status Register 2 (TRCSR2)
- Receive Data Register (RDR)
- Recevie Shift Register
- Transmit Data Register (TDR)
- Transmit Shift Register

To operate the SCI, initialize the RMCR and TRCSR2, after selecting the desirable operating mode and transfer format. Next, set the enable bit (TE or RE) of the TRCSR1. Operating mode and transfer format should be changed when the enable bit (TE, RE) is cleared. When setting the TE or RE again after changing the operating mode or transfer format, interval of more than a 1-bit cycle of the baud rate or bit rate is necessary. If a 1-bit cycle or more is not allowed, the SCI block may not be initialized.

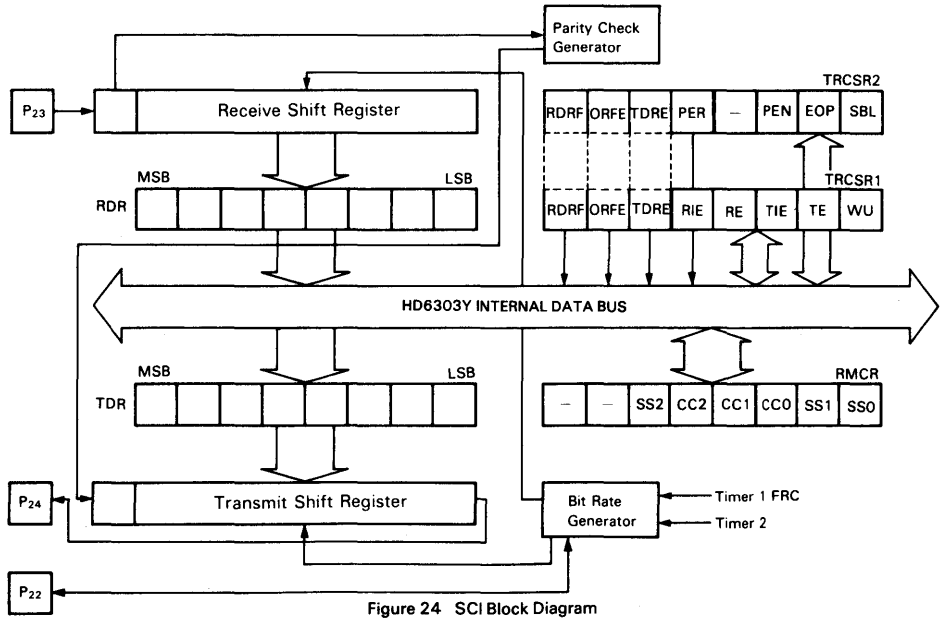


Figure 24 SCI Block Diagram

• Asynchronous Mode

Asynchronous mode contains 8 transfer formats as shown in Fig. 25.

Data transmission is enabled by setting TE bit of the TRCSR1, then port 2, bit 4 will unconditionally become a serial output independently of the corresponding DDR.

To transmit data, set the desirable transmit format with RMCR and TRCSR2. When the TE bit is set, the data can be transmitted after transmitting the one frame of preamble ("1").

The conditions at this stage are as follows.

1) If the TDR is empty (TDRE=1), consecutive 1's are produced to indicate the idle state.
2) If the TDR contains data (TDRE=0), data is sent to the Transmit Shift Register and data transmit starts.

During data transmit, a start bit of "0" is transmitted first. Then 7-bit or 8-bit data (starts from bit 0) is transmitted. With PEN=1, the parity bit, even or odd, selected by EOP bit is added, lastly the stop bit (1 bit or 2 bis) is sent.

When the TDR is "empty", hardware sets TDRE flag bit. If the CPU doesn't respond to the flag in proper timing (the TDRE is in set condition till the next normal data transfer starts from the transmit data register to the transmit sift register), "1" is transferred instead of the start bit "0" and continues to be transferred till data is provided to the data register. While the TDRE is "1", "0" is not transferred.

Data receive is possible by setting RE bit. This makes port 2, bit 3 a serial input. The operation mode of data receive is decided by the contents of the TRCSR2 and RMCR at first, and set RE bit of TRCSR1. The first "0" (space) synchronizes the receive bit flow. Each bit of the following data will be strobed in the middle. If a stop bit is not "1", a framing error assumed and ORFE is set.

When a framing error occurs, receive data is transferred to the Receive Data Register and the CPU can read the error-generating data. This makes it possible to detect a line break.

When PEN bit is set, the parity check is done. If the parity bit does not match the EOP bit, a parity error occurs and the PER bit is set, not the RDRF bit. Also, when the parity error occurs the receive data can be read just like in the case of the framing error.

The RDRF flag is set when the data is received without a framing error and a parity error.

If RDRF is still set when receiving the stop bit of the next data, ORFE is set to indicate the overrun generation. CPU can get the receive data by reading RDR. When 7 bit data format is selected, the 8th bit of RDR is "0".

When the CPU read the receive Data Register as a response to RDRF flag or ORFE flag after having read TRCSR, RDRF or ORFE is cleared.

(Note) Clock Source in Asynchronous Mode

If CC1:CC0=10, the internal bit rate clock is provided at P_{22} regardless of the values for TE or RE. Maximum clock rate is E÷16.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P_{22} at sixteen times (16×) the desired bit rate, but not greater than E.

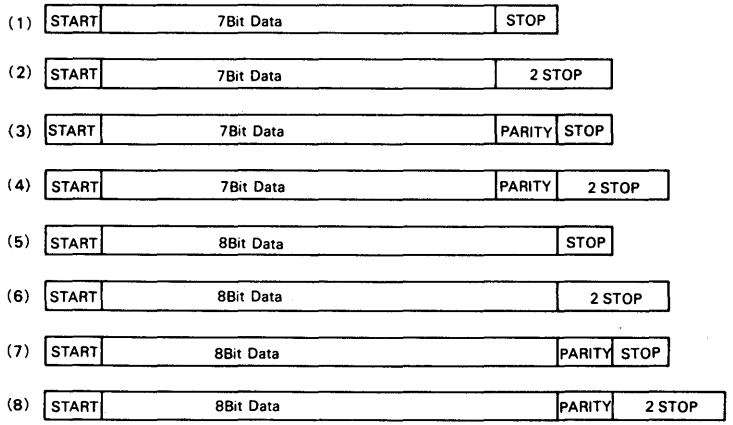


Figure 25 Asynchronous Mode Transfer Format

• Clocked Synchronous Mode

In the clocked synchronous mode, data transmit is synchronized with the clock pulse. The HD6303Y SCI provides functionally independent transmitter and receiver which makes full duplex operation possible in the asynchronous mode. But in the clocked synchronous mode an SCI clock I/O pin is only P_{22}, so the simultaneous receive and transmit operation is not available. In this mode, TE and RE should not be in set condition ("1") simultaneously. Fig. 26 gives a synchronous clock and a data format in the clocked synchronous mode.

1) Data transmit

Data transmit is realized by setting TE bit in the TRCSR1. Port 2, bit 4 becomes an output unconditionally independent of the value of the corresponding DDR.

Both the RMCR and TRCSR should be set in the desirable operating condition for data transmit.

When an external clock input is selected and the TDRE flag is "0", data transmit is performed from port 2, bit 4, synchronizing with 8 clock pulses input from external to port 2, bit 2.

Data is transmitted from bit 0 and the TDRE is set when the Transmit Shift Register (TSR) is "empty". More than 9th clock pulse of external are ignored.

When data transmit is selected to the clock output, the MPU produces transmit data and synchronous clock at TDRE flag clear.

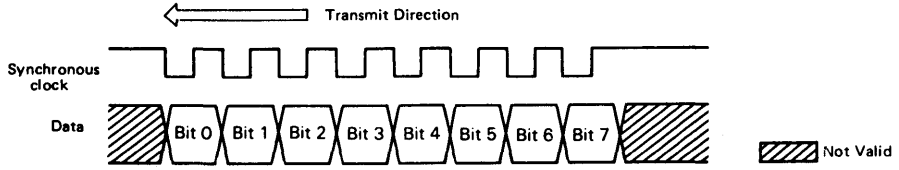
2) Data receive

Data receive is enabled by setting RE bit. Port 2, bit 3 will be a serial input. The operating mode of data receive is decided by the TRCSR1 and the RMCR.

If the external clock input is selected, 8 external clock pulses and the synchronized receive data are input to port 2, bit 2 and bit 3 respectively. The MPU put receive data into the receive data shift register by this clock and set the RDRF flag at the termination of 8 bit

data receive. More than 9th clock pulse of external input are ignored. When RDRF is cleared, the MPU starts receiving the next data instantly. So, RDRF should be cleared with P_{22} "High".

When data receive is selected with the clock output, 8 synchronous clocks are output to the external by setting RE bit. So receive data should be input from external synchronously with this clock. When the first byte data is received, the RDRF flag is set. After the second byte, receive operation is performed by sending the synchronous clock to the external after clearing the RDRF bit.



- Transmit data is produced from a falling edge of a synchronous clock to the next falling edge.
- Receive data is latched at the rising edge.

Figure 26 Clocked Synchronous Mode Format

- **Transmit/Receive Control Status Register (TRCSR1) ($0011)**

The TRCSR1 is composed of 8 bits which are all readable. Bits 0 to 4 are also writable. This register is initialized to $20 during reset. Each bit functions are as follows.

Transmit/Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU Wake-up

In a typical multi-processor configuration, the software protocol provides the destination address at the first byte of the message. In order to make uninterested MPU ignore the remaining message, a wake-up function is available. By this, uninterested MPU can inhibit all further receive processing till the next message starts.

Then wake-up function is triggered by consecutive 1's with 1 frame length. The software protocol should provide the idle time between messages.

By setting this bit, the MPU stops data receive till the next message. The receive of consecutive "1" with one frame length wakes up and clears this bit by hardware and then the MPU restarts receive operation. However, the RE flag should be already set before setting this bit. In the clocked synchronous mode WU is not available, so this bit should not be set.

Bit 1 TE Transmit Enable

When this bit is set, transmit data will appear at port 2, bit 4 after one frame preamble in asynchronous mode, while in clocked synchronous mode it appears immediately. This is executed regardless of the value of the corresponding DDR. When TE is cleared, the serial I/O doesn't affect port 2, bit 4.

Bit 2 TIE Transmit Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when TDRE (bit 5) is set. When cleared, the interrupt is inhibited.

Bit 3 RE Receive Enable

When set, a signal is input to the receiver from port 2, bit 3 regardless of the value of the DDR. When RE is cleared, the serial I/O doesn't afffect port 2, bit 3.

Bit 4 RIE Receive Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when RDRF (bit 7) or ORFE (bit 6) is set. When cleared, the interrupt is inhibited.

Bit 5 TDRE Transmit Data Register Empty

TDRE is set by hardware when the TDR is transferred to the Transmit Shift Register in the asynchronous mode, while in clocked synchronous mode when the TDSR is "empty". This bit is cleared by reading the TRCSR1 or TRCSR2 and writing new transmit data to the TDR when TDRE=1 TDRE is set to "1" during reset.

Bit 6 ORFE Overrun Framing Error

ORFE is set by hardware when an overrun or a framing error is generated (during data-receive only). An overrun error occurs when new receive data is ready to be transferred to the RDR during RDRF still being set. A framing error occurs when a stop bit is "0". But in clocked synchronous mode, this bit is not affected. This bit is cleared by reading the TRCSR1 or TRCSR2, and the RDR, when RDRF= 1. ORFE is cleared during reset.

Bit 7 RDRF Receive Data Register Full

RDRF is set by hardware when data is received normally and transferred from the Receive Shift Register (RSR) to the RDR. This bit is cleared by reading TRCSR1 or TRCSR2, and the RDR, when RDRF=1. This bit is cleared during reset.

- **Transmit Rate/Mode Control Register (RMCR)**

The RMCR controls the following serial I/O:

- Baud Rate
- Data Format
- Clock source
- Port 2, Bit 2 Function
- Operation Mode

All bits are readable/writable. Bit 0 to 5 of the RMCR are cleared during reset.

Transfer Rate/Mode Control Register

7	6	5	4	3	2	1	0	
–	–	SS2	CC2	CC1	CC0	SS1	SS0	$0010

Bit 0 SS0
Bit 1 SS1 Speed Select
Bit 5 SS2

These bits control the baud rate used for the SCI. Table 7 lists the available baud rates. The timer 1 FRC (SS2=0) and the timer 2 up counter (SS2=1) provide the internal clock to the SCI. When selecting the timer 2 as a baud rate clock source, it functions as a baud rate generator. The timer 2 generates the baud rate listed in Table 8 depending on the value of the TCONR.

(Note) When operating the SCI with internal clock, do not perform write operation to the timer/counter which is the

Table 7 SCI Bit Times and Transfer Rates

(1) Asynchronous Mode

SS2	SS1	SS0	XTAL / E	2.4576MHz / 614.4kHz	4.0MHz / 1.0MHz	4.9152MHz / 1.2288MHz
0	0	0	E÷16	26.μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

* When SS2 is "1", Timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32\,(N+1)} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

(2) Clocked Synchronous Mode *

SS2	SS1	SS0	XTAL / E	4.0MHz / 1.0MHz	6.0MHz / 1.5MHz	8.0MHz / 2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

* Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC ~ 1/2 system clock.

** The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate } (\mu\text{s/bit}) = \frac{4\,(N+1)}{f} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

Table 8 Baud Rate and Time Constant Register Example

Baud Rate (Baud) \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

* E/8 clock is input to the timer 2 up counter and E clock otherwise.

clock source of the SCI.

Bit 2 **CC0**
Bit 3 **CC1** Clock Control/Format Select*
Bit 4 **CC2**

These bits control the data format and the clock source (refer to Table 9).

* CC0, CC1 and CC2 are cleared during reset and the MPU goes to the clocked synchronous mode of the external clock operation. Then the MPU automatically set port 2, bit 2 into the clock input state. When using port 2, bit 2 as an output port, the DDR of port 2 should be set to "1" and CC1 and CC0 to "0" and "1" respectively.

Bit 6 Not Used.
Bit 7 Not Used

- **Transmit/Receive Control Status Register 2 (TRCSR2)**

The TRCSR2 is a 7-bit register which can select a data format in the asynchronous mode. The upper 3 bits are the same address as the TRCSR1. Therefore, the RDRF, ORFE and TDRE can be read by either the TRCSR1 or TRCSR2. Bits 0 to 2 of the TRCSR2 are used for read/write. Bits 4 to 7 are used only for read.

Transmit/Receive Control Status Register 2

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	PER	–	PEN	EOP	SBL	$001E

Bit 0 **SBL Stop Bit Length**
This bit selects the stop bit length in the asynchronous mode. If this bit is "0", the stop bit is 1-bit. If "1", the stop bit is 2-bit. This bit is cleared during reset.

Bit 1 **EOP Even/Odd Parity**
This bit selects the parity generated and checked when the PEN is "1". If this bit is "0", the parity is even. If "1", it is odd. This bit is cleared during reset.

Bit 2 **PEN Parity Enable**
This bit decides whether the parity bit should be generated and checked in the asynchronous mode or not. If this bit is "0", the parity bit is neither generated nor checked. If "1", it is generated and checked. This bit is cleared during reset.
The 3 bits above do not affect the SCI opertion in the clocked synchronous mode.

Bit 3 Not Used

Bit 4 **PER Parity Error**
This bit is set when the PEN is "1" and a parity error occurs. It is cleared by reading the RDR after reading the TRCSR2, when PER=1.

Bit 5 **TDRE**
Transmit Data Register Empty

Bit 6 **ORFE**
Overrun/Framing Error

Bit 7 **RDRF**
Receive Data Register Full

* Each flag of the TDRE, ORFE, and RDRF can be read from either the TRCSR1 or TRCSR2.

■ TIMER, SCI STATUS FLAG

Table 10 shows the set and reset conditions of each status flag in the timer 1, timer 2 and SCI.

Table 9 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock Source	Port 2, Bit 2	Port 2, Bit 3	Port 2, Bit 4
0	0	0	8-bit data	Clocked Synchronous	External	Input	When the TRCSR1, RE bit is "1", bit 3 is used as a serial input.	
0	0	1	8-bit data	Asynchronous	Internal	Not Used**		
0	1	0	8-bit data	Asynchronous	Internal	Output*		
0	1	1	8-bit data	Asynchronous	External	Input		
1	0	0	8-bit data	Clocked Synchronous	Internal	Output	When the TRCSR1, TE bit is "1", bit 4 is used as a serial output.	
1	0	1	7-bit data	Asynchronous	Internal	Not Used**		
1	1	0	7-bit data	Asynchronous	Internal	Output*		
1	1	1	7-bit data	Asynchronous	External	Input		

* Clock output regardless of the TRCSR1, bit RE and TE.
** Not used for the SCI.

Table 10 Timer 1, Timer 2 and SCI Status Flag

		Set Condition	Clear Condition
P6CSR	IS FLAG	Falling edge input to P_{54} ($\overline{IS}$)	1. Read the P6CSR then read or write the PORT6, when IS FLAG = 1 2. $\overline{RES}$ = 0
Timer 1	ICF	FRC → ICR by Rising or Falling edge input to P_{20} (Selecting with the IEDG bit)	1. Read the TCSR1 or TCSR2 then ICRH, when ICF = 1 2. $\overline{RES}$ = 0
	OCF1	OCR1 = FRC	1. Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1 = 1 2. $\overline{RES}$ = 0
	OCF2	OCR2 = FRC	1. Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2 = 1 2. $\overline{RES}$ = 0
	TOF	FRC = $FFFF + 1 cycle	1. Read the TCSR1 then FRCH, when TOF = 1 2. $\overline{RES}$ = 0
Timer 2	CMF	T2CNT = TCONR	1. Write "0" to CMF, when CMF = 1 2. $\overline{RES}$ = 0
SCI	RDRF	Receive Shift Register → RDR	1. Read the TRCSR1 or TRCSR2 then RDR, when RDRF = 1 2. $\overline{RES}$ = 0
	ORFE	1. Framing Error (Asynchronous Mode) Stop Bit = 0 2. Overrun Error (Asynchronous Mode) Receive Shift Register → RDR when RDRF = 1	1. Read the TRCSR1 or TRCSR2 then RDR, when ORFE = 1 2. $\overline{RES}$ = 0
	TDRE	1. Asynchronous Mode TDR → Transmit Shift Register 2. Clocked Synchronous Mode Transmit Shift Register is "empty" 3. $\overline{RES}$ = 0	Read the TRCSR1 or TRCSR2 then write to the TDR, when TDRE = 1
	PER	Parity when PEN=1	1. Read the TRCSR2 then RDR, when PER=1 2. $\overline{RES}$=0

(Note) → ; Transfer = ; equal

ICRH; Upper byte of ICR
OCR1H; Upper byte of OCR1
OCR2H; Upper byte of OCR2

OCR1L; Lower byte of OCR1
OCR2L; Lower byte of OCR2
FRCH; Upper byte of FRC

■ LOW POWER DISSIPATION MODE

The HD6303Y provides two low power dissipation modes; sleep and standby.

• Sleep Mode

The MPU goes to the sleep mode by SLP instruction execution. In the sleep mode, the CPU stops its operation, while the registers' contents are retained. In this mode, the peripherals except the CPU such as timers, SCI, etc. continue their functions. The power dissipation of sleep-condition is one fourth that of operating condition.

The MPU returns from this mode by an interrupt, $\overline{RES}$ or $\overline{STBY}$; it goes to the reset state by $\overline{RES}$ and the standby mode by $\overline{STBY}$. When the CPU acknowledges an interrupt request, it cancels the sleep mode, returns to the operation mode and branches to the interrupt routine. When the CPU masks this interrupt, it cancels the sleep mode and executes the next instruction. However, for example, if the timer 1 or 2 prohibits a timer interrupt, the CPU doesn't cancel the sleep mode because of no interrupt request.

This sleep mode is effective to reduce the power dissipation for a system with no need of the HD6303Y's consecutive operation.

• Standby Mode

The MPU goes to the standby mode with the $\overline{STBY}$ "Low" or by clearing the STBY flag. In this mode, the HD6303Y stops all the clocks and goes to the reset state. In this mode, the power dissipation is reduced to several μA. During standby, all pins, except the power supply (V_{CC}, V_{SS}), the $\overline{STBY}$, $\overline{RES}$ and XTAL (which outputs "0"), go to the high impedance state. In this mode, power (V_{CC}) is supplied to the HD6303Y, and the contents of RAM is retained. The MPU returns from this mode during reset. When the MPU goes to the standby mode with $\overline{STBY}$ "Low", it will restart at the timing shown in Fig. 27(a). When the MPU goes to the standby mode by clearing the STBY flag, it will restart only by keeping the $\overline{RES}$ "Low" for longer than the oscillating stabilization time. (Fig. 27(b))

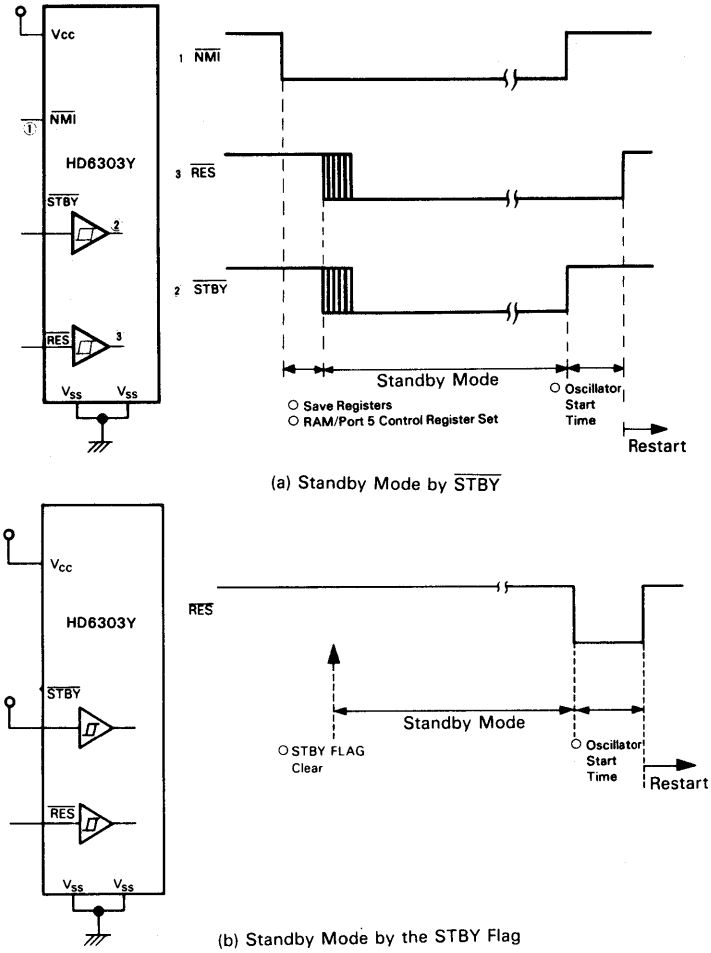


Figure 27 Standby Mode Timing

■ TRAP FUNCTION

The CPU generates an interrupt with the highest priority (TRAP) when fetching an undefined instruction or an instruction from non-memory space. The TRAP prevents the system-burst caused by noise or a program error.

• Op Code Error

When fetching an undefined op code, the CPU saves registers as well as a normal interrupt and branches to the TRAP ($FFEE, $FFEF). This has the priority next to reset.

• Address Error

When an instruction fetch is made from the address of internal register, the MPU generaters an interrupt as well as an op code error. But on the system with no memory in its external memory area, this function is not applicable if an instruction fetch is made from the external non-memory area. Addresses where an address error occurs are from $0000 to $0027.

This function is available only for an instruction fetch and is not applicable to the access of normal data read/write.

(Note) The TRAP interrupt provides a retry function differently from other interrupts. This is a program flow return to the address where the TRAP occurs when a sequence returns to a main routine from the TRAP interrupt routine by RTI. The retry can prevent the system burst caused by noise, etc.
However, if another TRAP occurs, the program repeats the TRAP interrupt forever, so the consideration is necessary in programming.

■ INSTRUCTION SET

The HD6303Y provides object code upward compatible with the HD6801 to utilize all instruction set of the HMCS6800. It also reduces the execution times of key instructions for throughput improvement.

Bit manipulation instruction, change instruction of the index register and accumulator and sleep instruction are also added.

The followings are explained here.

- CPU Programming Model (refer to Fig. 28)
- Addressing Mode
- Accumulator and Memory Manipulation Instruction (refer to Table 11)
- New Instruction
- Index Register and Stack Manipulation Instruction (refer to Table 12)
- Jump and Branch Instruction (refer to Table 13)
- Condition Code Register Manipulation (refer to Table 14)
- Op Code Map (refer to Table 15)

• Programming Model

Fig. 28 depicts the HD6303Y programming model. The double accumulator D consists of accumulator A and B, so when using the accumulator D, the contents of A and B are destroyed.

• CPU Addressing Mode

The HD6303Y provides 7 addressing modes. The addressing mode is determined by an instruction type and code. Tables 11 through 15 show addressing modes of each instruction with the execution times counted by the machine cycle.

When the clock frequency is 4MHz, the machine cycle time becomes microseconds directly.

Accumulator (ACCX) Addressing

Only an accumulator is addressed and the accumulator A or B is selected. This is a one-byte instruction.

Immediate Addressing

This addressing locates a data in the second byte of an instruction. However, LDS and LDX locate a data in the second and third byte exceptionally. This addressing is a 2 or 3-byte instruction.

Direct Addressing

In this addressing mode, the second byte of an instruction shows the address where a data is stored. 256 bytes ($0 through $255) can be addressed directly. Execution times can be reduced by storing data in this area so it is recommended to make it RAM for users' data area in configurating a system. This is a 2-byte instruction, while 3 byte with regard to AIM, OIM, EIM and TIM.

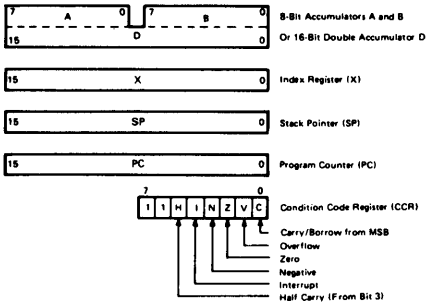


Figure 28 CPU Programming Model

Extended Addressing

In this mode, the second byte shows the upper 8 bit of the data stored address and the third byte the lower 8 bit. This indicates the absolute address of 3 byte instruction in the memory.

Indexed Addressing

The second byte of an instruction and the lower 8 bit of the index register are added in this mode. As for AIM, OIM, EIM and TIM, the third byte of an instruction and the lower 8 bits of the index register are added.

This carry is added to the upper 8 bit of the index register and the result is used for addressing the memory. The modified address is retained in the temporary address register, so the contents of the index register doesn't change. This is a 2-byte instruction except AIM, OIM, EIM and TIM (3-byte instruction).

Implied Addressing

An instruction itself specifies the address. This is, the instruction addresses a stack pointer, index register, etc. This is a one-byte instruction.

Relative Addressing

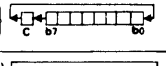
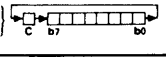
The second byte of an instruction and the lower 8 bits of the program counter are added. The carry or borrow is added to the upper 8 bit. So addressing from −126 to +129 byte of the current instruction is enabled. This is a 2-byte instruction.

(Note) CLI, SEI Instructions and Interrupt Operation
When accepting the IRQ at a preset timing with CLI and SEI instructions, more than 2 cycles are necessary between the CLI and SEI instructions. For example, the following program (a)(b) don't accept the IRQ but (c) accepts it.

(a)	(b)	(c)
·	·	·
·	·	·
·	·	·
·	·	CLI
CLI	CLI	NOP
SEI	NOP	NOP
·	SEI	SEI
·	·	·
·	·	·
·	·	·
·	·	·

The same thing can be said to the TAP instruction instead of the CLI and SEI instructions.

Table 11 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes														Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

(Note) Condition Code Register will be explained in Note of Table 14.

(continued)

Table 11 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	ACC A/ ACC B	•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	1	1	A	•	•	R	↕	⑥	↕
	LSRB													54	1	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	ACC A/ ACC B	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

(Note) Condition Code Register will be explained in Note of Table 14.

- **Additional Instruction**

In addition to the HD6801 instruction set, the HD6303Y prepares the following new instructions.

AIM ..... (M)·(IMM) → (M)
Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM ..... (M)+(IMM) → (M)
Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM ..... (M) ⊕ (IMM) → (M)
Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM ..... (M)·(IMM)
Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These are the 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX ..... (ACCD)↔(IX)
Exchanges the contents of accumulator and the index register.

SLP
Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 12 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD↔IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 14.

Table 13 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——		⑥		——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

(Note) * WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 14.

Table 14 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	⑩					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MCU Cycles
M_{SP} Contents of memory location pointed by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
– Arithmetic Minus
• Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3 to bit 4
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from/to bit 7
R Reset Always
S Set Always
↕ Set if true after test or clear
• Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕ C = 1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 15 OP-Code Map

OP CODE HI	LO	0000 0	0001 1	0010 2	0011 3	ACC A 0100 4	ACC B 0101 5	IND 0110 6	EXT / DIR* 0111 7	ACCA or SP IMM 1000 8	DIR 1001 9	IND 1010 A	EXT 1011 B	ACCB or X IMM 1100 C	DIR 1101 D	IND 1110 E	EXT 1111 F	LO
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

■ CPU OPERATION

• CPU Instruction Flow

When operating, the CPU fetches an instrution from a memory and executes the required function. This sequence starts with $\overline{RES}$ cancel and repeats itself limitlessly if not affected by a special instruction or a control signal. SWI, RTI, WAI and SLP instructions change this operation, while $\overline{NMI}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, IRQ_3, $\overline{HALT}$ and $\overline{STBY}$ control it. Fig. 29 gives the CPU mode transition and Fig. 30 the CPU system flow chart. Table 16 shows CPU operating states and port states.

• Operation at Each Instruction Cycle

Table 17 shows the operation at each instruction cycle. By the pipeline control of the HD6303Y, MULT, PUL, DAA and XGDX instructions, etc. prefetch the next instruction. So attention is necessary to the counting of the instruction cycles because it is different from the usual one—from op code fetch to the next instruction op code.

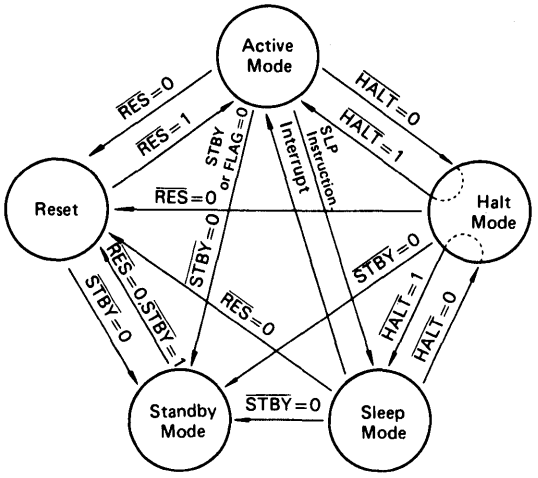
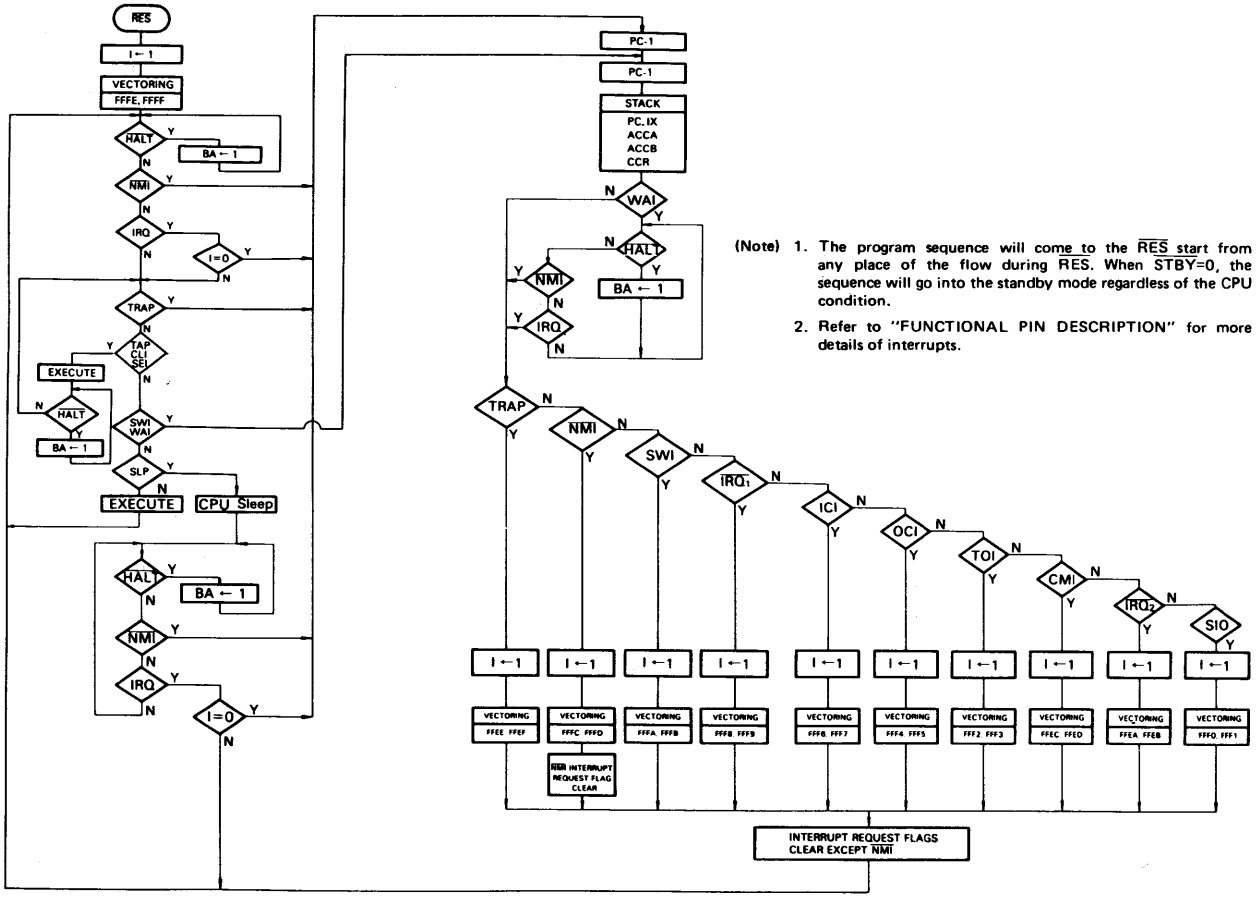


Figure 29 CPU Operation Mode Transition

Table 16 CPU Operation State and Port, Bus, Control Signal State

Port	Reset	STBY*3	HALT	Sleep
$A_0 \sim A_7$	H	T	T	H
Port 2	T	T	Keep	Keep
$D_0 \sim D_7$	T	T	T	T
$A_8 \sim A_{15}$	H	T	T	H
Port 5	T	T	Keep	Keep
Port 6	T	T	Keep	Keep
Control Signal	*1	T	*2	*1

*1 $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$ = H, BA = L
*2 $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$ = T, $\overline{LIR}$, BA = H
*3 E pin goes to high impedance state.



(Note) 1. The program sequence will come to the $\overline{RES}$ start from any place of the flow during $\overline{RES}$. When $\overline{STBY}$=0, the sequence will go into the standby mode regardless of the CPU condition.

2. Refer to "FUNCTIONAL PIN DESCRIPTION" for more details of interrupts.

Figure 30 HD6303Y System Flow Chart

Table 17 Cycle-by-Cycle Operation

Address Mode & Instructions		Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMMEDIATE									
ADC	ADD	2	1	Op Code Address+1	1	0	1	1	Operand Data
AND	BIT		2	Op Code Address+2	1	0	1	0	Next Op Code
CMP	EOR								
LDA	ORA								
SBC	SUB								
ADDD	CPX	3	1	Op Code Address+1	1	0	1	1	Operand Data (MSB)
LDD	LDS		2	Op Code Address+2	1	0	1	1	Operand Data (LSB)
LDX	SUBD		3	Op Code Address+3	1	0	1	0	Next Op Code
DIRECT									
ADC	ADD	3	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
AND	BIT		2	Address of Operand	1	0	1	1	Operand Data
CMP	EOR		3	Op Code Address+2	1	0	1	0	Next Op Code
LDA	ORA								
SBC	SUB								
STA		3	1	Op Code Address+1	1	0	1	1	Destination Address
			2	Destination Address	0	1	0	1	Accumulator Data
			3	Op Code Address+2	1	0	1	0	Next Op Code
ADDD	CPX	4	1	Op Code Address+1	1	0	1	1	Address of Operand (LSB)
LDD	LDS		2	Address of Operand	1	0	1	1	Operand Data (MSB)
LDX	SUBD		3	Address of Operand+1	1	0	1	1	Operand Data (LSB)
			4	Op Code Address+2	1	0	1	0	Next Op Code
STD	STS	4	1	Op Code Address+1	1	0	1	1	Destination Address (LSB)
STX			2	Destination Address	0	1	0	1	Register Data (MSB)
			3	Destination Address+1	0	1	0	1	Register Data (LSB)
			4	Op Code Address+2	1	0	1	0	Next Op Code
JSR		5	1	Op Code Address+1	1	0	1	1	Jump Address (LSB)
			2	FFFF	1	1	1	1	Restart Address (LSB)
			3	Stack Pointer	0	1	0	1	Return Address (LSB)
			4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
			5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM		4	1	Op Code Address+1	1	0	1	1	Immediate Data
			2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
			3	Address of Operand	1	0	1	1	Operand Data
			4	Op Code Address+3	1	0	1	0	Next Op Code
AIM	EIM	6	1	Op Code Address+1	1	0	1	1	Immediate Data
OIM			2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
			3	Address of Operand	1	0	1	1	Operand Data
			4	FFFF	1	1	1	1	Restart Address (LSB)
			5	Address of Operand	0	1	0	1	New Operand Data
			6	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
INDEXED								
JMP	3	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Accumulator Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data (MSB)
		4	IX+Offset+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Register Data (MSB)
		4	IX+Offset+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	IX+Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX+Offset	0	1	0	1	New Operand Data
		6	OP Code Address+2	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	IX+Offset	0	1	0	1	00
		5	Op Code Address+2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX+Offset	0	1	0	1	New Operand Data
		7	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
EXTEND								
JMP	3	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	0	1	0	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Accumulator Data
		4	Op Code Address+1	1	0	1	0	Next Op Code
PULX	4	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	0	1	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Index Register (LSB)
		4	Stack Pointer−1	0	1	0	1	Index Register (MSB)
		5	Op Code Address+1	1	0	1	0	Next Op Code
RTS	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Return Address (MSB)
		4	Stack Pointer+2	1	0	1	1	Return Address (LSB)
		5	Return Address	1	0	1	0	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	FFFF	1	1	1	1	Restart Address (LSB)
		7	FFFF	1	1	1	1	Restart Address (LSB)

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
WAI	9	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
RTI	10	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Conditional Code Register
		4	Stack Pointer+2	1	0	1	1	Accumulator B
		5	Stack Pointer+3	1	0	1	1	Accumulator A
		6	Stack Pointer+4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer+5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer+6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer+7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		Sleep						
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address+1	1	0	1	0	Next Op Code
RELATIVE								
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	0	1	1	Branch Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Branch Address ····· Test="1" Op Code Address+1 ··· Test="0"	1	0	1	0	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Branch Address	1	0	1	0	First Op Code of Subroutine

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 31, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD6303Y as possible.

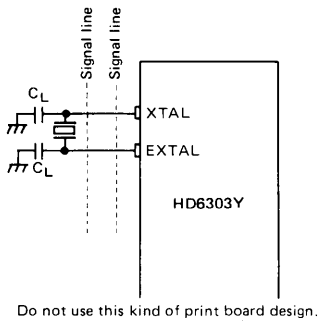


Figure 31 Precaution to the boad design of oscillation circuit

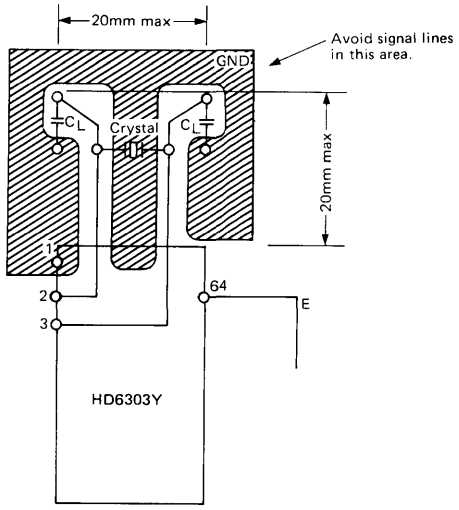


Figure 32 Example of Oscillation Circuits in Board Design

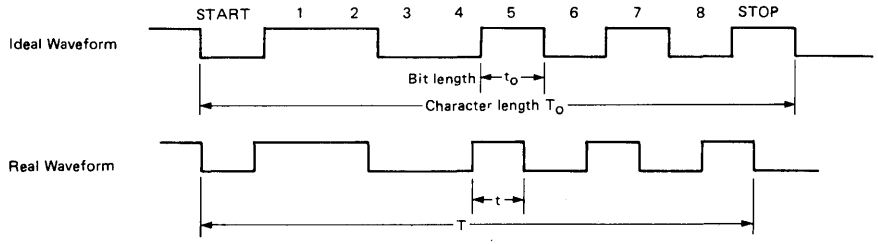
■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD6303Y is shown in Table 18.

Note: SCI = Serial Communication Interface

Table 18

	Bit distortion tolerance (t−to) /to	Character distortion tolerance (T−To) /To
HD6303Y	±43.7%	±4.37%



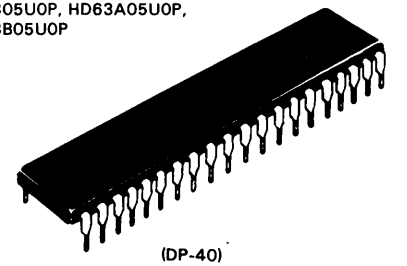
HD6305U0, HD63A05U0, HD63B05U0
CMOS MCU (Microcomputer Unit)

—PRELIMINARY—

The HD6305U0 is a CMOS 8-bit single-chip MCU which is similar to the HD6305X MCU family. This version is upward compatible with the HD6805 family in respect to instructions. A CPU, a clock generator, a 2k-byte ROM, a 128-byte RAM, 31 I/O terminals, two timers, and a serial communication interface (SCI) are incorporated in the HD6305U0. As a result of CMOS technology, the HD6305U0 consumes much less power than NMOS counterparts. In addition, three low power dissipation modes (stop, wait and standby) which further decreases power consumption, are included in the HD6305U0.

Other notable features include enhanced instruction cycle of the main instructions and the use of three additional instructions to improve system throughput.

HD6305U0P, HD63A05U0P, HD63B05U0P



(DP-40)

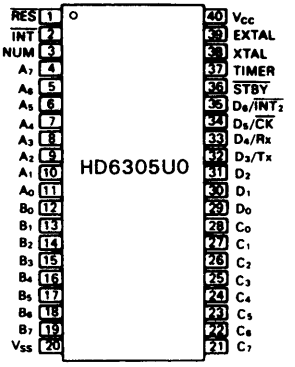
■ HARDWARE FEATURES

- CMOS 8-bit single-chip MCU
- 2048 bytes of ROM
- 128 bytes of RAM
- 31 bidirectional I/O terminals
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes
  - Wait..... In this mode, the clock oscillator is on and the CPU halts but the timer/serial/interrupt function is operatable. Also, all registers are held, except the I bit in the condition code register is cleared.
  - Stop..... In this mode, the clock stops but the RAM data, I/O status and registers are held. Except the timer control register (bits 6 and 7) and the I bit of the condition code register.
  - Standby.... In this mode, the clock stops, the RAM data is held, and the other internal condition is reset.
- Minimum instruction cycle time
  - HD6305U0 ..... 1 μs (f = 1 MHz)
  - HD63A05U0..... 0.67 μs (f = 1.5 MHz)
  - HD63B05U0 ..... 0.5 μs (f = 2 MHz)
- Wide operating range
  V_{CC} = 3 to 6V (f = 0.1 to 0.5 MHz)
  - HD6305U0 ..... f = 0.1 to 1 MHz (V_{CC} = 5V ± 10%)
  - HD63A05U0..... f = 0.1 to 1.5 MHz (V_{CC} = 5V ± 10%)
  - HD63B05U0 ..... f = 0.1 to 2 MHz (V_{CC} = 5V ± 10%)
- System development fully supported by an emulator

■ SOFTWARE FEATURES

- Similar to HD6800
- Byte efficient instruction set
- Powerful bit manipulation instructions (Bit Set, Bit Clear, and Bit Test and Branch usable for all RAM bits and all I/O terminals)

■ PIN ARRANGEMENT

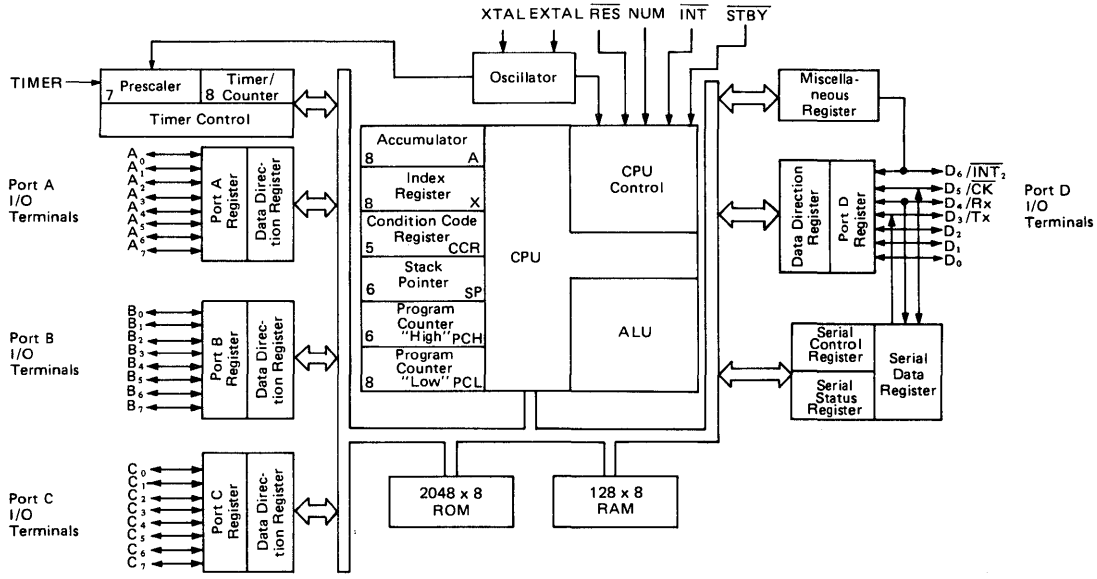


(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

- A variety of interrupt operations
- Index addressing mode useful for table processing
- A variety of conditional branch instructions
- Ten powerful addressing modes
- All addressing modes adaptable to RAM, and I/O instructions
- Three new instructions, Stop, Wait and DAA, added to the HD6805 family instruction set
- Instructions that are upward compatible with those of Motorola's MC6805P2 and MC146805G2
- Compatible instruction set with HD6305X

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 ~ +7.0	V
Input voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating temperature	T_{opr}	0 ~ +70	°C
Storage temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item		Symbol	Test condition	min	typ	max	Unit
Input voltage "High"	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC} - 0.5	–	V_{CC} + 0.3	V
	EXTAL			V_{CC} x 0.7	–	V_{CC} + 0.3	V
	Others			2.0	–	V_{CC} + 0.3	V
Input voltage "Low"	All Inputs	V_{IL}		-0.3	–	0.8	V
Current * dissipation	Operating	I_{CC}	f = 1MHz**	–	5	10	mA
	Wait			–	2	5	mA
	Stop			–	2	10	μA
	Standby			–	2	10	μA
Input leakage current	$\overline{TIMER}$, $\overline{INT}$, $\overline{STBY}$	$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} - 0.5V	–	–	1	μA
Three-state current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $D_0 \sim D_6$	$\|I_{TSI}\|$		–	–	1	μA
Input capacity	All terminals	C_{in}	f = 1MHz, V_{in} = 0V	–	–	12	pF

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8 V

** The value at f = xMHz can be calculated by the following equation: I_{CC} (f = xMHz) = I_{CC} (f = 1MHz) multiplied by x

• AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item	Symbol	Test condition	HD6305U0			HD63A05U0			HD63B05U0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock frequency	f_{cl}		0.4	–	4	0.4	–	6	0.4	–	8	MHz
Cycle time	t_{cyc}		1.0	–	10	0.666	–	10	0.5	–	10	μs
$\overline{INT}$ pulse width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ pulse width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ pulse width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
TIMER pulse width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation start time (crystal)	t_{OSC}	C_L = 22pF ± 20% R_s = 60Ω max	–	–	20	–	–	20	–	–	20	ms
Reset delay time	t_{RHL}	External cap. 2.2μF	80	–	–	80	–	–	80	–	–	ms

● **Port Electrical Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)**

Item		Symbol	Test condition	min	typ	max	Unit
Output voltage "High"	Ports A, B, C, D	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC} - 0.7	–	–	V
Output voltage "Low"		V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input voltage "High"	Ports A, B, C, D	V_{IH}		2.0	–	V_{CC} + 0.3	V
Input voltage "Low"		V_{IL}		- 0.3	–	0.8	V
Input leakage current		$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} - 0.5V	–	–	1	μA

● **SCI Timing (V_{CC} = 5.0V±10%, V_{SS} = GND and Ta = 0 ~ +70°C unless otherwise specified)**

Item	Symbol	Test Condition	HD6305U0			HD63A05U0			HD63B05U0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock Cycle	t_{Scyc}	Fig. 1 Fig. 2	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data Output Delay Time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data Set-up Time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data Hold Time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

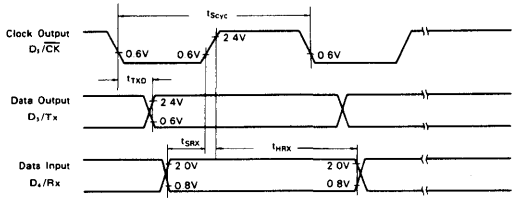


Figure 1 SCI Timing (Internal Clock)

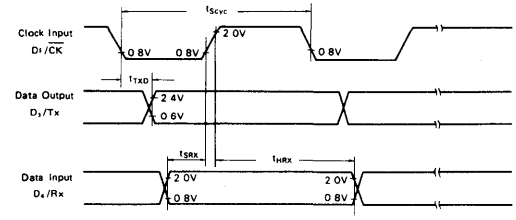
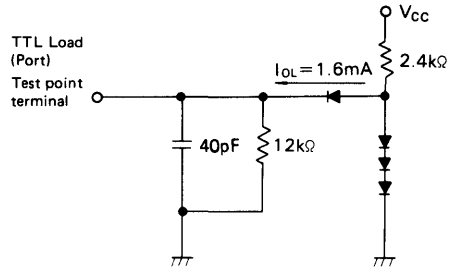


Figure 2 SCI Timing (External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ.

Figure 3 Test Load

■ DESCRIPTION OF TERMINAL FUNCTIONS

The input and output signals of the HD6305U0 are described here.

● **V_{CC}, V_{SS}**

Voltage is applied to the HD6305U0 through these two terminals. V_{CC} is 5.0V ± 10%, while V_{SS} is grounded.

● **$\overline{INT}$, $\overline{INT_2}$**

External interrupt request inputs to the HD6305U0. For details, refer to "INTERRUPTS". The $\overline{INT_2}$ terminal is also used as the port D_6 terminal.

- **XTAL, EXTAL**

These terminals provide input to the on-chip clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic filter is connected to the terminal. Refer to "INTERNAL OSCILLATOR" for using these input terminals.

- **TIMER**

This is an input terminal for event counter. Refer to "TIMER" for details.

- **$\overline{RES}$**

Used to reset the MCU. Refer to "RESET" for details.

- **NUM**

This terminal is not intended for user applications. It must be grounded to V_{SS}.

- **Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $D_0 \sim D_6$)**

These 31 terminals consist of three 8-bit I/O ports (A, B and C) and a 7-bit I/O port D. Each of these can be used as an input or output terminal on a bit basis through program control of the data direction register (DDR). For details, refer to "I/O PORTS."

Since port D_6 is also used for the $\overline{INT_2}$ input, in order to use port D_6 as an I/O port, the $\overline{INT_2}$ interrupt mask bit in the miscellaneous register should be set to "1" to disable the $\overline{INT_2}$ input.

- **$\overline{STBY}$**

This terminal is used to place the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and the internal condition is reset. For details, refer to "STANDBY MODE."

The terminals described in the following are I/O pins for serial communication interface (SCI). They are also used as ports D_3, D_4 and D_5. For details, refer to "SERIAL COMMUNICATION INTERFACE."

- **$\overline{CK}$ (D_5)**

Used to input or output clocks for serial operation.

- **Rx (D_4)**

Used to receive serial data.

- **Tx (D_3)**

Used to transmit serial data.

■ MEMORY MAP

The memory map of the HD6305U0 MCU is shown in Fig. 4. During interrupt processing, the contents of the MCU registers are saved into the stack in the sequence shown in Fig. 5. This saving begins with the lower byte (PCL) of the program counter. Then the value of the stack pointer is decremented and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CCR) are stacked in that order. In a subroutine call, only the contents of the program counter (PCH and PCL) are stacked.

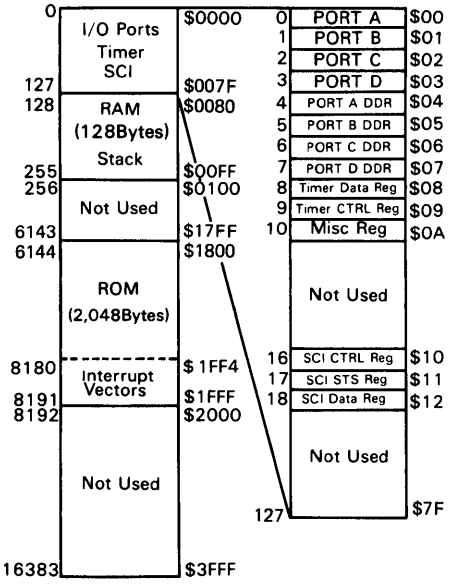
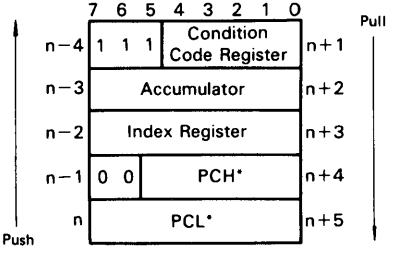


Figure 4 Memory Map of HD6305U0 MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 5 Sequence of Interrupt Stacking

■ REGISTERS

There are five registers which the programmer can operate.

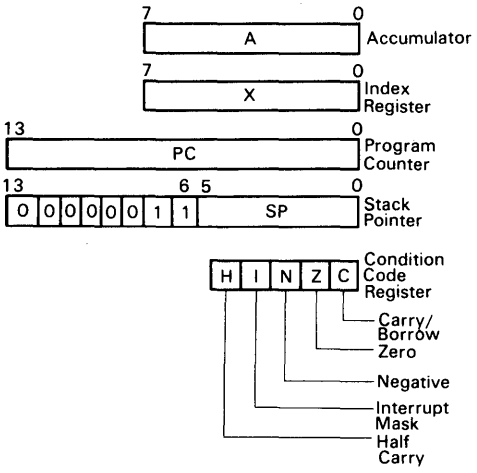


Figure 6 Programming Model

- **Accumulator (A)**

This accumulator is a general purpose 8-bit register which holds operands or the result of arithmetic operation or data processing.

- **Index Register (X)**

The index register is an 8-bit register, and is used for index addressing mode. Each of the addresses contained in the register consists of 8 bits which, combined with an offset value, provides an effective address.

In the case of a read/modify/write instruction, the index register can be used like an accumulator to hold operation data or the result of operation.

If not used in the index addressing mode, the register can be used to store data temporarily.

- **Program Counter (PC)**

The program counter is a 14-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is a 14-bit register that indicates the address of the next stacking space. Just after reset, the stack pointer is set at address $00FF. It is decremented when data is pushed, and incremented when pulled. The upper 8 bits of the stack pointer are fixed to 00000011. During the MCU being reset or during a reset stack pointer (RSP) instruction, the pointer is set to address $00FF. Since a subroutine or interrupt can use space up to address $00C1 for stacking, the subroutine can be used up to 31 levels and the interrupt up to 12 levels.

- **Condition Code Register (CCR)**

The condition code register is a 5-bit register, each bit indicating the result of the instruction just executed. The bits can be individually tested by conditional branch instructions. The CCR bits are as follows:

Half Carry (H): Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts, except a software interrupt, to be masked. If an interrupt occurs with the bit I set, it is latched. It will be processed the instant the interrupt mask bit is reset. (More specifically, it will enter the interrupt processing routine after the instruction following the CLI has been executed.)

Negative (N): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (bit 7 is logic "1").

Zero (Z): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is zero.

Carry/ Borrow (C): Represents a carry or borrow that occurred in the most recent arithmetic operation. This bit is also affected by the Bit Test and Branch instruction, a shift instruction and a Rotate instruction.

■ INTERRUPT

There are six different types of interrupt: external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER or the SCI and $TIMER_2$ generate the same vector address, respectively. Although, a different vector address is generated for a timer interrupt during the wait mode, as shown in Table 1.

When an interrupt occurs, the program in progress stops and then the CPU status is saved onto the stack. And then, the interrupt mask bit (I) of the condition code register is set and the start address of the interrupt processing routine is obtained from a particular interrupt vector address. Then the interrupt routine starts from the start address. System can exit from the interrupt routine by a RTI instruction. When this instruction is executed, the CPU status before the interrupt (saved onto the stack) is pulled and the CPU restarts the sequence with the instruction next to the one at which the interrupt occurred. Table 1 lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Priority	Interrupt	Vector Address	
1	$\overline{RES}$	$1FFE,	$1FFF
2	SWI	$1FFC,	$1FFD
3	$\overline{INT}$	$1FFA,	$1FFB
4	TIMER/$\overline{INT_2}$	$1FF8,	$1FF9
5	TIMER (WAIT)	$1FF6,	$1FF7
6	SCI/$TIMER_2$	$1FF4,	$1FF5

A flowchart of the interrupt sequence is shown in Fig. 7. A block diagram of the interrupt request source is shown in Fig. 8.

In the block diagram of Fig. 8, the external interrupt $\overline{INT_2}$ is a falling edge trigger input, whereas, the external interrupt $\overline{INT}$ can be configured as a falling edge trigger input or a combination of falling edge and low level trigger input, depending on the status of bit 5 in the miscellaneous register (MR). When an interrupt request is detected at the $\overline{INT_2}$ or $\overline{INT}$ inputs, an interrupt request is generated and latched. The $\overline{INT}$ interrupt request is automatically cleared if jumping is made to the $\overline{INT}$ processing routine. Meanwhile, the $\overline{INT_2}$ request is cleared if "0" is written in bit 7 of the miscellaneous register.

For the external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$) and serial interrupt (SCI), each interrupt request is held, but not processed, if the I bit of the condition code register is set. Immediately after the I bit is cleared, the corresponding interrupt processing starts according to the priority.

The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by setting bit 6 of the timer control register; the SCI interrupt by setting bit 5 of the serial status register; and the $TIMER_2$ interrupt by setting bit 4 of the serial status register.

The status of the $\overline{INT}$ terminal can be tested by a BIL or BIH instruction. The $\overline{INT}$ falling edge and falling edge/low level detec-

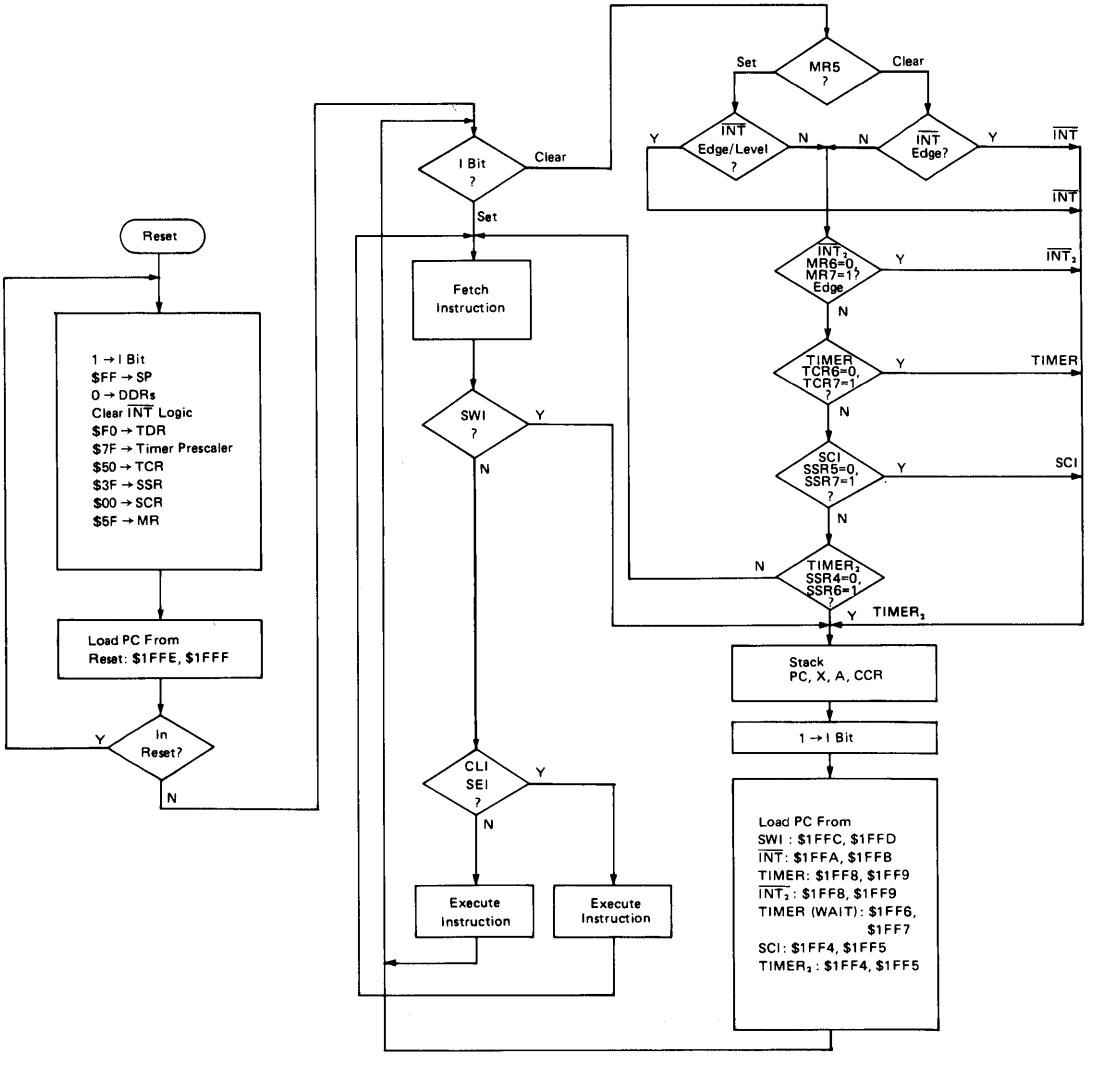


Figure 7 Interrupt Flowchart

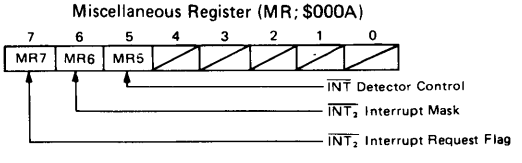
tor circuit and its latching circuit are independent of testing by these instructions. This is also true with the status of the $\overline{INT_2}$ terminal.

- **Miscellaneous Register (MR; $000A)**

The external interrupt $\overline{INT_2}$ and the TIMER interrupt have identical interrupt vector addresses, as shown in Table 1. For this reason, bits 6 and 7 of a special register called the miscellaneous register (MR: $000A) are available to control the $\overline{INT_2}$ interrupt. Moreover, bit 5 of the MR controls the sensing mode for the $\overline{INT}$ interrupt detector (falling edge detector or falling edge/low level detector).

Bit 7 of the MR is the $\overline{INT_2}$ interrupt request flag. When a falling edge is detected at the $\overline{INT_2}$ terminal, bit 7 is set to "1" Then the interrupt routine software (vector addresses: $1FF8, $1FF9) checks bit 7 to see if an $\overline{INT_2}$ interrupt occurred. Bit 7 can be reset by software.

Miscellaneous Register (MR; $000A)



Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If this bit is set to "1", then the $\overline{INT_2}$ interrupt is disabled. Both read and write are possible with bit 7, but "1" cannot be written in this bit by software. In other words, an $\overline{INT_2}$ interrupt request by software is not possible.

Bit 5 is the control bit for $\overline{INT}$ interrupt detection. If this bit is reset to "0", the detection logic will detect a falling edge. When this bit is set to "1", the detection logic will detect a falling edge or a low level.

When reset, bit 7 is cleared to "0", bit 6 is set to "1" and bit 5 is cleared to "0".

■ TIMER

Figure 9 shows an MCU timer block diagram. The timer data register is loaded by software and, upon receipt of a clock input, begins to count down. When the timer data register (TDR) becomes "0", the timer interrupt request bit (bit 7) in the timer control register is set. In response to the interrupt request, the MCU saves its status into the stack and fetches the timer interrupt routine addresses $1FF8 and $1FF9 (or $1FF6 and $1FF7 when the timer interrupt occures during the wait mode) and executes the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also mask the timer interrupt.

The source clock to the timer can be either an external signal from the timer input terminal or the internal E signal (the oscillator clock divided by 4). If the E signal is used as the source, the clock input can be gated by the input to the timer input terminal.

Once the timer count has reached 0, it starts counting down with "FF". The count can be monitored whenever desired by reading the timer data register. This permits the program to know the length of time having passed after the occurrence of a timer interrupt, without disturbing the contents of the counter.

When the MCU is reset or placed in the stop mode, the timer data register (TDR) is initialized to $F0. The timer interrupt request bit (bit 7) then is cleared and the timer interrupt mask bit (bit 6) is set.

To clear the timer interrupt request bit (bit 7), it is necessary to write "0" in that bit.

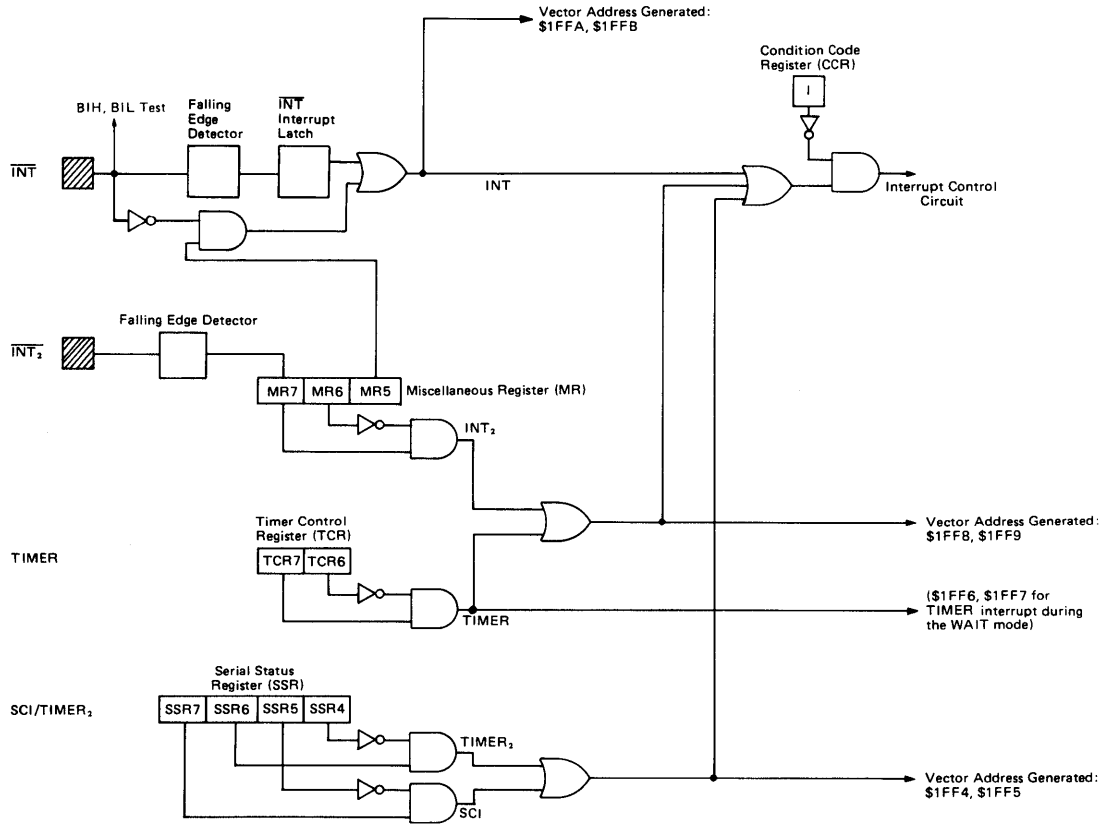


Figure 8 Interrupt Request Generation Circuitry

TCR7	Timer interrupt request
0	Absent
1	Present

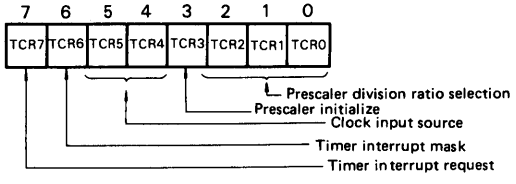
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

- **Timer Control Register (TCR; $0009)**

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After reset, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$F0" immediately after reset.

When "1" is written in bit 3, the prescaler is initialized to "$7F". This bit always shows "0" when read.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under TIMER terminal control
1	0	No clock input (counting stopped)
1	1	Event input from TIMER terminal

A prescaler division ratio is selected by the combination of three bits (bits 0, 1 and 2) of the timer control register (see Table 3). There are eight different division ratios: ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128. After reset, the TCR is set to the ÷1 mode.

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

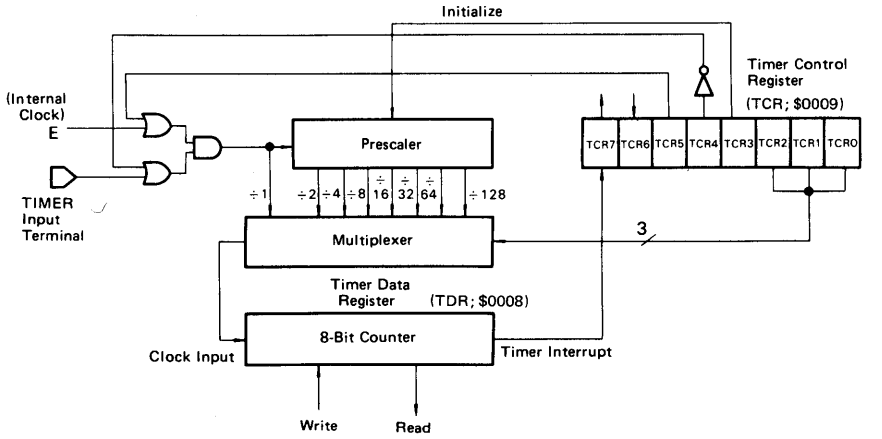


Figure 9 Timer Block Diagram

A timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. This bit can be cleared by writing "0" in that bit.

■ SERIAL COMMUNICATION INTERFACE (SCI)

This interface is used for serial transmission or reception of 8-bit data. Sixteen transfer rates are available in the range from 1 μs to approx. 32 ms (for oscillation at 4 MHz).

The SCI consists of three registers, one eighth counter and one prescaler. (See Fig. 10.) SCI communicates with the CPU via the data bus, and with the outside world through bits 3, 4 and 5 of port D. Described below are the operations of each register and data transfer.

● SCI Control Register (SCR; $0010)

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

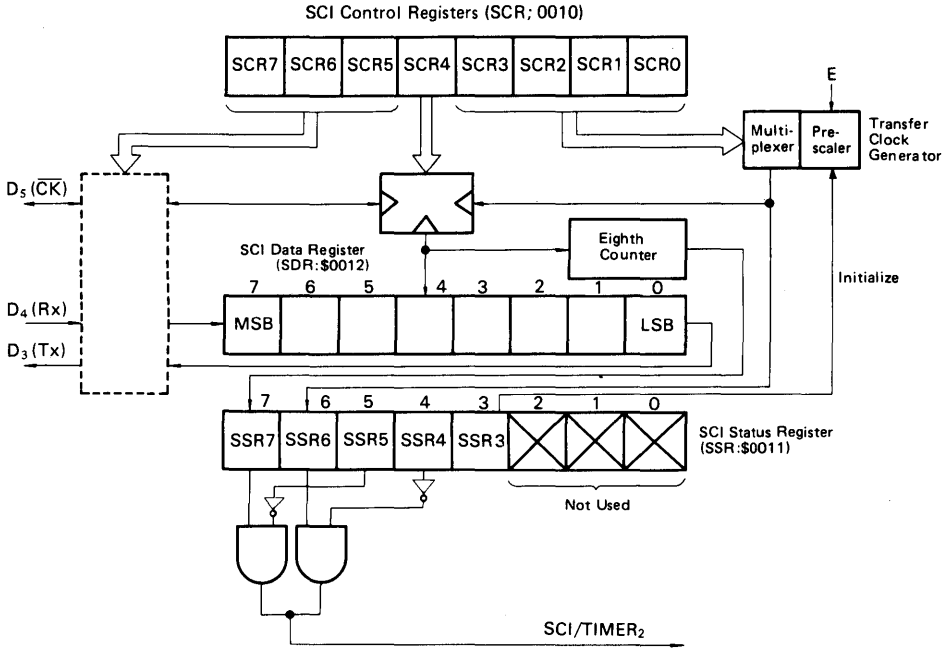


Figure 10 SCI Block Diagram

SCR7	D_3 terminal
0	Used as I/O terminal (by DDR)
1	Serial data output (DDR output)

SCR6	D_4 terminal
0	Used as I/O terminal (by DDR)
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	D_5 terminal
0	0	–	Used as I/O terminal (by DDR)
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)

When this bit is set, the DDR corresponding to the D_3 becomes "1" and this terminal serves for output of SCI data. After reset, the bit is cleared to "0".

Bit 6 (SCR6)

When this bit is set, the DDR corresponding to the D_4 becomes "0" and this terminal serves for input of SCI data. After reset, the bit is cleared to "0".

Bits 5 and 4 (SCR5, SCR4)

These bits are used to select a clock source. After reset, the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)

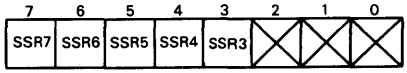
These bits are used to select a transfer clock rate. After reset, the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

- **SCI Data Register (SDR; $0012)**

A serial-parallel conversion register that is used for transfer of data.

- **SCI Status Register (SSR; $0011)**



Bit 7 (SSR7)

Bit 7 is the SCI interrupt request bit which is set upon completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5 = "1". The bit can also be cleared by writing "0" in it.

Bit 6 (SSR6)

Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is used commonly with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When reset, the bit is cleared. It also be cleared by writing "0" in it. (For details, see $TIMER_2$.)

Bit 5 (SSR5)

Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When reset, it is set to "1".

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

Bit 4 (SSR4)

Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When reset, it is set to "1".

Bit 3 (SSR3)

When "1" is written in this bit, the prescaler of the transfer clock generator is initialized. When read, the bit always is "0".

Bits 2 ~ 0

Not used.

- **Data Transmission**

By writing the desired control bits into the SCI control registers, a transfer rate and a source of transfer clock are determined and bits 3 and 5 of port D are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the D_3/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock. (See Fig. 11.) When 8 bits of data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the D_3/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the $D_3/\overline{CK}$ terminal is set as input. If the internal clock has been selected, the $D_3/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

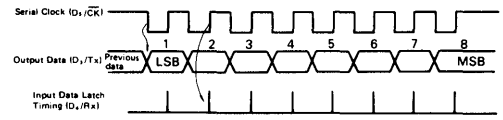


Figure 11 SCI Timing Chart

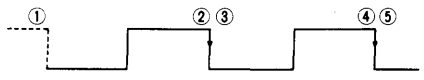
- **Data Reception**

By writing the desired control bits into the SCI control register, a transfer rate and a source of transfer clock are determined and bits 4 and 5 of Port D are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed for the second and subsequent data receptions. It must be taken only after resetting.)

The data from the D_4/Rx terminal is input to the SCI data register synchronously with the leading edge of the serial clock (see Fig. 11). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored and the data is received synchronously with the clock from the $D_3/\overline{CK}$ terminal. If the internal clock has been selected, the $D_3/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

- **$TIMER_2$**

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 ~ 0 of the SCI control register (4 μs ~ approx. 32 ms (for oscillation at 4 MHz)) is input to bit 6 of the SCI status register and the $TIMER_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, $TIMER_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : $TIMER_2$ interrupt request
③, ⑤ : $TIMER_2$ interrupt request bit cleared

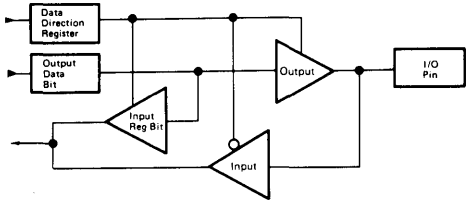
$TIMER_2$ is commonly used with the SCI transfer clock generator. If wanting to use $TIMER_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■ I/O PORTS

There are 31 input/output terminals (ports A, B, C, D). Each I/O terminal can be selected for either input or output by the data direction register. More specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B, C or D reads latched data if it has been programmed as output, even with the output level being fluctuated by the output load. (See Fig. 12.)

When reset, the data direction register goes "0" and all the input/output terminals are used as input.



Bit of data direction register	Bit of output data	Status of output	Input to CPU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

Figure 12 Input/Output Port (Ports A, B, C and D) Diagram

All input/output terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite that they are not used.

■ RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 13.) On power up, the reset input must be held "Low" for at least t_{OSC} to assure that the internal oscillator is stabilized. A sufficient time of delay can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 14.

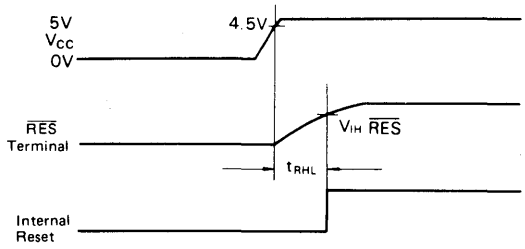


Figure 13 Power On and Reset Timing

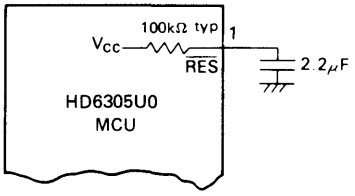


Figure 14 Input Reset Delay Circuit

■ INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0 MHz) or ceramic oscillator between pins 38 and 39 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 15. Figs. 16 and 17 illustrate the specifications and typical arrangement of the crystal, respectively.

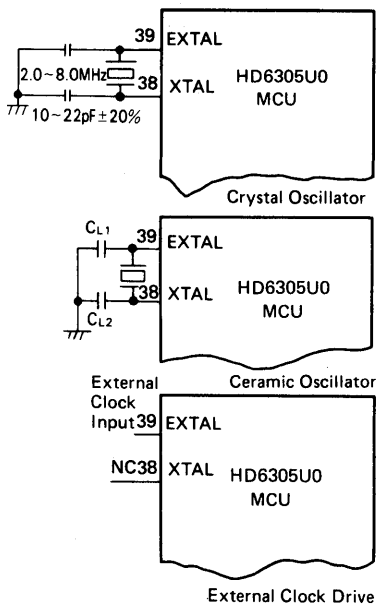


Figure 15 Internal Oscillator Circuit

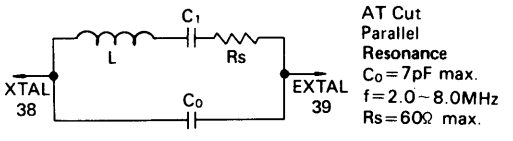
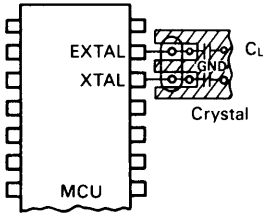


Figure 16 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 17 Typical Crystal Arrangement

■ LOW POWER DISSIPATION MODE

The HD6305U0 has three low power dissipation modes: wait, stop and standby.

● Wait Mode

When WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers (except the I bit of the condition code register which is cleared), RAM and I/O terminals hold their condition just before entering into the wait mode.

The escape from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$) is masked by the timer control register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 18 shows a flowchart for the wait function.

● Stop Mode

When STOP instruction being executed, MCU enters into the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive, but the RAM, registers (except bits 6 and 7 of the timer control register and the I bit of the condition code register) and I/O terminals hold their condition just before entering the stop mode. Bits 6 and 7 of the timer control register are initialized to "1" and "0", bits 7, 6, 5 and 4 of SCI status register are initialized "0", "0", "1", "1", respectively, and the I bit of the condition code register is cleared.

The escape from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings into the standby mode.

When interrupt is requested to the CPU and accepted, the stop mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 19 shows a flowchart for the stop function. Fig. 20 shows a timing chart of return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$ = "0" must exceed t_{osc} to assure stabilized oscillation.

● Standby Mode

The MCU enters into the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are hold. The I/O terminals turn to high-impedance state. The standby mode should escape by bringing $\overline{STBY}$ "High". The CPU must be restarted by reset. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 21.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 22.

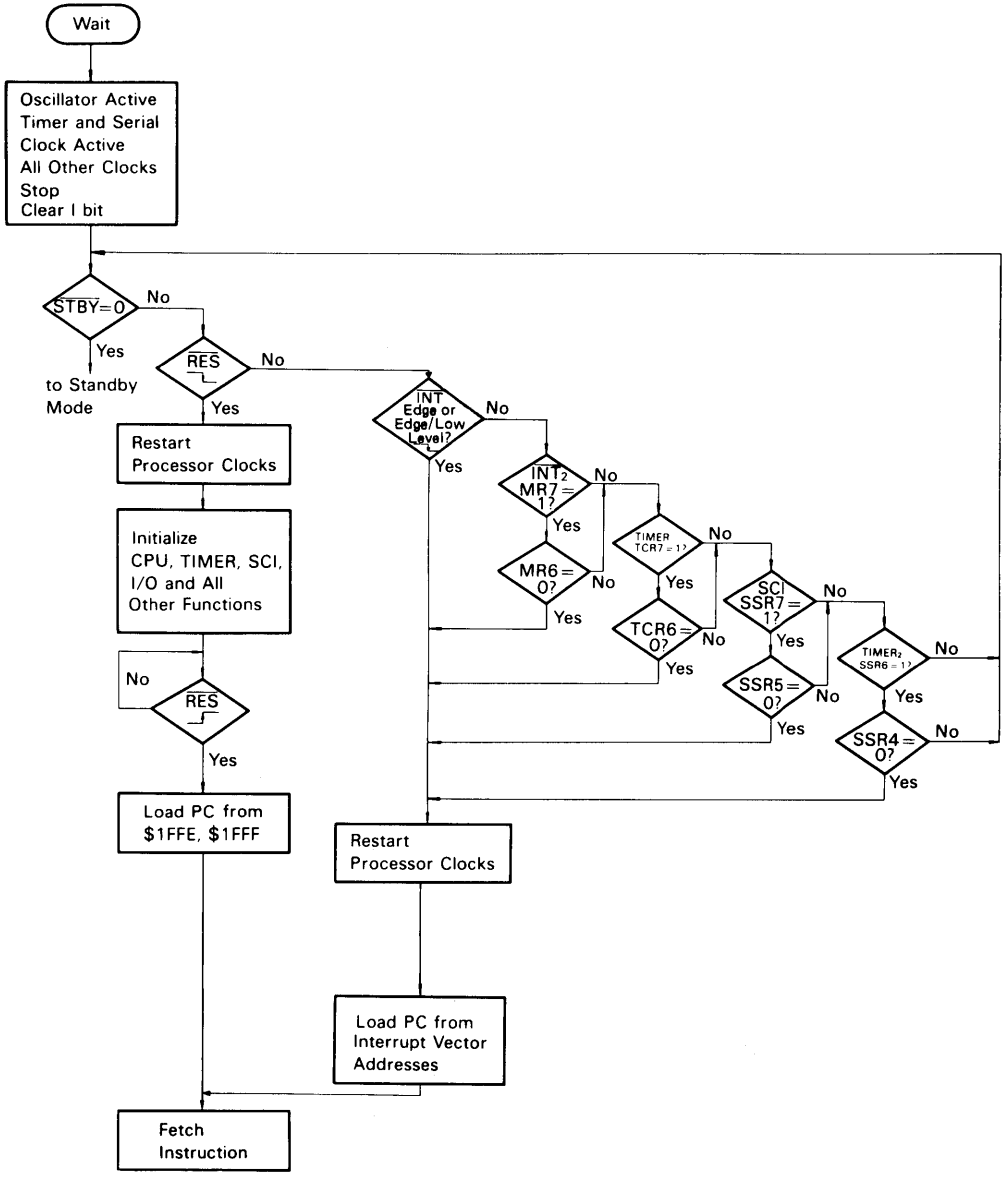


Figure 18 Wait Mode Flowchart

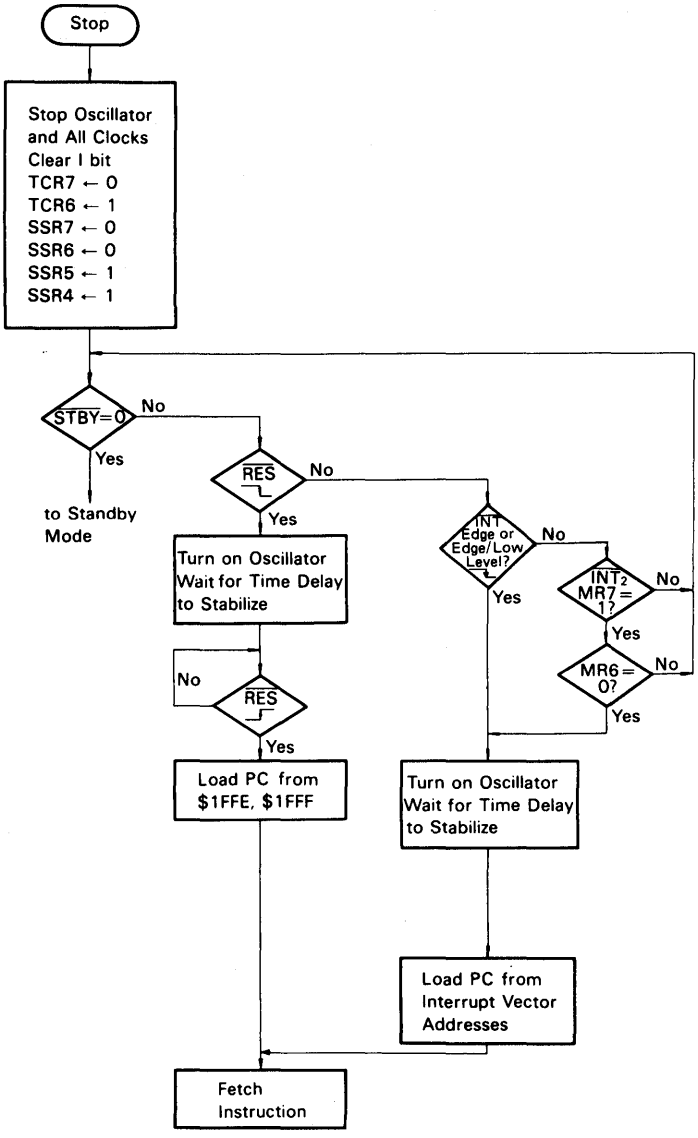
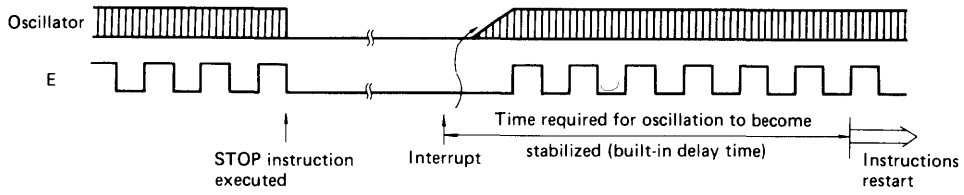
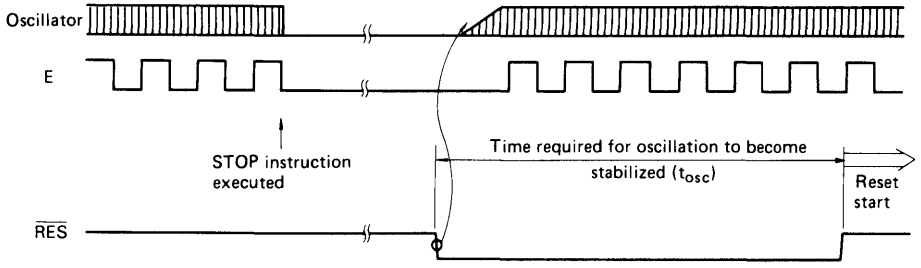


Figure 19 Stop Mode Flowchart



(a) Restart by Interrupt



(b) Restart by Reset

Figure 20 Timing Chart of Releasing from Stop Mode

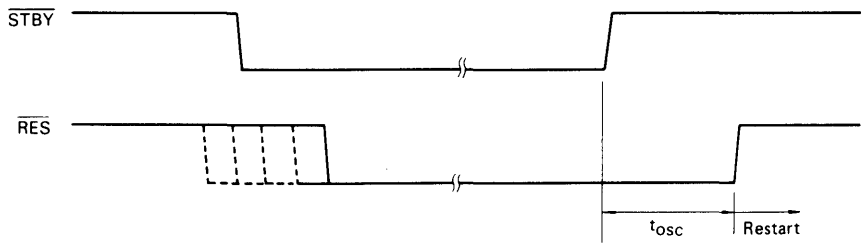


Figure 21 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register*	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Keep	High impedance	$\overline{STBY}$="High"

* Register in the CPU (except I bit in the CCR)

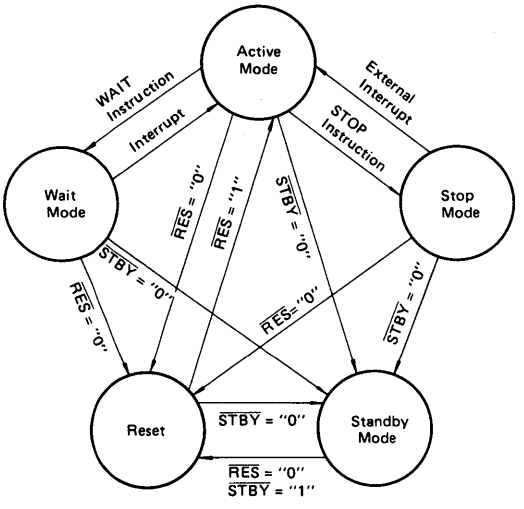


Figure 22 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■ BIT MANIPULATION

The HD6305U0 MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM or an I/O port. Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM, or I/O can be manipulated, the user may use a bit within the RAM as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 23 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the ROM. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
        :
SELF 1. BRCLR 0, PORT A, SELF 1
        BSET 1, PORT A
        BCLR 1, PORT A
        :
```

Figure 23 Example of Bit Manipulation

■ ADDRESSING MODES

Ten different addressing modes are available to the HD6305U0 MCU.

• Immediate

See Fig. 24. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 25. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. All RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 26. the extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 27. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 28. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 29. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 30. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 31. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 32. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The result of the test is written in the carry bit of the condition code register. (Set if true, cleared otherwise.)

• Implied

See Fig. 33. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

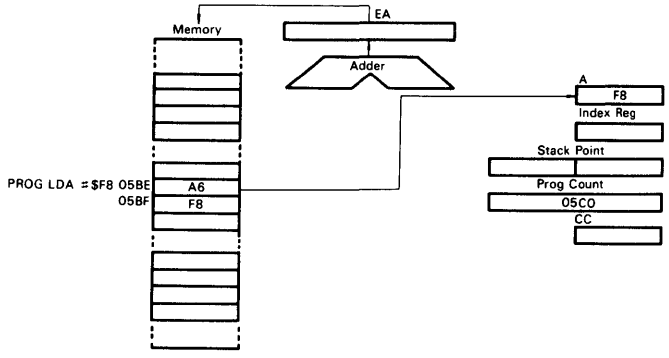


Figure 24 Example of Immediate Addressing

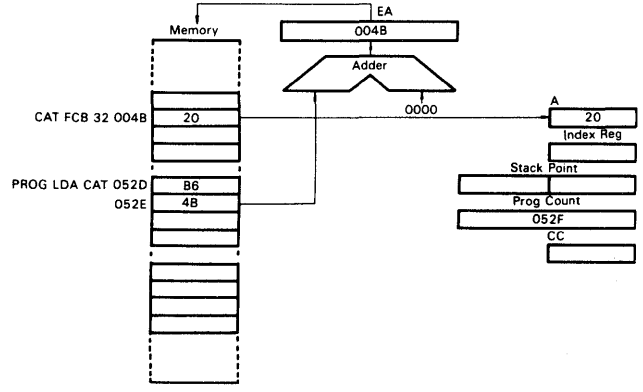


Figure 25 Example of Direct Addressing

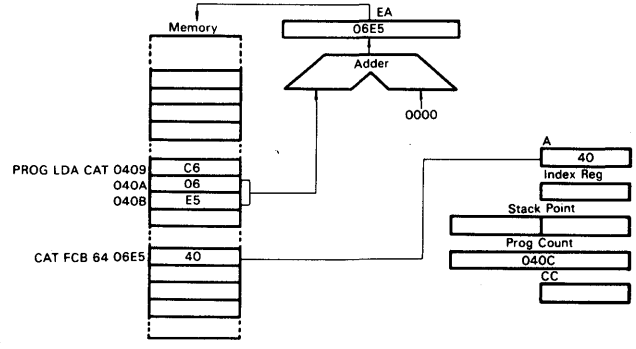


Figure 26 Example of Extended Addressing

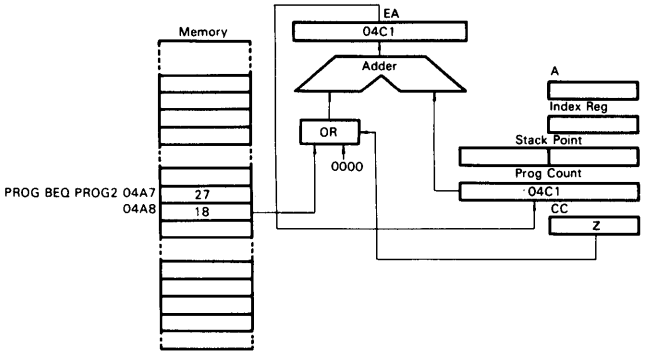


Figure 27 Example of Relative Addressing

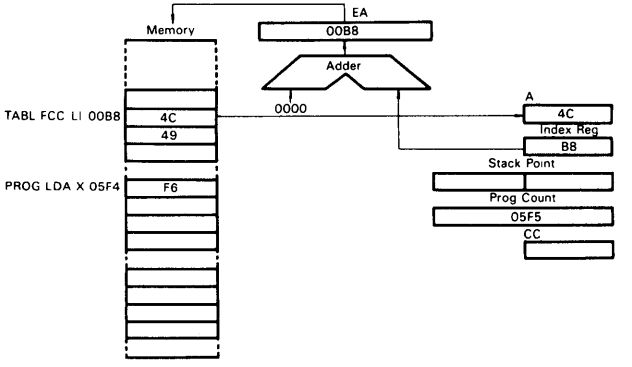


Figure 28 Example of Indexed (No Offset) Addressing

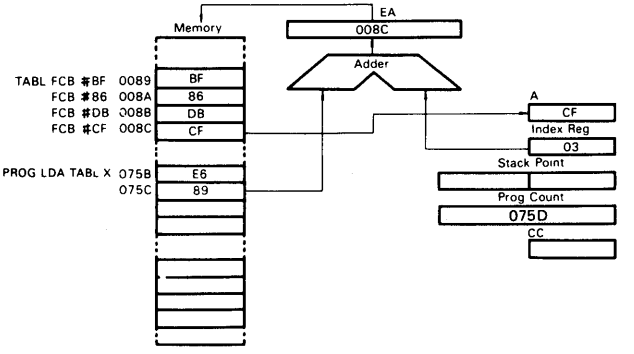


Figure 29 Example of Indexed (8-bit Offset) Addressing

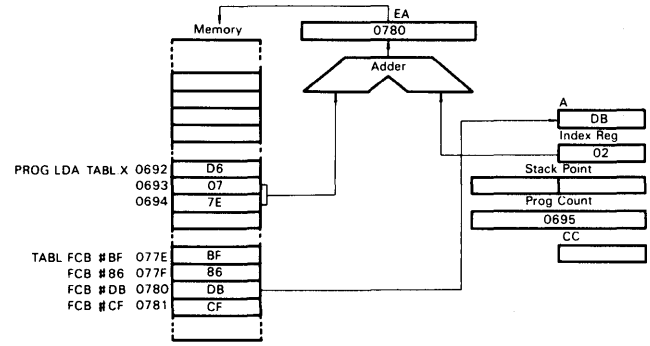


Figure 30 Example of Indexed (16-bit Offset) Addressing

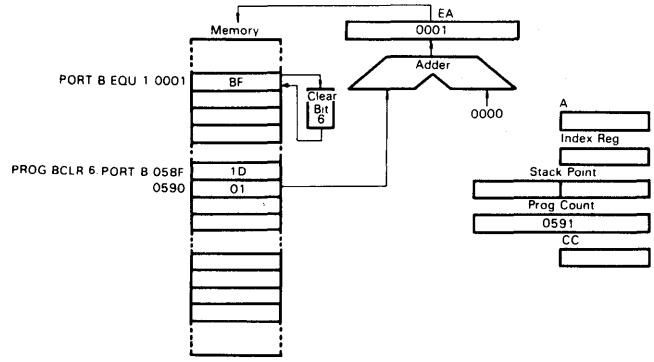


Figure 31 Example of Bit Set/Clear Addressing

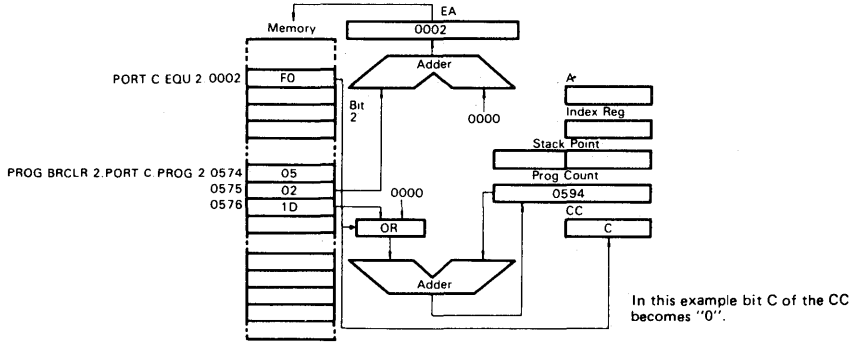


Figure 32 Example of Bit Test and Branch Addressing

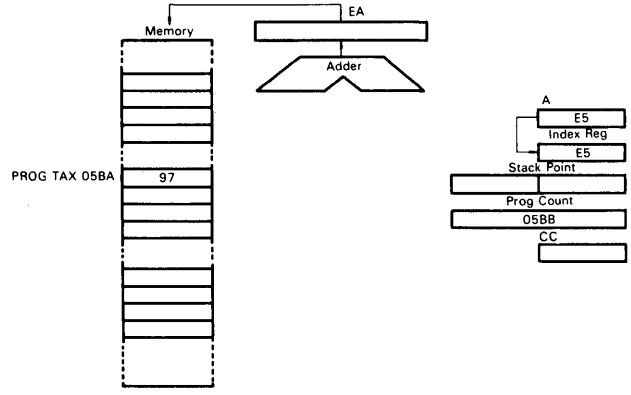


Figure 33 Example of Implied Addressing

■ INSTRUCTION SET

There are 62 basic instructions available to the HD6305U0 MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

● Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the HD6305U0 MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

● Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

● Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

● Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

● Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

● List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the HD6305U0 MCU in the alphabetical order.

● Operation Code Map

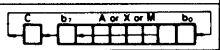
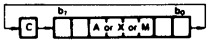
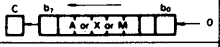
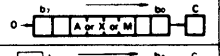
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)			Boolean/ Arithmetic Operation	Condition Code				
		OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~		H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	∧	∧	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	∧	∧	•
Store A in Memory	STA	—	—	—	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	∧	∧	•
Store X in Memory	STX	—	—	—	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	∧	∧	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	∧	•	∧	∧	∧
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	∧	•	∧	∧	∧
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	∧	∧	∧
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	∧	∧	∧
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	∧	∧	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	∧	∧	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A+M→A	•	•	∧	∧	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	∧	∧	∧
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	∧	∧	∧
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	∧	∧	•
Jump Unconditional	JMP	—	—	—	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR	—	—	—	BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A)			Implied(X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)			Boolean/Arithmetic Operation	Condition Code				
		OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~		H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	∧	∧	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	∧	∧	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	∧	∧	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	∧	∧	∧
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	∧	∧	∧
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	∧	∧	∧
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	∧	∧	∧
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	∧	∧
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	∧	∧	∧
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	∧	∧	∧
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	∧	∧	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	●	●	●	●	●
Branch Never	BRN	21	2	3	None	●	●	●	●	●
Branch IF Higher	BHI	22	2	3	C+Z=0	●	●	●	●	●
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	●	●	●	●	●
Branch IF Carry Clear	BCC	24	2	3	C=0	●	●	●	●	●
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	●	●	●	●	●
Branch IF Carry Set	BCS	25	2	3	C=1	●	●	●	●	●
(Branch IF Lower)	(BLO)	25	2	3	C=1	●	●	●	●	●
Branch IF Not Equal	BNE	26	2	3	Z=0	●	●	●	●	●
Branch IF Equal	BEQ	27	2	3	Z=1	●	●	●	●	●
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	●	●	●	●	●
Branch IF Half Carry Set	BHCS	29	2	3	H=1	●	●	●	●	●
Branch IF Plus	BPL	2A	2	3	N=0	●	●	●	●	●
Branch IF Minus	BMI	2B	2	3	N=1	●	●	●	●	●
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	●	●	●	●	●
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	●	●	●	●	●
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	●	●	●	●	●
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	●	●	●	●	●
Branch to Subroutine	BSR	AD	2	5	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	●	●	●	●	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	●	●	●	●	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	●	●	●	●	●
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes Implied OP	#	~	Boolean Operation	Condition Code H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	●	●	●	●	●
Transfer X to A	TXA	9F	1	2	X→A	●	●	●	●	●
Set Carry Bit	SEC	99	1	1	1→C	●	●	●	●	1
Clear Carry Bit	CLC	98	1	1	0→C	●	●	●	●	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	●	1	●	●	●
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	●	0	●	●	●
Software Interrupt	SWI	83	1	10		●	1	●	●	●
Return from Subroutine	RTS	81	1	5		●	●	●	●	●
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	●	●	●	●	●
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	●	●	●	●	●
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	●	●	^	^	^*
Stop	STOP	8E	1	4		●	0	●	●	●
Wait	WAIT	8F	1	4		●	0	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	Condition Code H	I	N	Z	C
ADC		×	×	×		×	×	×			^	●	^	^	^
ADD		×	×	×		×	×	×			^	●	^	^	^
AND		×	×	×		×	×	×			●	●	^	^	●
ASL	×		×			×	×				●	●	^	^	^
ASR	×		×			×	×				●	●	^	^	^
BCC					×						●	●	●	●	●
BCLR									×		●	●	●	●	●
BCS					×						●	●	●	●	●
BEQ					×						●	●	●	●	●
BHCC					×						●	●	●	●	●
BHCS					×						●	●	●	●	●
BHI					×						●	●	●	●	●
(BHS)					×						●	●	●	●	●
BIH					×						●	●	●	●	●
BIL					×						●	●	●	●	●
BIT		×	×	×		×	×	×			●	●	^	^	●
(BLO)					×						●	●	●	●	●
BLS					×						●	●	●	●	●
BMC					×						●	●	●	●	●
BMI					×						●	●	●	●	●
BMS					×						●	●	●	●	●
BNE					×						●	●	●	●	●
BPL					×						●	●	●	●	●
BRA					×						●	●	●	●	●

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						•	•	•	•	•
BRCLR										×	•	•	•	•	∧
BRSET										×	•	•	•	•	∧
BSET									×		•	•	•	•	•
BSR					×						•	•	•	•	•
CLC	×										•	•	•	•	0
CLI	×										•	0	•	•	•
CLR	×		×			×	×				•	•	0	1	•
CMP		×	×	×		×	×	×			•	•	∧	∧	∧
COM	×		×			×	×				•	•	∧	∧	1
CPX		×	×	×		×	×	×			•	•	∧	∧	∧
DAA	×										•	•	∧	∧	∧
DEC	×		×			×	×				•	•	∧	∧	•
EOR		×	×	×		×	×	×			•	•	∧	∧	•
INC	×		×			×	×				•	•	∧	∧	•
JMP			×	×		×	×	×			•	•	•	•	•
JSR			×	×		×	×	×			•	•	•	•	•
LDA		×	×	×		×	×	×			•	•	∧	∧	•
LDX		×	×	×		×	×	×			•	•	∧	∧	•
LSL	×		×			×	×				•	•	∧	∧	∧
LSR	×		×			×	×				•	•	0	∧	∧
NEG	×		×			×	×				•	•	∧	∧	∧
NOP	×										•	•	•	•	•
ORA		×	×	×		×	×	×			•	•	∧	∧	•
ROL	×		×			×	×				•	•	∧	∧	∧
ROR	×		×			×	×				•	•	∧	∧	∧
RSP	×										•	•	•	•	•
RTI	×										?	?	?	?	?
RTS	×										•	•	•	•	•
SBC		×	×	×		×	×	×			•	•	∧	∧	∧
SEC	×										•	•	•	•	1
SEI	×										•	1	•	•	•
STA			×	×		×	×	×			•	•	∧	∧	•
STOP	×										•	0	•	•	•
STX			×	×		×	×	×			•	•	∧	∧	•
SUB		×	×	×		×	×	×			•	•	∧	∧	∧
SWI	×										•	1	•	•	•
TAX	×										•	•	•	•	•
TST	×		×			×	×				•	•	∧	∧	•
TXA	×										•	•	•	•	•
WAIT	×										•	0	•	•	•

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI	—					—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

LOW

(NOTES) 1. "—" is an undefined operation code.
2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD6305U0:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

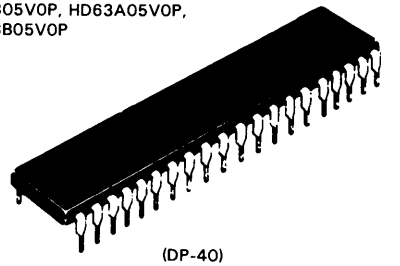
HD6305V0, HD63A05V0, HD63B05V0
CMOS MCU (Microcomputer Unit)

—PRELIMINARY—

The HD6305V0 is a CMOS 8-bit single-chip MCU which is similar to the HD6305X MCU family. This version is upward compatible with the HD6805 family in respect to instructions. A CPU, a clock generator, a 4k-byte ROM, a 192-byte RAM, 31 I/O terminals, two timers, and a serial communication interface (SCI) are incorporated in the HD6305V0. As a result of CMOS technology, the HD6305V0 consumes much less power than NMOS counterparts. In addition, three low power dissipation modes (stop, wait and standby) which further decreases power consumption, are included in the HD6305V0.

Other notable features include enhanced instruction cycle of the main instructions and the use of three additional instructions to improve system throughput.

HD6305V0P, HD63A05V0P, HD63B05V0P



(DP-40)

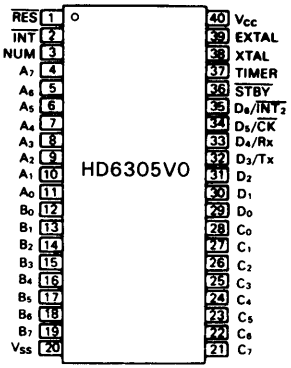
■ HARDWARE FEATURES

- CMOS 8-bit single-chip MCU
- 4096 bytes of ROM
- 192 bytes of RAM
- 31 bidirectional I/O terminals
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes
  - Wait..... In this mode, the clock oscillator is on and the CPU halts but the timer/serial/interrupt function is operatable. Also, all registers are held, except the I bit in the condition code register is cleared.
  - Stop..... In this mode, the clock stops but the RAM data, I/O status and registers are held. Except the timer control register (bits 6 and 7) and the I bit of the condition code register.
  - Standby.... In this mode, the clock stops, the RAM data is held, and the other internal condition is reset.
- Minimum instruction cycle time
  - HD6305V0..... 1 μs (f = 1 MHz)
  - HD63A05V0 ..... 0.67 μs (f = 1.5 MHz)
  - HD63B05V0..... 0.5 μs (f = 2 MHz)
- Wide operating range
  V_{CC} = 3 to 6V (f = 0.1 to 0.5 MHz)
  - HD6305V0..... f = 0.1 to 1 MHz (V_{CC} = 5V ± 10%)
  - HD63A05V0 ..... f = 0.1 to 1.5 MHz (V_{CC} = 5V ± 10%)
  - HD63B05V0..... f = 0.1 to 2 MHz (V_{CC} = 5V ± 10%)
- System development fully supported by an emulator

■ SOFTWARE FEATURES

- Similar to HD6800
- Byte efficient instruction set
- Powerful bit manipulation instructions (Bit Set, Bit Clear, and Bit Test and Branch usable for all RAM bits and all I/O terminals)

■ PIN ARRANGEMENT

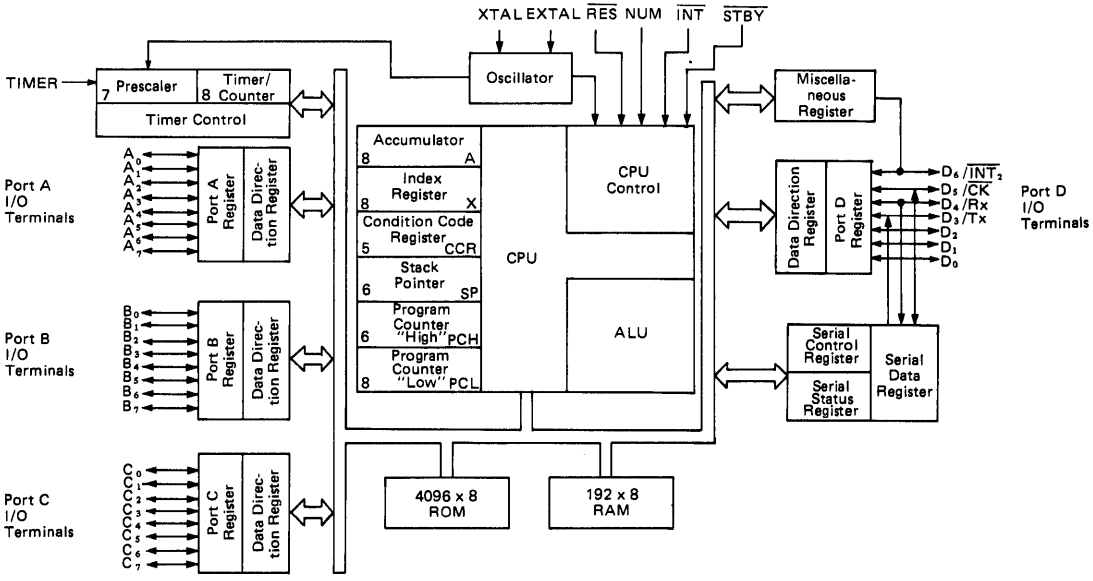


(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

- A variety of interrupt operations
- Index addressing mode useful for table processing
- A variety of conditional branch instructions
- Ten powerful addressing modes
- All addressing modes adaptable to RAM, and I/O instructions
- Three new instructions, Stop, Wait and DAA, added to the HD6805 family instruction set
- Instructions that are upward compatible with those of Motorola's MC6805P2 and MC146805G2
- Compatible instruction set with HD6305X

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 ~ +7.0	V
Input voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating temperature	T_{opr}	0 ~ +70	°C
Storage temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ELECTRICAL CHARACTERISTICS

• DC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item		Symbol	Test condition	min	typ	max	Unit
Input voltage "High"	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC} - 0.5	–	V_{CC} + 0.3	V
	EXTAL			V_{CC} x 0.7	–	V_{CC} + 0.3	V
	Others			2.0	–	V_{CC} + 0.3	V
Input voltage "Low"	All Inputs	V_{IL}		-0.3	–	0.8	V
Current * dissipation	Operating	I_{CC}	f = 1MHz**	–	5	10	mA
	Wait			–	2	5	mA
	Stop			–	2	10	μA
	Standby			–	2	10	μA
Input leakage current	$\overline{TIMER}$, $\overline{INT}$, $\overline{STBY}$	$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} - 0.5V	–	–	1	μA
Three-state current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $D_0 \sim D_6$	$\|I_{TSI}\|$		–	–	1	μA
Input capacity	All terminals	C_{in}	f = 1MHz, V_{in} = 0V	–	–	12	pF

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8 V

**The value at f = xMHz can be calculated by the following equation: I_{CC} (f = xMHz) = I_{CC} (f = 1MHz) multiplied by x

• AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item	Symbol	Test condition	HD6305V0			HD63A05V0			HD63B05V0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock frequency	f_{cl}		0.4	–	4	0.4	–	6	0.4	–	8	MHz
Cycle time	t_{cyc}		1.0	–	10	0.666	–	10	0.5	–	10	μs
$\overline{INT}$ pulse width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ pulse width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ pulse width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
TIMER pulse width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation start time (crystal)	t_{OSC}	C_L = 22pF ± 20% R_s = 60Ω max	–	–	20	–	–	20	–	–	20	ms
Reset delay time	t_{RHL}	External cap. 2.2μF	80	–	–	80	–	–	80	–	–	ms

● **Port Electrical Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)**

Item		Symbol	Test condition	min	typ	max	Unit
Output voltage "High"	Ports A, B, C, D	V_{OH}	I_{OH} = −200μA	2.4	–	–	V
			I_{OH} = −10μA	V_{CC} − 0.7	–	–	V
Output voltage "Low"		V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input voltage "High"	Ports A, B, C, D	V_{IH}		2.0	–	V_{CC} + 0.3	V
Input voltage "Low"		V_{IL}		- 0.3	–	0.8	V
Input leakage current		$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} − 0.5V	–	–	1	μA

● **SCI Timing (V_{CC} = 5.0V±10%, V_{SS} = GND and Ta = 0 ~ +70°C unless otherwise specified)**

Item	Symbol	Test Condition	HD6305V0			HD63A05V0			HD63B05V0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock Cycle	t_{Scyc}	Fig. 1 Fig. 2	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data Output Delay Time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data Set-up Time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data Hold Time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

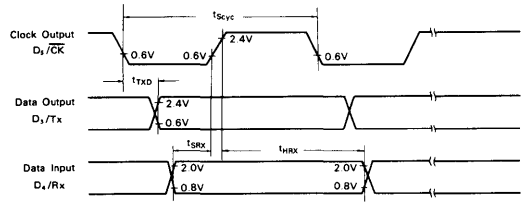


Figure 1 SCI Timing (Internal Clock)

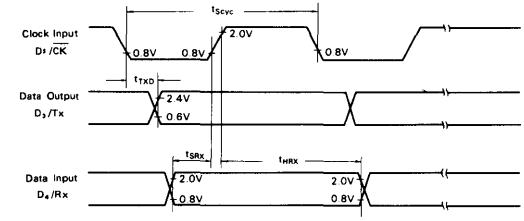
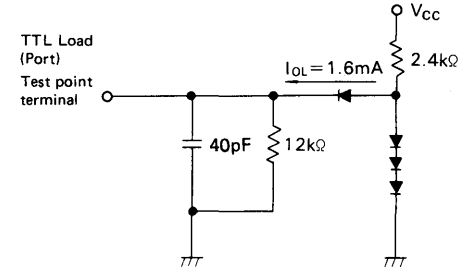


Figure 2 SCI Timing (External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ.

Figure 3 Test Load

■ DESCRIPTION OF TERMINAL FUNCTIONS

The input and output signals of the HD6305V0 are described here.

● **V_{CC}, V_{SS}**

Voltage is applied to the HD6305V0 through these two terminals. V_{CC} is 5.0V ± 10%, while V_{SS} is grounded.

● **$\overline{INT_1}$, $\overline{INT_2}$**

External interrupt request inputs to the HD6305V0. For details, refer to "INTERRUPTS". The $\overline{INT_2}$ terminal is also used as the port D_6 terminal.

- **XTAL, EXTAL**
  These terminals provide input to the on-chip clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic filter is connected to the terminal. Refer to "INTERNAL OSCILLATOR" for using these input terminals.

- **TIMER**
  This is an input terminal for event counter. Refer to "TIMER" for details.

- **$\overline{RES}$**
  Used to reset the MCU. Refer to "RESET" for details.

- **NUM**
  This terminal is not intended for user applications. It must be grounded to V_{SS}.

- **Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $D_0 \sim D_6$)**
  These 31 terminals consist of three 8-bit I/O ports (A, B and C) and a 7-bit I/O port D. Each of these can be used as an input or output terminal on a bit basis through program control of the data direction register (DDR). For details, refer to "I/O PORTS."
  Since port D_6 is also used for the $\overline{INT_2}$ input, in order to use port D_6 as an I/O port, the $\overline{INT_2}$ interrupt mask bit in the miscellaneous register should be set to "1" to disable the $\overline{INT_2}$ input.

- **$\overline{STBY}$**
  This terminal is used to place the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and the internal condition is reset. For details, refer to "STANDBY MODE."
  The terminals described in the following are I/O pins for serial communication interface (SCI). They are also used as ports D_3, D_4 and D_5. For details, refer to "SERIAL COMMUNICATION INTERFACE."

- **$\overline{CK}$ (D_5)**
  Used to input or output clocks for serial operation.

- **Rx (D_4)**
  Used to receive serial data.

- **Tx (D_3)**
  Used to transmit serial data.

■ MEMORY MAP

The memory map of the HD6305V0 MCU is shown in Fig. 4. During interrupt processing, the contents of the MCU registers are saved into the stack in the sequence shown in Fig. 5. This saving begins with the lower byte (PCL) of the program counter. Then the value of the stack pointer is decremented and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CCR) are stacked in that order. In a subroutine call, only the contents of the program counter (PCH and PCL) are stacked.

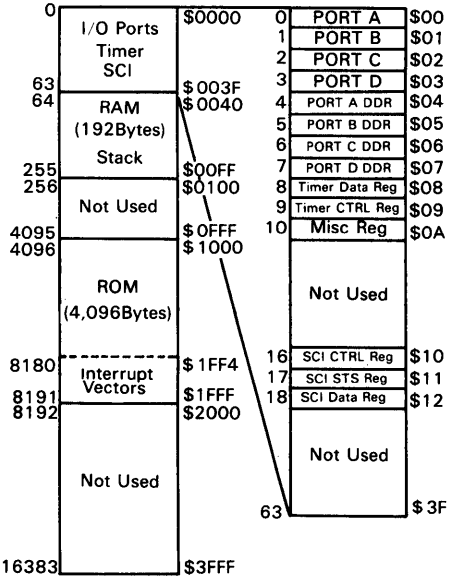
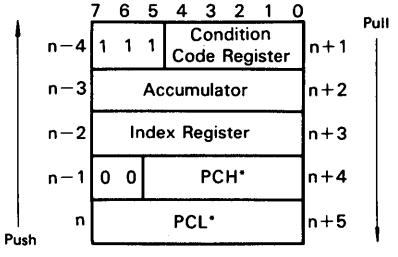


Figure 4 Memory Map of HD6305V0 MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 5 Sequence of Interrupt Stacking

■ REGISTERS

There are five registers which the programmer can operate.

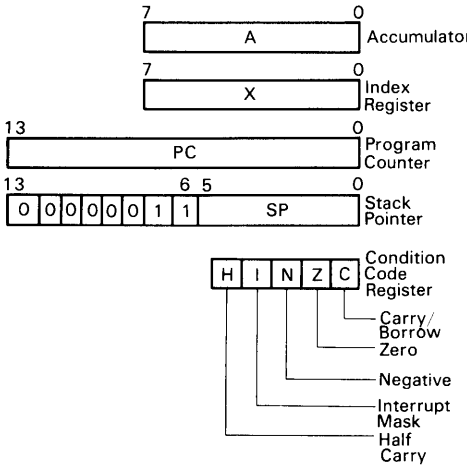


Figure 6 Programming Model

- **Accumulator (A)**

This accumulator is a general purpose 8-bit register which holds operands or the result of arithmetic operation or data processing.

- **Index Register (X)**

The index register is an 8-bit register, and is used for index addressing mode. Each of the addresses contained in the register consists of 8 bits which, combined with an offset value, provides an effective address.

In the case of a read/modify/write instruction, the index register can be used like an accumulator to hold operation data or the result of operation.

If not used in the index addressing mode, the register can be used to store data temporarily.

- **Program Counter (PC)**

The program counter is a 14-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is a 14-bit register that indicates the address of the next stacking space. Just after reset, the stack pointer is set at address $00FF. It is decremented when data is pushed, and incremented when pulled. The upper 8 bits of the stack pointer are fixed to 00000011. During the MCU being reset or during a reset stack pointer (RSP) instruction, the pointer is set to address $00FF. Since a subroutine or interrupt can use space up to address $00C1 for stacking, the subroutine can be used up to 31 levels and the interrupt up to 12 levels.

- **Condition Code Register (CCR)**

The condition code register is a 5-bit register, each bit indicating the result of the instruction just executed. The bits can be individually tested by conditional branch instructions. The CCR bits are as follows:

Half Carry (H): Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts, except a software interrupt, to be masked. If an interrupt occurs with the bit I set, it is latched. It will be processed the instant the interrupt mask bit is reset. (More specifically, it will enter the interrupt processing routine after the instruction following the CLI has been executed.)

Negative (N): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (bit 7 is logic "1").

Zero (Z): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is zero.

Carry/ Borrow (C): Represents a carry or borrow that occurred in the most recent arithmetic operation. This bit is also affected by the Bit Test and Branch instruction, a shift instruction and a Rotate instruction.

■ INTERRUPT

There are six different types of interrupt: external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER or the SCI and $TIMER_2$ generate the same vector address, respectively. Although, a different vector address is generated for a timer interrupt during the wait mode, as shown in Table 1.

When an interrupt occurs, the program in progress stops and then the CPU status is saved onto the stack. And then, the interrupt mask bit (I) of the condition code register is set and the start address of the interrupt processing routine is obtained from a particular interrupt vector address. Then the interrupt routine starts from the start address. System can exit from the interrupt routine by a RTI instruction. When this instruction is executed, the CPU status before the interrupt (saved onto the stack) is pulled and the CPU restarts the sequence with the instruction next to the one at which the interrupt occurred. Table 1 lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Priority	Interrupt	Vector Address	
1	$\overline{RES}$	$1FFE,	$1FFF
2	SWI	$1FFC,	$1FFD
3	$\overline{INT}$	$1FFA,	$1FFB
4	TIMER/$\overline{INT_2}$	$1FF8,	$1FF9
5	TIMER (WAIT)	$1FF6,	$1FF7
6	SCI/$TIMER_2$	$1FF4,	$1FF5

A flowchart of the interrupt sequence is shown in Fig. 7. A block diagram of the interrupt request source is shown in Fig. 8.

In the block diagram of Fig. 8, the external interrupt $\overline{INT_2}$ is a falling edge trigger input, whereas, the external interrupt $\overline{INT}$ can be configured as a falling edge trigger input or a combination of falling edge and low level trigger input, depending on the status of bit 5 in the miscellaneous register (MR). When an interrupt request is detected at the $\overline{INT_2}$ or $\overline{INT}$ inputs, an interrupt request is generated and latched. The $\overline{INT}$ interrupt request is automatically cleared if jumping is made to the $\overline{INT}$ processing routine. Meanwhile, the $\overline{INT_2}$ request is cleared if "0" is written in bit 7 of the miscellaneous register.

For the external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$) and serial interrupt (SCI), each interrupt request is held, but not processed, if the I bit of the condition code register is set. Immediately after the I bit is cleared, the corresponding interrupt processing starts according to the priority.

The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by setting bit 6 of the timer control register; the SCI interrupt by setting bit 5 of the serial status register; and the $TIMER_2$ interrupt by setting bit 4 of the serial status register.

The status of the $\overline{INT}$ terminal can be tested by a BIL or BIH instruction. The $\overline{INT}$ falling edge and falling edge/low level detec-

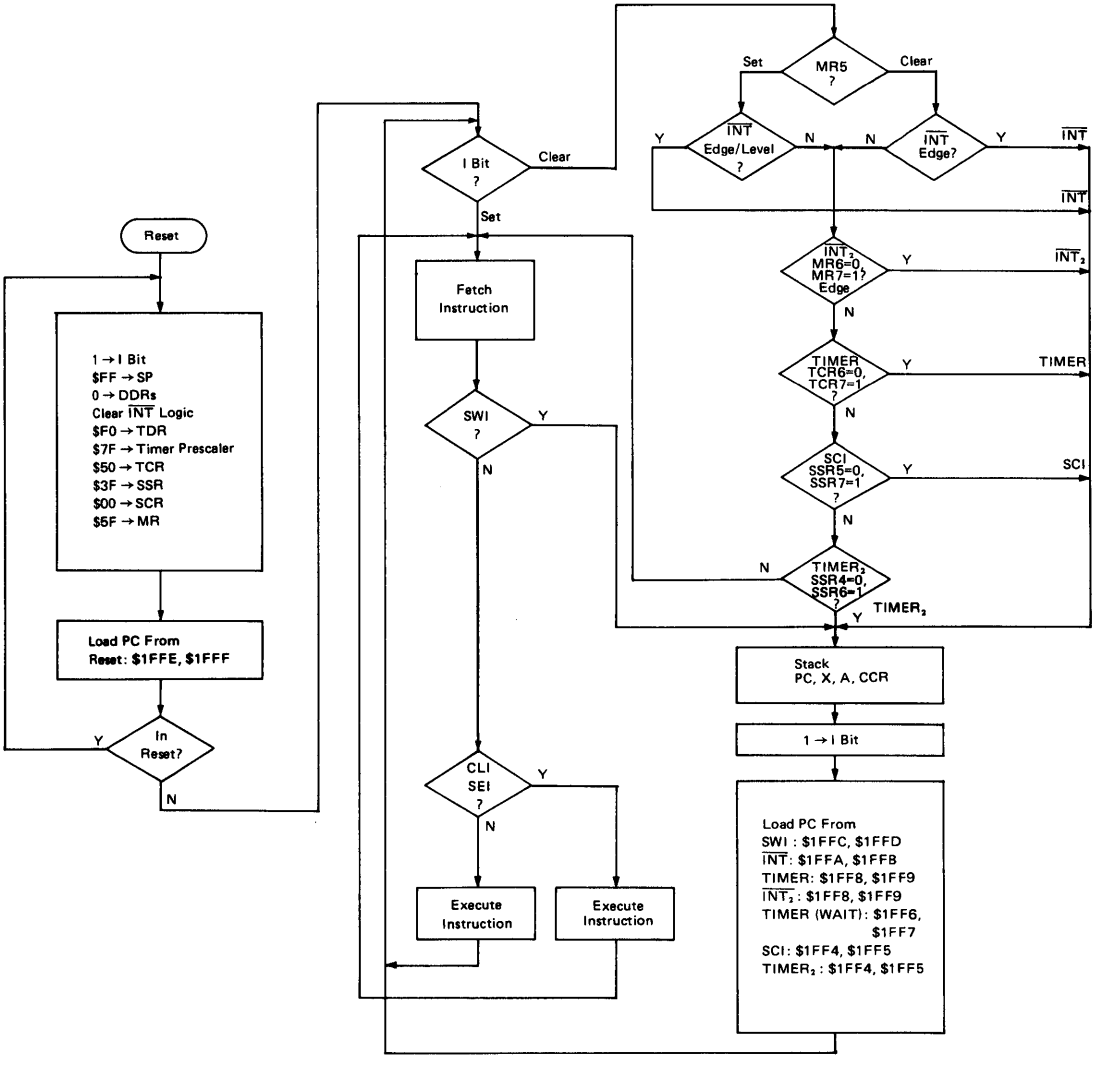


Figure 7 Interrupt Flowchart

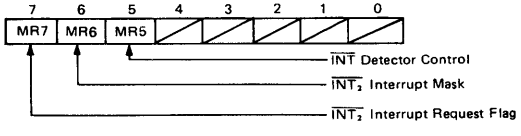
tor circuit and its latching circuit are independent of testing by these instructions. This is also true with the status of the $\overline{INT_2}$ terminal.

- **Miscellaneous Register (MR; $000A)**

The external interrupt $\overline{INT_2}$ and the TIMER interrupt have identical interrupt vector addresses, as shown in Table 1. For this reason, bits 6 and 7 of a special register called the miscellaneous register (MR: $000A) are available to control the $\overline{INT_2}$ interrupt. Moreover, bit 5 of the MR controls the sensing mode for the $\overline{INT}$ interrupt detector (falling edge detector or falling edge/low level detector).

Bit 7 of the MR is the $\overline{INT_2}$ interrupt request flag. When a falling edge is detected at the $\overline{INT_2}$ terminal, bit 7 is set to "1" Then the interrupt routine software (vector addresses: $1FF8, $1FF9) checks bit 7 to see if an $\overline{INT_2}$ interrupt occurred. Bit 7 can be reset by software.

Miscellaneous Register (MR; $000A)



Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If this bit is set to "1", then the $\overline{INT_2}$ interrupt is disabled. Both read and write are possible with bit 7, but "1" cannot be written in this bit by software. In other words, an $\overline{INT_2}$ interrupt request by software is not possible.

Bit 5 is the control bit for $\overline{INT}$ interrupt detection. If this bit is reset to "0", the detection logic will detect a falling edge. When this bit is set to "1", the detection logic will detect a falling edge or a low level.

When reset, bit 7 is cleared to "0", bit 6 is set to "1" and bit 5 is cleared to "0".

■ TIMER

Figure 9 shows an MCU timer block diagram. The timer data register is loaded by software and, upon receipt of a clock input, begins to count down. When the timer data register (TDR) becomes "0", the timer interrupt request bit (bit 7) in the timer control register is set. In response to the interrupt request, the MCU saves its status into the stack and fetches the timer interrupt routine addresses $1FF8 and $1FF9 (or $1FF6 and $1FF7 when the timer interrupt occures during the wait mode) and executes the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also mask the timer interrupt.

The source clock to the timer can be either an external signal from the timer input terminal or the internal E signal (the oscillator clock divided by 4). If the E signal is used as the source, the clock input can be gated by the input to the timer input terminal.

Once the timer count has reached 0, it starts counting down with "FF". The count can be monitored whenever desired by reading the timer data register. This permits the program to know the length of time having passed after the occurrence of a timer interrupt, without disturbing the contents of the counter.

When the MCU is reset or placed in the stop mode, the timer data register (TDR) is initialized to $F0. The timer interrupt request bit (bit 7) then is cleared and the timer interrupt mask bit (bit 6) is set.

To clear the timer interrupt request bit (bit 7), it is necessary to write "0" in that bit.

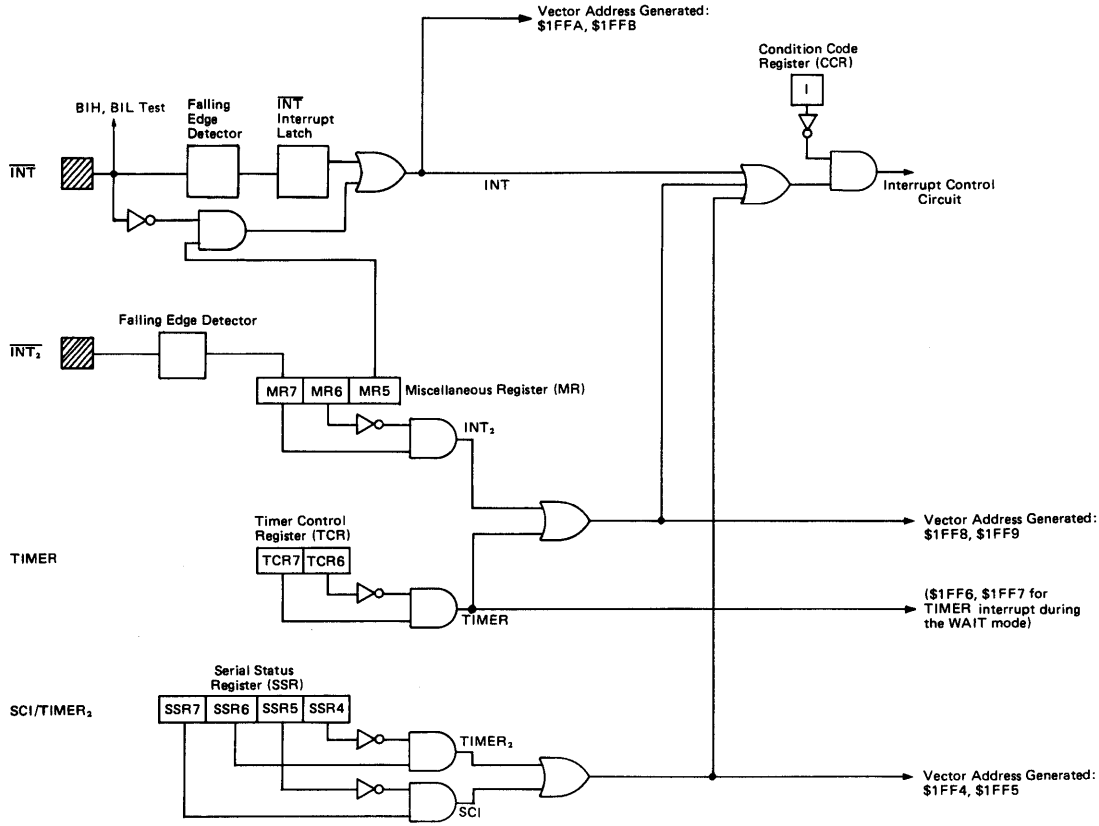


Figure 8 Interrupt Request Generation Circuitry

TCR7	Timer interrupt request
0	Absent
1	Present

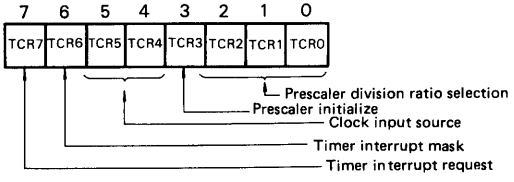
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

- **Timer Control Register (TCR; $0009)**

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After reset, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$F0" immediately after reset.

When "1" is written in bit 3, the prescaler is initialized to "$7F". This bit always shows "0" when read.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under TIMER terminal control
1	0	No clock input (counting stopped)
1	1	Event input from TIMER terminal

A prescaler division ratio is selected by the combination of three bits (bits 0, 1 and 2) of the timer control register (see Table 3). There are eight different division ratios: ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128. After reset, the TCR is set to the ÷1 mode.

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

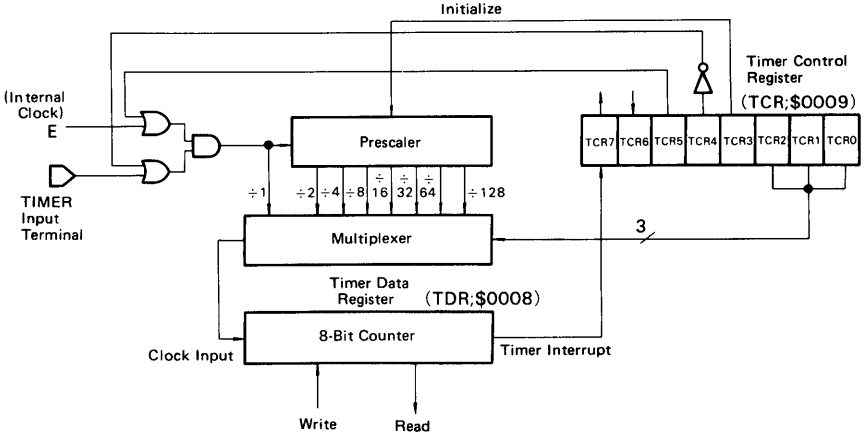


Figure 9 Timer Block Diagram

A timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. This bit can be cleared by writing "0" in that bit.

■ SERIAL COMMUNICATION INTERFACE (SCI)

This interface is used for serial transmission or reception of 8-bit data. Sixteen transfer rates are available in the range from 1 μs to approx. 32 ms (for oscillation at 4 MHz).

The SCI consists of three registers, one eighth counter and one prescaler. (See Fig. 10.) SCI communicates with the CPU via the data bus, and with the outside world through bits 3, 4 and 5 of port D. Described below are the operations of each register and data transfer.

● SCI Control Register (SCR; $0010)

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

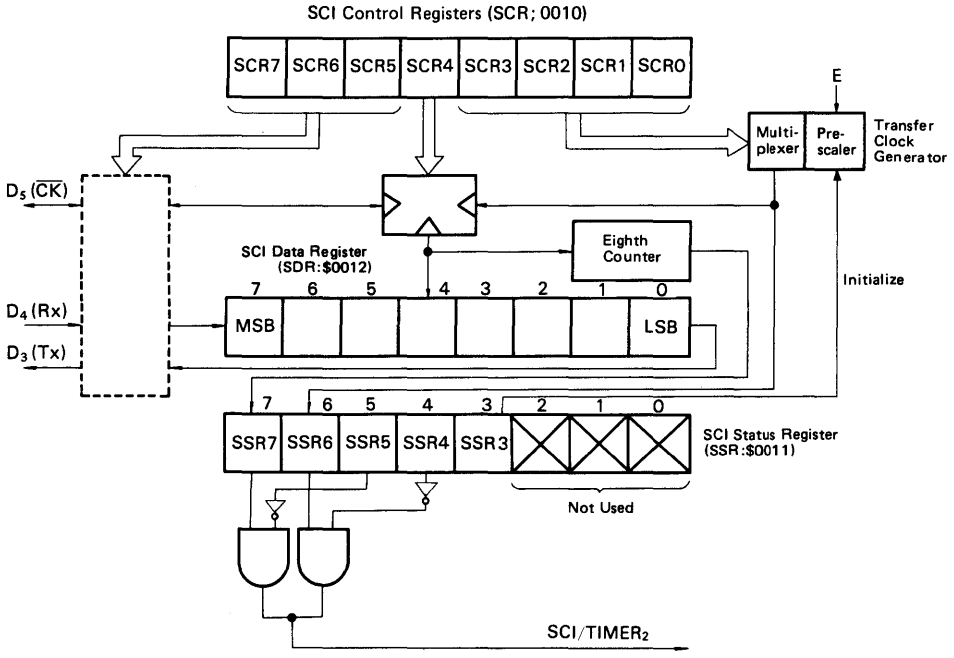


Figure 10 SCI Block Diagram

SCR7	D_3 terminal
0	Used as I/O terminal (by DDR)
1	Serial data output (DDR output)

SCR6	D_4 terminal
0	Used as I/O terminal (by DDR)
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	D_5 terminal
0	0	–	Used as I/O terminal (by DDR)
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)

When this bit is set, the DDR corresponding to the D_3 becomes "1" and this terminal serves for output of SCI data. After reset, the bit is cleared to "0".

Bit 6 (SCR6)

When this bit is set, the DDR corresponding to the D_4 becomes "0" and this terminal serves for input of SCI data. After reset, the bit is cleared to "0".

Bits 5 and 4 (SCR5, SCR4)

These bits are used to select a clock source. After reset, the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)

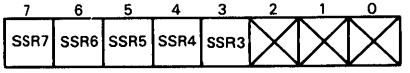
These bits are used to select a transfer clock rate. After reset, the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

- **SCI Data Register (SDR; $0012)**

A serial-parallel conversion register that is used for transfer of data.

- **SCI Status Register (SSR; $0011)**



Bit 7 (SSR7)

Bit 7 is the SCI interrupt request bit which is set upon completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5 = "1". The bit can also be cleared by writing "0" in it.

Bit 6 (SSR6)

Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is used commonly with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When reset, the bit is cleared. It also be cleared by writing "0" in it. (For details, see $TIMER_2$.)

Bit 5 (SSR5)

Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When reset, it is set to "1".

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

Bit 4 (SSR4)

Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When reset, it is set to "1".

Bit 3 (SSR3)

When "1" is written in this bit, the prescaler of the transfer clock generator is initialized. When read, the bit always is "0".

Bits 2 ~ 0

Not used.

- **Data Transmission**

By writing the desired control bits into the SCI control registers, a transfer rate and a source of transfer clock are determined and bits 3 and 5 of port D are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the D_3/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock. (See Fig. 11.) When 8 bits of data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the D_3/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the $D_5/\overline{CK}$ terminal is set as input. If the internal clock has been selected, the $D_5/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

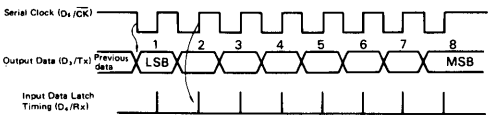


Figure 11 SCI Timing Chart

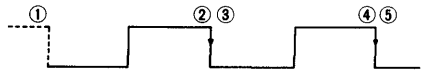
- **Data Reception**

By writing the desired control bits into the SCI control register, a transfer rate and a source of transfer clock are determined and bits 4 and 5 of Port D are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed for the second and subsequent data receptions. It must be taken only after resetting.)

The data from the D_4/Rx terminal is input to the SCI data register synchronously with the leading edge of the serial clock (see Fig. 11). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored and the data is received synchronously with the clock from the $D_5/\overline{CK}$ terminal. If the internal clock has been selected, the $D_5/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

- **$TIMER_2$**

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 ~ 0 of the SCI control register (4 μs ~ approx. 32 ms (for oscillation at 4 MHz)) is input to bit 6 of the SCI status register and the $TIMER_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, $TIMER_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : $TIMER_2$ interrupt request
③, ⑤ : $TIMER_2$ interrupt request bit cleared

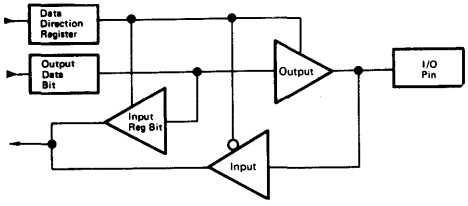
$TIMER_2$ is commonly used with the SCI transfer clock generator. If wanting to use $TIMER_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■ I/O PORTS

There are 31 input/output terminals (ports A, B, C, D). Each I/O terminal can be selected for either input or output by the data direction register. More specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B, C or D reads latched data if it has been programmed as output, even with the output level being fluctuated by the output load. (See Fig. 12.)

When reset, the data direction register goes "0" and all the input/output terminals are used as input.



Bit of data direction register	Bit of output data	Status of output	Input to CPU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

Figure 12 Input/Output Port (Ports A, B, C and D) Diagram

All input/output terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite that they are not used.

■ RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 13.) On power up, the reset input must be held "Low" for at least t_{OSC} to assure that the internal oscillator is stabilized. A sufficient time of delay can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 14.

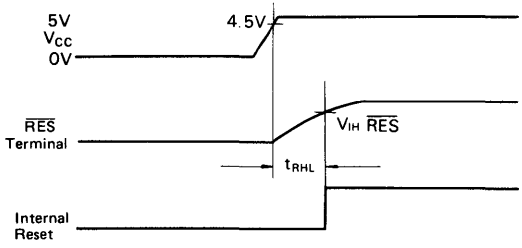


Figure 13 Power On and Reset Timing

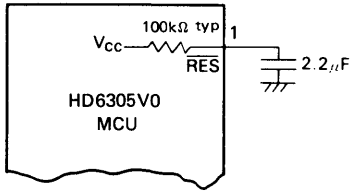


Figure 14 Input Reset Delay Circuit

■ INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0 MHz) or ceramic oscillator between pins 38 and 39 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 15. Figs. 16 and 17 illustrate the specifications and typical arrangement of the crystal, respectively.

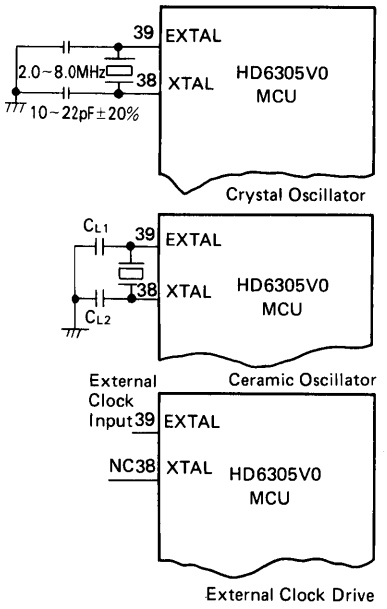


Figure 15 Internal Oscillator Circuit

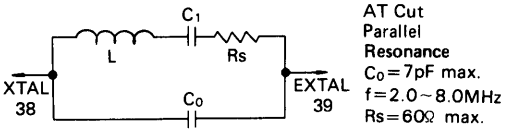
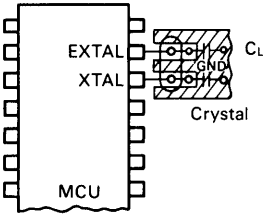


Figure 16 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 17 Typical Crystal Arrangement

■ LOW POWER DISSIPATION MODE

The HD6305V0 has three low power dissipation modes: wait, stop and standby.

● Wait Mode

When WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions — the timer and the serial communication interface — stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers (except the I bit of the condition code register which is cleared), RAM and I/O terminals hold their condition just before entering into the wait mode.

The escape from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$) is masked by the timer control register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 18 shows a flowchart for the wait function.

● Stop Mode

When STOP instruction being executed, MCU enters into the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive, but the RAM, registers (except bits 6 and 7 of the timer control register and the I bit of the condition code register) and I/O terminals hold their condition just before entering the stop mode. Bits 6 and 7 of the timer control register are initialized to "1" and "0", bits 7, 6, 5 and 4 of SCI status register are initialized "0", "0", "1", "1", respectively, and the I bit of the condition code register is cleared.

The escape from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings into the standby mode.

When interrupt is requested to the CPU and accepted, the stop mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 19 shows a flowchart for the stop function. Fig. 20 shows a timing chart of return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$ = "0" must exceed t_{osc} to assure stabilized oscillation.

● Standby Mode

The MCU enters into the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are hold. The I/O terminals turn to high-impedance state. The standby mode should escape by bringing $\overline{STBY}$ "High". The CPU must be restarted by reset. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 21.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 22.

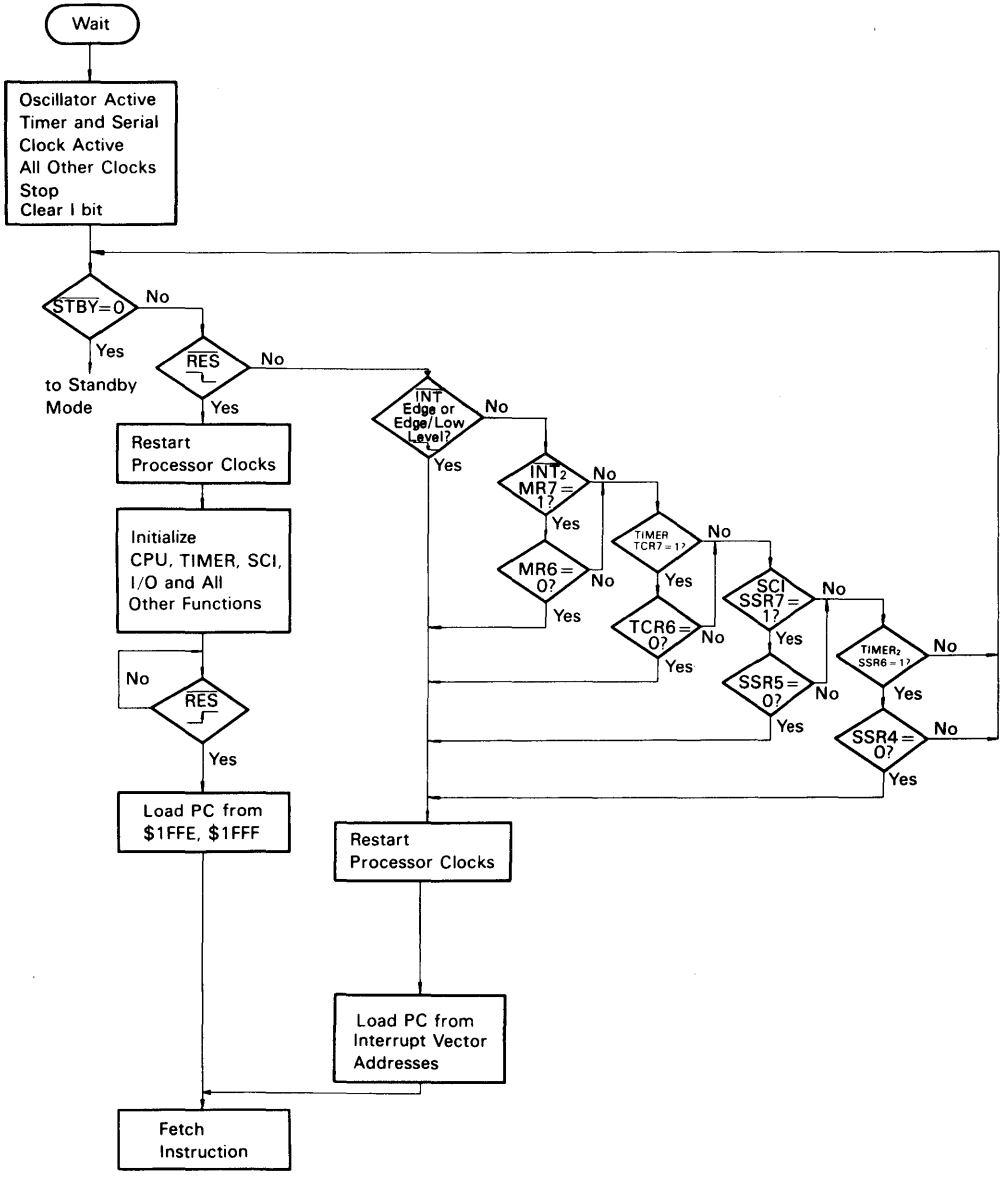


Figure 18 Wait Mode Flowchart

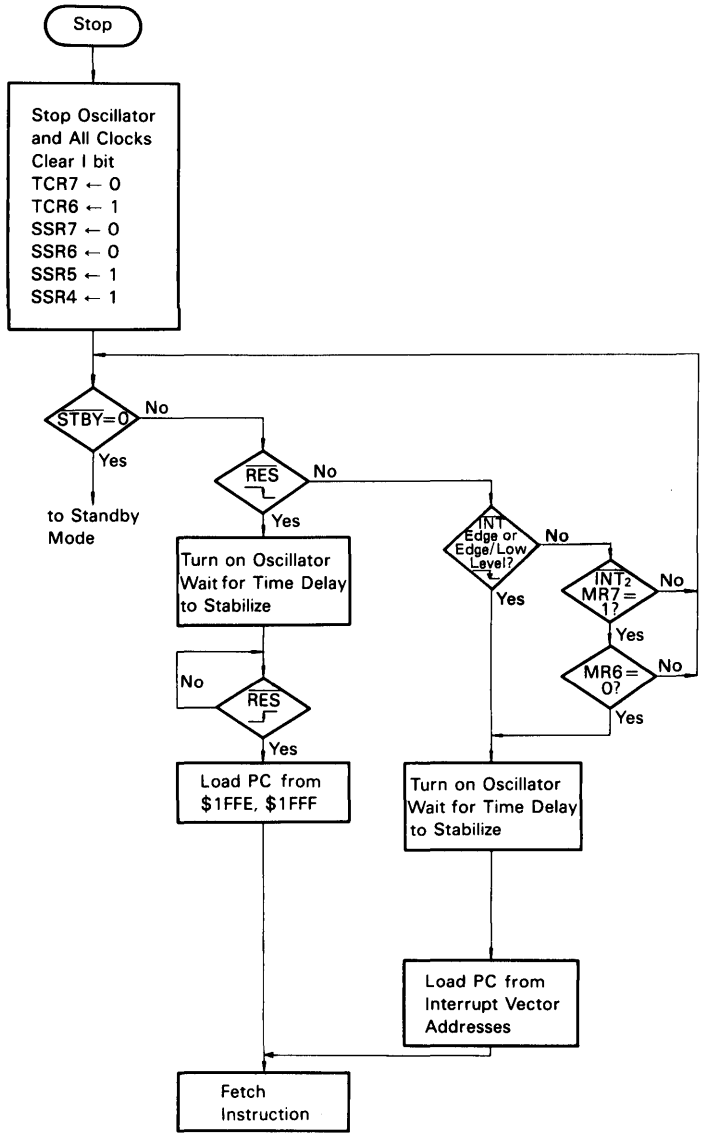
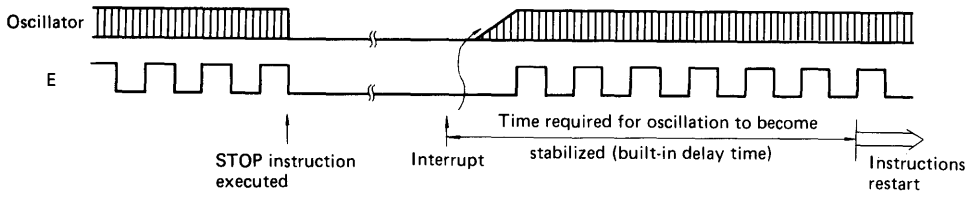
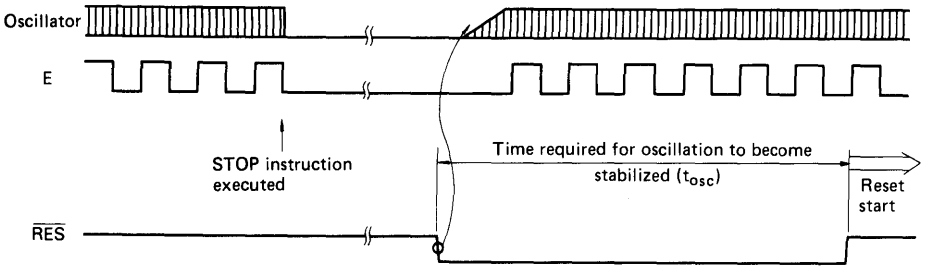


Figure 19 Stop Mode Flowchart



(a) Restart by Interrupt



(b) Restart by Reset

Figure 20 Timing Chart of Releasing from Stop Mode

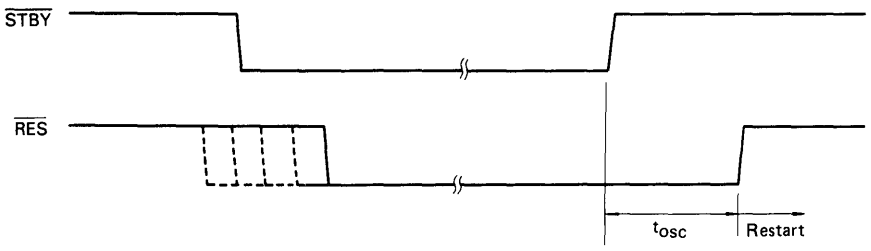


Figure 21 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register*	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Keep	High impedance	STBY="High"

* Register in the CPU (except I bit in the CCR

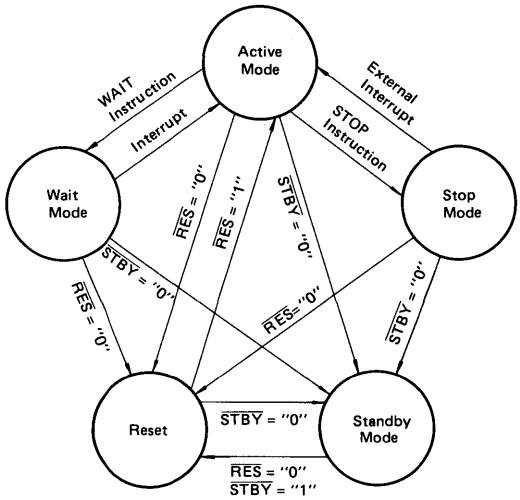


Figure 22 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■ BIT MANIPULATION

The HD6305V0 MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM or an I/O port. Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM, or I/O can be manipulated, the user may use a bit within the RAM as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 23 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the ROM. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
          :
SELF 1.   BRCLR 0, PORT A, SELF 1
          BSET 1, PORT A
          BCLR 1, PORT A
          :
```

Figure 23 Example of Bit Manipulation

■ ADDRESSING MODES

Ten different addressing modes are available to the HD6305V0 MCU.

● Immediate

See Fig. 24. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

● Direct

See Fig. 25. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. All RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

● Extended

See Fig. 26. the extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

● Relative

See Fig. 27. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

● Indexed (No Offset)

See Fig. 28. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

● Indexed (8-bit Offset)

See Fig. 29. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

● Indexed (16-bit Offset)

See Fig. 30. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

● Bit Set/Clear

See Fig. 31. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

● Bit Test and Branch

See Fig. 32. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The result of the test is written in the carry bit of the condition code register. (Set if true, cleared otherwise.)

● Implied

See Fig. 33. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

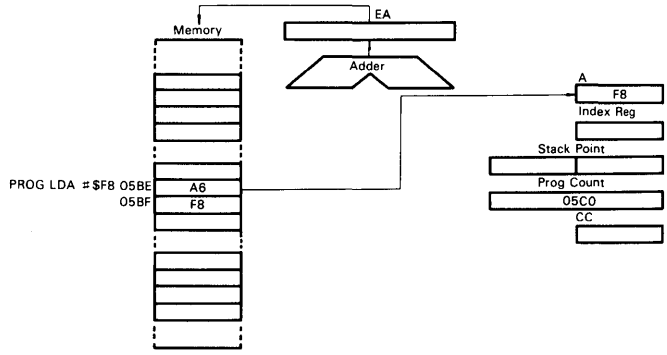


Figure 24 Example of Immediate Addressing

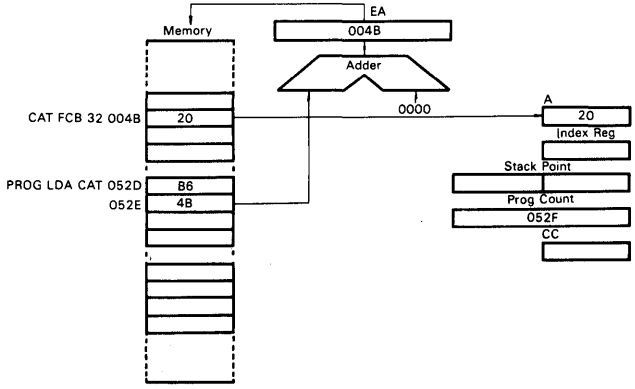


Figure 25 Example of Direct Addressing

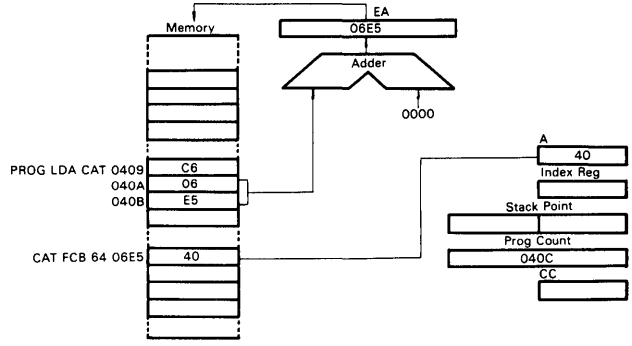


Figure 26 Example of Extended Addressing

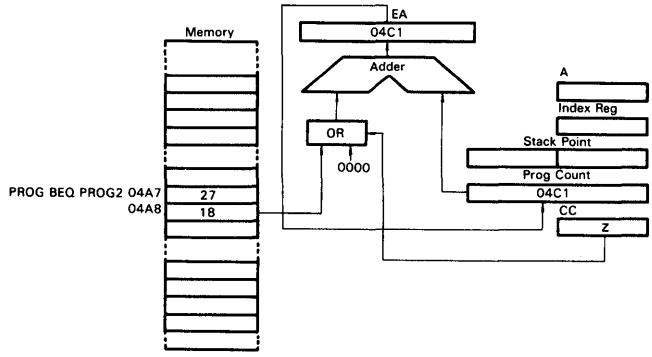


Figure 27 Example of Relative Addressing

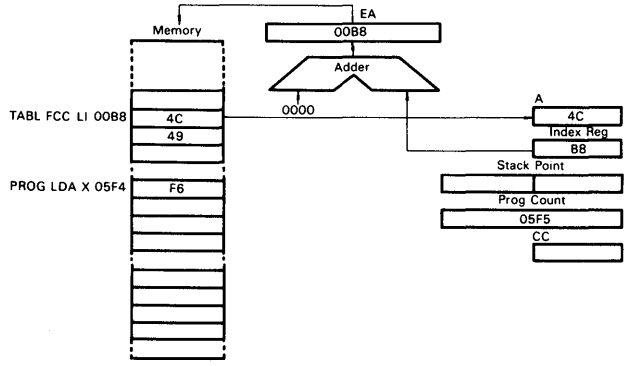


Figure 28 Example of Indexed (No Offset) Addressing

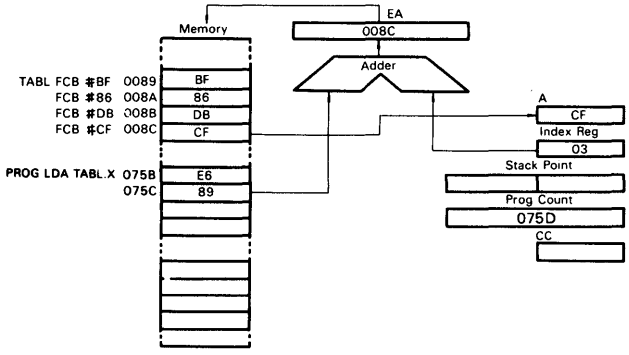


Figure 29 Example of Indexed (8-bit Offset) Addressing

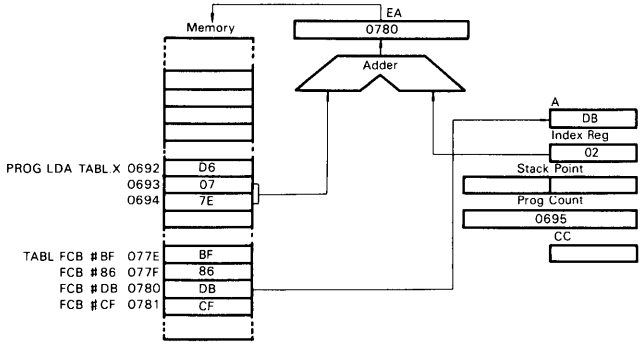


Figure 30 Example of Indexed (16-bit Offset) Addressing

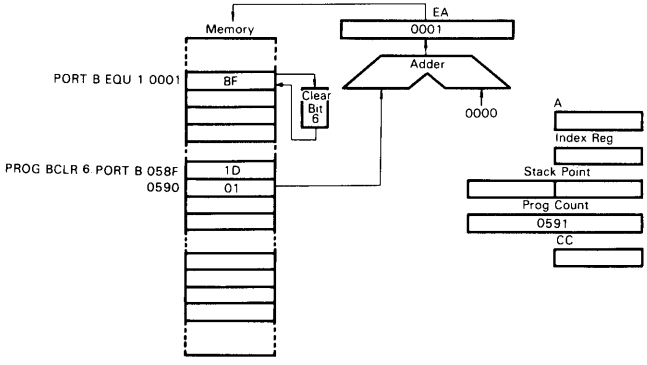


Figure 31 Example of Bit Set/Clear Addressing

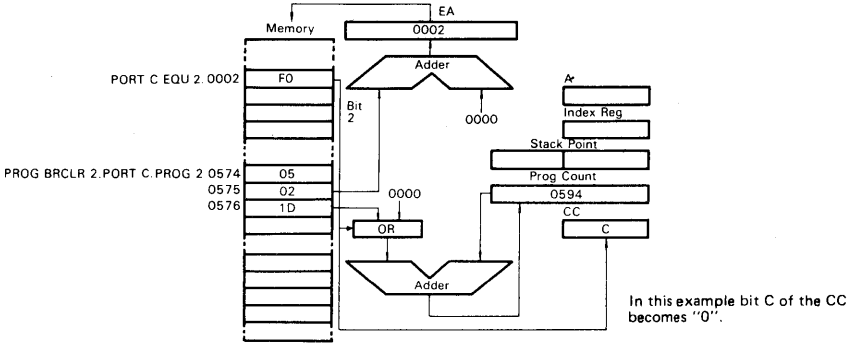


Figure 32 Example of Bit Test and Branch Addressing

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

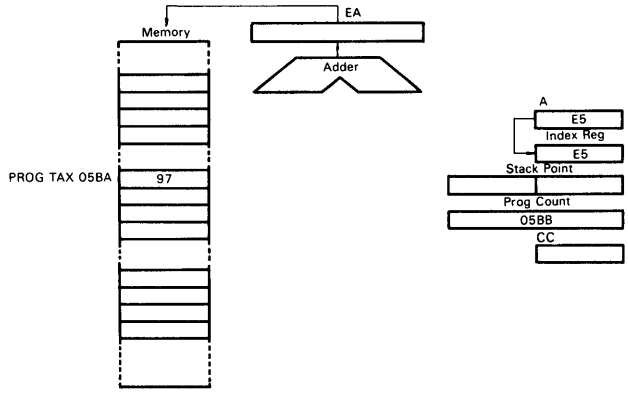


Figure 33 Example of Implied Addressing

■ INSTRUCTION SET

There are 62 basic instructions available to the HD6305V0 MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the HD6305V0 MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the HD6305V0 MCU in the alphabetical order.

• Operation Code Map

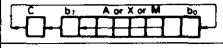
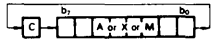
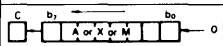
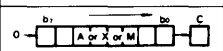
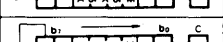
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes																Boolean/ Arithmetic Operation	Condition Code				
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)							
		OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~		H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	∧	∧	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	∧	∧	•
Store A in Memory	STA	—	—	—	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	∧	∧	•
Store X in Memory	STX	—	—	—	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	∧	∧	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	∧	•	∧	∧	∧
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	∧	•	∧	∧	∧
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	∧	∧	∧
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	∧	∧	∧
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	∧	∧	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	∧	∧	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A+M→A	•	•	∧	∧	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	∧	∧	∧
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	∧	∧	∧
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	∧	∧	•
Jump Unconditional	JMP	—	—	—	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR	—	—	—	BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes												Boolean/Arithmetic Operation	Condition Code				
		Implied(A)			Implied(X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)					
		OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~		H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	∧	∧	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	∧	∧	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	∧	∧	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	∧	∧	∧
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	∧	∧	∧
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	∧	∧	∧
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	∧	∧	∧
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	∧	∧
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	∧	∧	∧
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	∧	∧	∧
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	∧	∧	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	•	•	•	•	•
Branch Never	BRN	21	2	3	None	•	•	•	•	•
Branch IF Higher	BHI	22	2	3	C+Z=0	•	•	•	•	•
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	•	•	•	•	•
Branch IF Carry Clear	BCC	24	2	3	C=0	•	•	•	•	•
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	•	•	•	•	•
Branch IF Carry Set	BCS	25	2	3	C=1	•	•	•	•	•
(Branch IF Lower)	(BLO)	25	2	3	C=1	•	•	•	•	•
Branch IF Not Equal	BNE	26	2	3	Z=0	•	•	•	•	•
Branch IF Equal	BEQ	27	2	3	Z=1	•	•	•	•	•
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	•	•	•	•	•
Branch IF Half Carry Set	BHCS	29	2	3	H=1	•	•	•	•	•
Branch IF Plus	BPL	2A	2	3	N=0	•	•	•	•	•
Branch IF Minus	BMI	2B	2	3	N=1	•	•	•	•	•
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	•	•	•	•	•
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	•	•	•	•	•
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	•	•	•	•	•
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	•	•	•	•	•
Branch to Subroutine	BSR	AD	2	5	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	•	•	•	•	
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	•	•	•	•	
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	•	•	•	•	•
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code				
		Implied								
		OP	#	~		H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	●	●	●	●	●
Transfer X to A	TXA	9F	1	2	X→A	●	●	●	●	●
Set Carry Bit	SEC	99	1	1	1→C	●	●	●	●	1
Clear Carry Bit	CLC	98	1	1	0→C	●	●	●	●	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	●	1	●	●	●
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	●	0	●	●	●
Software Interrupt	SWI	83	1	10		●	1	●	●	●
Return from Subroutine	RTS	81	1	5		●	●	●	●	●
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	●	●	●	●	●
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	●	●	●	●	●
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	●	●	∧	∧	∧*
Stop	STOP	8E	1	4		●	0	●	●	●
Wait	WAIT	8F	1	4		●	0	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		×	×	×		×	×	×			∧	●	∧	∧	∧
ADD		×	×	×		×	×	×			∧	●	∧	∧	∧
AND		×	×	×		×	×	×			●	●	∧	∧	●
ASL	×		×			×	×				●	●	∧	∧	∧
ASR	×		×			×	×				●	●	∧	∧	∧
BCC					×						●	●	●	●	●
BCLR									×		●	●	●	●	●
BCS					×						●	●	●	●	●
BEQ					×						●	●	●	●	●
BHCC					×						●	●	●	●	●
BHCS					×						●	●	●	●	●
BHI					×						●	●	●	●	●
(BHS)					×						●	●	●	●	●
BIH					×						●	●	●	●	●
BIL					×						●	●	●	●	●
BIT		×	×	×		×	×	×			●	●	∧	∧	●
(BLO)					×						●	●	●	●	●
BLS					×						●	●	●	●	●
BMC					×						●	●	●	●	●
BMI					×						●	●	●	●	●
BMS					×						●	●	●	●	●
BNE					×						●	●	●	●	●
BPL					×						●	●	●	●	●
BRA					×						●	●	●	●	●

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						●	●	●	●	●
BRCLR										×	●	●	●	●	∧
BRSET										×	●	●	●	●	∧
BSET									×		●	●	●	●	●
BSR					×						●	●	●	●	●
CLC	×										●	●	●	●	0
CLI	×										●	0	●	●	●
CLR	×		×			×	×				●	●	0	1	●
CMP		×	×	×		×	×	×			●	●	∧	∧	∧
COM	×		×			×	×				●	●	∧	∧	1
CPX		×	×	×		×	×	×			●	●	∧	∧	∧
DAA	×										●	●	∧	∧	∧
DEC	×		×			×	×				●	●	∧	∧	●
EOR		×	×	×		×	×	×			●	●	∧	∧	●
INC	×		×			×	×				●	●	∧	∧	●
JMP			×	×		×	×	×			●	●	●	●	●
JSR			×	×		×	×	×			●	●	●	●	●
LDA		×	×	×		×	×	×			●	●	∧	∧	●
LDX		×	×	×		×	×	×			●	●	∧	∧	●
LSL	×		×			×	×				●	●	∧	∧	∧
LSR	×		×			×	×				●	●	0	∧	∧
NEG	×		×			×	×				●	●	∧	∧	∧
NOP	×										●	●	●	●	●
ORA		×	×	×		×	×	×			●	●	∧	∧	●
ROL	×		×			×	×				●	●	∧	∧	∧
ROR	×		×			×	×				●	●	∧	∧	∧
RSP	×										●	●	●	●	●
RTI	×										?	?	?	?	?
RTS	×										●	●	●	●	●
SBC		×	×	×		×	×	×			●	●	∧	∧	∧
SEC	×										●	●	●	●	1
SEI	×										●	1	●	●	●
STA			×	×		×	×	×			●	●	∧	∧	●
STOP	×										●	0	●	●	●
STX			×	×		×	×	×			●	●	∧	∧	●
SUB		×	×	×		×	×	×			●	●	∧	∧	∧
SWI	×										●	1	●	●	●
TAX	×										●	●	●	●	●
TST	×		×			×	×				●	●	∧	∧	●
TXA	×										●	●	●	●	●
WAIT	×										●	0	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI	—					—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

LOW

(NOTES) 1. "—" is an undefined operation code.
2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD6305V0:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

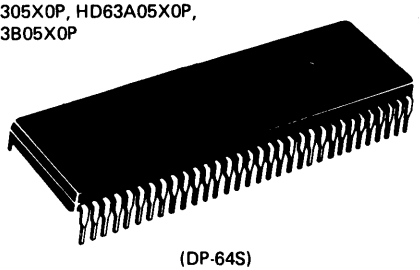
STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

HD6305X0, HD63A05X0, HD63B05X0 CMOS MCU (Microcomputer Unit)

The HD6305X0 is a CMOS 8-bit single-chip microcomputer. The CMOS unit is upward compatible with the HD6805 family in respect to instructions. On the chip of the HD6305X0, a CPU, a clock generator, a 4kB ROM, a 128-byte RAM, 55 I/O terminals, two timers and a serial communication interface (SCI) are built in. Because of the CMOS process, the HD6305X0 consumes less power than the NMOS. In addition, three low power dissipation modes (stop, wait, and standby) support the low power operating.

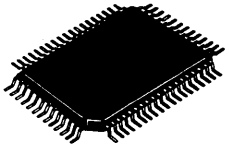
Other distinguished features include enhanced instruction cycle of the main instructions and the use of three additional instructions to obtain more improved system throughput.

HD6305X0P, HD63A05X0P, HD63B05X0P



(DP-64S)

HD6305X0F, HD63A05X0F, HD63B05X0F



(FP-64)

■ HARDWARE FEATURES

- 8-bit based MCU
- 4096-bytes of ROM
- 128-bytes of RAM
- A total of 55 terminals, including 32 I/O's, 7 inputs and 16 outputs
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes
  - Wait . . . . In this mode, the clock oscillator is on and the CPU halts but the timer/serial/interrupt function is operatable.
  - Stop . . . . In this mode, the clock stops but the RAM data, I/O status and registers are held.
  - Standby. . In this mode, the clock stops, the RAM data is held, and the other internal condition is reset.
- Minimum instruction cycle time
  - HD6305X0 . . . . . 1 μs (f = 1 MHz)
  - HD63A05X0 . . . . 0.67 μs (f = 1.5 MHz)
  - HD63B05X0 . . . . 0.5 μs (f = 2 MHz)
- Wide operating range
  V_{CC} = 3 to 6V (f = 0.1 to 0.5 MHz)
  - HD6305X0 . . . . . f = 0.1 to 1 MHz (V_{CC} = 5V ± 10%)
  - HD63A05X0 . . . . f = 0.1 to 1.5 MHz (V_{CC} = 5V ± 10%)
  - HD63B05X0 . . . . f = 0.1 to 2 MHz (V_{CC} = 5V ± 10%)
- System development fully supported by an evaluation kit

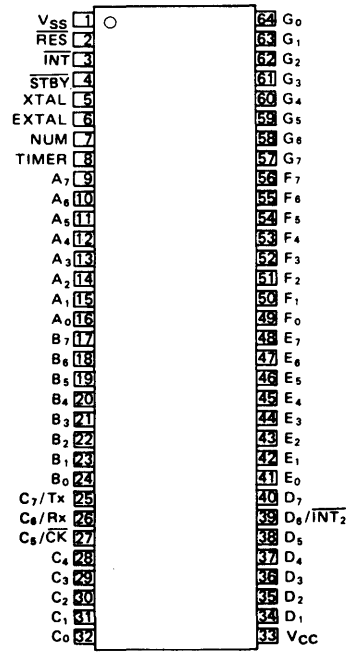
■ SOFTWARE FEATURES

- Similar to HD6800
- Byte efficient instruction set
- Powerful bit manipulation instructions (Bit Set, Bit Clear, and Bit Test and Branch usable for all RAM bits and all I/O terminals)
- A variety of interrupt operations
- Index addressing mode useful for table processing
- A variety of conditional branch instructions
- Ten powerful addressing modes
- All addressing modes adaptable to RAM, and I/O instructions
- Three new instructions, STOP, WAIT and DAA, added to the HD6805 family instruction set
- Instructions that are upward compatible with those of Motorola's MC6805P2 and MC146805G2

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

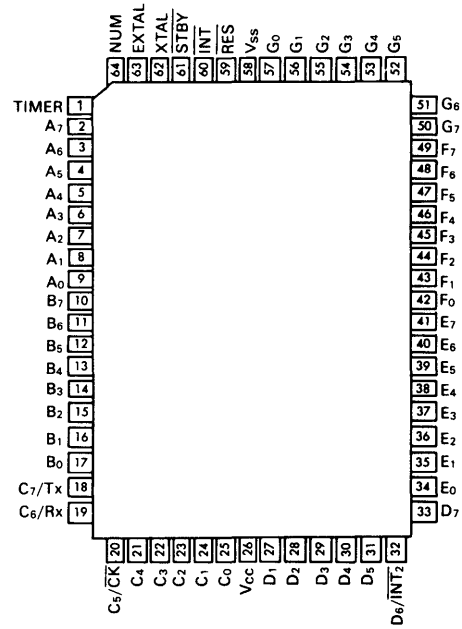
■ PIN ARRANGEMENT

● HD6305X0P, HD63A05X0P, HD63B05X0P



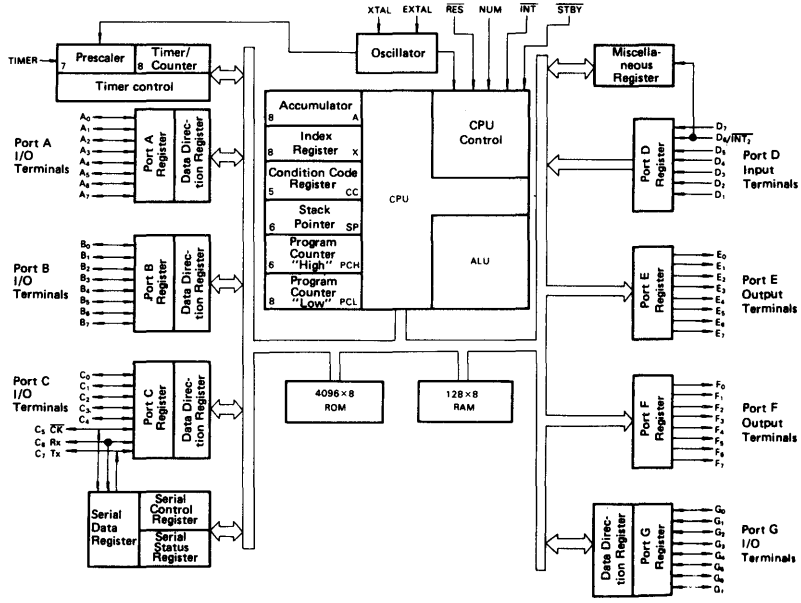
(Top View)

● HD6305X0F, HD63A05X0F, HD63B05X0F



(Top View)

■ BLOCK DIAGRAM



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 ~ +7.0	V
Input voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating temperature	T_{opr}	0 ~ +70	°C
Storage temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item		Symbol	Test condition	min	typ	max	Unit
Input voltage "High"	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC} − 0.5	−	V_{CC} + 0.3	V
	EXTAL			V_{CC} x 0.7	−	V_{CC} + 0.3	V
	Others			2.0	−	V_{CC} + 0.3	V
Input voltage "Low"	All Inputs	V_{IL}		−0.3	−	0.8	V
Current * dissipation	Operating	I_{CC}	f = 1MHz**	−	5	10	mA
	Wait			−	2	5	mA
	Stop			−	2	10	μA
	Standby			−	2	10	μA
Input leakage current	$\overline{TIMER}$, $\overline{INT}$, $D_1 \sim D_7$, $\overline{STBY}$	$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} − 0.5V	−	−	1	μA
Three-state current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $G_0 \sim G_7$, $E_0 \sim E_7$***, $F_0 \sim F_7$***	$\|I_{TSI}\|$		−	−	1	μA
Input capacity	All terminals	C_{in}	f = 1MHz, V_{in} = 0V	−	−	12	pF

* V_{IH} min = V_{CC}−1.0V, V_{IL} max = 0.8 V
**The value at f = xMHz can be calculated by the following equation: I_{CC} (f = xMHz) = I_{CC} (f = 1MHz) multiplied by x
***At standby mode

● **AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)**

Item	Symbol	Test condition	HD6305X0			HD63A05X0			HD63B05X0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock frequency	f_{cl}		0.4	–	4	0.4	–	6	0.4	–	8	MHz
Cycle time	t_{cyc}		1.0	–	10	0.666	–	10	0.5	–	10	μs
$\overline{INT}$ pulse width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ pulse width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ pulse width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
TIMER pulse width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation start time (crystal)	t_{OSC}	C_L = 22pF ± 20% R_s = 60Ω max	–	–	20	–	–	20	–	–	20	ms
Reset delay time	t_{RHL}	External cap. 2.2μF	80	–	–	80	–	–	80	–	–	ms

● **Port Electrical Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)**

Item		Symbol	Test condition	min	typ	max	Unit
Output voltage "High"	Ports A, B, C, G, E, F	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC} - 0.7	–	–	V
Output voltage "Low"		V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input voltage "High"	Ports A, B, C, D, G	V_{IH}		2.0	–	V_{CC} + 0.3	V
Input voltage "Low"		V_{IL}		- 0.3	–	0.8	V
Input leakage current		I_{IL}	V_{in} = 0.5 ~ V_{CC} - 0.5V	-1	–	1	μA

● **SCI Timing (V_{CC} = 5.0V±10%, V_{SS} = 0V and Ta = 0 ~ +70°C unless otherwise specified)**

Item	Symbol	Test Condition	HD6305X0			HD63A05X0			HD63B05X0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock Cycle	t_{Scyc}	Fig. 1 Fig. 2	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data Output Delay Time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data Set-up Time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data Hold Time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

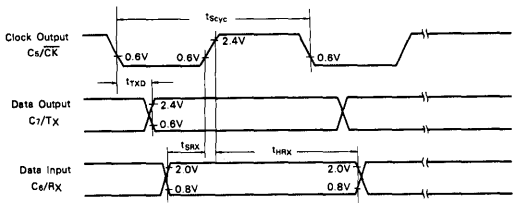


Figure 1 SCI Timing (Internal Clock)

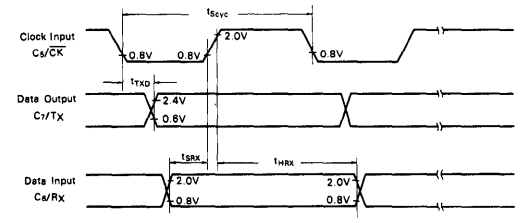
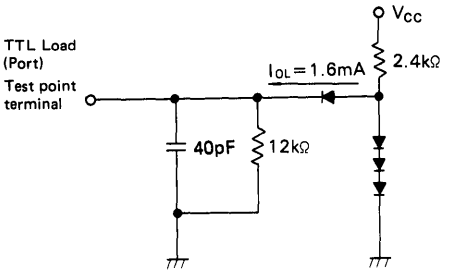


Figure 2 SCI Timing (External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ.

Figure 3 Test Load

■ DESCRIPTION OF TERMINAL FUNCTIONS

The input and output signals of the HD6305X0 are described here.

• V_{CC}, V_{SS}

Voltage is applied to MCU through these two terminals. V_{CC} is 5.0V ± 10%, while V_{SS} is grounded.

• $\overline{INT}$, $\overline{INT_2}$

External interrupt request inputs to MCU. For details, refer to "INTERRUPTS". The $\overline{INT_2}$ terminal is also used as the port D_6 terminal.

• XTAL, EXTAL

These terminals provide input to the on-chip clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic filter is connected to the terminal. Refer to "INTERNAL OSCILLATOR" for using these input terminals.

• TIMER

This is an input terminal for event counter. Refer to "TIMER" for details.

• $\overline{RES}$

Used to reset the MCU. Refer to "RESET" for details.

• NUM

This terminal is not intended for user applications. It should be grounded to V_{SS}.

• Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $G_0 \sim G_7$)

These 32 terminals consist of four 8-bit I/O ports (A, B, C, G). Each of them can be used as an input or output terminal on a bit through program control of the data direction register. For details, refer to "I/O PORTS".

• Input Terminals ($D_1 \sim D_7$)

These seven input-only terminals are TTL or CMOS compatible. Of the port D's, D_6 is also used as $\overline{INT_2}$. If D_6 is used as a port, the $\overline{INT_2}$ interrupt mask bit of the miscellaneous register must be set to "1" to prevent an $\overline{INT_2}$ interrupt from being accidentally accepted.

• Output Terminals ($E_0 \sim E_7$, $F_0 \sim F_7$)

These 16 output-only terminals are TTL or CMOS compatible.

• $\overline{STBY}$

This terminal is used to place the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and the internal condition is reset. For details, refer to "Standby Mode".

The terminals described in the following are I/O pins for serial communication interface (SCI). They are also used as ports C_5, C_6 and C_7. For details, refer to "SERIAL COMMUNICATION INTERFACE."

• $\overline{CK}$ (C_5)

Used to input or output clocks for serial operation.

• Tx (C_7)

Used to transmit serial data.

• Rx (C_6)

Used to receive serial data.

■MEMORY MAP

The memory map of the HD6305X0 MCU is shown in Fig. 4. During interrupt processing, the contents of the CPU registers are saved into the stack in the sequence shown in Fig. 5. This saving begins with the lower byte (PCL) of the program counter. Then the value of the stack pointer is decremented and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CC) are stacked in that order. In a subroutine call, only the contents of the program counter (PCH and PCL) are stacked.

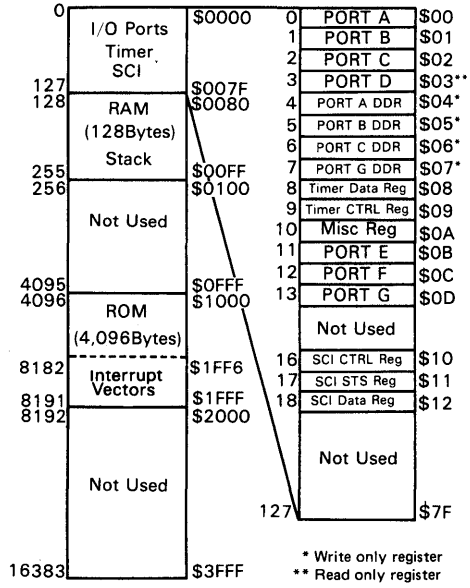
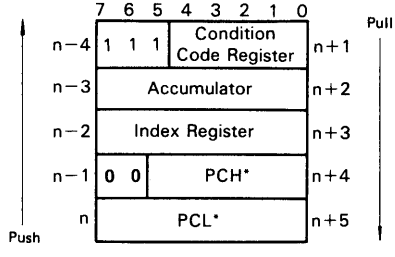


Figure 4 Memory Map of HD6305X0 MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 5 Sequence of Interrupt Stacking

■REGISTERS

There are five registers which the programmer can operate.

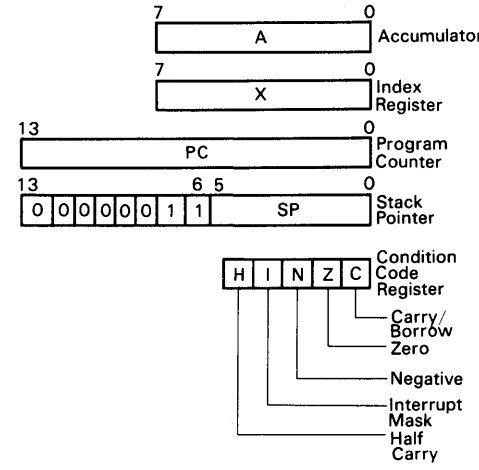


Figure 6 Programming Model

• Accumulator (A)

This accumulator is an ordinary 8-bit register which holds operands or the result of arithmetic operation or data processing.

• Index Register (X)

The index register is an 8-bit register, and is used for index addressing mode. Each of the addresses contained in the register consists of 8 bits which, combined with an offset value, provides an effective address.

In the case of a read/modify/write instruction, the index register can be used like an accumulator to hold operation data or the result of operation.

If not used in the index addressing mode, the register can be used to store data temporarily.

• Program Counter (PC)

The program counter is a 14-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 14-bit register that indicates the address of the next stacking space. Just after reset, the stack pointer is set at address $00FF. It is decremented when data is pushed, and incremented when pulled. The upper 8 bits of the stack pointer are fixed to 00000011. During the MCU being reset or during a reset stack pointer (RSP) instruction, the pointer is set to address $00FF. Since a subroutine or interrupt can use space up to address $00C1 for stacking, the subroutine can be used up to 31 levels and the interrupt up to 12 levels.

• Condition Code Register (CC)

The condition code register is a 5-bit register, each bit indicating the result of the instruction just executed. The bits can be individually tested by conditional branch instruc-

tions. The CC bits are as follows:

Half Carry (H): Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts, except a software interrupt, to be masked. If an interrupt occurs with the bit I set, it is latched. It will be processed the instant the interrupt mask bit is reset. (More specifically, it will enter the interrupt processing routine after the instruction following the CLI has been executed.)

Negative (N): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (bit 7 is logic "1").

Zero (Z): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is zero.

Carry/ Borrow (C): Represents a carry or borrow that occurred in the most recent arithmetic operation. This bit is also affected by the Bit Test and Branch instruction and a Rotate instruction.

■ INTERRUPT

There are six different types of interrupt: external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER or the SCI and $TIMER_2$ generate the same vector address, respectively.

When an interrupt occurs, the program in progress stops and the then CPU status is saved onto the stack. And then, the interrupt mask bit (I) of the condition code register is set and the start address of the interrupt processing routine is obtained from a particular interrupt vector address. Then the interrupt routine starts from the start address. System can exit from the interrupt routine by an RTI instruction. When this instruction is executed, the CPU status before the interrupt (saved onto the stack) is pulled and the CPU restarts the sequence with the instruction next to the one at which the interrupt occurred. Table 1 lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$1FFE, $1FFF
SWI	2	$1FFC, $1FFD
$\overline{INT}$	3	$1FFA, $1FFB
TIMER/$\overline{INT_2}$	4	$1FF8, $1FF9
SCI/$TIMER_2$	5	$1FF6, $1FF7

A flowchart of the interrupt sequence is shown in Fig. 7. A block diagram of the interrupt request source is shown in Fig. 8.

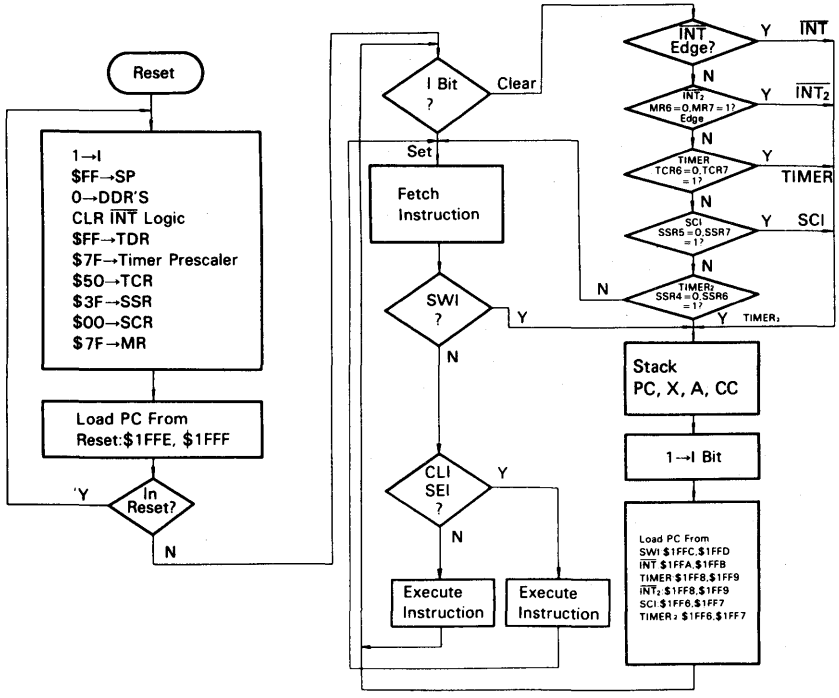


Figure 7 Interrupt Flowchart

In the block diagram, both the external interrupts $\overline{INT}$ and $\overline{INT_2}$ are edge trigger inputs. At the falling edge of each input, an interrupt request is generated and latched. The $\overline{INT}$ interrupt request is automatically cleared if jumping is made to the $\overline{INT}$ processing routine. Meanwhile, the $\overline{INT_2}$ request is cleared if "0" is written in bit 7 of the miscellaneous register.

For the external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$) and serial interrupt (SCI), each interrupt request is held, but not processed, if the I bit of the condition code register is set. Immediately after the I bit is cleared, the corresponding interrupt processing starts according to the priority.

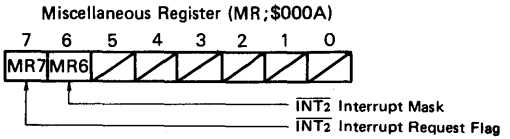
The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by setting bit 6 of the timer control register; the SCI interrupt by setting bit 5 of the serial status register; and the $TIMER_2$ interrupt by setting bit 4 of the serial status register.

The status of the $\overline{INT}$ terminal can be tested by a BIL or BIH instruction. The $\overline{INT}$ falling edge detector circuit and its latching circuit are independent of testing by these instructions. This is also true with the status of the $\overline{INT_2}$ terminal.

• Miscellaneous Register (MR; $000A)

The interrupt vector address for the external interrupt $\overline{INT_2}$ is the same as that for the TIMER interrupt, as shown in Table 1. For this reason, a special register called the miscellaneous register (MR; $000A) is available to control the $\overline{INT_2}$ interrupts.

Bit 7 of this register is the $\overline{INT_2}$ interrupt request flag. When the falling edge is detected at the $\overline{INT_2}$ terminal, "1" is set in bit 7. Then the software in the interrupt routine (vector addresses: $1FF8, $1FF9) checks bit 7 to see if it is $\overline{INT_2}$ interrupt. Bit 7 can be reset by software.



Miscellaneous Register (MR; $000A)

Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If this bit is set to "1", then the $\overline{INT_2}$ interrupt is disabled. Both read and write are possible with bit 7 but "1" cannot be written in this bit by software. This means that an interrupt request by software is impossible.

When reset, bit 7 is cleared to "0" and bit 6 is set to "1".

■ TIMER

Figure 9 shows an MCU timer block diagram. The timer data register is loaded by software and, upon receipt of a clock input, begins to count down. When the timer data

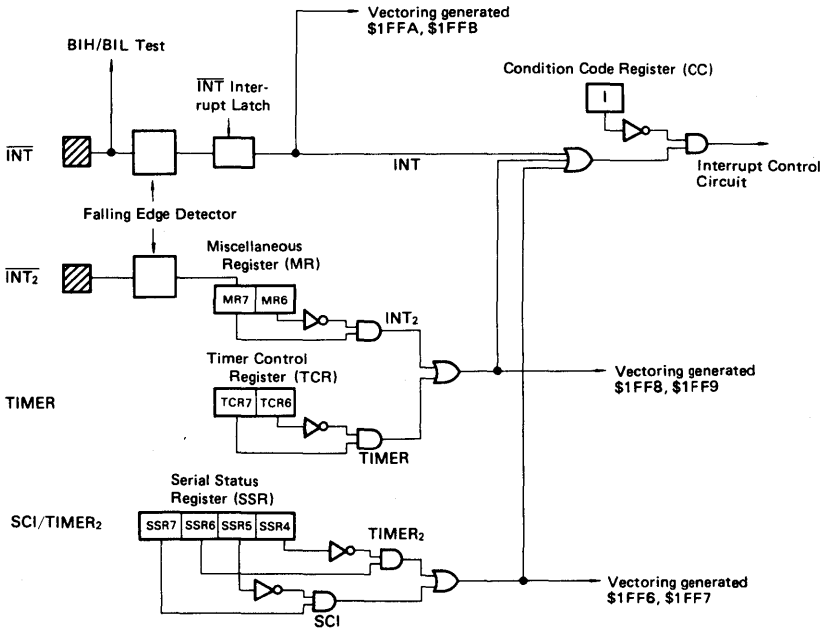


Figure 8 Interrupt Request Generation Circuitry

register (TDR) becomes "0", the timer interrupt request bit (bit 7) in the timer control register is set. In response to the interrupt request, the MCU saves its status into the stack and fetches timer interrupt routine address from addresses $1FF8 and $1FF9 and execute the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also mask the timer interrupt.

The source clock to the timer can be either an external signal from the timer input terminal or the internal E signal (the oscillator clock divided by 4). If the E signal is used as the source, the clock input can be gated by the input to the timer input terminal.

Once the timer count has reached "0", it starts counting down with "$FF". The count can be monitored whenever desired by reading the timer data register. This permits the program to know the length of time having passed after the occurrence of a timer interrupt, without disturbing the contents of the counter.

When the MCU is reset, both the prescaler and counter are initialized to logic "1". The timer interrupt request bit (bit 7) then is cleared and the timer interrupt mask bit (bit 6) is set.

To clear the timer interrupt request bit (bit 7), it is necessary to write "0" in that bit.

TCR7	Timer interrupt request
0	Absent
1	Present

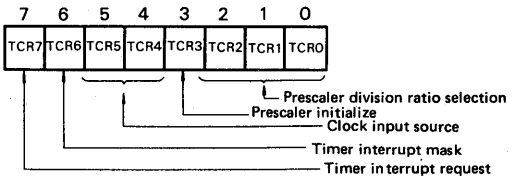
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

• Timer Control Register (TCR; $0009)

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After reset, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$FF" immediately after reset.

When "1" is written in bit 3, the prescaler is initialized. This bit always shows "0" when read.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under timer terminal control
1	0	No clock input (counting stopped)
1	1	Event input from timer terminal

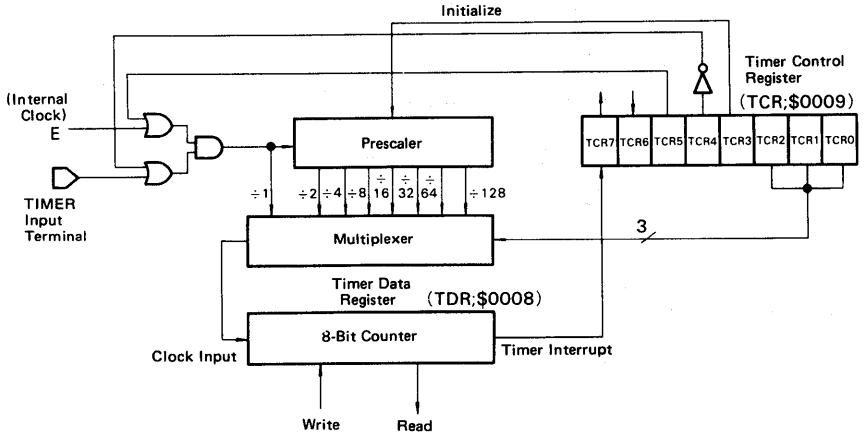


Figure 9 Timer Block Diagram

A prescaler division ratio is selected by the combination of three bits (bits 0, 1 and 2) of the timer control register (see Table 3). There are eight different division ratios: ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128. After reset, the TCR is set to the ÷1 mode.

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

A timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. This bit can be cleared by writing "0" in that bit.

■SERIAL COMMUNICATION INTERFACE (SCI)

This interface is used for serial transmission or reception of 8-bit data. Sixteen transfer rates are available in the range from 1 μs to approx. 32 ms (for oscillation at 4 MHz).

The SCI consists of three registers, one eighth counter and one prescaler. (See Fig. 10.) SCI communicates with the CPU via the data bus, and with the outside world through bits 5, 6 and 7 of port C. Described below are the operations of each register and data transfer.

●SCI Control Register (SCR; $0010)

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

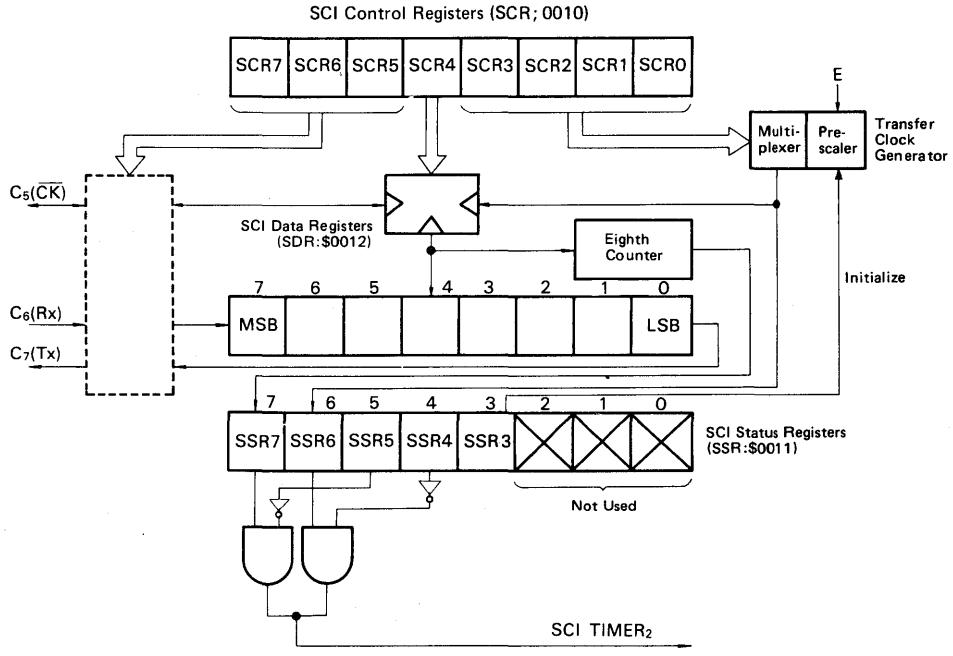


Figure 10 SCI Block Diagram

SCR7	C_7 terminal
0	Used as I/O terminal (by DDR).
1	Serial data output (DDR output)

SCR6	C_6 terminal
0	Used as I/O terminal (by DDR).
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	C_5 terminal
0	0	–	Used as I/O terminal (by DDR).
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)
When this bit is set, the DDR corresponding to the C_7 becomes "1" and this terminal serves for output of SCI data. After reset, the bit is cleared to "0".

Bit 6 (SCR6)
When this bit is set, the DDR corresponding to the C_6 becomes "0" and this terminal serves for input of SCI data. After reset, the bit is cleared to "0".

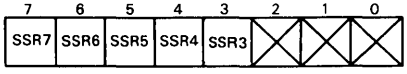
Bits 5 and 4 (SCR5, SCR4)
These bits are used to select a clock source. After reset, the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)
These bits are used to select a transfer clock rate. After reset, the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

•SCI Data Register (SDR; $0012)
A serial-parallel conversion register that is used for transfer of data.

•SCI Status Register (SSR; $0011)



Bit 7 (SSR7)
Bit 7 is the SCI interrupt request bit which is set upon completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5="1". The bit can also be cleared by writing "0" in it.

Bit 6 (SSR6)
Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is used commonly with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When reset, the bit is cleared. It also be cleared by writing "0" in it. (For details, see $TIMER_2$.)

Bit 5 (SSR5)
Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When reset, it is set to "1".

Bit 4 (SSR4)
Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When reset, it is set to "1".

Bit 3 (SSR3)
When "1" is written in this bit, the prescaler of the transfer clock generator is initialized. When read, the bit always is "0".

Bits 2 ~ 0
Not used.

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

• Data Transmission
By writing the desired control bits into the SCI control registers, a transfer rate and a source of transfer clock are determined and bits 7 and 5 of port C are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the C_7/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock. (See Fig.11.) When 8 bits of

data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register with the rising edge of the last serial clock. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the C_7/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the C_5/$\overline{CK}$ terminal is set as input. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

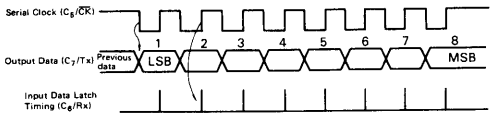


Figure 11 SCI Timing Chart

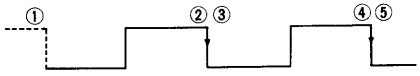
• Data Reception

By writing the desired control bits into the SCI control register, a transfer rate and a source of transfer clock are determined and bit 6 and 5 of port C are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed after reading subseqent received data. It must be taken after reset and after not reading subsequent received data.)

The data from the C_6/Rx terminal is input to the SCI data register synchronously with the rising edge of the serial clock (see Fig. 11). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored and the data is received synchronously with the clock from the C_5/$\overline{CK}$ terminal. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

• TIMER$_2$

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 ~ 0 of the SCI control register (4 μs ~ approx. 32 ms (for oscillation at 4 MHz)) is input to bit 6 of the SCI status register and the TIMER$_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, TIMER$_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : TIMER$_2$ interrupt request
③, ⑤ : TIMER$_2$ interrupt request bit cleared

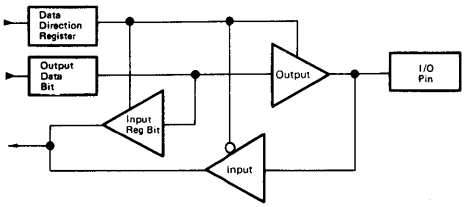
TIMER$_2$ is commonly used with the SCI transfer clock generator. If wanting to use TIMER$_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■ I/O PORTS

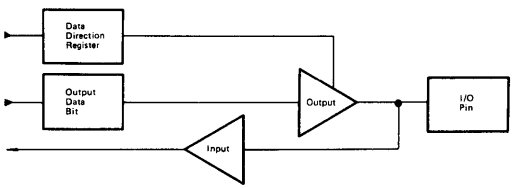
There are 32 input/output terminals (ports A, B, C, G). Each I/O terminal can be selected for either input or output by the data direction register. More specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B or C reads latched data if it has been programmed as output, even with the output level being fluctuated by the output load. (See Fig. 12-a.) For port G, in such a case, the level of the pin is always read when it is read. (See Fig. 12-b.) This implies that, even when "1" is being output, port G may read "0" if the load condition causes the output voltage to decrease to below 2.0V.

When reset, the data direction register and data register go to "0" and all the input/output terminals are used as input.



Bit of data direction register	Bit of output data	Status of output	Input to CPU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

a. Ports A, B and C



b. Port G

Figure 12 Input/Output Port Diagram

There are 16 output-only terminals (ports E and F). Each of them can also read. In this case, latched data is read even with the output terminal level being fluctuated by the output load (as with ports A, B and C).
When reset, "Low" level is output from each output terminal.

Seven input-only terminals are available (port D). Writing to an input terminal is invalid.

All input/output terminals, output terminals and input terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports or input ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite that they are not used.

■RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 13.) On power up, the reset input must be held "Low" for at least t_{OSC} to assure that the internal oscillator is stabilized. A sufficient time of delay can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 14.

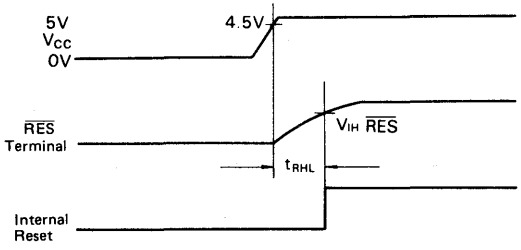


Figure 13 Power On and Reset Timing

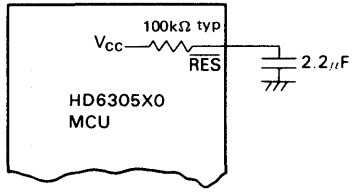


Figure 14 Input Reset Delay Circuit

■INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0MHz) or ceramic oscillator between pins 5 and 6 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 15. Figs. 16 and 17 illustrate the specifications and typical arrangement of the crystal, respectively.

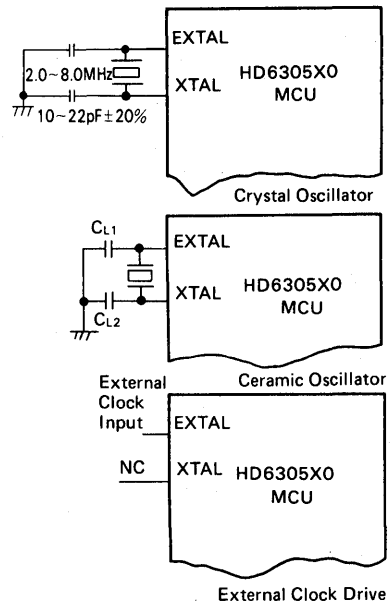


Figure 15 Internal Oscillator Circuit

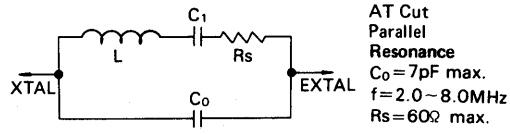
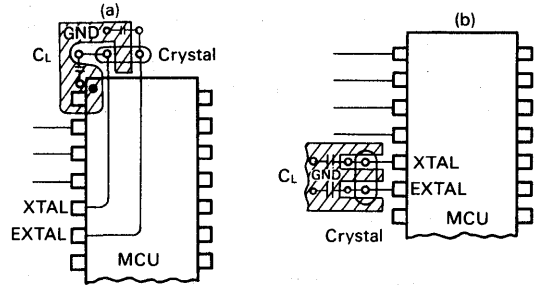


Figure 16 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 17 Typical Crystal Arrangement

■LOW POWER DISSIPATION MODE

The HD6305X0 has three low power dissipation modes: wait, stop and standby.

• Wait Mode

When WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers, RAM and I/O terminals hold their condition just before entering into the wait mode.

The escape from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the wait mode the MCU executes the instruction next to the WAIT. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$) is masked by the timer control register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 18 shows a flowchart for the wait function.

• Stop Mode

When STOP instruction being executed, MCU enters into the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive but the RAM, registers and I/O terminals hold their condition just before entering into the stop mode.

The escape from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings into the standby mode.

When interrupt is requested to the CPU and accepted, the stop mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the stop mode, the MCU executes the instruction next to the STOP. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 19 shows a flowchart for the stop function. Fig. 20 shows a timing chart of return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$="0" must exceed t_{osc} to assure stabilized oscillation.

• Standby Mode

The MCU enters into the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are hold. The I/O terminals turn to high-impedance state. The standby mode should escape by bringing $\overline{STBY}$ "High". The CPU must be restarted by reset. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 21.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 22.

(Note)

When I bit of condition code register is "1" and interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$, SCI/TIMER$_2$) is held, MCU does not enter WAIT mode by the execution of WAIT instruction.

In that case, after the 4 dummy cycles MCU executes the next instruction.

In the same way, when external interrupts ($\overline{INT}$, $\overline{INT_2}$) are held at the bit I set, MCU does not enter STOP mode by the execution of STOP instruction. In that case, also, MCU executes the next instruction after the 4 dummy cycles.

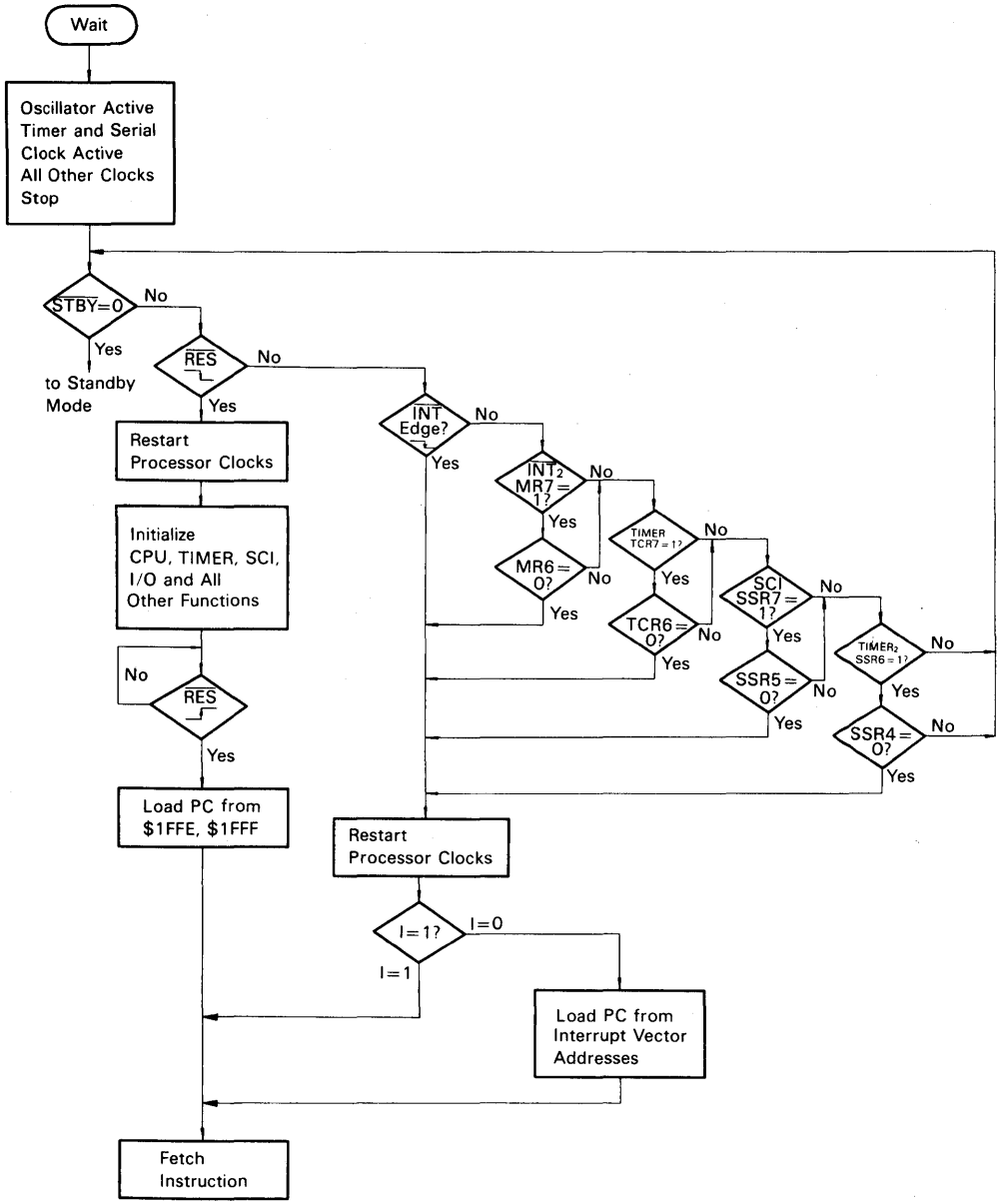


Figure 18 Wait Mode Flowchart

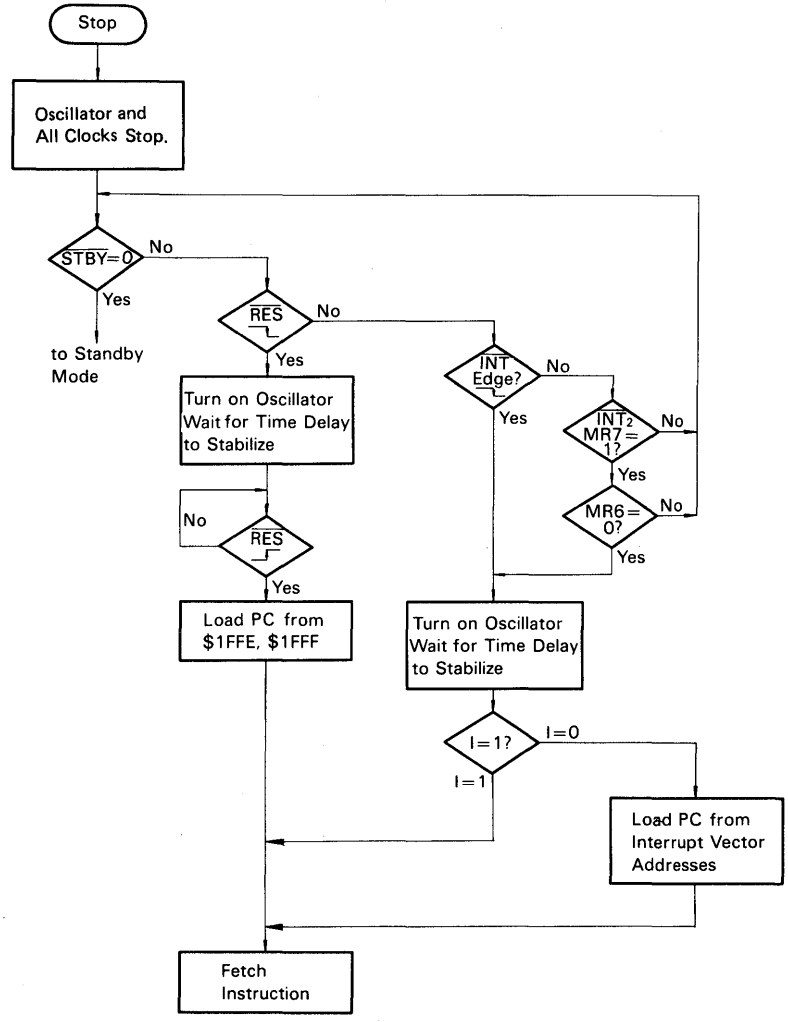
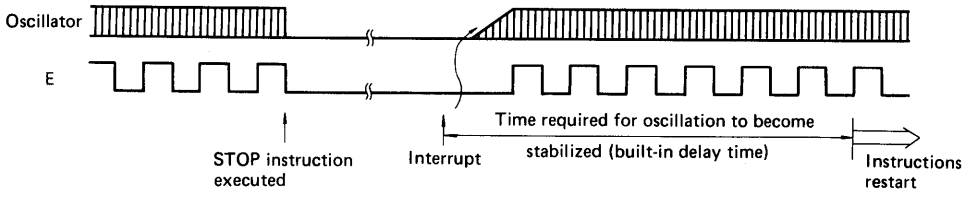
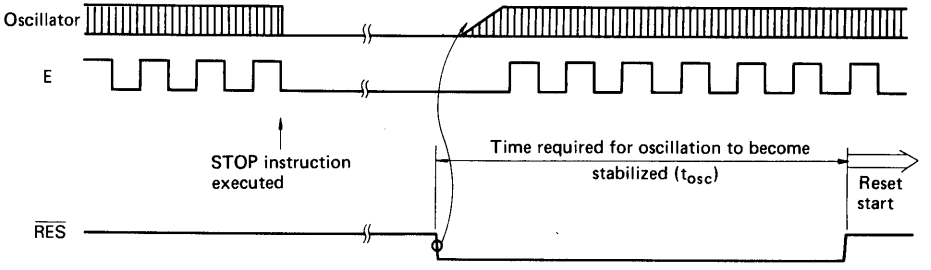


Figure 19 Stop Mode Flowchart



(a) Restart by Interrupt



(b) Restart by Reset

Figure 20 Timing Chart of Releasing from Stop Mode

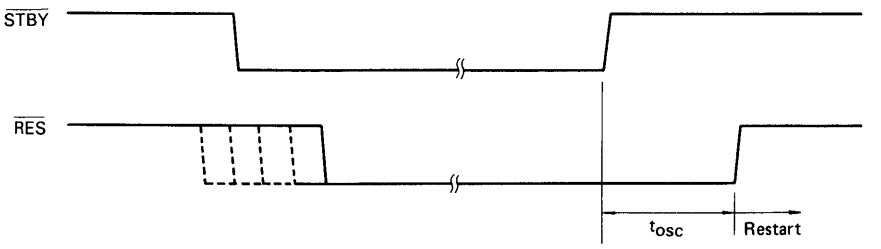


Figure 21 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Keep	High impedance	$\overline{STBY}$="High"

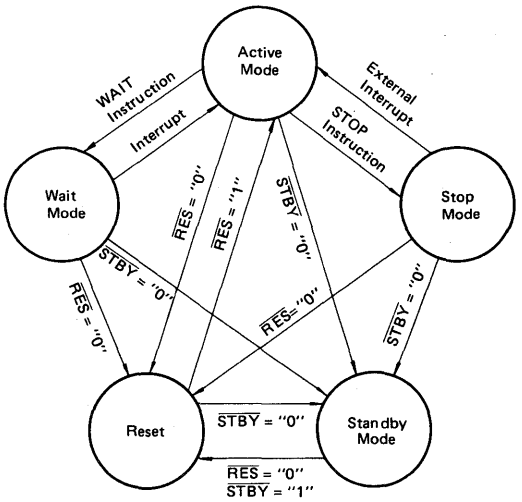


Figure 22 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■BIT MANIPULATION

The HD6305X0 MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM or an I/O port (except the write-only registers such as the data direction register). Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM, or I/O can be manipulated, the user may use a bit within the RAM as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 23 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the ROM. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
        :
SELF 1. BRCLR 0, PORT A, SELF 1
        BSET 1, PORT A
        BCLR 1, PORT A
        :
```

Figure 23 Example of Bit Manipulation

■ADDRESSING MODES

Ten different addressing modes are available to the HD6305X0 MCU.

• Immediate

See Fig. 24. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 25. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. All RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 26. The extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 27. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 28. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 29. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 30. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 31. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 32. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The value of the test bit is written in the carry bit of the condition code register.

• Implied

See Fig. 33. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

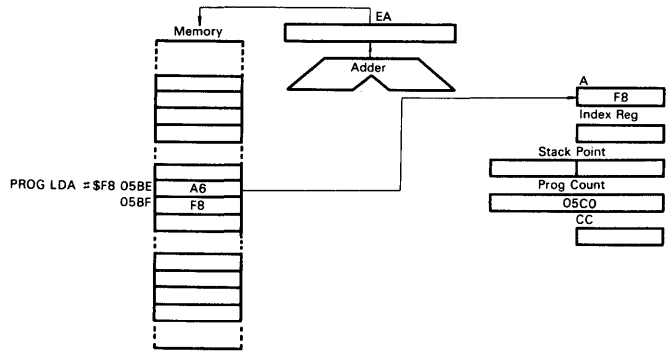


Figure 24 Example of Immediate Addressing

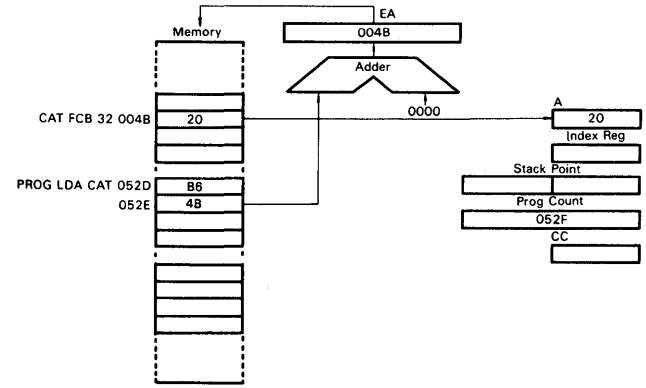


Figure 25 Example of Direct Addressing

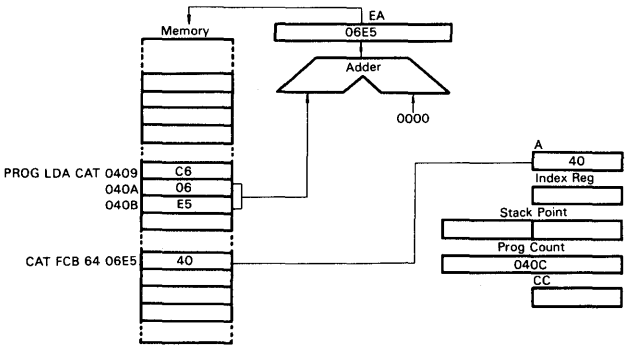


Figure 26 Example of Extended Addressing

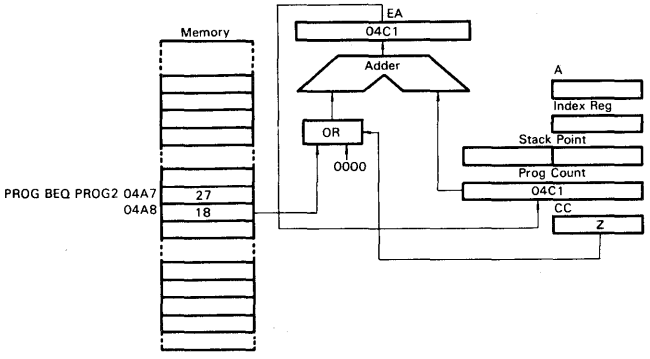


Figure 27 Example of Relative Addressing

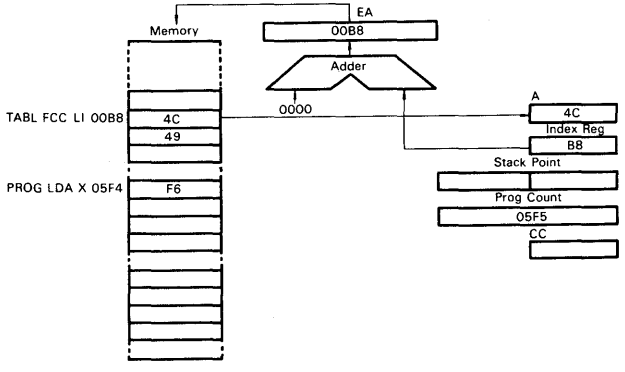


Figure 28 Example of Indexed (No Offset) Addressing

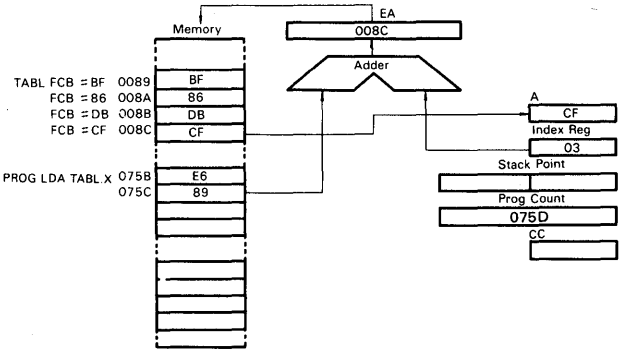


Figure 29 Example of Index (8-bit Offset) Addressing

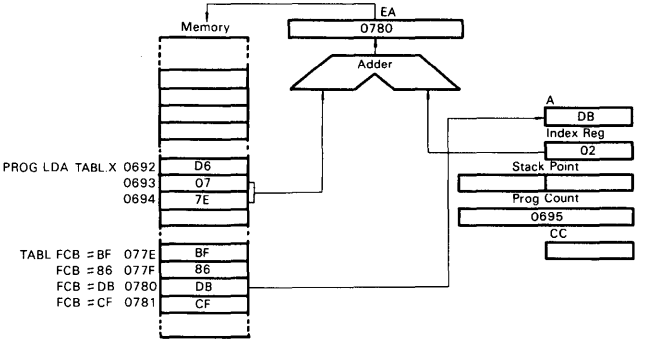


Figure 30 Example of Index (16-bit Offset) Addressing

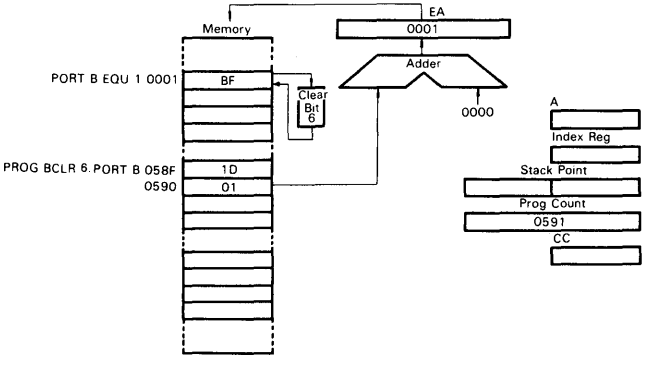


Figure 31 Example of Bit Set/Clear Addressing

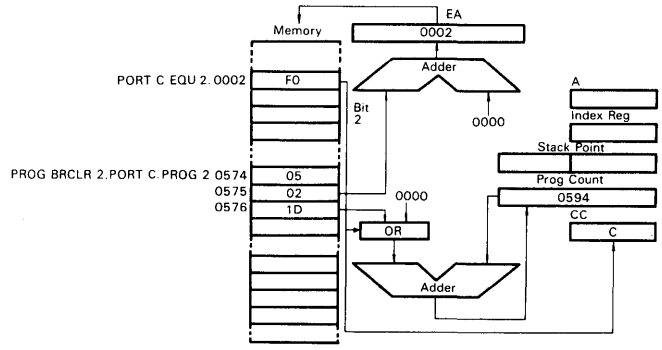


Figure 32 Example of Bit Test and Branch Addressing

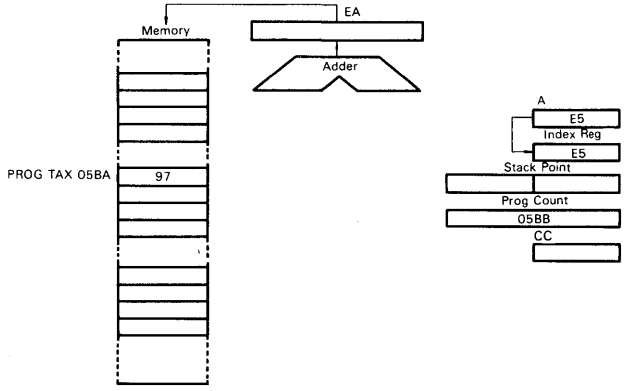


Figure 33 Example of Implied Addressing

■INSTRUCTION SET

There are 62 basic instructions available to the HD6305X0 MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the HD6305X0 MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the HD6305X0 MCU in the alphabetical order.

• Operation Code Map

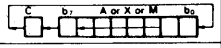
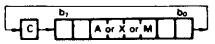
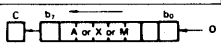
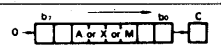
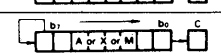
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate OP	#	~	Direct OP	#	~	Extended OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Indexed (16-Bit Offset) OP	#	~	Boolean/ Arithmetic Operation	Condition Code H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	^	^	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	^	^	•
Store A in Memory	STA	–	–	–	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	^	^	•
Store X in Memory	STX	–	–	–	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	^	^	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	^	•	^	^	^
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	^	•	^	^	^
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	^	^	^
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	^	^	^
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	^	^	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	^	^	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	^	^	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	^	^	^
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	^	^	^
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	^	^	•
Jump Unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A) OP	#	~	Implied(X) OP	#	~	Direct OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Boolean/Arithmetic Operation	Condition Code H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	^	^	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	^	^	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	^	^	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	^	^	^
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	^	^	^
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	^	^	^
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	^	^	^
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	^	^
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	^	^	^
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	^	^	^
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	^	^	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	●	●	●	●	●
Branch Never	BRN	21	2	3	None	●	●	●	●	●
Branch IF Higher	BHI	22	2	3	C+Z=0	●	●	●	●	●
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	●	●	●	●	●
Branch IF Carry Clear	BCC	24	2	3	C=0	●	●	●	●	●
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	●	●	●	●	●
Branch IF Carry Set	BCS	25	2	3	C=1	●	●	●	●	●
(Branch IF Lower)	(BLO)	25	2	3	C=1	●	●	●	●	●
Branch IF Not Equal	BNE	26	2	3	Z=0	●	●	●	●	●
Branch IF Equal	BEQ	27	2	3	Z=1	●	●	●	●	●
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	●	●	●	●	●
Branch IF Half Carry Set	BHCS	29	2	3	H=1	●	●	●	●	●
Branch IF Plus	BPL	2A	2	3	N=0	●	●	●	●	●
Branch IF Minus	BMI	2B	2	3	N=1	●	●	●	●	●
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	●	●	●	●	●
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	●	●	●	●	●
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	●	●	●	●	●
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	●	●	●	●	●
Branch to Subroutine	BSR	AD	2	5	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	●	●	●	●	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	●	●	●	●	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	●	●	●	●	●
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes Implied OP	#	~	Boolean Operation	Condition Code H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	•	•	•	•	•
Transfer X to A	TXA	9F	1	2	X→A	•	•	•	•	•
Set Carry Bit	SEC	99	1	1	1→C	•	•	•	•	1
Clear Carry Bit	CLC	98	1	1	0→C	•	•	•	•	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	•	1	•	•	•
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	•	0	•	•	•
Software Interrupt	SWI	83	1	10		•	1	•	•	•
Return from Subroutine	RTS	81	1	5		•	•	•	•	•
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	•	•	•	•	•
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	•	•	•	•	•
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	•	•	∧	∧	∧*
Stop	STOP	8E	1	4		•	•	•	•	•
Wait	WAIT	8F	1	4		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	Condition Code H	I	N	Z	C
ADC		×	×	×		×	×	×			∧	•	∧	∧	∧
ADD		×	×	×		×	×	×			∧	•	∧	∧	∧
AND		×	×	×		×	×	×			•	•	∧	∧	•
ASL	×		×			×	×				•	•	∧	∧	∧
ASR	×		×			×	×				•	•	∧	∧	∧
BCC					×						•	•	•	•	•
BCLR									×		•	•	•	•	•
BCS					×						•	•	•	•	•
BEQ					×						•	•	•	•	•
BHCC					×						•	•	•	•	•
BHCS					×						•	•	•	•	•
BHI					×						•	•	•	•	•
(BHS)					×						•	•	•	•	•
BIH					×						•	•	•	•	•
BIL					×						•	•	•	•	•
BIT		×	×	×		×	×	×			•	•	∧	∧	•
(BLO)					×						•	•	•	•	•
BLS					×						•	•	•	•	•
BMC					×						•	•	•	•	•
BMI					×						•	•	•	•	•
BMS					×						•	•	•	•	•
BNE					×						•	•	•	•	•
BPL					×						•	•	•	•	•
BRA					×						•	•	•	•	•

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						●	●	●	●	●
BRCLR										×	●	●	●	●	∧
BRSET										×	●	●	●	●	∧
BSET									×		●	●	●	●	●
BSR					×						●	●	●	●	●
CLC	×										●	●	●	●	0
CLI	×										●	0	●	●	●
CLR	×		×			×	×				●	●	0	1	●
CMP		×	×	×		×	×	×			●	●	∧	∧	∧
COM	×		×			×	×				●	●	∧	∧	1
CPX		×	×	×		×	×	×			●	●	∧	∧	∧
DAA	×										●	●	∧	∧	∧
DEC	×		×			×	×				●	●	∧	∧	●
EOR		×	×	×		×	×	×			●	●	∧	∧	●
INC	×		×			×	×				●	●	∧	∧	●
JMP			×	×		×	×	×			●	●	●	●	●
JSR			×	×		×	×	×			●	●	●	●	●
LDA		×	×	×		×	×	×			●	●	∧	∧	●
LDX		×	×	×		×	×	×			●	●	∧	∧	●
LSL	×		×			×	×				●	●	∧	∧	∧
LSR	×		×			×	×				●	●	0	∧	∧
NEG	×		×			×	×				●	●	∧	∧	∧
NOP	×										●	●	●	●	●
ORA		×	×	×		×	×	×			●	●	∧	∧	●
ROL	×		×			×	×				●	●	∧	∧	∧
ROR	×		×			×	×				●	●	∧	∧	∧
RSP	×										●	●	●	●	●
RTI	×										?	?	?	?	?
RTS	×										●	●	●	●	●
SBC		×	×	×		×	×	×			●	●	∧	∧	∧
SEC	×										●	●	●	●	1
SEI	×										●	1	●	●	●
STA			×	×		×	×	×			●	●	∧	∧	●
STOP	×										●	●	●	●	●
STX			×	×		×	×	×			●	●	∧	∧	●
SUB		×	×	×		×	×	×			●	●	∧	∧	∧
SWI	×										●	1	●	●	●
TAX	×										●	●	●	●	●
TST	×		×			×	×				●	●	∧	∧	●
TXA	×										●	●	●	●	●
WAIT	×										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI	—					—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

LOW

(NOTES) 1. "—" is an undefined operation code.
2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD6305X0:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

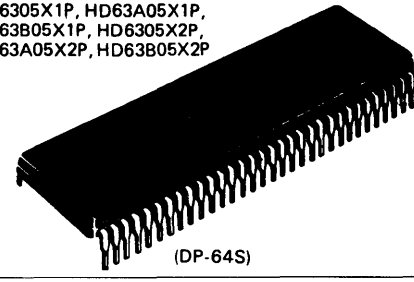
STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

HD6305X1, HD63A05X1, HD63B05X1
HD6305X2, HD63A05X2, HD63B05X2
CMOS MCU (Microcomputer Unit)

The HD6305X1 and the HD6305X2 are memory expandable versions of the HD6305X0, which is CMOS 8-bit single chip microcomputer. A CPU, a clock generator, a 128-byte RAM, I/O terminals, two timers and a serial communication interface (SCI) are built in both chip of the HD6305X1 and the HD 6305X2. Their memory spaces are expandable to 16k bytes externally.

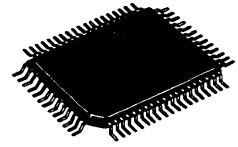
The HD6305X1 and the HD6305X2 have the same functions as the HD6305X0's except for the number of I/O terminals. The HD6305X1 has a 4k byte ROM and its memory space is expandable to 12k bytes externally. The HD6305X2 is a microcomputer unit which includes no ROM and its memory space is expandable to 16k bytes externally.

HD6305X1P, HD63A05X1P,
HD63B05X1P, HD6305X2P,
HD63A05X2P, HD63B05X2P



(DP-64S)

HD6305X1F, HD63A05X1F,
HD63B05X1F, HD6305X2F,
HD63A05X2F, HD63B05X2F



(FP-64)

■ HARDWARE FEATURES

- 8-bit based MCU
- 4k-bytes of internal ROM (HD6305X1)
  No internal ROM (HD6305X2)
- 128-bytes of RAM
- A total of 31 terminals, including 24 I/O's, 7 inputs
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes
  - Wait . . . . In this mode, the clock oscillator is on and the CPU halts but the timer/serial/interrupt function is operatable.
  - Stop . . . . In this mode, the clock stops but the RAM data, I/O status and registers are held.
  - Standby. . In this mode, the clock stops, the RAM data is held, and the other internal condition is reset.
- Minimum instruction cycle time
  - HD6305X1/X2 . . 1 μs (f = 1 MHz)
  - HD63A05X1/X2. . 0.67 μs (f = 1.5 MHz)
  - HD63B05X1/X2. . 0.5 μs (f = 2 MHz)
- Wide operating range
  V_{CC} = 3 to 6V (f = 0.1 to 0.5 MHz)
  - HD6305X1/X2 . . f = 0.1 to 1 MHz (V_{CC} = 5V ± 10%)
  - HD63A05X1/X2. . f = 0.1 to 1.5 MHz (V_{CC} = 5V ± 10%)
  - HD63B05X1/X2. . f = 0.1 to 2 MHz (V_{CC} = 5V ± 10%)
- System development fully supported by an evaluation kit

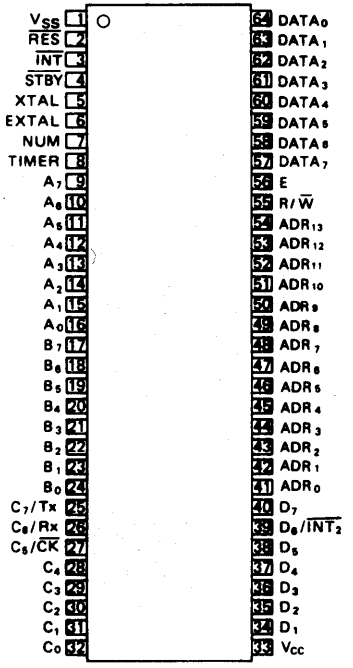
■ SOFTWARE FEATURES

- Similar to HD6800
- Byte efficient instruction set
- Powerful bit manipulation instructions (Bit Set, Bit Clear, and Bit Test and Branch usable for all RAM bits and all I/O terminals)
- A variety of interrupt operations
- Index addressing mode useful for table processing
- A variety of conditional branch instructions
- Ten powerful addressing modes
- All addressing modes adaptable to RAM, and I/O instructions
- Three new instructions, STOP, WAIT and DAA, added to the HD6805 family instruction set
- Instructions that are upward compatible with those of Motorola's MC6805P2 and MC146805G2

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

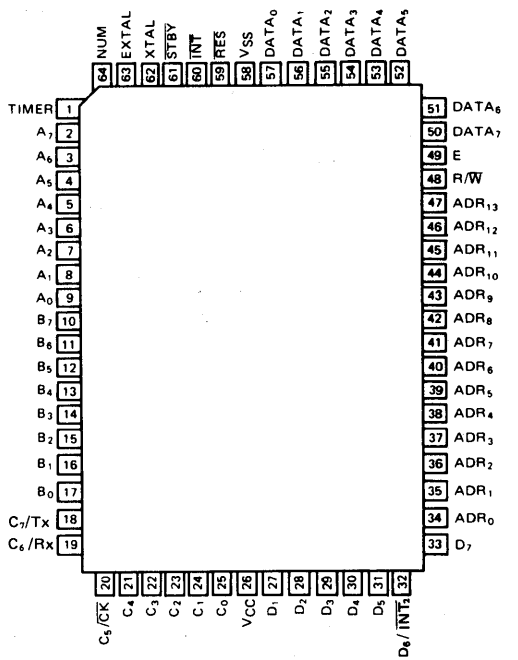
■ PIN ARRANGEMENT

- HD6305X1P, HD63A05X1P, HD63B05X1P, HD6305X2P, HD63A05X2P, HD63B05X2P



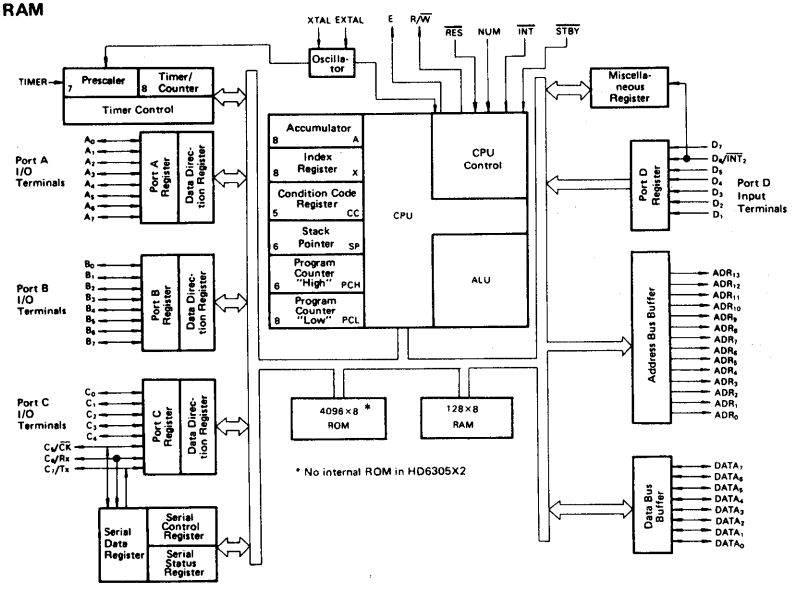
(Top View)

- **HD6305X1F, HD63A05X1F, HD63B05X1F, HD6305X2F, HD63A05X2F, HD63B05X2F**



(Top View)

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 ~ +7.0	V
Input Voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC}+0.3	V
	EXTAL			V_{CC}×0.7	–	V_{CC}+0.3	
	Other Inputs			2.0	–	V_{CC}+0.3	
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input Leakage Current	TIMER, $\overline{INT}$, $D_1 \sim D_7$, $\overline{STBY}$	$\|I_{IL}\|$	Vin = 0.5 ~ V_{CC}-0.5	–	–	1.0	μA
Three-state Current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $ADR_0 \sim ADR_{13}$*, $DATA_0 \sim DATA_7$, E*, R/$\overline{W}$*	$\|I_{TSI}\|$		–	–	1.0	μA
Current Dissipation**	Operating	I_{CC}	f = 1MHz***	–	5	10	mA
	Wait			–	2	5	mA
	Stop			–	2	10	μA
	Standby			–	2	10	μA
Input Capacitance	All Terminals	Cin	f = 1MHz, Vin = 0V	–	–	12	pF

* Only at standby
** V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V
*** The value at f = xMHz is given by using
I_{CC} (f = xMHz) = I_{CC} (f = 1MHz) × x

• AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	HD6305X1/X2			HD63A05X1/X2			HD63B05X1/X2			Unit
			min	typ	max	min	typ	max	min	typ	max	
Cycle Time	t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time	t_{Er}		–	–	20	–	–	20	–	–	20	ns
Enable Fall Time	t_{Ef}		–	–	20	–	–	20	–	–	20	ns
Enable Pulse Width("High" Level)	PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width("Low" Level)	PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address Delay Time	t_{AD}		–	–	250	–	–	190	–	–	180	ns
Address Hold Time	t_{AH}		40	–	–	30	–	–	20	–	–	ns
Data Delay Time	t_{DW}		–	–	200	–	–	160	–	–	120	ns
Data Hold Time (Write)	t_{HW}		40	–	–	30	–	–	20	–	–	ns
Data Set-up Time (Read)	t_{DSR}		80	–	–	60	–	–	50	–	–	ns
Data Hold Time (Read)	t_{HR}		0	–	–	0	–	–	0	–	–	ns

- **PORT TIMING (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD6305X1/X2			HD63A05X1/X2			HD63B05X1/X2			Unit
			min	typ	max	min	typ	max	min	typ	max	
Port Data Set-up Time (Port A, B, C, D)	t_{PDS}	Fig. 2	200	–	–	200	–	–	200	–	–	ns
Port Data Hold Time (Port A, B, C, D)	t_{PDH}		200	–	–	200	–	–	200	–	–	ns
Port Data Delay Time (Port A, B, C)	t_{PDW}	Fig. 3	–	–	300	–	–	300	–	–	300	ns

- **CONTROL SIGNAL TIMING (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD6305X1/X2			HD63A05X1/X2			HD63B05X1/X2			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{INT}$ Pulse Width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ Pulse Width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ Pulse Width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
Control Set-up Time	t_{CS}	Fig. 5	250	–	–	250	–	–	250	–	–	ns
Timer Pulse Width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation Start Time (Crystal)	t_{OSC}	Fig.5, Fig.20*	–	–	20	–	–	20	–	–	20	ms
Reset Delay Time	t_{RHL}	Fig. 19	80	–	–	80	–	–	80	–	–	ms

* C_L = 22pF ±20%, R_s = 60Ω max.

- **SCI TIMING (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD6305X1/X2			HD63A05X1/X2			HD63B05X1/X2			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock Cycle	t_{Scyc}	Fig. 6, Fig. 7	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data Output Delay Time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data Set-up Time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data Hold Time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

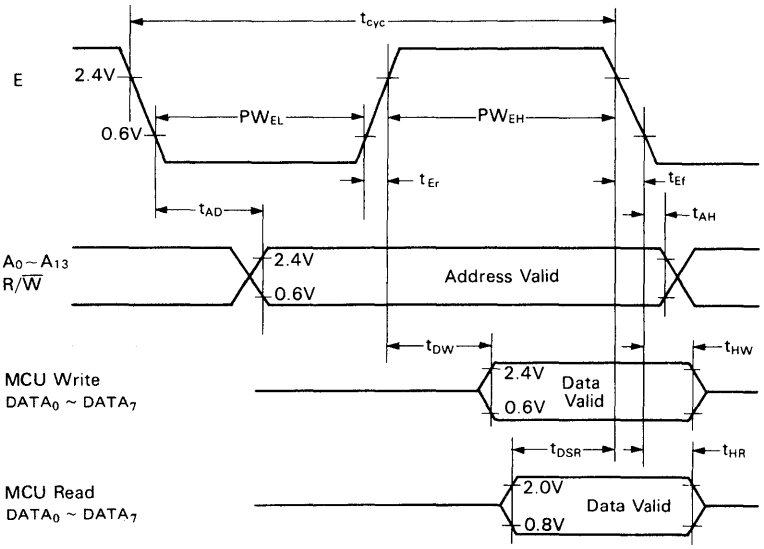


Figure 1 Bus Timing

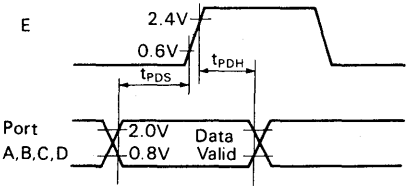


Figure 2 Port Data Set-up and Hold Times (MCU Read)

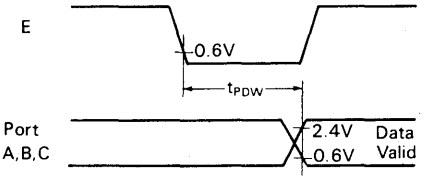


Figure 3 Port Data Delay Time (MCU Write)

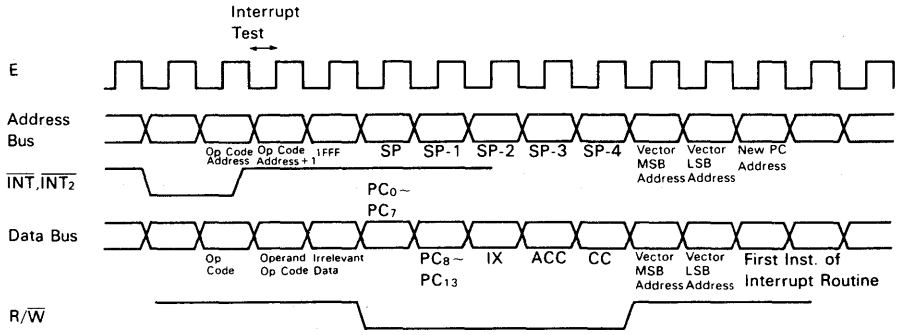


Figure 4 Interrupt Sequence

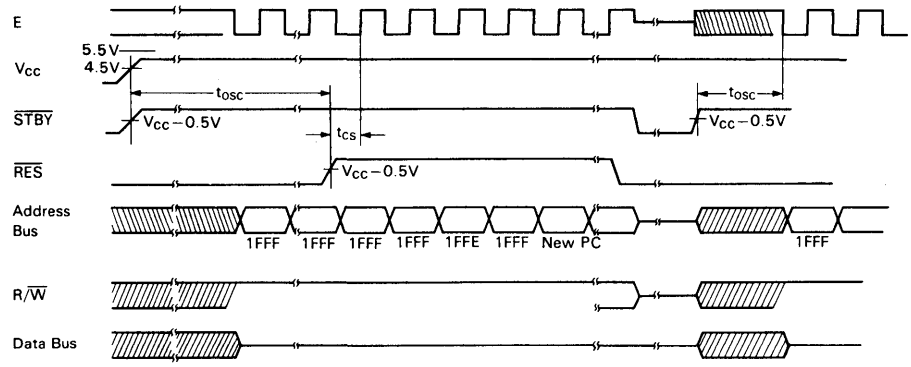


Figure5 Reset Timing

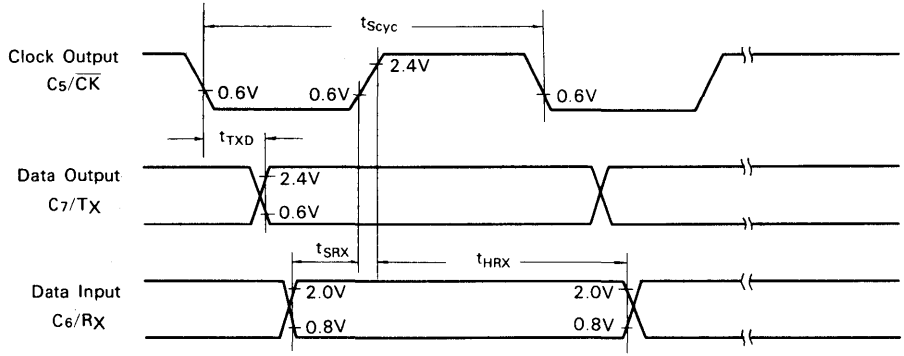


Figure6 SCI Timing (Internal Clock)

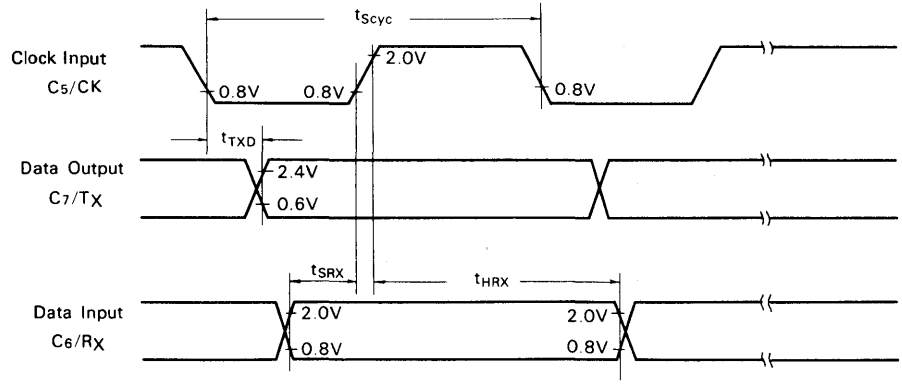
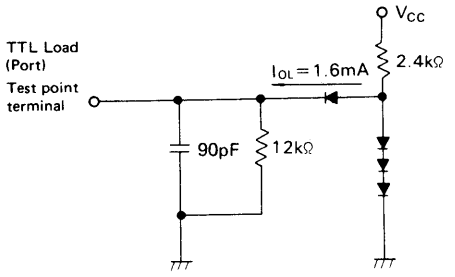


Figure7 SCI Timing(External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ.

Figure 8 Test Load

■ DESCRIPTION OF TERMINAL FUNCTIONS

The input and output signals of the MCU are described here.

• V_{CC}, V_{SS}

Voltage is applied to the MCU through these two terminals. V_{CC} is 5.0V ± 10%, while V_{SS} is grounded.

• $\overline{INT}$, $\overline{INT_2}$

External interrupt request inputs to the MCU. For details, refer to "INTERRUPT". The $\overline{INT_2}$ terminal is also used as the port D_6 terminal.

• XTAL, EXTAL

These terminals provide input to the on-chip clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic filter is connected to the terminal. Refer to "INTERNAL OSCILLATOR" for using these input terminals.

• TIMER

This is an input terminal for event counter. Refer to "TIMER" for details.

• $\overline{RES}$

Used to reset the MCU. Refer to "RESET" for details.

• NUM

This terminal is not for user application. In case of the HD6305X1, this terminal should be connected to V_{CC} through 10kΩ resistance. In case of the HD6305X2, this terminal should be connected to V_{SS}.

• Enable (E)

This output terminal supplies E clock. Output is a single-phase, TTL compatible and 1/4 crystal oscillation frequency or 1/4 external clock frequency. It can drive one TTL load and a 90pF condenser.

• Read/Write (R/$\overline{W}$)

This TTL compatible output signal indicates to peripheral and memory devices whether MCU is in Read ("High"), or in Write ("Low"). The normal standby state is Read ("High"). Its output can drive one TTL load and a 90pF condenser.

• Data Bus ($DATA_0 \sim DATA_7$)

This TTL compatible three-state buffer can drive one TTL load and 90pF.

• Address Bus ($ADR_0 \sim ADR_{13}$)

Each terminal is TTL compatible and can drive one TTL load and 90pF.

• Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$)

These 24 terminals consist of three 8-bit I/O ports (A, B, C). Each of them can be used as an input or output terminal on a bit through program control of the data direction register. For details, refer to "I/O PORTS."

• Input Terminals ($D_1 \sim D_7$)

These seven input-only terminals are TTL or CMOS compatible. Of the port D's, D_6 is also used as $\overline{INT_2}$. If D_6 is used as a port, the $\overline{INT_2}$ interrupt mask bit of the miscellaneous register must be set to "1" to prevent an $\overline{INT_2}$ interrupt from being accidentally accepted.

• $\overline{STBY}$

This terminal is used to place the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and the internal condition is reset. For details, refer to "Stand-by Mode."

The terminals described in the following are I/O pins for serial communication interface (SCI). They are also used as ports C_5, C_6 and C_7. For details, refer to "SERIAL COMMUNICATION INTERFACE."

• $\overline{CK}$ (C_5)

Used to input or output clocks for serial operation.

• Rx (C_6)

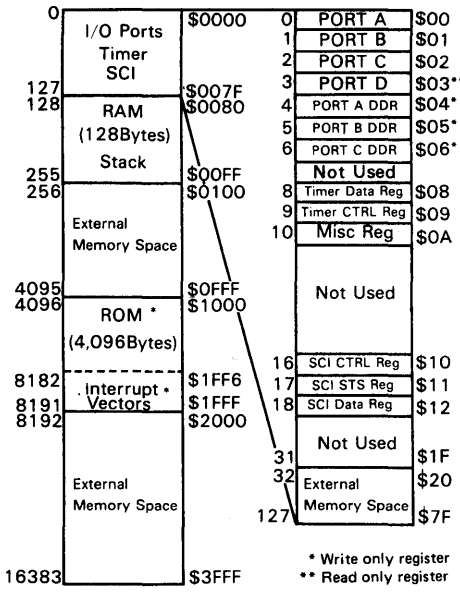
Used to receive serial data.

• Tx (C_7)

Used to transmit serial data.

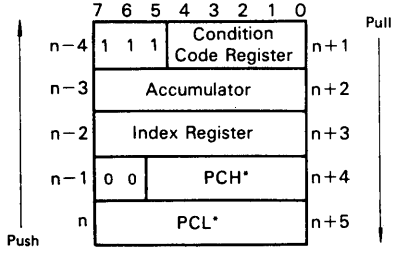
■ MEMORY MAP

The memory map of the MCU is shown in Fig. 9. $1000 ~ $1FFF of the HD6305X2 are external addresses. However, care should be taken to assign vector addresses to $1FF6 ~ $1FFF. During interrupt processing, the contents of the CPU registers are saved into the stack in the sequence shown in Fig. 10. This saving begins with the lower byte (PCL) of the program counter. Then the value of the stack pointer is decremented and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CC) are stacked in that order. In a subroutine call, only the contents of the program counter (PCH and PCL) are stacked.



* ROM area ($1000 ~ $1FFF) in the HD6305X2 is changed into External Memory Space.

Figure 9 Memory Map of MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 10 Sequence of Interrupt Stacking

■REGISTERS

There are five registers which the programmer can operate.

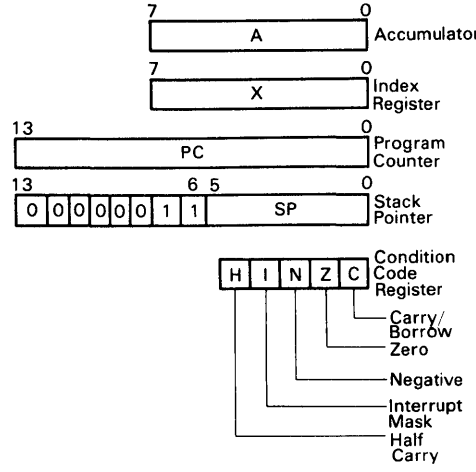


Figure 11 Programming Model

• Accumulator (A)

This accumulator is an ordinary 8-bit register which holds operands or the result of arithmetic operation or data processing.

• Index Register (X)

The index register is an 8-bit register, and is used for index addressing mode. Each of the addresses contained in the register consists of 8 bits which, combined with an offset value, provides an effective address.

In the case of a read/modify/write instruction, the index register can be used like an accumulator to hold operation data or the result of operation.

If not used in the index addressing mode, the register can be used to store data temporarily.

• Program Counter (PC)

The program counter is a 14-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 14-bit register that indicates the address of the next stacking space. Just after reset, the stack pointer is set at address $00FF. It is decremented when data is pushed, and incremented when pulled. The upper 8 bits of the stack pointer are fixed to 00000011. During the MCU being reset or during a reset stack pointer (RSP) instruction, the pointer is set to address $00FF. Since a subroutine or interrupt can use space up to address $00C1 for stacking, the subroutine can be used up to 31 levels and the interrupt up to 12 levels.

• Condition Code Register (CC)

The condition code register is a 5-bit register, each bit indicating the result of the instruction just executed. The bits can be individually tested by conditional branch instruc-

tions. The CC bits are as follows:

Half Carry (H): Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts, except a software interrupt, to be masked. If an interrupt occurs with the bit I set, it is latched. It will be processed the instant the interrupt mask bit is reset. (More specifically, it will enter the interrupt processing routine after the instruction following the CLI has been executed.)

Negative (N): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (bit 7 is logic "1").

Zero (Z): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is zero.

Carry/ Borrow (C): Represents a carry or borrow that occurred in the most recent arithmetic operation. This bit is also affected by the Bit Test and Branch instruction and a Rotate instruction.

■ INTERRUPT

There are six different types of interrupt: external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER or the SCI and $TIMER_2$ generate the same vector address, respectively.

When an interrupt occurs, the program in progress stops and the then CPU status is saved onto the stack. And then, the interrupt mask bit (I) of the condition code register is set and the start address of the interrupt processing routine is obtained from a particular interrupt vector address. Then the interrupt routine starts from the start address. System can exit from the interrupt routine by an RTI instruction. When this instruction is executed, the CPU status before the interrupt (saved onto the stack) is pulled and the CPU restarts the sequence with the instruction next to the one at which the interrupt occurred. Table 1 lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$1FFE, $1FFF
SWI	2	$1FFC, $1FFD
$\overline{INT}$	3	$1FFA, $1FFB
TIMER/$\overline{INT_2}$	4	$1FF8, $1FF9
SCI/$TIMER_2$	5	$1FF6, $1FF7

A flowchart of the interrupt sequence is shown in Fig. 12. A block diagram of the interrupt request source is shown in Fig. 13.

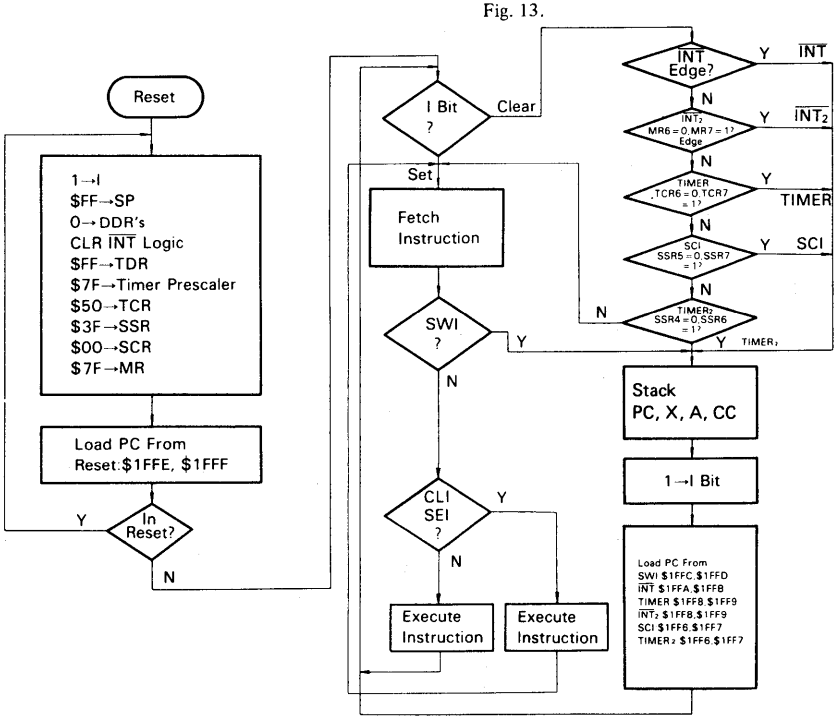


Figure 12 Interrupt Flow Chart

In the block diagram, both the external interrupts $\overline{INT}$ and $\overline{INT_2}$ are edge trigger inputs. At the falling edge of each input, an interrupt request is generated and latched. The $\overline{INT}$ interrupt request is automatically cleared if jumping is made to the $\overline{INT}$ processing routine. Meanwhile, the $\overline{INT_2}$ request is cleared if "0" is written in bit 7 of the miscellaneous register.

For the external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, TIMER$_2$) and serial interrupt (SCI), each interrupt request is held, but not processed, if the I bit of the condition code register is set. Immediately after the I bit is cleared, the corresponding interrupt processing starts according to the priority.

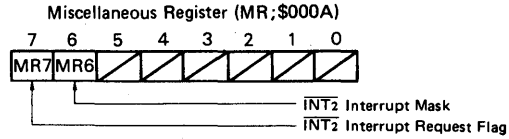
The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by setting bit 6 of the timer control register; the SCI interrupt by setting bit 5 of the serial status register; and the TIMER$_2$ interrupt by setting bit 4 of the serial status register.

The status of the $\overline{INT}$ terminal can be tested by a BIL or BIH instruction. The $\overline{INT}$ falling edge detector circuit and its latching circuit are independent of testing by these instructions. This is also true with the status of the $\overline{INT_2}$ terminal.

• Miscellaneous Register (MR; $000A)

The interrupt vector address for the external interrupt $\overline{INT_2}$ is the same as that for the TIMER interrupt, as shown in Table 1. For this reason, a special register called the miscellaneous register (MR; $000A) is available to control the $\overline{INT_2}$ interrupts.

Bit 7 of this register is the $\overline{INT_2}$ interrupt request flag. When the falling edge is detected at the $\overline{INT_2}$ terminal, "1" is set in bit 7. Then the software in the interrupt routine (vector addresses: $1FF8, $1FF9) checks bit 7 to see if it is $\overline{INT_2}$ interrupt. Bit 7 can be reset by software.



Miscellaneous Register (MR; $000A)

Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If this bit is set to "1", then the $\overline{INT_2}$ interrupt is disabled. Both read and write are possible with bit 7 but "1" cannot be written in this bit by software. This means that an interrupt request by software is impossible.

When reset, bit 7 is cleared to "0" and bit 6 is set to "1"

■ TIMER

Figure 14 shows a MCU timer block diagram. The timer data register is loaded by software and, upon receipt of a clock input, begins to count down. When the timer data

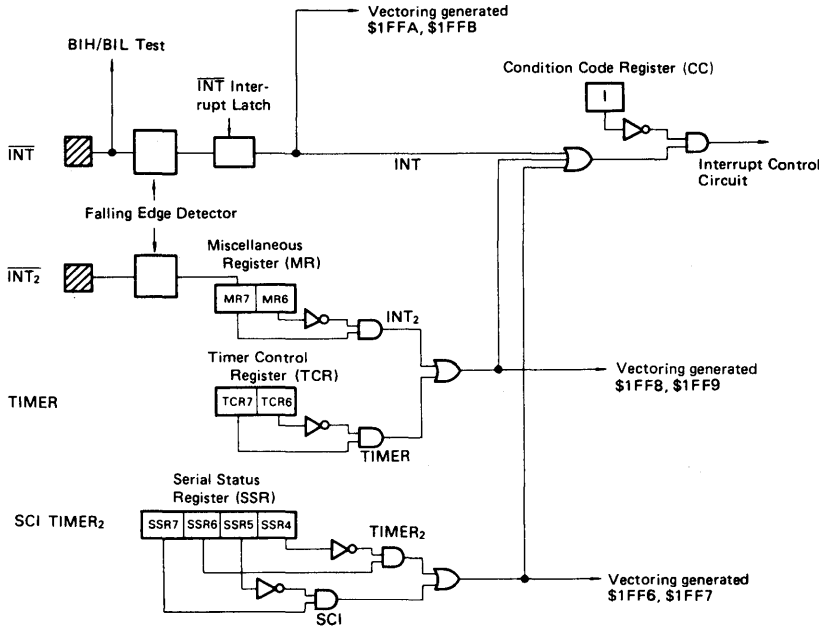


Figure 13 Interrupt Request Generation Circuitry

register (TDR) becomes "0", the timer interrupt request bit (bit 7) in the timer control register is set. In response to the interrupt request, the CPU saves its status into the stack and fetches timer interrupt routine address from addresses $1FF8 and $1FF9 and execute the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also mask the timer interrupt.

The source clock to the timer can be either an external signal from the timer input terminal or the internal E signal (the oscillator clock divided by 4). If the E signal is used as the source, the clock input can be gated by the input to the timer input terminal.

Once the timer count has reached "0", it starts counting down with "$FF". The count can be monitored whenever desired by reading the timer data register. This permits the program to know the length of time having passed after the occurrence of a timer interrupt, without disturbing the contents of the counter.

When the MCU is reset, both the prescaler and counter are initialized to logic "1". The timer interrupt request bit (bit 7) then is cleared and the timer interrupt mask bit (bit 6) is set.

To clear the timer interrupt request bit (bit 7), it is necessary to write "0" in that bit.

TCR7	Timer interrupt request
0	Absent
1	Present

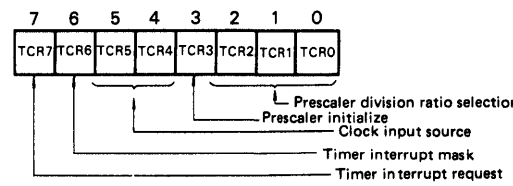
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

• Timer Control Register (TCR; $0009)

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After reset, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$FF" immediately after reset.

When "1" is written in bit 3, the prescaler is initialized. This bit always shows "0" when read.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under timer terminal control
1	0	No clock input (counting stopped)
1	1	Event input from timer terminal

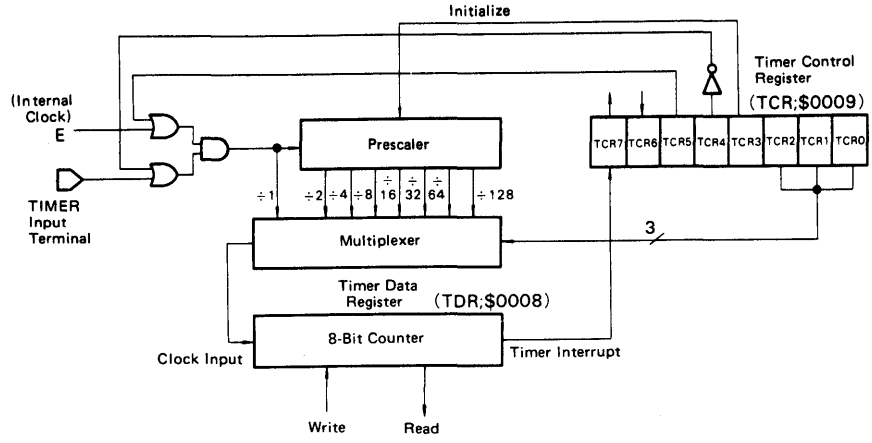


Figure 14 Timer Block Diagram

A prescaler division ratio is selected by the combination of three bits (bits 0, 1 and 2) of the timer control register (see Table 3). There are eight different division ratios: ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128. After reset, the TCR is set to the ÷1 mode.

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

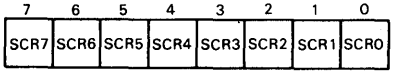
A timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. This bit can be cleared by writing "0" in that bit.

■SERIAL COMMUNICATION INTERFACE (SCI)

This interface is used for serial transmission or reception of 8-bit data. Sixteen transfer rates are available in the range from 1 μs to approx. 32 ms (for oscillation at 4 MHz).

The SCI consists of three registers, one eighth counter and one prescaler. (See Fig. 15.) SCI communicates with the CPU via the data bus, and with the outside world through bits 5, 6 and 7 of port C. Described below are the operations of each register and data transfer.

●SCI Control Register (SCR; $0010)



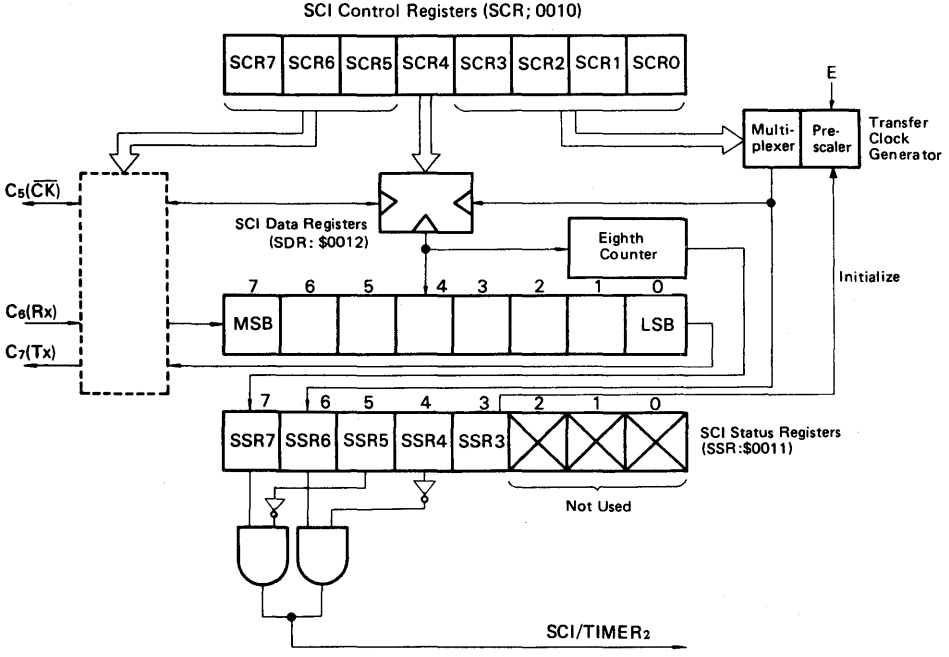


Figure 15 SCI Block Diagram

SCR7	C_7 terminal
0	Used as I/O terminal (by DDR).
1	Serial data output (DDR output)

SCR6	C_6 terminal
0	Used as I/O terminal (by DDR).
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	C_5 terminal
0	0	–	Used as I/O terminal (by DDR).
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)
When this bit is set, the DDR corresponding to the C_7 becomes "1" and this terminal serves for output of SCI data. After reset, the bit is cleared to "0".

Bit 6 (SCR6)
When this bit is set, the DDR corresponding to the C_6 becomes "0" and this terminal serves for input of SCI data. After reset, the bit is cleared to "0".

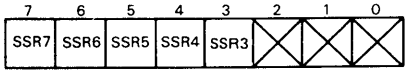
Bits 5 and 4 (SCR5, SCR4)
These bits are used to select a clock source. After reset, the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)
These bits are used to select a transfer clock rate. After reset, the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

•SCI Data Register (SDR; $0012)
A serial-parallel conversion register that is used for transfer of data.

•SCI Status Register (SSR; $0011)



Bit 7 (SSR7)
Bit 7 is the SCI interrupt request bit which is set upon completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5="1". The bit can also be cleared by writing "0" in it.

Bit 6 (SSR6)
Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is used commonly with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When reset, the bit is cleared. It also be cleared by writing "0" in it. (For details, see $TIMER_2$.)

Bit 5 (SSR5)
Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When reset, it is set to "1".

Bit 4 (SSR4)
Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When reset, it is set to "1".

Bit 3 (SSR3)
When "1" is written in this bit, the prescaler of the transfer clock generator is initialized. When read, the bit always is "0".

Bits 2 ~ 0
Not used.

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

• Data Transmission
By writing the desired control bits into the SCI control registers, a transfer rate and a source of transfer clock are determined and bits 7 and 5 of port C are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the C_7/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock. (See Fig. 16.) When 8 bit of

data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register with the rising edge of the last serial clock. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the C_7/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the C_5/$\overline{CK}$ terminal is set as input. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

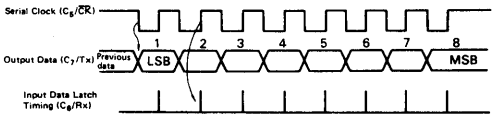


Figure 16 SCI Timing Chart

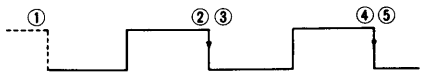
• Data Reception

By writing the desired control bits into the SCI control register, a transfer rate and a source of transfer clock are determined and bit 6 and 5 of port C are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed after reading subsequent received data. It must be taken after reset and after not reading subsequent received data.

The data from the C_6/Rx terminal is input to the SCI data register synchronously with the rising edge of the serial clock (see Fig. 16). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored and the data is received synchronously with the clock from the C_5/$\overline{CK}$ terminal. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

• TIMER$_2$

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 ~ 0 of the SCI control register (4 μs ~ approx. 32 ms (for oscillation at 4 MHz)) is input to bit 6 of the SCI status register and the TIMER$_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, TIMER$_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : TIMER$_2$ interrupt request
③, ⑤ : TIMER$_2$ interrupt request bit cleared

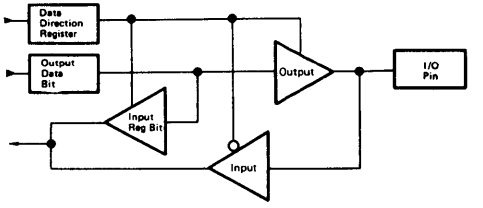
TIMER$_2$ is commonly used with the SCI transfer clock generator. If wanting to use TIMER$_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■I/O PORTS

There are 24 input/output terminals (ports A, B, C). Each I/O terminal can be selected for either input or output by the data direction register. More specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B or C reads latched data if it has been programmed as output, even with the output level being fluctuated by the output load. (See Fig. 17.)

When reset, the data direction register and data register go to "0" and all the input/output terminals are used as input.



Bit of data direction register	Bit of output data	Status of output	Input to CPU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

Figure 17 Input/Output Port Diagram

Seven input-only terminals are available (port D). Writing to an input terminal is invalid.

All input/output terminals and input terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports or input ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite that they are not used.

■RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 18.) On power up, the reset input must be held "Low" for at least t_{OSC} to assure that the internal oscillator is stabilized. A sufficient time of delay can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 19.

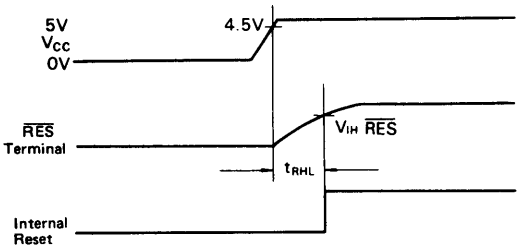


Figure 18 Power On and Reset Timing

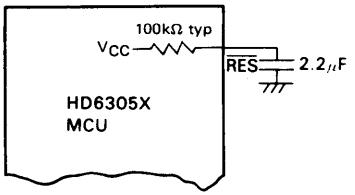


Figure 19 Input Reset Delay Circuit

■INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0MHz) or ceramic oscillator between pins 5 and 6 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 20. Figs. 21 and 22 illustrate the specifications and typical arrangement of the crystal, respectively.

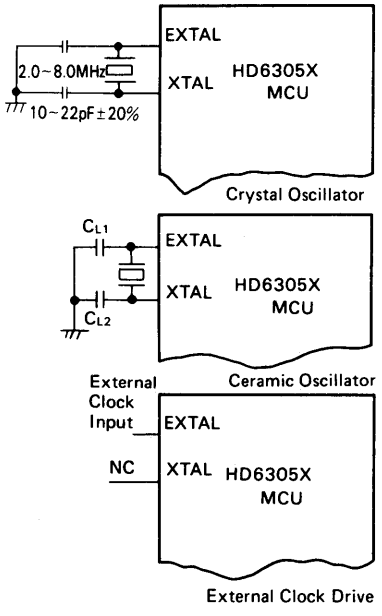


Figure 20 Internal Oscillator Circuit

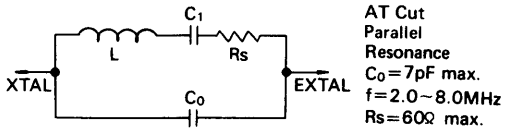
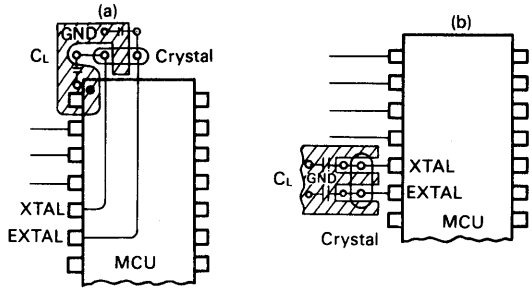


Figure 21 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 22 Typical Crystal Arrangement

■LOW POWER DISSIPATION MODE

The HD6305X has three low power dissipation modes: wait, stop and standby.

• Wait Mode

When WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers, RAM and I/O terminals hold their condition just before entering into the wait mode.

The escape from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the wait mode the MCU executes the instruction next to the WAIT. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$) is masked by the timer control

register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 23 shows a flowchart for the wait function.

• Stop Mode

When STOP instruction being executed, MCU enters into the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive but the RAM, registers and I/O terminals hold their condition just before entering into the stop mode.

The escape from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings into the standby mode.

When interrupt is requested to the CPU and accepted, the stop mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the stop mode, the MCU executes the instruction next to the STOP. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 24 shows a flowchart for the stop function. Fig. 25 shows a timing chart of return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$="0" must exceed t_{osc} to assure stabilized oscillation.

• Standby Mode

The MCU enters into the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are hold. The I/O terminals turn to high-impedance state. The standby mode should escape by bringing $\overline{STBY}$ "High". The CPU must be restarted by reset. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 26.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 27.

(Note)

When I bit of condition code register is "1" and interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$, SCI/$TIMER_2$) is held, MCU does not enter WAIT mode by the execution of WAIT instruction.

In that case, after the 4 dummy cycles MCU executes the next instruction.

In the same way, when external interrupts ($\overline{INT}$, $\overline{INT_2}$) are held at the bit I set, MCU does not enter STOP mode by the execution of STOP instruction. In that case, also, MCU executes the next instruction after the 4 dummy cycles.

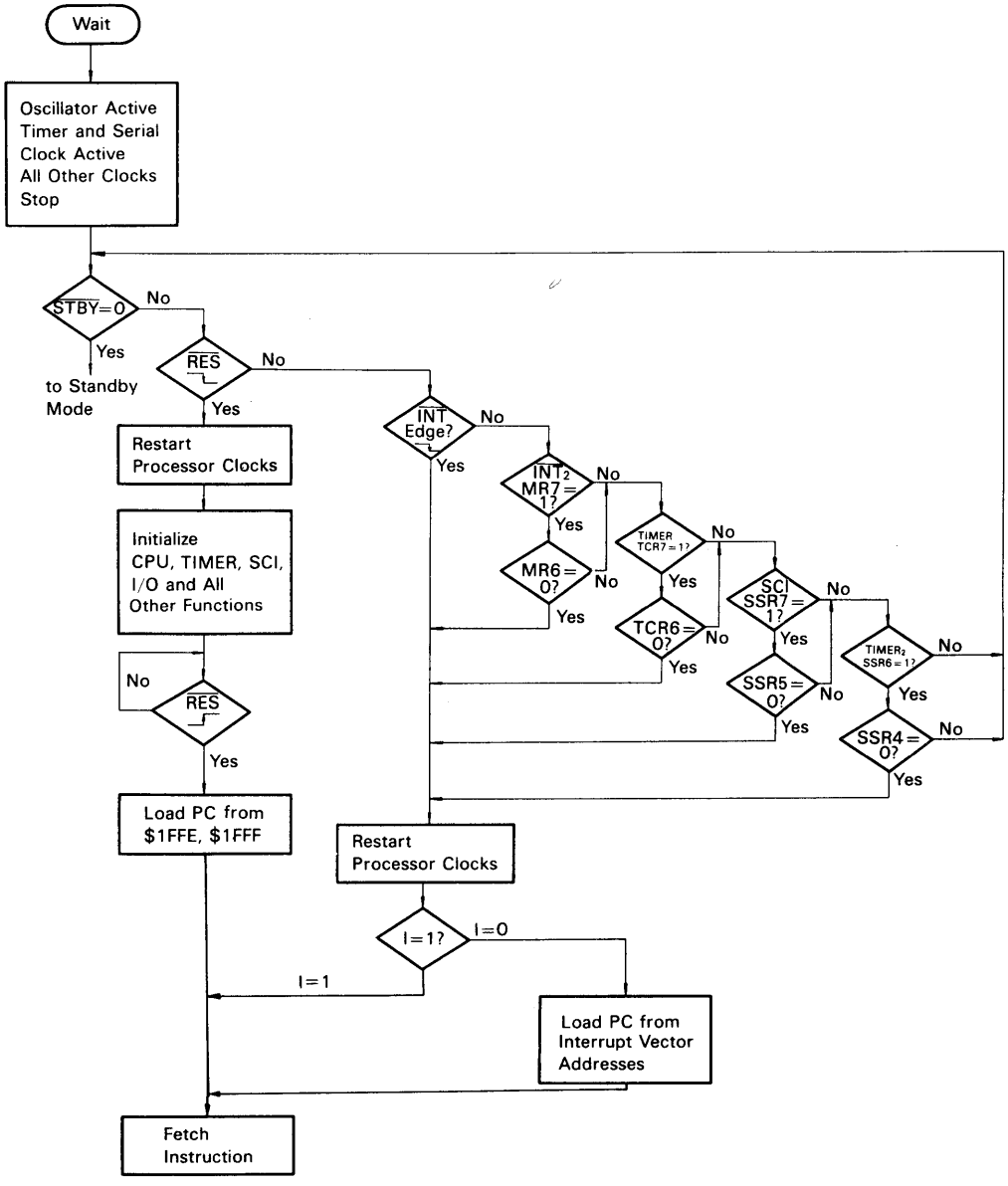


Figure 23 Wait Mode Flow Chart

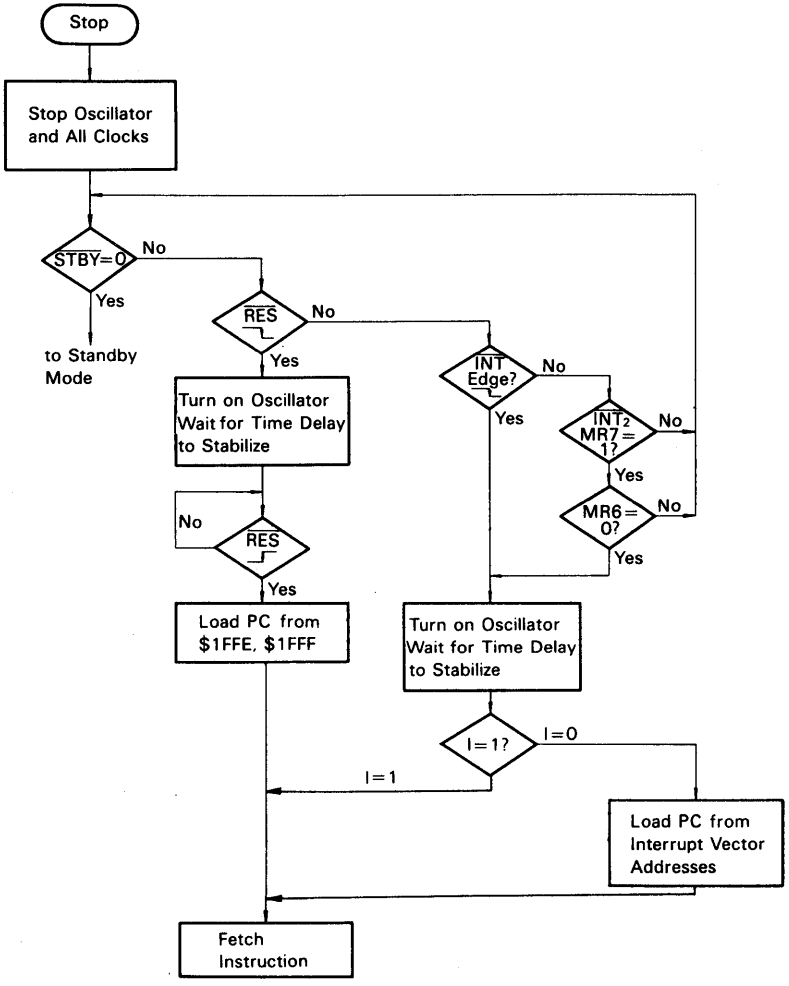
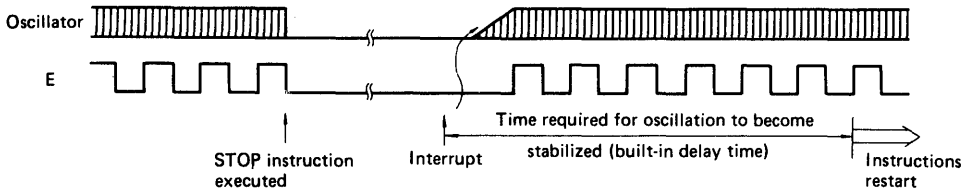
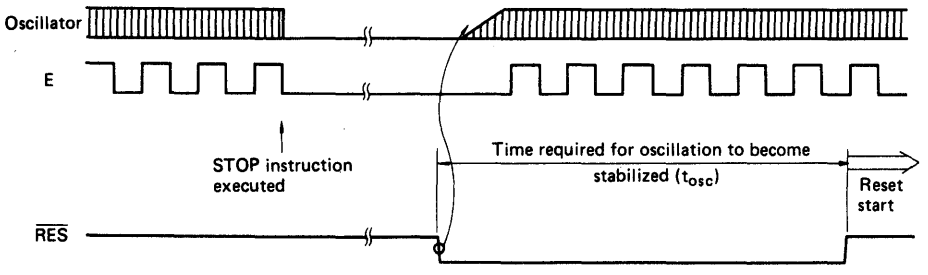


Figure 24 Stop Mode Flow Chart



(a) Restart by Interrupt



(b) Restart by Reset

Figure 25 Timing Chart of Releasing from Stop Mode

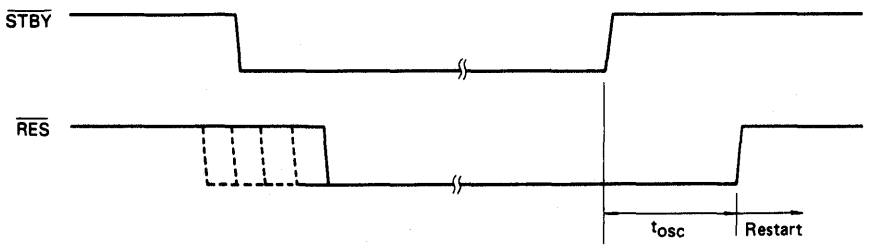


Figure 26 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Keep	High impedance	$\overline{STBY}$="High"

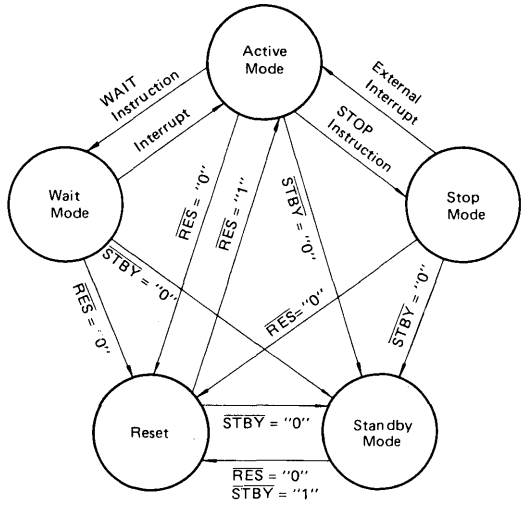


Figure 27 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■BIT MANIPULATION

The MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM or an I/O port (except the write-only registers such as the data direction register). Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM, or I/O can be manipulated, the user may use a bit within the RAM as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 28 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the ROM. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
          ⋮
SELF 1.   BRCLR 0, PORT A, SELF 1
          BSET 1, PORT A
          BCLR 1, PORT A
          ⋮
```

Figure 28 Example of Bit Manipulation

■ADDRESSING MODES

Ten different addressing modes are available to the MCU.

• Immediate

See Fig. 29. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 30. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. All RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 31. The extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 32. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 33. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 34. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 35. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 36. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 37. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The value of the test bit is written in the carry bit of the condition code register.

• Implied

See Fig. 38. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

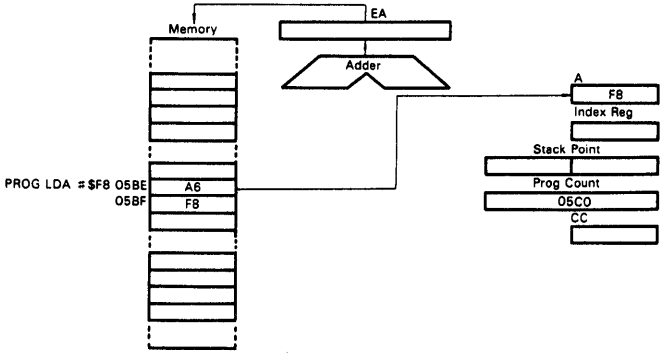


Figure 29 Example of Immediate Addressing

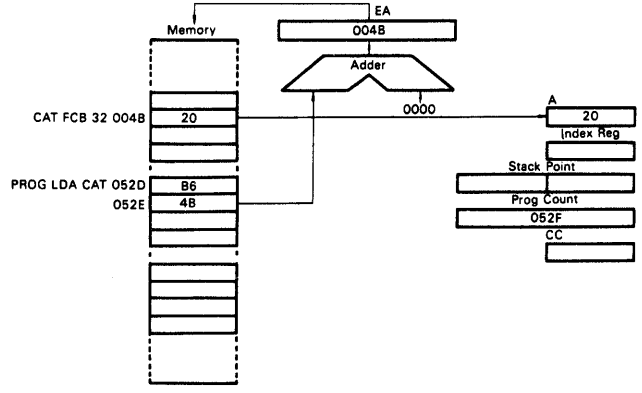


Figure 30 Example of Direct Addressing

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

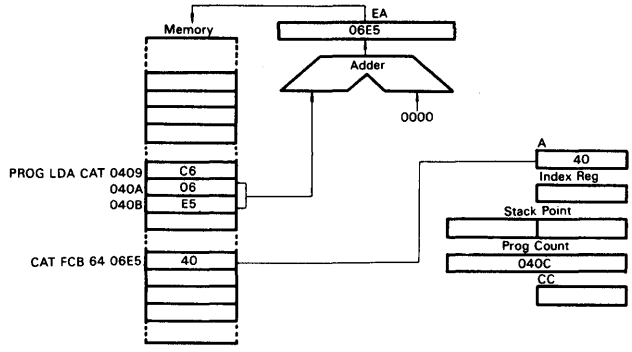


Figure 31 Example of Extended Addressing

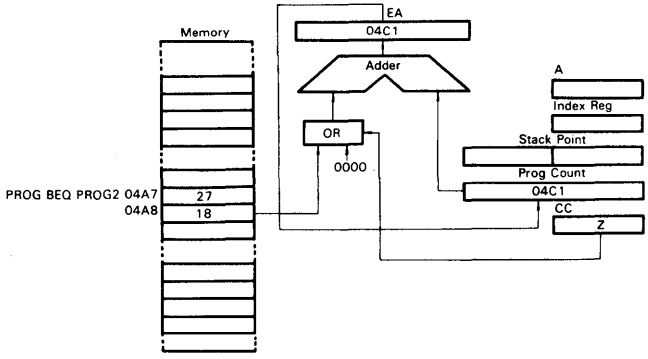


Figure 32 Example of Relative Addressing

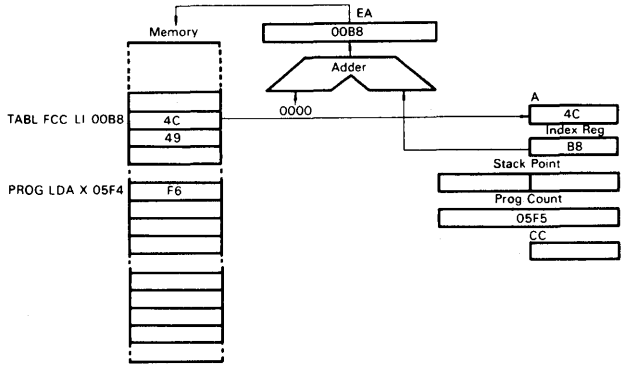


Figure 33 Example of Indexed (No Offset) Addressing

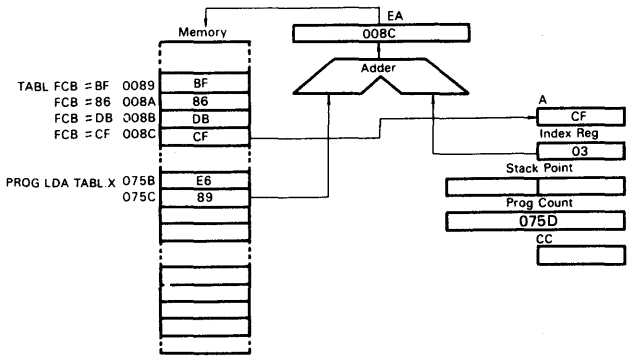


Figure 34 Example of Index (8-bit Offset) Addressing

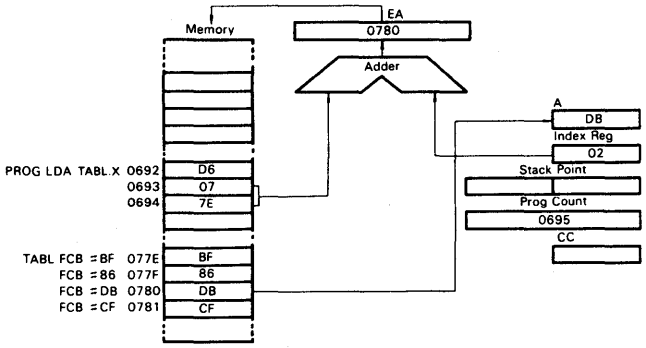


Figure 35 Example of Index (16-bit Offset) Addressing

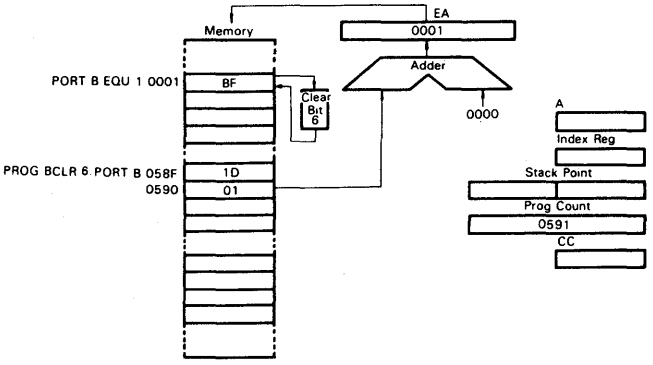


Figure 36 Example of Bit Set/Clear Addressing

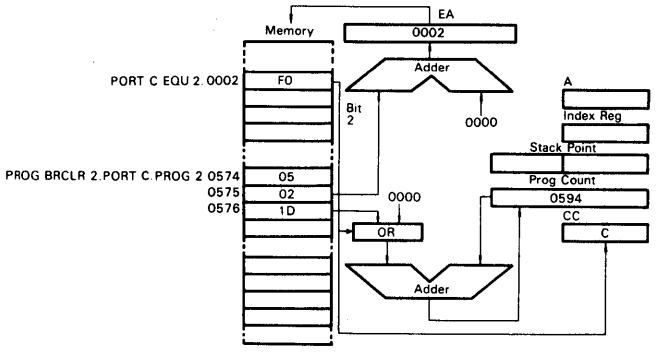


Figure 37 Example of Bit Test and Branch Addressing

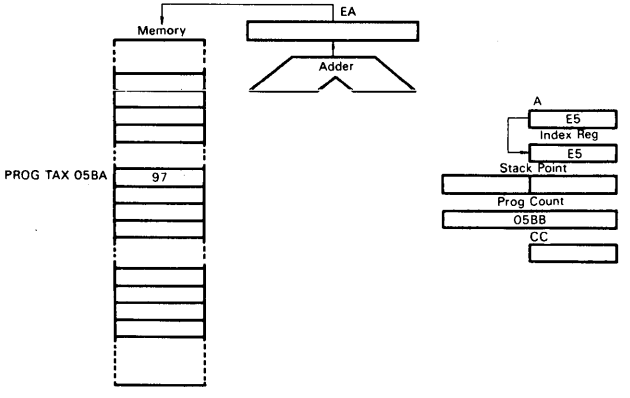


Figure 38 Example of Implied Addressing

■INSTRUCTION SET

There are 62 basic instructions available to the HD6305X MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the HD6305X MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the HD6305X MCU in the alphabetical order.

• Operation Code Map

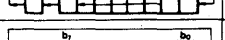
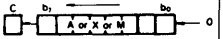
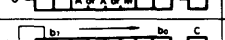
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate OP	#	~	Direct OP	#	~	Extended OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Indexed (16-Bit Offset) OP	#	~	Boolean/ Arithmetic Operation	Condition Code H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	^	^	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	^	^	•
Store A in Memory	STA	–	–	–	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	^	^	•
Store X in Memory	STX	–	–	–	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	^	^	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	^	•	^	^	^
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	^	•	^	^	^
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	^	^	^
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	^	^	^
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	^	^	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	^	^	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	^	^	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	^	^	^
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	^	^	^
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	^	^	•
Jump Unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A) OP	#	~	Implied(X) OP	#	~	Direct OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Boolean/Arithmetic Operation	Condition Code H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	^	^	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	^	^	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	^	^	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	^	^	^
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	^	^	^
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	^	^	^
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	^	^	^
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	^	^
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	^	^	^
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	^	^	^
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	^	^	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	●	●	●	●	●
Branch Never	BRN	21	2	3	None	●	●	●	●	●
Branch IF Higher	BHI	22	2	3	C+Z=0	●	●	●	●	●
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	●	●	●	●	●
Branch IF Carry Clear	BCC	24	2	3	C=0	●	●	●	●	●
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	●	●	●	●	●
Branch IF Carry Set	BCS	25	2	3	C=1	●	●	●	●	●
(Branch IF Lower)	(BLO)	25	2	3	C=1	●	●	●	●	●
Branch IF Not Equal	BNE	26	2	3	Z=0	●	●	●	●	●
Branch IF Equal	BEQ	27	2	3	Z=1	●	●	●	●	●
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	●	●	●	●	●
Branch IF Half Carry Set	BHCS	29	2	3	H=1	●	●	●	●	●
Branch IF Plus	BPL	2A	2	3	N=0	●	●	●	●	●
Branch IF Minus	BMI	2B	2	3	N=1	●	●	●	●	●
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	●	●	●	●	●
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	●	●	●	●	●
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	●	●	●	●	●
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	●	●	●	●	●
Branch to Subroutine	BSR	AD	2	5	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	●	●	●	●	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	●	●	●	●	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	●	●	●	●	●
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes: Implied OP	#	~	Boolean Operation	Condition Code: H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	●	●	●	●	●
Transfer X to A	TXA	9F	1	2	X→A	●	●	●	●	●
Set Carry Bit	SEC	99	1	1	1→C	●	●	●	●	1
Clear Carry Bit	CLC	98	1	1	0→C	●	●	●	●	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	●	1	●	●	●
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	●	0	●	●	●
Software Interrupt	SWI	83	1	10		●	1	●	●	●
Return from Subroutine	RTS	81	1	5		●	●	●	●	●
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	●	●	●	●	●
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	●	●	●	●	●
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	●	●	∧	∧	∧*
Stop	STOP	8E	1	4		●	●	●	●	●
Wait	WAIT	8F	1	4		●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes: Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	Condition Code: H	I	N	Z	C
ADC		×	×	×		×	×	×			∧	●	∧	∧	∧
ADD		×	×	×		×	×	×			∧	●	∧	∧	∧
AND		×	×	×		×	×	×			●	●	∧	∧	●
ASL	×		×			×	×				●	●	∧	∧	∧
ASR	×		×			×	×				●	●	∧	∧	∧
BCC					×						●	●	●	●	●
BCLR									×		●	●	●	●	●
BCS					×						●	●	●	●	●
BEQ					×						●	●	●	●	●
BHCC					×						●	●	●	●	●
BHCS					×						●	●	●	●	●
BHI					×						●	●	●	●	●
(BHS)					×						●	●	●	●	●
BIH					×						●	●	●	●	●
BIL					×						●	●	●	●	●
BIT		×	×	×		×	×	×			●	●	∧	∧	●
(BLO)					×						●	●	●	●	●
BLS					×						●	●	●	●	●
BMC					×						●	●	●	●	●
BMI					×						●	●	●	●	●
BMS					×						●	●	●	●	●
BNE					×						●	●	●	●	●
BPL					×						●	●	●	●	●
BRA					×						●	●	●	●	●

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						●	●	●	●	●
BRCLR										×	●	●	●	●	^
BRSET										×	●	●	●	●	^
BSET									×		●	●	●	●	●
BSR					×						●	●	●	●	●
CLC	×										●	●	●	●	0
CLI	×										●	0	●	●	●
CLR	×		×			×	×				●	●	0	1	●
CMP		×	×	×		×	×	×			●	●	^	^	^
COM	×		×			×	×				●	●	^	^	1
CPX		×	×	×		×	×	×			●	●	^	^	^
DAA	×										●	●	^	^	^
DEC	×		×			×	×				●	●	^	^	●
EOR		×	×	×		×	×	×			●	●	^	^	●
INC	×		×			×	×				●	●	^	^	●
JMP			×	×		×	×	×			●	●	●	●	●
JSR			×	×		×	×	×			●	●	●	●	●
LDA		×	×	×		×	×	×			●	●	^	^	●
LDX		×	×	×		×	×	×			●	●	^	^	●
LSL	×		×			×	×				●	●	^	^	^
LSR	×		×			×	×				●	●	0	^	^
NEG	×		×			×	×				●	●	^	^	^
NOP	×										●	●	●	●	●
ORA		×	×	×		×	×	×			●	●	^	^	●
ROL	×		×			×	×				●	●	^	^	^
ROR	×		×			×	×				●	●	^	^	^
RSP	×										●	●	●	●	●
RTI	×										?	?	?	?	?
RTS	×										●	●	●	●	●
SBC		×	×	×		×	×	×			●	●	^	^	^
SEC	×										●	●	●	●	1
SEI	×										●	1	●	●	●
STA			×	×		×	×	×			●	●	^	^	●
STOP	×										●	●	●	●	●
STX			×	×		×	×	×			●	●	^	^	●
SUB		×	×	×		×	×	×			●	●	^	^	^
SWI	×										●	1	●	●	●
TAX	×										●	●	●	●	●
TST	×		×			×	×				●	●	^	^	●
TXA	×										●	●	●	●	●
WAIT	×										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI	—					—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

LOW

(NOTES) 1. "—" is an undefined operation code.

2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD6305X:

DAA Converts the contents of the accumulator into BCD code.

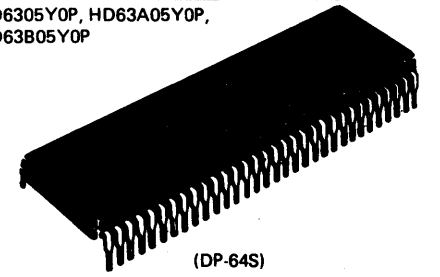
WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

HD6305Y0, HD63A05Y0, HD63B05Y0
CMOS MCU (Microcomputer Unit)

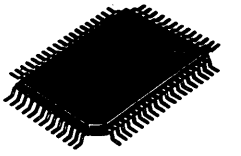
HD6305Y0 is a CMOS 8-bit single-chip microcomputer which includes a CPU upward compatible with the HD6305X0. On the chip of the HD6305Y0, 7872 byte ROM, 256 byte RAM, 55 I/O terminals, two timers and a serial communication interface (SCI) are built in. And three low power dissipation modes (stop, wait and standby) support the low power operating. Instruction set is upward compatible with the HD6805 family.

HD6305Y0P, HD63A05Y0P, HD63B05Y0P



(DP-64S)

HD6305Y0F, HD63A05Y0F, HD63B05Y0F



(FP-64)

■ HARDWARE FEATURES

- 8-bit based MCU
- 7872 bytes of ROM
- 256 bytes of RAM
- A total of 55 terminals, including 32 I/O's, 7 inputs and 16 outputs
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes
  - Wait .... In this mode, the clock oscillator is on and the CPU halts but the timer/serial/interrupt function is operatable.
  - Stop .... In this mode, the clock stops but the RAM data, I/O status and registers are held.
  - Standby.. In this mode, the clock stops, the RAM data is held, and the other internal condition is reset.
- Minimum instruction cycle time
  - HD6305Y0 .... 1 μs (f = 1 MHz)
  - HD63A05Y0 .... 0.67 μs (f = 1.5 MHz)
  - HD63B05Y0 .... 0.5 μs (f = 2 MHz)
- Wide operating range
  V_{CC} = 3 to 6V (f = 0.1 to 0.5 MHz)
  - HD6305Y0 .... f = 0.1 to 1 MHz (V_{CC} = 5V ± 10%)
  - HD63A05Y0 .... f = 0.1 to 1.5 MHz (V_{CC} = 5V ± 10%)
  - HD63B05Y0 .... f = 0.1 to 2 MHz (V_{CC} = 5V ± 10%)
- System development fully supported by an evaluation kit

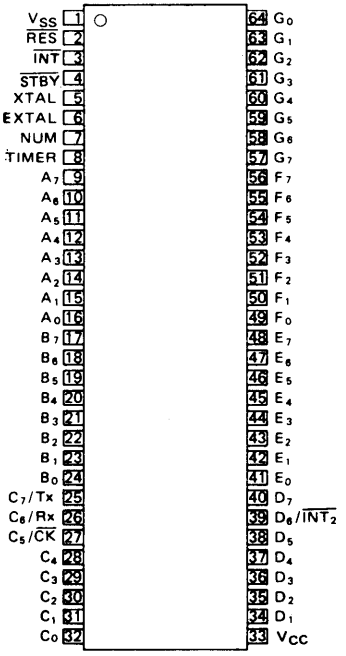
■ SOFTWARE FEATURES

- Similar to HD6800
- Byte efficient instruction set
- Powerful bit manipulation instructions (Bit Set, Bit Clear, and Bit Test and Branch usable for 192 byte RAM bits within page 0 and all I/O terminals)
- A variety of interrupt operations
- Index addressing mode useful for table processing
- A variety of conditional branch instructions
- Ten powerful addressing modes
- All addressing modes adaptable to RAM, and I/O instructions
- Three new instructions, STOP, WAIT and DAA, added to the HD6805 family instruction set
- Instructions that are upward compatible with those of Motorola's MC6805P2 and MC146805G2

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

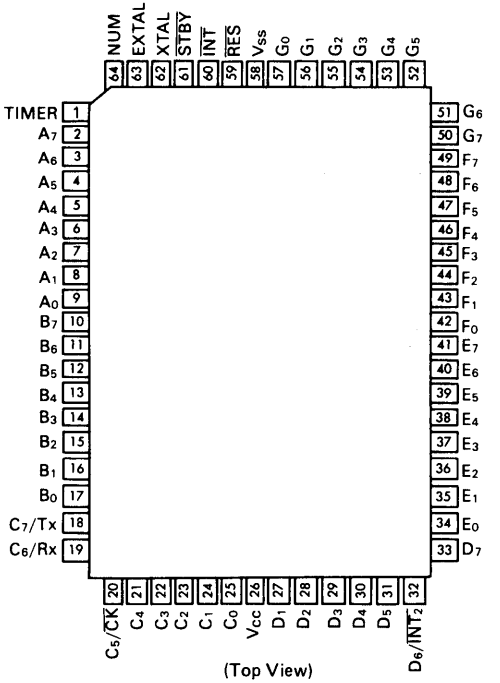
■ PIN ARRANGEMENT

● HD6305Y0P, HD63A05Y0P, HD63B05Y0P



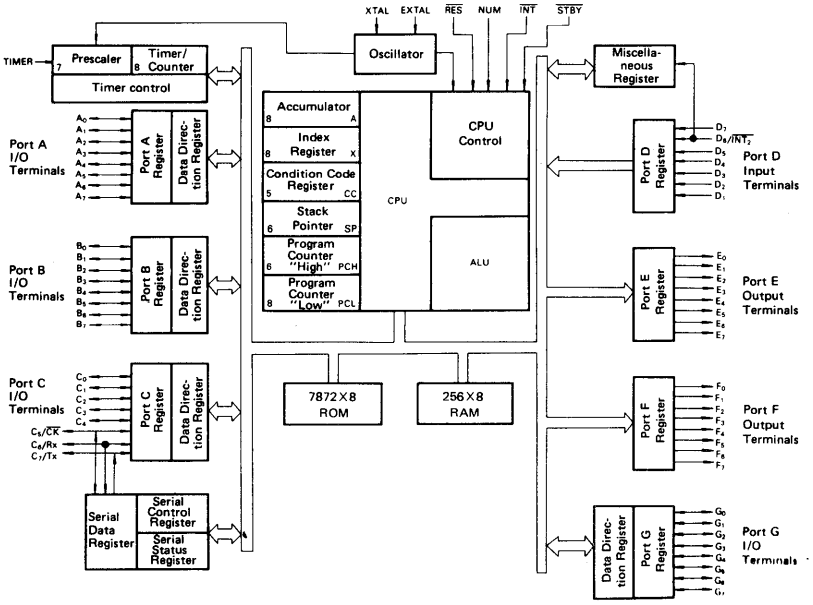
(Top View)

● HD6305Y0F, HD63A05Y0F, HD63B05Y0F



(Top View)

■ BLOCK DIAGRAM



Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 ~ +7.0	V
Input voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating temperature	T_{opr}	0 ~ +70	°C
Storage temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• **DC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)**

Item		Symbol	Test condition	min	typ	max	Unit
Input voltage "High"	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC} − 0.5	−	V_{CC} + 0.3	V
	EXTAL			V_{CC} x 0.7	−	V_{CC} + 0.3	V
	Others			2.0	−	V_{CC} + 0.3	V
Input voltage "Low"	All Input	V_{IL}		−0.3	−	0.8	V
Current dissipation	Operating	I_{CC}	f = 1MHz*	−	5	10	mA
	Wait			−	2	5	mA
	Stop			−	2	10	μA
	Standby			−	2	10	μA
Input leakage current	$\overline{TIMER}$, $\overline{INT}$, $D_1 \sim D_7$, $\overline{STBY}$	$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} − 0.5V	−	−	1	μA
Three-state current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $G_0 \sim G_7$, $E_0 \sim E_7$**, $F_0 \sim F_7$**	$\|I_{TSI}\|$		−	−	1	μA
Input capacity	All terminals	C_{in}	f = 1MHz, V_{in} = 0V	−	−	12	pF

* The value at f = xMHz can be calculated by the following equation: I_{CC} (f = xMHz) = I_{CC} (f = 1MHz) multiplied by x
**At standby mode

• AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item	Symbol	Test condition	HD6305Y0			HD63A05Y0			HD63B05Y0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock frequency	f_{cl}		0.4	–	4	0.4	–	6	0.4	–	8	MHz
Cycle time	t_{cyc}		1.0	–	10	0.666	–	10	0.5	–	10	μs
$\overline{INT}$ pulse width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ pulse width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ pulse width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
TIMER pulse width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation start time (crystal)	t_{OSC}	C_L = 22pF ± 20% R_s = 60Ω max	–	–	20	–	–	20	–	–	20	ms
Reset delay time	t_{RHL}	External cap. 2.2μF	80	–	–	80	–	–	80	–	–	ms

• Port Electrical Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item		Symbol	Test condition	min	typ	max	Unit
Output voltage "High"	Ports A, B, C, G, E, F	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC} - 0.7	–	–	V
Output voltage "Low"		V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input voltage "High"	Ports A, B, C, D, G	V_{IH}		2.0	–	V_{CC} + 0.3	V
Input voltage "Low"		V_{IL}		- 0.3	–	0.8	V
Input leakage current		I_{IL}	V_{in} = 0.5 ~ V_{CC} - 0.5V	-1	–	1	μA

• SCI Timing (V_{CC} = 5.0V±10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item	Symbol	Test Condition	HD6305Y0			HD63A05Y0			HD63B05Y0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock Cycle	t_{Scyc}	Fig. 1, Fig. 2	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data Output Delay Time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data Set-up Time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data Hold Time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

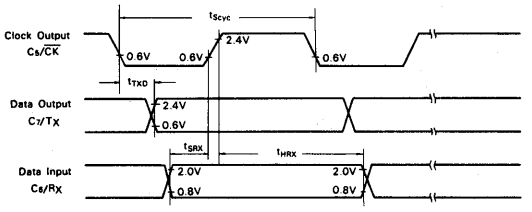


Figure 1 SCI Timing (Internal Clock)

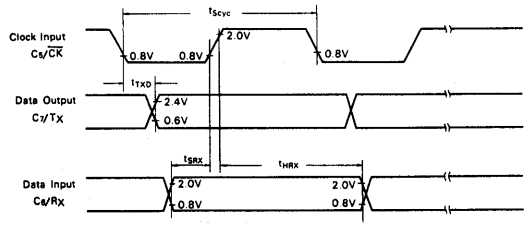
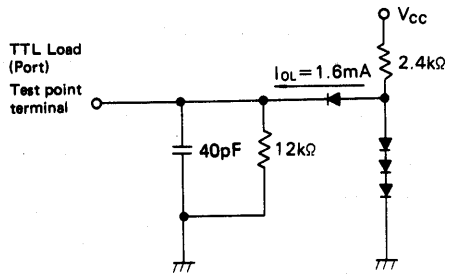


Figure2 SCI Timing(External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ.

Figure 3 Test Load

■ DESCRIPTION OF TERMINAL FUNCTIONS

The input and output signals of the HD6305Y0 are described here.

•V_{CC}, V_{SS}
Voltage is applied to the HD6305Y0 through these two terminals. V_{CC} is 5.0V ± 10%, while V_{SS} is grounded.

• $\overline{INT}$, $\overline{INT_2}$
External interrupt request inputs to the HD6305Y0. For details, refer to "INTERRUPTS". The $\overline{INT_2}$ terminal is also used as the port D_6 terminal.

•XTAL, EXTAL
These terminals provide input to the on-chip clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic filter is connected to the terminal. Refer to "INTERNAL OSCILLATOR" for using these input terminals.

•TIMER
This is an input terminal for event counter. Refer to "TIMER" for details.

• $\overline{RES}$
Used to reset the MCU. Refer to "RESET" for details.

•NUM
This terminal is not intended for user applications. It must be grounded to V_{SS}.

• Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $G_0 \sim G_7$)
These 32 terminals consist of four 8-bit I/O ports (A, B, C, G). Each of them can be used as an input or output terminal on a bit through program control of the data direction register. For details, refer to "I/O PORTS."

• Input Terminals ($D_1 \sim D_7$)
These seven input-only terminals are TTL or CMOS compatible. Of the port D's, D_6 is also used as $\overline{INT_2}$. If D_6 is used as a port, the $\overline{INT_2}$ interrupt mask bit of the miscellaneous register must be set to "1" to prevent an $\overline{INT_2}$ interrupt from being accidentally accepted.

• Output Terminals ($E_0 \sim E_7$, $F_0 \sim F_7$)
These 16 output-only terminals are TTL or CMOS compatible.

• $\overline{STBY}$
This terminal is used to place the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and the internal condition is reset. For details, refer to "Standby Mode."

The terminals described in the following are I/O pins for serial communication interface (SCI). They are also used as ports C_5, C_6 and C_7. For details, refer to "SERIAL COMMUNICATION INTERFACE."

• $\overline{CK}$ (C_5)
Used to input or output clocks for serial operation.

• Rx (C_6)
Used to receive serial data.

• Tx (C_7)
Used to transmit serial data.

■MEMORY MAP

The memory map of the HD6305Y0 MCU is shown in Fig. 4. During interrupt processing, the contents of the MCU registers are saved into the stack in the sequence shown in Fig. 5. This saving begins with the lower byte (PCL) of the program counter. Then the value of the stack pointer is decremented and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CC) are stacked in that order. In a subroutine call, only the contents of the program counter (PCH and PCL) are stacked.

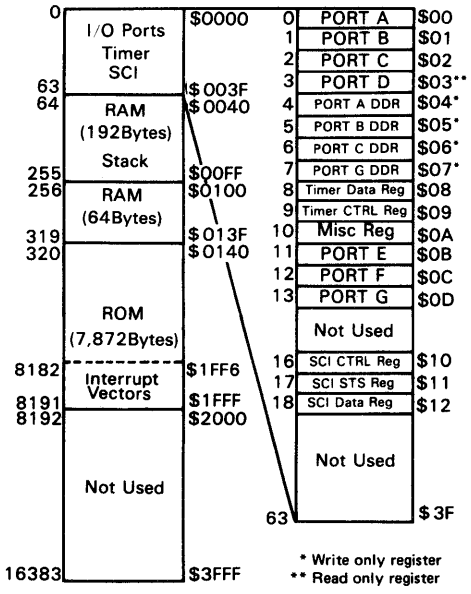
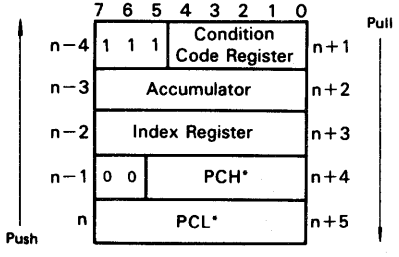


Figure 4 Memory Map of HD6305Y0 MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 5 Sequence of Interrupt Stacking

■REGISTERS

There are five registers which the programmer can operate.

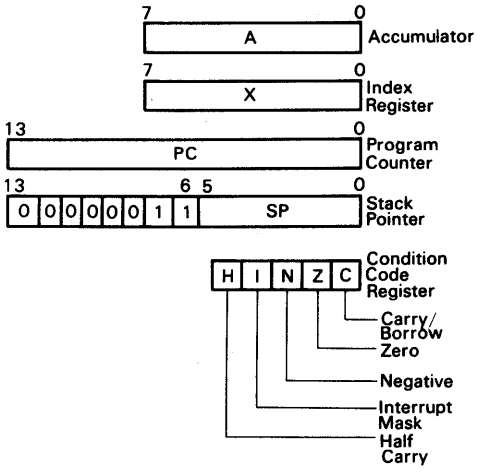


Figure 6 Programming Model

• Accumulator (A)

This accumulator is an ordinary 8-bit register which holds operands or the result of arithmetic operation or data processing.

• Index Register (X)

The index register is an 8-bit register, and is used for index addressing mode. Each of the addresses contained in the register consists of 8 bits which, combined with an offset value, provides an effective address.

In the case of a read/modify/write instruction, the index register can be used like an accumulator to hold operation data or the result of operation.

If not used in the index addressing mode, the register can be used to store data temporarily.

• Program Counter (PC)

The program counter is a 14-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 14-bit register that indicates the address of the next stacking space. Just after reset, the stack pointer is set at address $00FF. It is decremented when data is pushed, and incremented when pulled. The upper 8 bits of the stack pointer are fixed to 00000011. During the MCU being reset or during a reset stack pointer (RSP) instruction, the pointer is set to address $00FF. Since a subroutine or interrupt can use space up to address $00C1 for stacking, the subroutine can be used up to 31 levels and the interrupt up to 12 levels.

• Condition Code Register (CC)

The condition code register is a 5-bit register, each bit indicating the result of the instruction just executed. The bits can be individually tested by conditional branch instruc-

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

tions. The CC bits are as follows:

Half Carry (H): Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts, except a software interrupt, to be masked. If an interrupt occurs with the bit I set, it is latched. It will be processed the instant the interrupt mask bit is reset. (More specifically, it will enter the interrupt processing routine after the instruction following the CLI has been executed.)

Negative (N): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (bit 7 is logic "1").

Zero (Z): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is zero.

Carry/ Borrow (C): Represents a carry or borrow that occurred in the most recent arithmetic operation. This bit is also affected by the Bit Test and Branch instruction and a Rotate instruction.

■ INTERRUPT

There are six different types of interrupt: external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER' or the SCI and $TIMER_2$ generate the same vector address, respectively.

When an interrupt occurs, the program in progress stops and the then CPU status is saved onto the stack. And then, the interrupt mask bit (I) of the condition code register is set and the start address of the interrupt processing routine is obtained from a particular interrupt vector address. Then the interrupt routine starts from the start address. System can exit from the interrupt routine by an RTI instruction. When this instruction is executed, the CPU status before the interrupt (saved onto the stack) is pulled and the CPU restarts the sequence with the instruction next to the one at which the interrupt occurred. Table 1 lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$1FFE, $1FFF
SWI	2	$1FFC, $1FFD
$\overline{INT}$	3	$1FFA, $1FFB
TIMER/$\overline{INT_2}$	4	$1FF8, $1FF9
SCI/$TIMER_2$	5	$1FF6, $1FF7

A flowchart of the interrupt sequence is shown in Fig. 7. A block diagram of the interrupt request source is shown in Fig. 8.

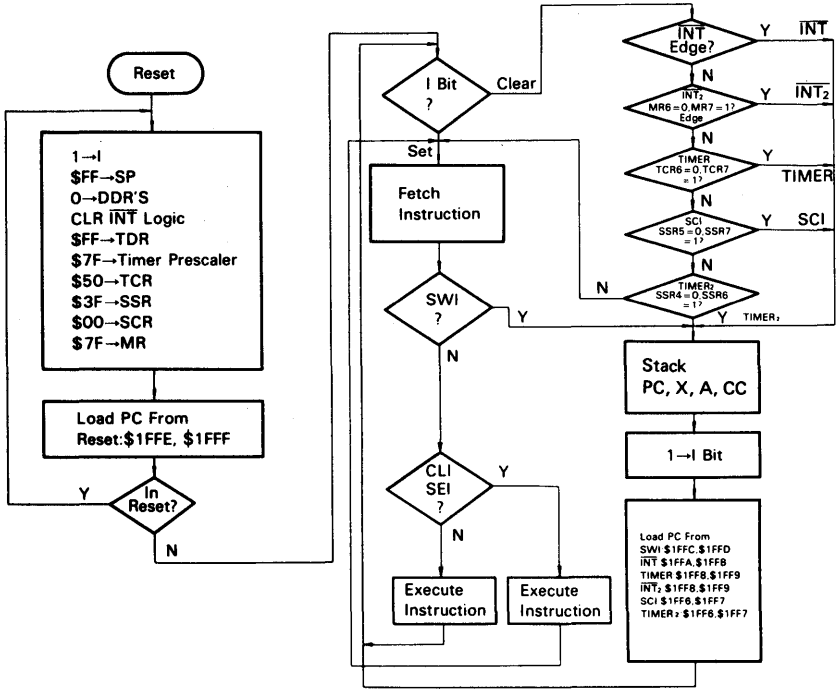


Figure 7 Interrupt Flowchart

In the block diagram, both the external interrupts $\overline{INT}$ and $\overline{INT_2}$ are edge trigger inputs. At the falling edge of each input, an interrupt request is generated and latched. The $\overline{INT}$ interrupt request is automatically cleared if jumping is made to the $\overline{INT}$ processing routine. Meanwhile, the $\overline{INT_2}$ request is cleared if "0" is written in bit 7 of the miscellaneous register.

For the external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$) and serial interrupt (SCI), each interrupt request is held, but not processed, if the I bit of the condition code register is set. Immediately after the I bit is cleared, the corresponding interrupt processing starts according to the priority.

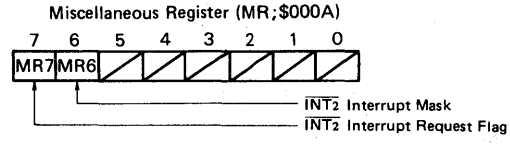
The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by setting bit 6 of the timer control register; the SCI interrupt by setting bit 5 of the serial status register; and the $TIMER_2$ interrupt by setting bit 4 of the serial status register.

The status of the $\overline{INT}$ terminal can be tested by a BIL or BIH instruction. The $\overline{INT}$ falling edge detector circuit and its latching circuit are independent of testing by these instructions. This is also true with the status of the $\overline{INT_2}$ terminal.

• Miscellaneous Register (MR; $000A)

The interrupt vector address for the external interrupt $\overline{INT_2}$ is the same as that for the TIMER interrupt, as shown in Table 1. For this reason, a special register called the miscellaneous register (MR; $000A) is available to control the $\overline{INT_2}$ interrupts.

Bit 7 of this register is the $\overline{INT_2}$ interrupt request flag. When the falling edge is detected at the $\overline{INT_2}$ terminal, "1" is set in bit 7. Then the software in the interrupt routine (vector addresses: $1FF8, $1FF9) checks bit 7 to see if it is $\overline{INT_2}$ interrupt. Bit 7 can be reset by software.



Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If this bit is set to "1", then the $\overline{INT_2}$ interrupt is disabled. Both read and write are possible with bit 7 but "1" cannot be written in this bit by software. This means that an interrupt request by software is impossible.

When reset, bit 7 is cleared to "0" and bit 6 is set to "1".

■ TIMER

Figure 9 shows an MCU timer block diagram. The timer data register is loaded by software and, upon receipt of a clock input, begins to count down. When the timer data

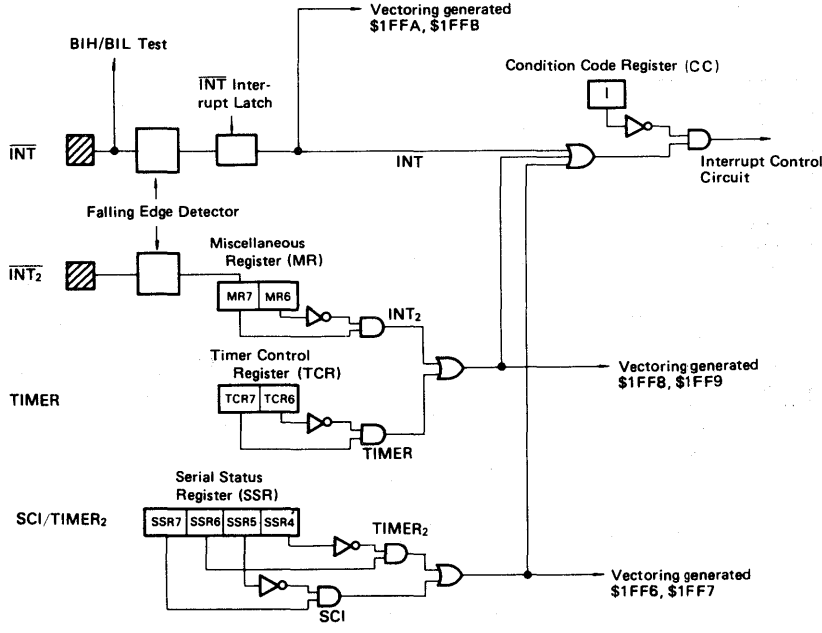


Figure 8 Interrupt Request Generation Circuitry

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

register (TDR) becomes "0", the timer interrupt request bit (bit 7) in the timer control register is set. In response to the interrupt request, the MCU saves its status into the stack and fetches timer interrupt routine address from addresses $1FF8 and $1FF9 and execute the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also mask the timer interrupt.

The source clock to the timer can be either an external signal from the timer input terminal or the internal E signal (the oscillator clock divided by 4). If the E signal is used as the source, the clock input can be gated by the input to the timer input terminal.

Once the timer count has reached "0", it starts counting down with "$FF". The count can be monitored whenever desired by reading the timer data register. This permits the program to know the length of time having passed after the occurrence of a timer interrupt, without disturbing the contents of the counter.

When the MCU is reset, both the prescaler and counter are initialized to logic "1". The timer interrupt request bit (bit 7) then is cleared and the timer interrupt mask bit (bit 6) is set.

To clear the timer interrupt request bit (bit 7), it is necessary to write "0" in that bit.

TCR7	Timer interrupt request
0	Absent
1	Present

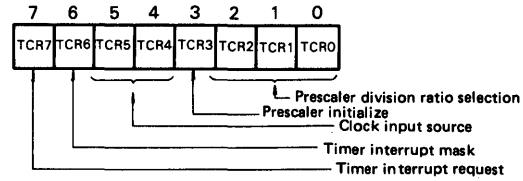
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

• Timer Control Register (TCR, $0009)

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After reset, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$FF" immediately after reset.

When "1" is written in bit 3, the prescaler is initialized. This bit always shows "0" when read.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under timer terminal control
1	0	No clock input (counting stopped)
1	1	Event input from timer terminal

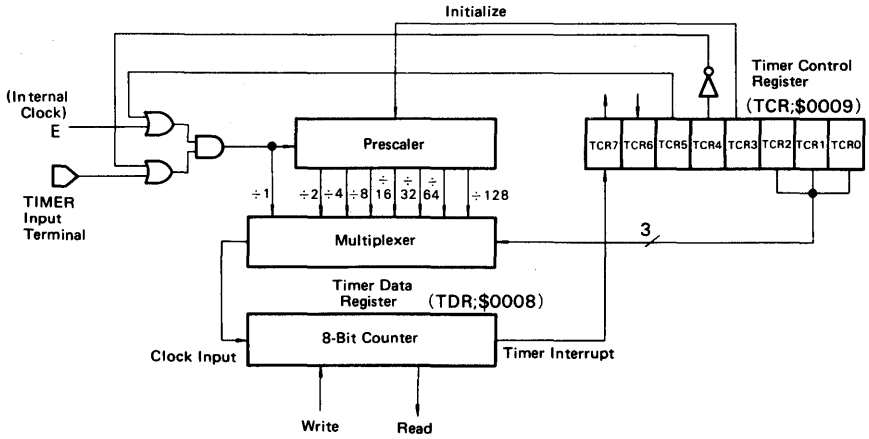


Figure 9 Timer Block Diagram

A prescaler division ratio is selected by the combination of three bits (bits 0, 1 and 2) of the timer control register (see Table 3). There are eight different division ratios: ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128. After reset, the TCR is set to the ÷1 mode.

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

A timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. This bit can be cleared by writing "0" in that bit.

■SERIAL COMMUNICATION INTERFACE (SCI)

This interface is used for serial transmission or reception of 8-bit data. Sixteen transfer rates are available in the range from 1 μs to approx. 32 ms (for oscillation at 4 MHz).

The SCI consists of three registers, one eighth counter and one prescaler. (See Fig. 10.) SCI communicates with the CPU via the data bus, and with the outside world through bits 5, 6 and 7 of port C. Described below are the operations of each register and data transfer.

●SCI Control Register (SCR; $0010)

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

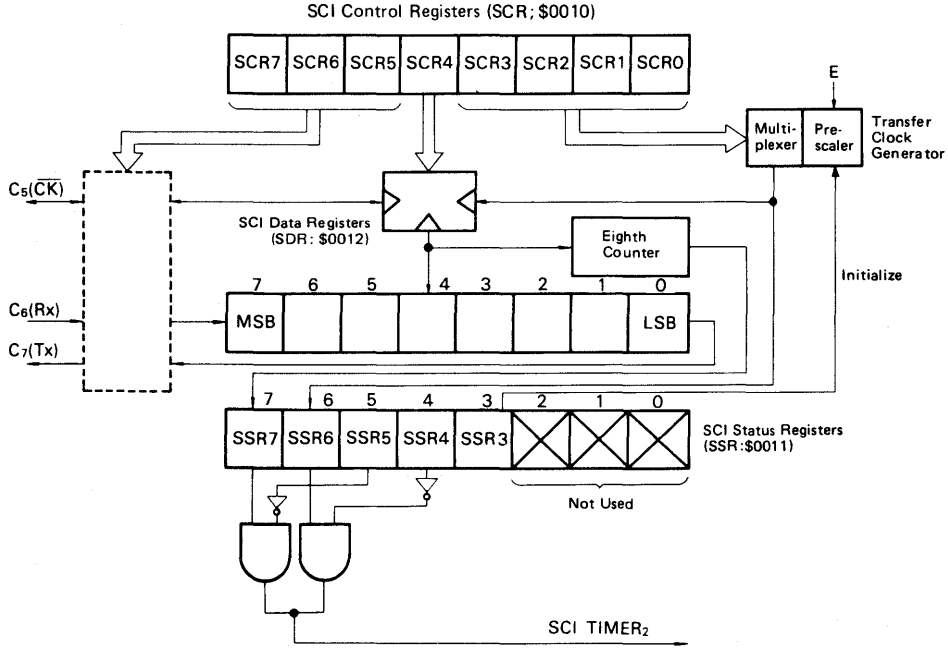


Figure 10 SCI Block Diagram

SCR7	C_7 terminal
0	Used as I/O terminal (by DDR).
1	Serial data output (DDR output)

SCR6	C_6 terminal
0	Used as I/O terminal (by DDR).
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	C_5 terminal
0	0	–	Used as I/O terminal (by DDR).
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)
When this bit is set, the DDR corresponding to the C_7 becomes "1" and this terminal serves for output of SCI data. After reset, the bit is cleared to "0".

Bit 6 (SCR6)
When this bit is set, the DDR corresponding to the C_6 becomes "0" and this terminal serves for input of SCI data. After reset, the bit is cleared to "0".

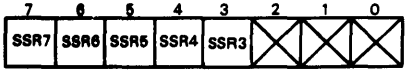
Bits 5 and 4 (SCR5, SCR4)
These bits are used to select a clock source. After reset, the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)
These bits are used to select a transfer clock rate. After reset, the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

•SCI Data Register (SDR; $0012)
A serial-parallel conversion register that is used for transfer of data.

•SCI Status Register (SSR; $0011)



Bit 7 (SSR7)
Bit 7 is the SCI interrupt request bit which is set upon completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5="1". The bit can also be cleared by writing "0" in it.

Bit 6 (SSR6)
Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is used commonly with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When reset, the bit is cleared. It also be cleared by writing "0" in it. (For details, see $TIMER_2$.)

Bit 5 (SSR5)
Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When reset, it is set to "1".

Bit 4 (SSR4)
Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When reset, it is set to "1"

Bit 3 (SSR3)
When "1" is written in this bit, the prescaler of the transfer clock generator is initialized. When read, the bit always is "0".

Bits 2 ~ 0
Not used.

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

• Data Transmission
By writing the desired control bits into the SCI control registers, a transfer rate and a source of transfer clock are determined and bits 7 and 5 of port C are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the C_7/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock. (See Fig. 11.) When 8 bits of

data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register with the rising edge of the last serial clock. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the C_7/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the C_5/$\overline{CK}$ terminal is set as input. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

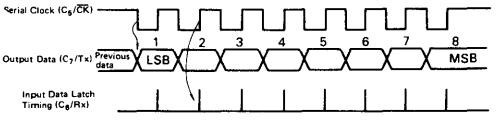


Figure 11 SCI Timing Chart

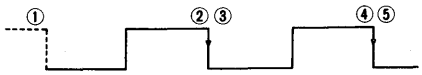
• Data Reception

By writing the desired control bits into the SCI control register, a transfer rate and a source of transfer clock are determined and bit 6 and 5 of port C are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed after reading subsequent received data. It must be taken after reset and after not reading subsequent received data.)

The data from the C_6/Rx terminal is input to the SCI data register synchronously with the rising edge of the serial clock (see Fig. 11). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored and the data is received synchronously with the clock from the C_5/$\overline{CK}$ terminal. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

• TIMER$_2$

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 ~ 0 of the SCI control register (4 μs ~ approx. 32 ms (for oscillation at 4 MHz)) is input to bit 6 of the SCI status register and the TIMER$_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, TIMER$_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : TIMER$_2$ interrupt request
③, ⑤ : TIMER$_2$ interrupt request bit cleared

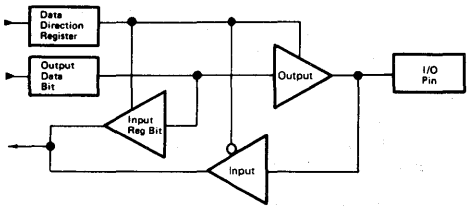
TIMER$_2$ is commonly used with the SCI transfer clock generator. If wanting to use TIMER$_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■ I/O PORTS

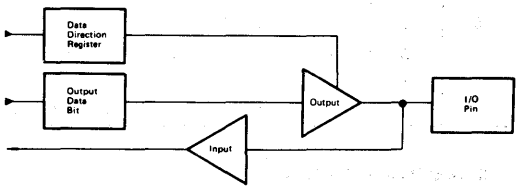
There are 32 input/output terminals (ports A, B, C, G). Each I/O terminal can be selected for either input or output by the data direction register. More specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B or C reads latched data if it has been programmed as output, even with the output level being fluctuated by the output load. (See Fig. 12-a.) For port G, in such a case, the level of the pin is always read when it is read. (See Fig. 12-b.) This implies that, even when "1" is being output, port G may read "0" if the load condition causes the output voltage to decrease to below 2.0V.

When reset, the data direction register and data register go to "0" and all the input/output terminals are used as input.



Bit of data direction register	Bit of output data	Status of output	Input to CPU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

a. Ports A, B and C



b. Port G

Figure 12 Input/Output Port Diagram

There are 16 output-only terminals (ports E and F). Each of them can also read. In this case, latched data is read even with the output terminal level being fluctuated by the output load (as with ports A, B and C).
When reset, "Low" level is output from each output terminal.

Seven input-only terminals are available (port D). Writing to an input terminal is invalid.

All input/output terminals, output terminals and input terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports or input ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite that they are not used.

■RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 13.) On power up, the reset input must be held "Low" for at least 30 ms to assure that the internal oscillator is stabilized. A sufficient time of delay can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 14.

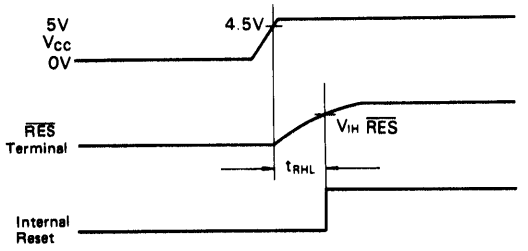


Figure 13 Power On and Reset Timing

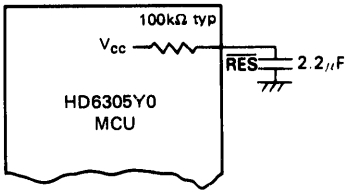


Figure 14 Input Reset Delay Circuit

■INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0MHz) or ceramic oscillator between pins 5 and 6 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 15. Figs. 16 and 17 illustrate the specifications and typical arrangement of the crystal, respectively.

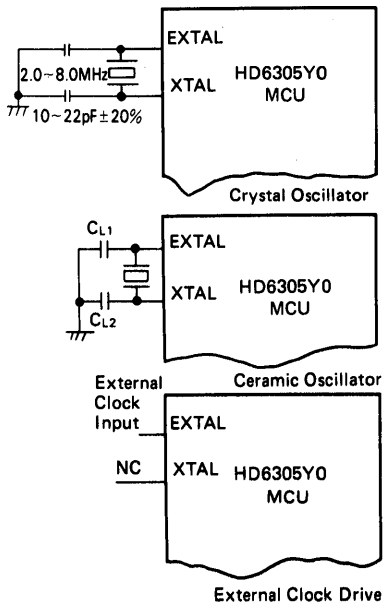


Figure 15 Internal Oscillator Circuit

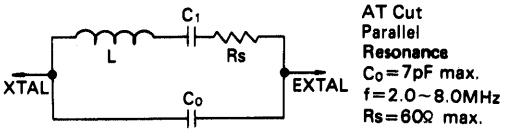
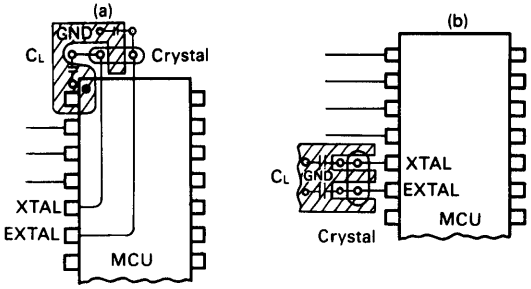


Figure 16 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 17 Typical Crystal Arrangement

■LOW POWER DISSIPATION MODE

The HD6305Y0 has three low power dissipation modes: wait, stop and standby.

• Wait Mode

When WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers, RAM and I/O terminals hold their condition just before entering into the wait mode.

The escape from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the wait mode the MCU executes the instruction next to the WAIT. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$) is masked by the timer control register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 18 shows a flowchart for the wait function.

• Stop Mode

When STOP instruction being executed, MCU enters into the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive but the RAM, registers and I/O terminals hold their condition just before entering into the stop mode.

The escape from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings into the standby mode.

When interrupt is requested to the CPU and accepted, the stop mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the stop mode, the MCU executes the instruction next to the STOP. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 19 shows a flowchart for the stop function. Fig. 20 shows a timing chart of return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$="0" must exceed 30 ms to assure stabilized oscillation.

• Standby Mode

The MCU enters into the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are hold. The I/O terminals turn to high-impedance state. The standby mode should escape by bringing $\overline{STBY}$ "High". The CPU must be restarted by reset. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 21.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 22.

(Note)

When I bit of condition code register is "1" and interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$, SCI/TIMER$_2$) is held, MCU does not enter WAIT mode by the execution of WAIT instruction.

In that case, after the 4 dummy cycles MCU executes the next instruction.

In the same way, when external interrupts ($\overline{INT}$, $\overline{INT_2}$) are held at the bit I set, MCU does not enter STOP mode by the execution of STOP instruction. In that case, also, MCU executes the next instruction after the 4 dummy cycles.

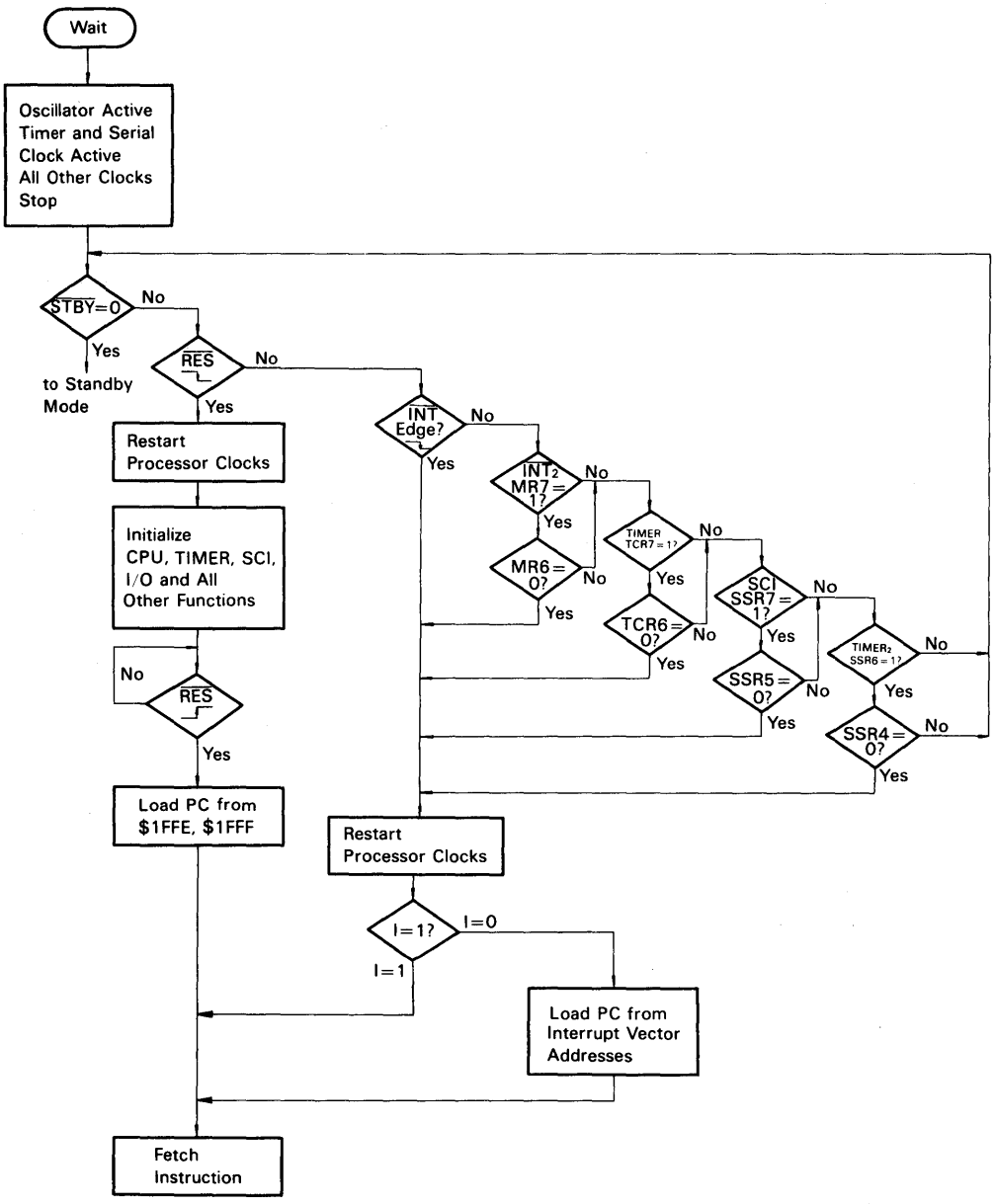


Figure 18 Wait Mode Flow Chart

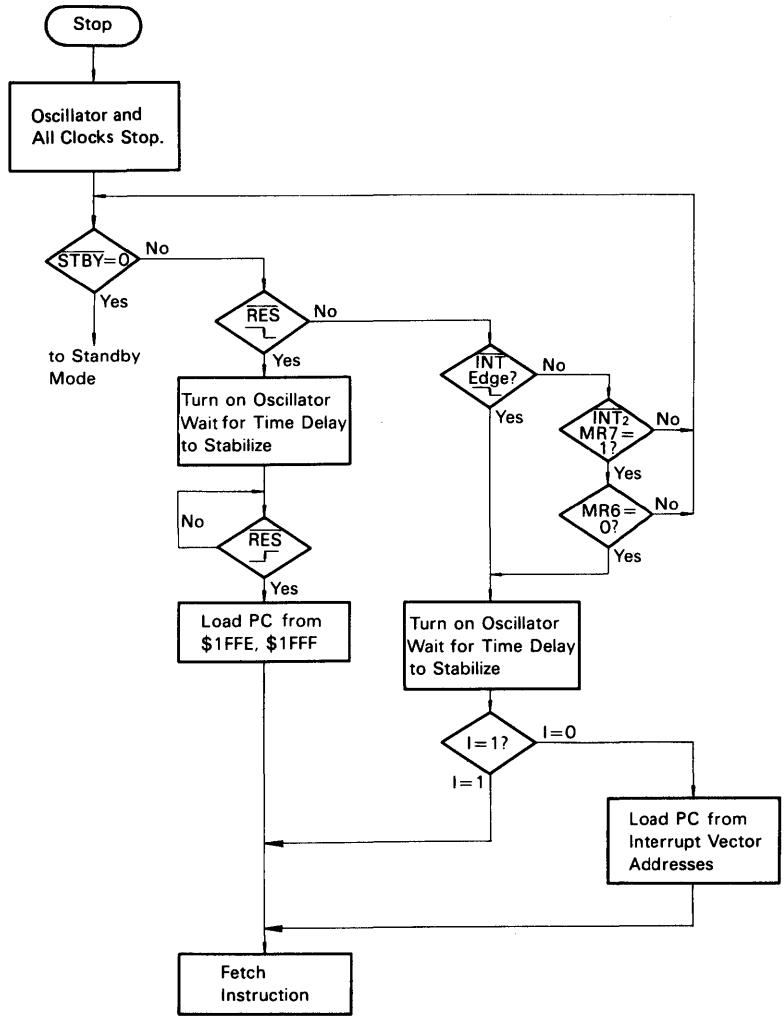
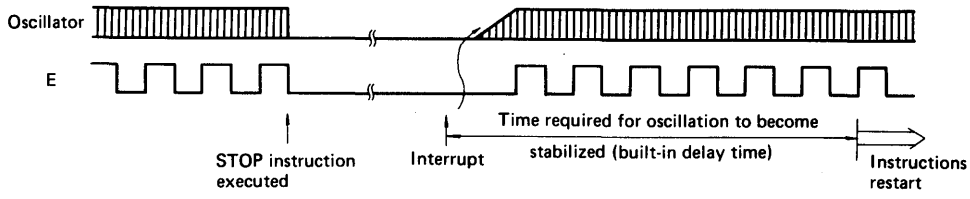
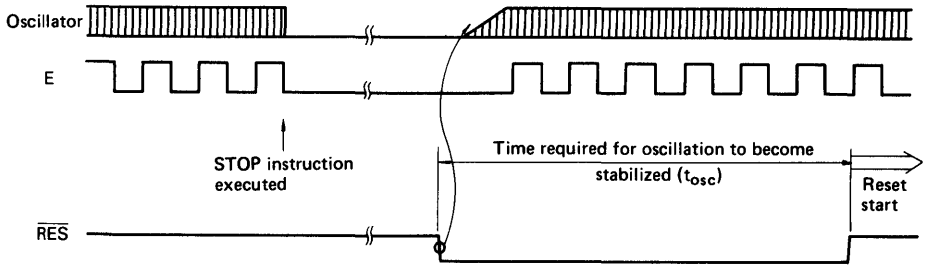


Figure 19 Stop Mode Flow Chart



(a) Restart by Interrupt



(b) Restart by Reset

Figure 20 Timing Chart of Releasing from Stop Mode

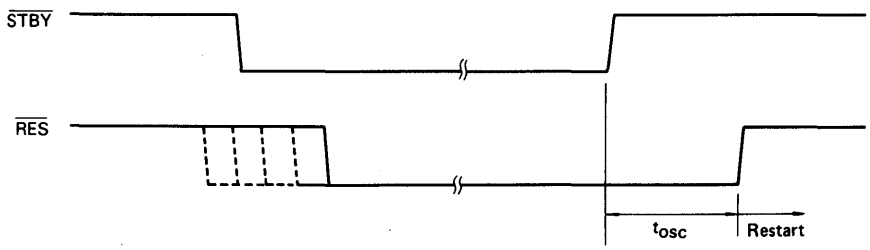


Figure 21 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Keep	High impedance	$\overline{STBY}$="High"

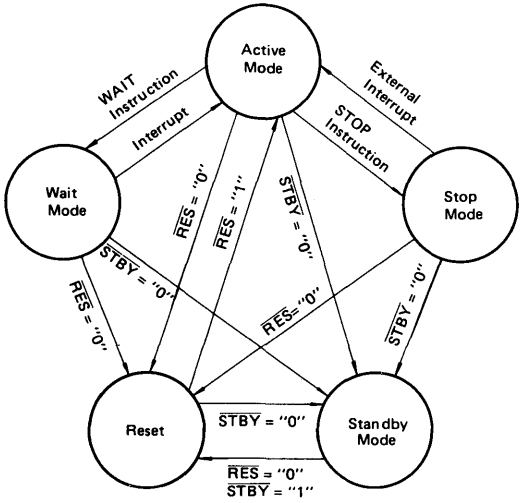


Figure 22 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■BIT MANIPULATION

The HD6305Y0 MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM within page 0 or an I/O port (except the write-only registers such as the data direction register). Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM on page 0, or I/O can be manipulated, the user may use a bit within the RAM on page 0 as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 23 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the ROM. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
        :
SELF 1. BRCLR 0, PORT A, SELF 1
        BSET 1, PORT A
        BCLR 1, PORT A
        :
```

Figure 23 Example of Bit Manipulation

■ADDRESSING MODES

Ten different addressing modes are available to the HD6305Y0 MCU.

• Immediate

See Fig. 24. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 25. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. 192 byte RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 26. The extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 27. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 28. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 29. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 30. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 31. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 32. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The value of the test bit is written in the carry bit of the condition code register.

• Implied

See Fig. 33. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

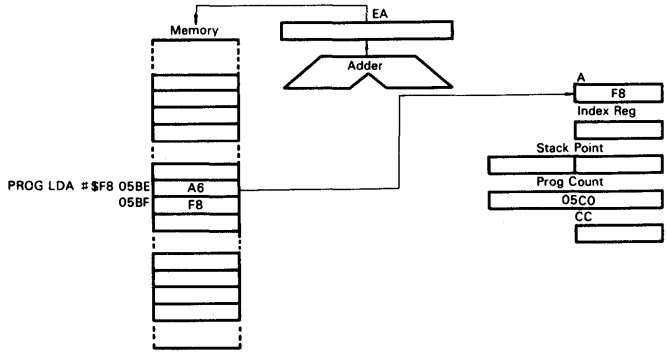


Figure 24 Example of Immediate Addressing

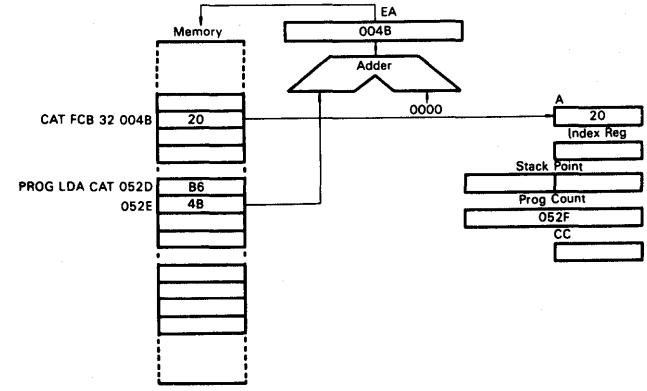


Figure 25 Example of Direct Addressing

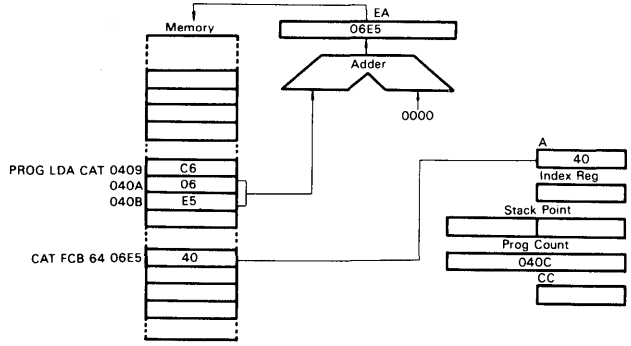


Figure 26 Example of Extended Addressing

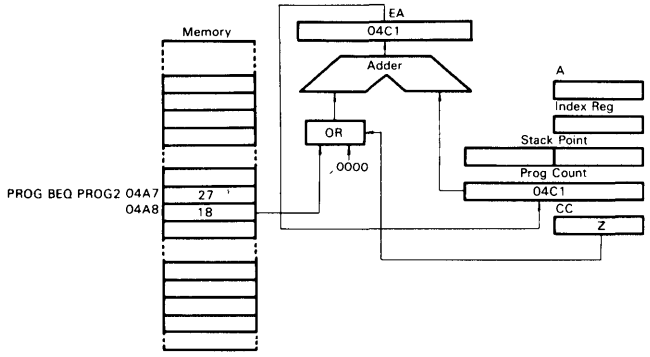


Figure 27 Example of Relative Addressing

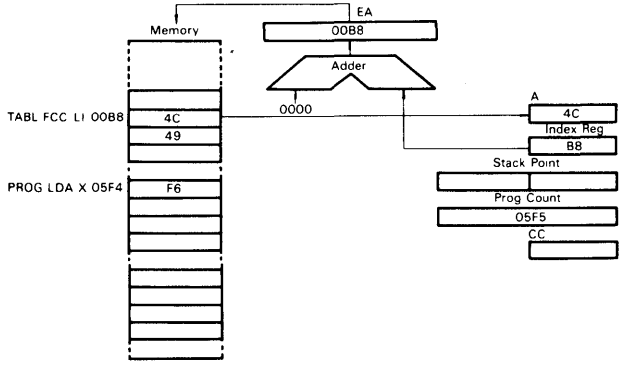


Figure 28 Example of Indexed (No Offset) Addressing

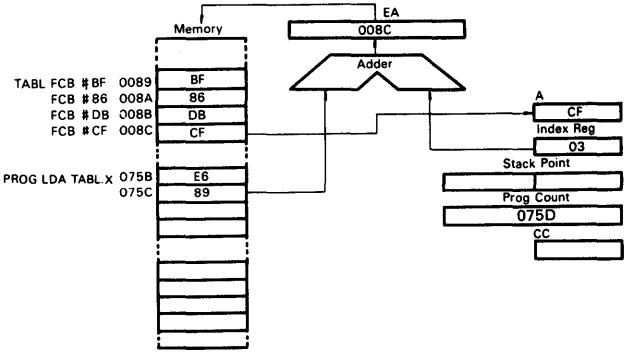


Figure 29 Example of Index (8-bit Offset) Addressing

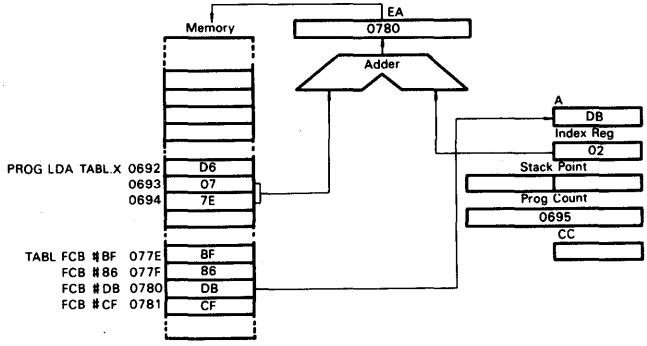


Figure 30 Example of Index (16-bit Offset) Addressing

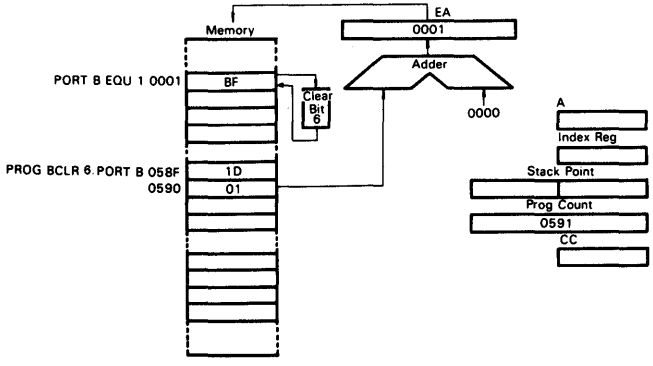


Figure 31 Example of Bit Set/Clear Addressing

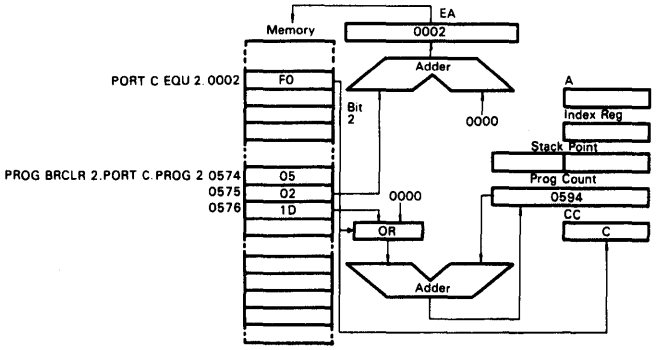


Figure 32 Example of Bit Test and Branch Addressing

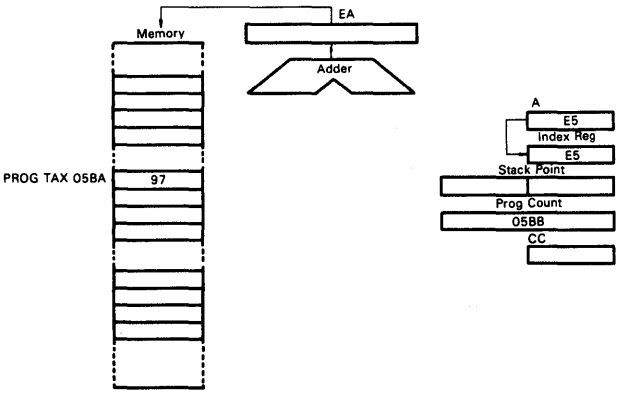


Figure 33 Example of Implied Addressing

■INSTRUCTION SET

There are 62 basic instructions available to the HD6305Y0 MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the HD6305Y0 MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the HD6305Y0 MCU in the alphabetical order.

• Operation Code Map

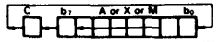
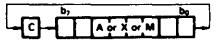
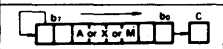
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate OP	#	~	Direct OP	#	~	Extended OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Indexed (16-Bit Offset) OP	#	~	Boolean/ Arithmetic Operation	Condition Code H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	^	^	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	^	^	•
Store A in Memory	STA	–	–	–	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	^	^	•
Store X in Memory	STX	–	–	–	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	^	^	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	^	•	^	^	^
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	^	•	^	^	^
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	^	^	^
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	^	^	^
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	^	^	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	^	^	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	^	^	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	^	^	^
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	^	^	^
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	^	^	•
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR	–	–	–	BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A) OP	#	~	Implied(X) OP	#	~	Direct OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Boolean/Arithmetic Operation	Condition Code H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	^	^	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	^	^	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	^	^	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	^	^	^
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	^	^	^
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	^	^	^
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	^	^	^
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	^	^
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	^	^	^
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	^	^	^
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	^	^	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	●	●	●	●	●
Branch Never	BRN	21	2	3	None	●	●	●	●	●
Branch IF Higher	BHI	22	2	3	C+Z=0	●	●	●	●	●
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	●	●	●	●	●
Branch IF Carry Clear	BCC	24	2	3	C=0	●	●	●	●	●
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	●	●	●	●	●
Branch IF Carry Set	BCS	25	2	3	C=1	●	●	●	●	●
(Branch IF Lower)	(BLO)	25	2	3	C=1	●	●	●	●	●
Branch IF Not Equal	BNE	26	2	3	Z=0	●	●	●	●	●
Branch IF Equal	BEQ	27	2	3	Z=1	●	●	●	●	●
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	●	●	●	●	●
Branch IF Half Carry Set	BHCS	29	2	3	H=1	●	●	●	●	●
Branch IF Plus	BPL	2A	2	3	N=0	●	●	●	●	●
Branch IF Minus	BMI	2B	2	3	N=1	●	●	●	●	●
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	●	●	●	●	●
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	●	●	●	●	●
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	●	●	●	●	●
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	●	●	●	●	●
Branch to Subroutine	BSR	AD	2	5	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	●	●	●	●	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	●	●	●	●	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	●	●	●	●	●
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes			Boolean Operation	Condition Code				
		Implied								
		OP	#	~		H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	●	●	●	●	●
Transfer X to A	TXA	9F	1	2	X→A	●	●	●	●	●
Set Carry Bit	SEC	99	1	1	1→C	●	●	●	●	1
Clear Carry Bit	CLC	98	1	1	0→C	●	●	●	●	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	●	1	●	●	●
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	●	0	●	●	●
Software Interrupt	SWI	83	1	10		●	1	●	●	●
Return from Subroutine	RTS	81	1	5		●	●	●	●	●
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	●	●	●	●	●
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	●	●	●	●	●
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	●	●	∧	∧	∧*
Stop	STOP	8E	1	4		●	●	●	●	●
Wait	WAIT	8F	1	4		●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		×	×	×		×	×	×			∧	●	∧	∧	∧
ADD		×	×	×		×	×	×			∧	●	∧	∧	∧
AND		×	×	×		×	×	×			●	●	∧	∧	●
ASL	×		×			×	×				●	●	∧	∧	∧
ASR	×		×			×	×				●	●	∧	∧	∧
BCC					×						●	●	●	●	●
BCLR									×		●	●	●	●	●
BCS					×						●	●	●	●	●
BEQ					×						●	●	●	●	●
BHCC					×						●	●	●	●	●
BHCS					×						●	●	●	●	●
BHI					×						●	●	●	●	●
(BHS)					×						●	●	●	●	●
BIH					×						●	●	●	●	●
BIL					×						●	●	●	●	●
BIT		×	×	×		×	×	×			●	●	∧	∧	●
(BLO)					×						●	●	●	●	●
BLS					×						●	●	●	●	●
BMC					×						●	●	●	●	●
BMI					×						●	●	●	●	●
BMS					×						●	●	●	●	●
BNE					×						●	●	●	●	●
BPL					×						●	●	●	●	●
BRA					×						●	●	●	●	●

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						●	●	●	●	●
BRCLR										×	●	●	●	●	^
BRSET										×	●	●	●	●	^
BSET									×		●	●	●	●	●
BSR					×						●	●	●	●	●
CLC	×										●	●	●	●	0
CLI	×										●	0	●	●	●
CLR	×		×			×	×				●	●	0	1	●
CMP		×	×	×		×	×	×			●	●	^	^	^
COM	×		×			×	×				●	●	^	^	1
CPX		×	×	×		×	×	×			●	●	^	^	^
DAA	×										●	●	^	^	^
DEC	×		×			×	×				●	●	^	^	●
EOR		×	×	×		×	×	×			●	●	^	^	●
INC	×		×			×	×				●	●	^	^	●
JMP			×	×		×	×	×			●	●	●	●	●
JSR			×	×		×	×	×			●	●	●	●	●
LDA		×	×	×		×	×	×			●	●	^	^	●
LDX		×	×	×		×	×	×			●	●	^	^	●
LSL	×		×			×	×				●	●	^	^	^
LSR	×		×			×	×				●	●	0	^	^
NEG	×		×			×	×				●	●	^	^	^
NOP	×										●	●	●	●	●
ORA		×	×	×		×	×	×			●	●	^	^	●
ROL	×		×			×	×				●	●	^	^	^
ROR	×		×			×	×				●	●	^	^	^
RSP	×										●	●	●	●	●
RTI	×										?	?	?	?	?
RTS	×										●	●	●	●	●
SBC		×	×	×		×	×	×			●	●	^	^	^
SEC	×										●	●	●	●	1
SEI	×										●	1	●	●	●
STA			×	×		×	×	×			●	●	^	^	●
STOP	×										●	●	●	●	●
STX			×	×		×	×	×			●	●	^	^	●
SUB		×	×	×		×	×	×			●	●	^	^	^
SWI	×										●	1	●	●	●
TAX	×										●	●	●	●	●
TST	×		×			×	×				●	●	^	^	●
TXA	×										●	●	●	●	●
WAIT	×										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI	—					—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

↑ LOW

(NOTES) 1. "—" is an undefined operation code.
2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD6305Y0:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

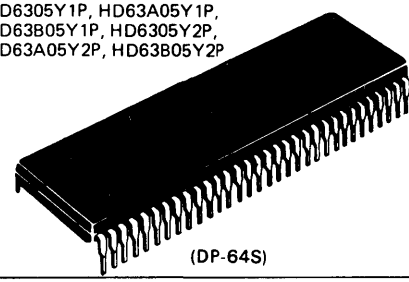
STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

HD6305Y1, HD63A05Y1, HD63B05Y1
HD6305Y2, HD63A05Y2, HD63B05Y2
CMOS MCU (Microcomputer Unit)

The HD6305Y1 and the HD6305Y2 are CMOS 8-bit single chip microcomputers. A CPU, a clock generator, a 256 byte RAM, I/O terminals, two timers and a serial communication interface (SCI) are built in both chip of the HD6305Y1 and the HD6305Y2. Their memory spaces are expandable to 16k bytes externally.

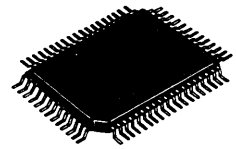
The HD6305Y1 and the HD6305Y2 have the same functions as the HD6305Y0's except for the number of I/O terminals. The HD6305Y1 has 7872 byte ROM and its memory space is expandable to 8k bytes externally. The HD6305Y2 is a microcomputer unit which includes no ROM and its memory space is expandable to 16k bytes externally.

HD6305Y1P, HD63A05Y1P,
HD63B05Y1P, HD6305Y2P,
HD63A05Y2P, HD63B05Y2P



(DP-64S)

HD6305Y1F, HD63A05Y1F,
HD63B05Y1F, HD6305Y2F,
HD63A05Y2F, HD63B05Y2F



(FP-64)

■ HARDWARE FEATURES

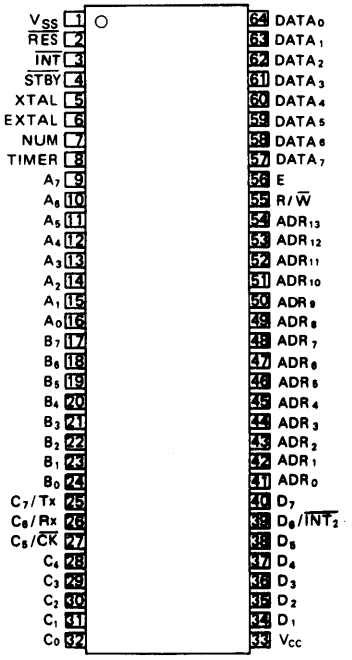
- 8-bit based MCU
- 7872 bytes of internal ROM (HD6305Y1)
  No internal ROM (HD6305Y2)
- 256 bytes of RAM
- A total of 31 terminals, including 24 I/O's, 7 inputs
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes
  - Wait . . . . In this mode, the clock oscillator is on and the CPU halts but the timer/serial/interrupt function is operatable.
  - Stop . . . . In this mode, the clock stops but the RAM data, I/O status and registers are held.
  - Standby. . In this mode, the clock stops, the RAM data is held, and the other internal condition is reset.
- Minimum instruction cycle time
  - HD6305Y1/Y2 . . 1 μs (f = 1 MHz)
  - HD63A05Y1/Y2. . 0.67 μs (f = 1.5 MHz)
  - HD63B05Y1/Y2. . 0.5 μs (f = 2 MHz)
- Wide operating range
  V_{CC} = 3 to 6V (f = 0.1 to 0.5 MHz)
  - HD6305Y1/Y2 . . f = 0.1 to 1 MHz (V_{CC} = 5V ± 10%)
  - HD63A05Y1/Y2. . f = 0.1 to 1.5 MHz (V_{CC} = 5V ± 10%)
  - HD63B05Y1/Y2. . f = 0.1 to 2 MHz (V_{CC} = 5V ± 10%)
- System development fully supported by an evaluation kit

■ SOFTWARE FEATURES

- Similar to HD6800
- Byte efficient instruction set
- Powerful bit manipulation instructions (Bit Set, Bit Clear, and Bit Test and Branch usable for 192 byte RAM bits within page 0 and all I/O terminals)
- A variety of interrupt operations
- Index addressing mode useful for table processing
- A variety of conditional branch instructions
- Ten powerful addressing modes
- All addressing modes adaptable to RAM, and I/O instructions
- Three new instructions, STOP, WAIT and DAA, added to the HD6805 family instruction set
- Instructions that are upward compatible with those of Motorola's MC6805P2 and MC146805G2

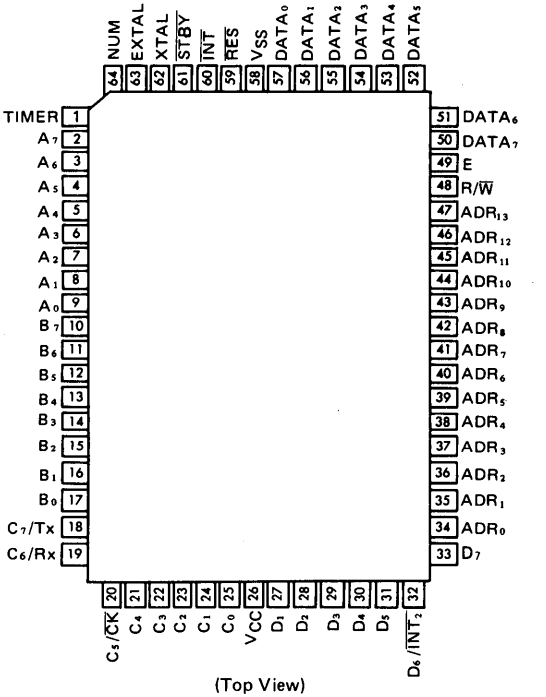
■ PIN ARRANGEMENT

- **HD6305Y1P, HD6A05Y1P, HD63B05Y1P, HD6305Y2P, HD63A05Y2P, HD63B05Y2P**



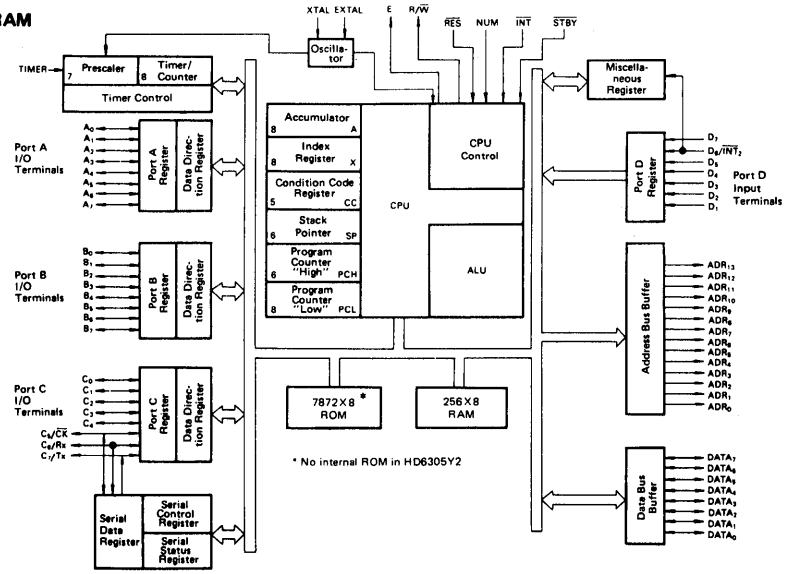
(Top View)

- **HD6305Y1F, HD63A05Y1F, HD63B05Y1F, HD6305Y2F, HD63A05Y2F, HD63B05Y2F**



(Top View)

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	−0.3 ~ +7.0	V
Input Voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC}+0.3	V
	EXTAL			V_{CC}x0.7	–	V_{CC}+0.3	
	Other Inputs			2.0	–	V_{CC}+0.3	
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input Leakage Current	TIMER, $\overline{INT}$, $D_1 \sim D_7$, $\overline{STBY}$	$\|I_{IL}\|$	Vin = 0.5 ~ V_{CC}-0.5	–	–	1.0	μA
Three-state Current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $ADR_0 \sim ADR_{13}$*, $DATA_0 \sim DATA_7$, E*, R/$\overline{W}$*	$\|I_{TSI}\|$		–	–	1.0	μA
Current Dissipation**	Operating	I_{CC}	f = 1MHz***	–	5	10	mA
	Wait			–	2	5	mA
	Stop			–	2	10	μA
	Standby			–	2	10	μA
Input Capacitance	All Terminals	Cin	f = 1MHz, Vin = 0V	–	–	12	pF

* Only at standby
** V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V
*** The value at f = xMHz is given by using.
I_{CC} (f = xMHz) = I_{CC} (f = 1MHz) x x

• AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	HD6305Y1/Y2			HD63A05Y1/Y2			HD63B05Y1/Y2			Unit
			min	typ	max	min	typ	max	min	typ	max	
Cycle Time	t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time	t_{Er}		–	–	20	–	–	20	–	–	20	ns
Enable Fall Time	t_{Ef}		–	–	20	–	–	20	–	–	20	ns
Enable Pulse Width("High" Level)	PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width("Low" Level)	PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address Delay Time	t_{AD}		–	–	250	–	–	190	–	–	180	ns
Address Hold Time	t_{AH}		40	–	–	30	–	–	20	–	–	ns
Data Delay Time	t_{DW}		–	–	200	–	–	160	–	–	120	ns
Data Hold Time (Write)	t_{HW}		40	–	–	30	–	–	20	–	–	ns
Data Set-up Time (Read)	t_{DSR}		80	–	–	60	–	–	50	–	–	ns
Data Hold Time (Read)	t_{HR}		0	–	–	0	–	–	0	–	–	ns

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

- **PORT TIMING (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD6305Y1/Y2			HD63A05Y1/Y2			HD63B05Y1/Y2			Unit
			min	typ	max	min	typ	max	min	typ	max	
Port Data Set-up Time (Port A, B, C, D)	t_{PDS}	Fig. 2	200	–	–	200	–	–	200	–	–	ns
Port Data Hold Time (Port A, B, C, D)	t_{PDH}		200	–	–	200	–	–	200	–	–	ns
Port Data Delay Time (Port A, B, C)	t_{PDW}	Fig. 3	–	–	300	–	–	300	–	–	300	ns

- **CONTROL SIGNAL TIMING (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD6305Y1/Y2			HD63A05Y1/Y2			HD63B05Y1/Y2			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{INT}$ Pulse Width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ Pulse Width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ Pulse Width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
Control Set-up Time	t_{CS}	Fig. 5	250	–	–	250	–	–	250	–	–	ns
Timer Pulse Width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation Start Time (Crystal)	t_{OSC}	Fig.5, Fig.20*	–	–	20	–	–	20	–	–	20	ms
Reset Delay Time	t_{RHL}	Fig. 19	80	–	–	80	–	–	80	–	–	ms

* C_L = 22pF ±20%, R_s = 60Ω max.

- **SCI TIMING (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD6305Y1/Y2			HD63A05Y1/Y2			HD63B05Y1/Y2			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock Cycle	t_{Scyc}	Fig. 6, Fig. 7	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data Output Delay Time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data Set-up Time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data Hold Time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

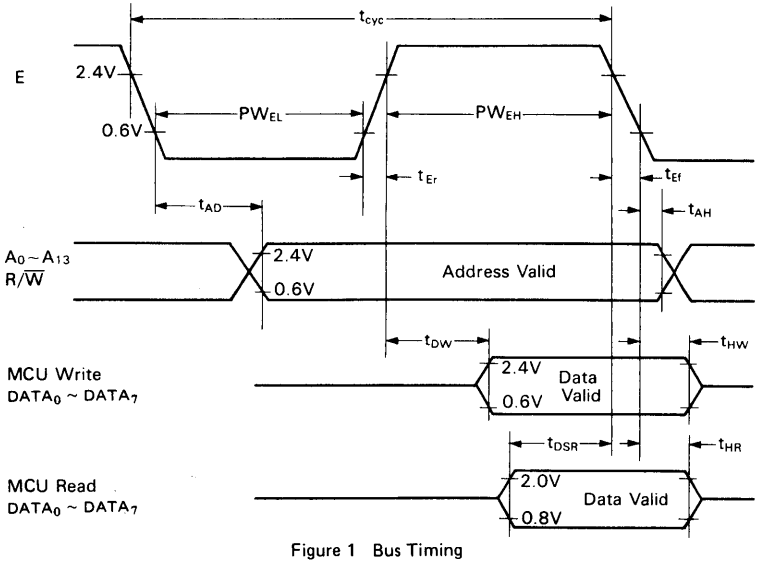


Figure 1 Bus Timing

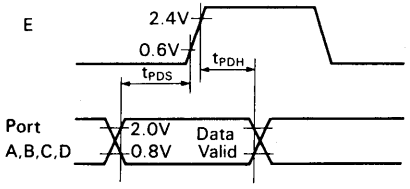


Figure 2 Port Data Set-up and Hold Times (MCU Read)

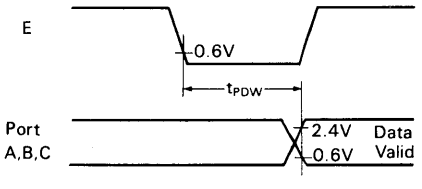


Figure 3 Port Data Delay Time (MCU Write)

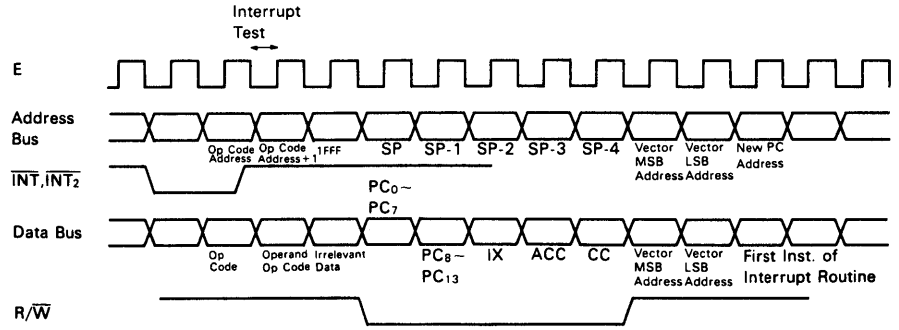


Figure 4 Interrupt Sequence

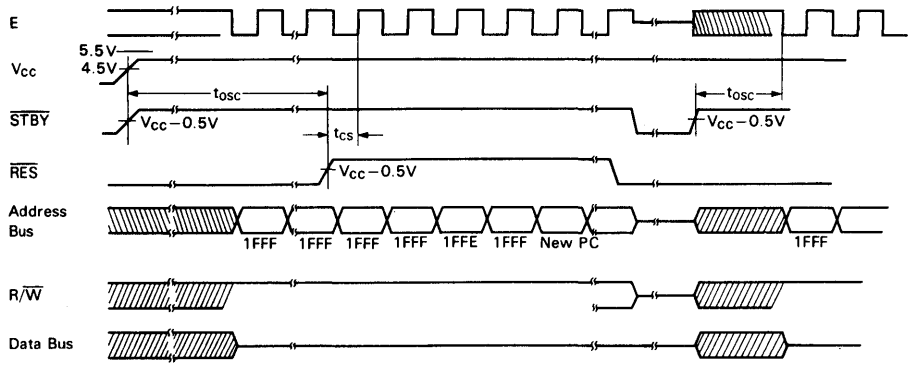


Figure5 Reset Timing

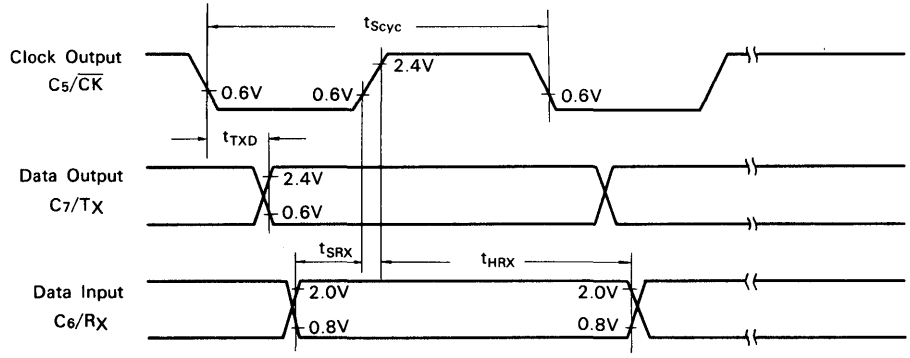


Figure6 SCI Timing (Internal Clock)

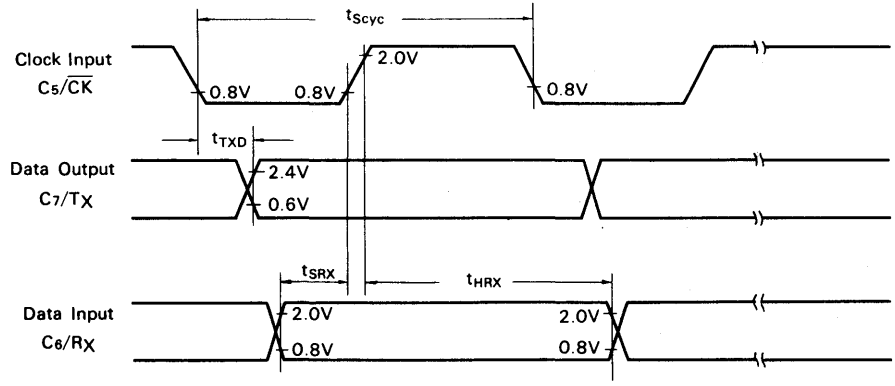
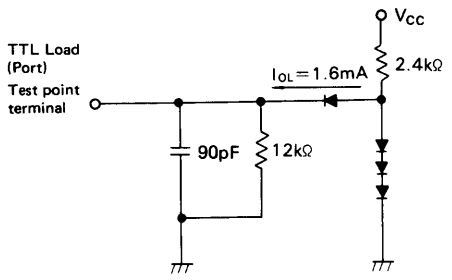


Figure7 SCI Timing(External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ

Figure 8 Test Load

■ DESCRIPTION OF TERMINAL FUNCTIONS

The input and output signals of the MCU are described here.

• V_{CC}, V_{SS}

Voltage is applied to the MCU through these two terminals. V_{CC} is 5.0V ± 10%, while V_{SS} is grounded.

• $\overline{INT}$, $\overline{INT_2}$

External interrupt request inputs to the MCU. For details, refer to "INTERRUPT". The $\overline{INT_2}$ terminal is also used as the port D_6 terminal.

• XTAL, EXTAL

These terminals provide input to the on-chip clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic filter is connected to the terminal. Refer to "INTERNAL OSCILLATOR" for using these input terminals.

• TIMER

This is an input terminal for event counter. Refer to "TIMER" for details.

• $\overline{RES}$

Used to reset the MCU. Refer to "RESET" for details.

• NUM

This terminal is not for user application. In case of the HD6305Y1, this terminal should be connected to V_{CC} through 10kΩ resistance. In case of the HD6305Y2, this terminal should be connected to V_{SS}.

• Enable (E)

This output terminal supplies E clock. Output is a single-phase, TTL compatible and 1/4 crystal oscillation frequency or 1/4 external clock frequency. It can drive one TTL load and a 90pF condenser.

• Read/Write (R/$\overline{W}$)

This TTL compatible output signal indicates to peripheral and memory devices whether MCU is in Read ("High"), or in Write ("Low"). The normal standby state is Read ("High"). Its output can drive one TTL load and a 90pF condenser.

• Data Bus ($DATA_0$ ~ $DATA_7$)

This TTL compatible three-state buffer can drive one TTL load and 90pF.

• Address Bus (ADR_0 ~ ADR_{13})

Each terminal is TTL compatible and can drive one TTL load and 90pF.

• Input/Output Terminals (A_0 ~ A_7, B_0 ~ B_7, C_0 ~ C_7)

These 24 terminals consist of three 8-bit I/O ports (A, B, C). Each of them can be used as an input or output terminal on a bit through program control of the data direction register. For details, refer to "I/O PORTS."

• Input Terminals (D_1 ~ D_7)

These seven input-only terminals are TTL or CMOS compatible. Of the port D's, D_6 is also used as $\overline{INT_2}$. If D_6 is used as a port, the $\overline{INT_2}$ interrupt mask bit of the miscellaneous register must be set to "1" to prevent an $\overline{INT_2}$ interrupt from being accidentally accepted.

• $\overline{STBY}$

This terminal is used to place the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and the internal condition is reset. For details, refer to "Stand-by Mode."

The terminals described in the following are I/O pins for serial communication interface (SCI). They are also used as ports C_5, C_6 and C_7. For details, refer to "SERIAL COMMUNICATION INTERFACE."

• $\overline{CK}$ (C_5)

Used to input or output clocks for serial operation.

• Rx (C_6)

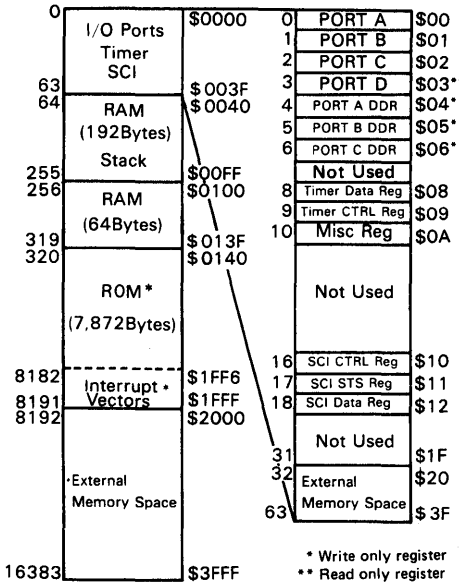
Used to receive serial data.

• Tx (C_7)

Used to transmit serial data.

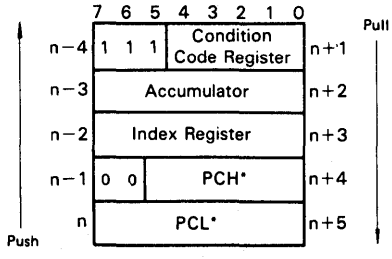
■ MEMORY MAP

The memory map of the MCU is shown in Fig. 9. $0140 ~ $1FFF of the HD6305Y2 are external addresses. However, care should be taken to assign vector addresses to $1FF6 ~ $1FFF. During interrupt processing, the contents of the CPU registers are saved into the stack in the sequence shown in Fig. 10. This saving begins with the lower byte (PCL) of the program counter. Then the value of the stack pointer is decremented and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CC) are stacked in that order. In a subroutine call, only the contents of the program counter (PCH and PCL) are stacked.



* ROM area ($0140 ~ $1FFF) in the HD6305Y2 is changed into External Memory Space.

Figure 9 Memory Map of MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 10 Sequence of Interrupt Stacking

■REGISTERS

There are five registers which the programmer can operate.

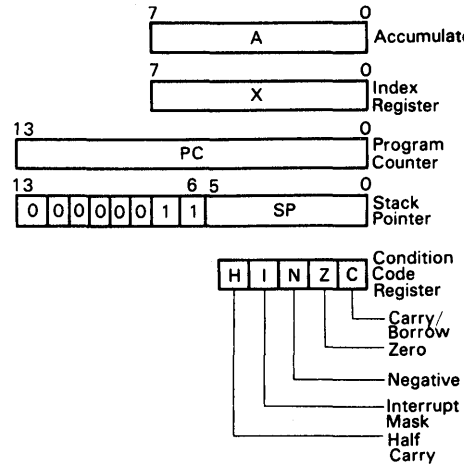


Figure 11 Programming Model

• Accumulator (A)

This accumulator is an ordinary 8-bit register which holds operands or the result of arithmetic operation or data processing.

• Index Register (X)

The index register is an 8-bit register, and is used for index addressing mode. Each of the addresses contained in the register consists of 8 bits which, combined with an offset value, provides an effective address.

In the case of a read/modify/write instruction, the index register can be used like an accumulator to hold operation data or the result of operation.

If not used in the index addressing mode, the register can be used to store data temporarily.

• Program Counter (PC)

The program counter is a 14-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 14-bit register that indicates the address of the next stacking space. Just after reset, the stack pointer is set at address $00FF. It is decremented when data is pushed, and incremented when pulled. The upper 8 bits of the stack pointer are fixed to 00000011. During the MCU being reset or during a reset stack pointer (RSP) instruction, the pointer is set to address $00FF. Since a subroutine or interrupt can use space up to address $00C1 for stacking, the subroutine can be used up to 31 levels and the interrupt up to 12 levels.

• Condition Code Register (CC)

The condition code register is a 5-bit register, each bit indicating the result of the instruction just executed. The bits can be individually tested by conditional branch instruc-

tions. The CC bits are as follows:

Half Carry (H): Used to indicate that a carry occurred between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts, except a software interrupt, to be masked. If an interrupt occurs with the bit I set, it is latched. It will be processed the instant the interrupt mask bit is reset. (More specifically, it will enter the interrupt processing routine after the instruction following the CLI has been executed.)

Negative (N): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is negative (bit 7 is logic "1").

Zero (Z): Used to indicate that the result of the most recent arithmetic operation, logical operation or data processing is zero.

Carry/ Borrow (C): Represents a carry or borrow that occurred in the most recent arithmetic operation. This bit is also affected by the Bit Test and Branch instruction and a Rotate instruction.

■ INTERRUPT

There are six different types of interrupt: external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER or the SCI and $TIMER_2$ generate the same vector address, respectively.

When an interrupt occurs, the program in progress stops and the then CPU status is saved onto the stack. And then, the interrupt mask bit (I) of the condition code register is set and the start address of the interrupt processing routine is obtained from a particular interrupt vector address. Then the interrupt routine starts from the start address. System can exit from the interrupt routine by an RTI instruction. When this instruction is executed, the CPU status before the interrupt (saved onto the stack) is pulled and the CPU restarts the sequence with the instruction next to the one at which the interrupt occurred. Table 1 lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$1FFE, $1FFF
SWI	2	$1FFC, $1FFD
$\overline{INT}$	3	$1FFA, $1FFB
TIMER/$\overline{INT_2}$	4	$1FF8, $1FF9
SCI/$TIMER_2$	5	$1FF6, $1FF7

A flowchart of the interrupt sequence is shown in Fig. 12. A block diagram of the interrupt request source is shown in Fig. 13.

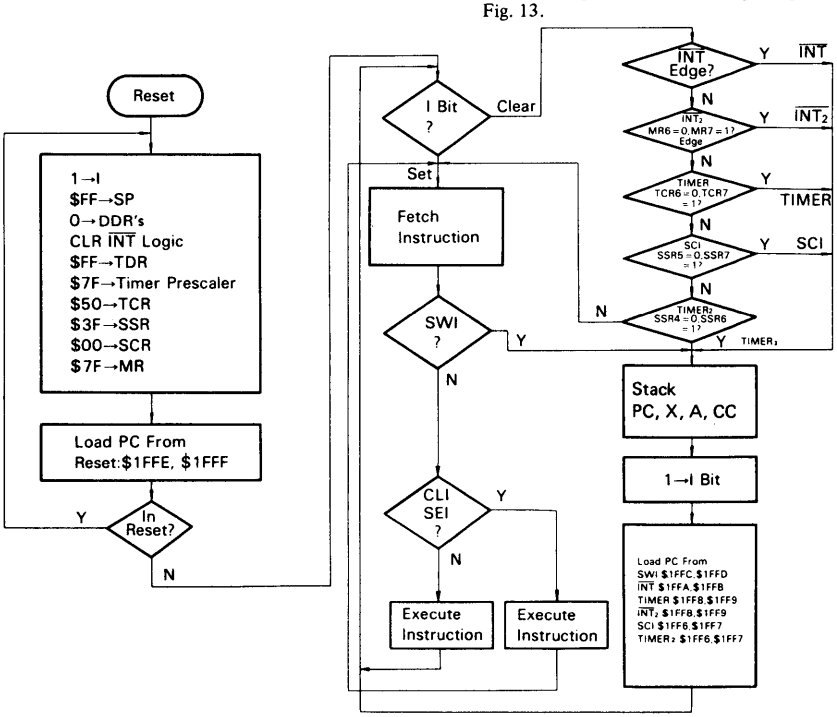


Figure 12 Interrupt Flow Chart

In the block diagram, both the external interrupts $\overline{INT}$ and $\overline{INT_2}$ are edge trigger inputs. At the falling edge of each input, an interrupt request is generated and latched. The $\overline{INT}$ interrupt request is automatically cleared if jumping is made to the $\overline{INT}$ processing routine. Meanwhile, the $\overline{INT_2}$ request is cleared if "0" is written in bit 7 of the miscellaneous register.

For the external interrupts ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, $TIMER_2$) and serial interrupt (SCI), each interrupt request is held, but not processed, if the I bit of the condition code register is set. Immediately after the I bit is cleared, the corresponding interrupt processing starts according to the priority.

The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by setting bit 6 of the timer control register; the SCI interrupt by setting bit 5 of the serial status register; and the $TIMER_2$ interrupt by setting bit 4 of the serial status register.

The status of the $\overline{INT}$ terminal can be tested by a BIL or BIH instruction. The $\overline{INT}$ falling edge detector circuit and its latching circuit are independent of testing by these instructions. This is also true with the status of the $\overline{INT_2}$ terminal.

• Miscellaneous Register (MR; $000A)

The interrupt vector address for the external interrupt $\overline{INT_2}$ is the same as that for the TIMER interrupt, as shown in Table 1. For this reason, a special register called the miscellaneous register (MR; $000A) is available to control the $\overline{INT_2}$ interrupts.

Bit 7 of this register is the $\overline{INT_2}$ interrupt request flag. When the falling edge is detected at the $\overline{INT_2}$ terminal, "1" is set in bit 7. Then the software in the interrupt routine (vector addresses: $1FF8, $1FF9) checks bit 7 to see if it is $\overline{INT_2}$ interrupt. Bit 7 can be reset by software.

Miscellaneous Register (MR;$000A)

7	6	5	4	3	2	1	0
MR7	MR6						

$\overline{INT_2}$ Interrupt Mask

$\overline{INT_2}$ Interrupt Request Flag

Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If this bit is set to "1", then the $\overline{INT_2}$ interrupt is disabled. Both read and write are possible with bit 7 but "1" cannot be written in this bit by software. This means that an interrupt request by software is impossible.

When reset, bit 7 is cleared to "0" and bit 6 is set to "1".

■ TIMER

Figure 14 shows a MCU timer block diagram. The timer data register is loaded by software and, upon receipt of a clock input, begins to count down. When the timer data

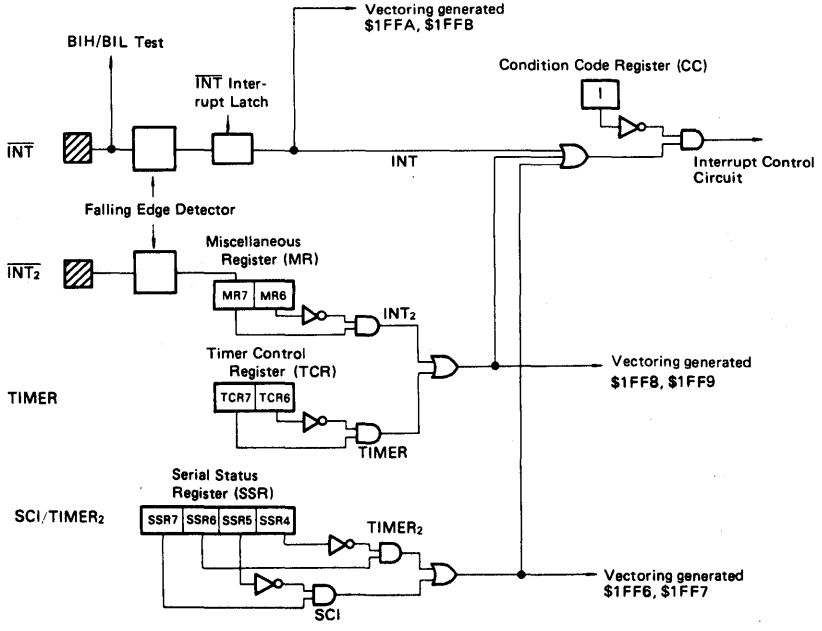


Figure 13 Interrupt Request Generation Circuitry

register (TDR) becomes "0", the timer interrupt request bit (bit 7) in the timer control register is set. In response to the interrupt request, the CPU saves its status into the stack and fetches timer interrupt routine address from addresses $1FF8 and $1FF9 and execute the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also mask the timer interrupt.

The source clock to the timer can be either an external signal from the timer input terminal or the internal E signal (the oscillator clock divided by 4). If the E signal is used as the source, the clock input can be gated by the input to the timer input terminal.

Once the timer count has reached "0", it starts counting down with "$FF". The count can be monitored whenever desired by reading the timer data register. This permits the program to know the length of time having passed after the occurrence of a timer interrupt, without disturbing the contents of the counter.

When the MCU is reset, both the prescaler and counter are initialized to logic "1". The timer interrupt request bit (bit 7) then is cleared and the timer interrupt mask bit (bit 6) is set.

To clear the timer interrupt request bit (bit 7), it is necessary to write "0" in that bit.

TCR7	Timer interrupt request
0	Absent
1	Present

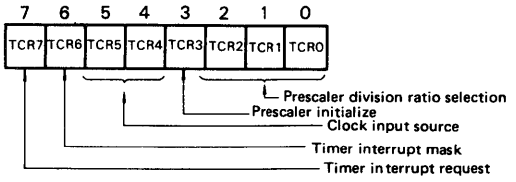
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

• Timer Control Register (TCR; $0009)

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After reset, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$FF" immediately after reset.

When "1" is written in bit 3, the prescaler is initialized. This bit always shows "0" when read.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under timer terminal control
1	0	No clock input (counting stopped)
1	1	Event input from timer terminal

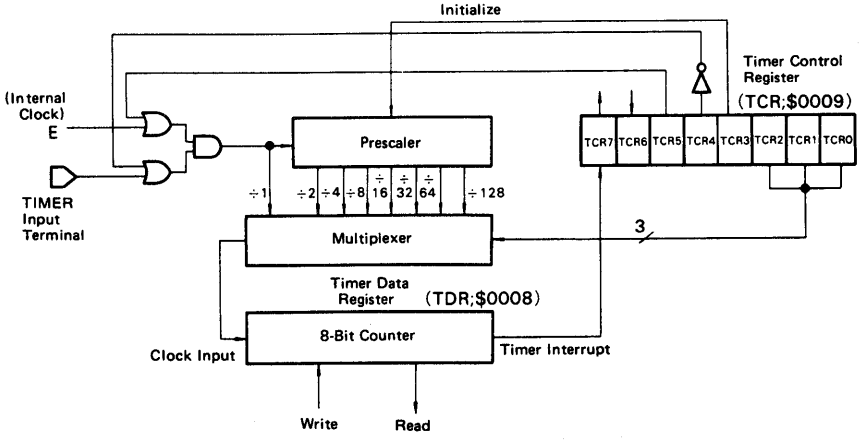


Figure 14 Timer Block Diagram

A prescaler division ratio is selected by the combination of three bits (bits 0, 1 and 2) of the timer control register (see Table 3). There are eight different division ratios: ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128. After reset, the TCR is set to the ÷1 mode.

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

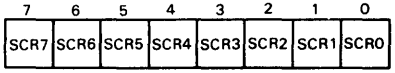
A timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. This bit can be cleared by writing "0" in that bit.

■SERIAL COMMUNICATION INTERFACE (SCI)

This interface is used for serial transmission or reception of 8-bit data. Sixteen transfer rates are available in the range from 1 μs to approx. 32 ms (for oscillation at 4 MHz).

The SCI consists of three registers, one eighth counter and one prescaler. (See Fig. 15.) SCI communicates with the CPU via the data bus, and with the outside world through bits 5, 6 and 7 of port C. Described below are the operations of each register and data transfer.

•SCI Control Register (SCR; $0010)



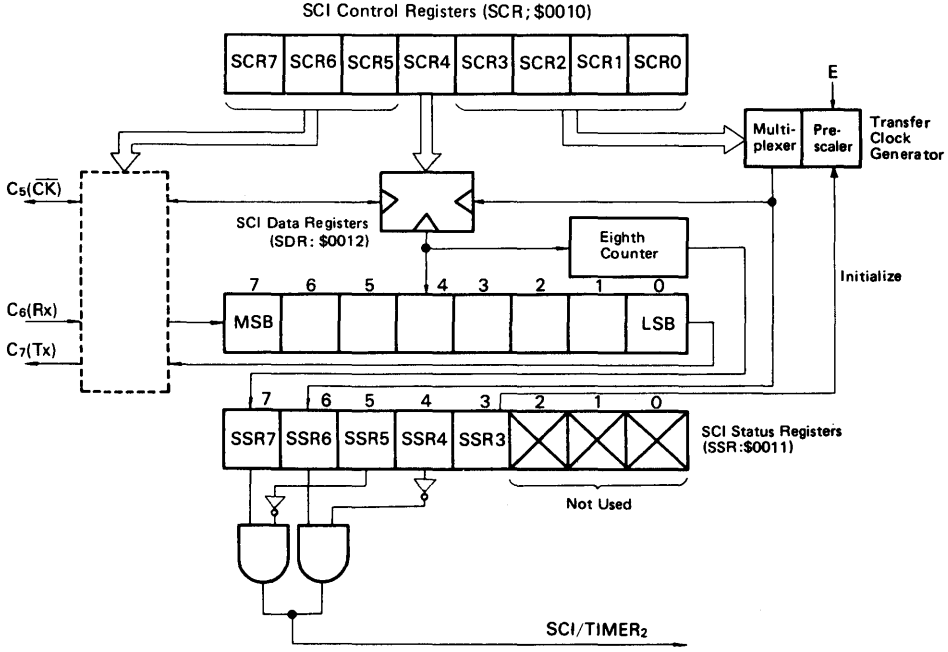


Figure 15 SCI Block Diagram

SCR7	C_7 terminal
0	Used as I/O terminal (by DDR).
1	Serial data output (DDR output)

SCR6	C_6 terminal
0	Used as I/O terminal (by DDR).
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	C_5 terminal
0	0	–	Used as I/O terminal (by DDR).
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)
When this bit is set, the DDR corresponding to the C_7 becomes "1" and this terminal serves for output of SCI data. After reset, the bit is cleared to "0".

Bit 6 (SCR6)
When this bit is set, the DDR corresponding to the C_6 becomes "0" and this terminal serves for input of SCI data. After reset, the bit is cleared to "0".

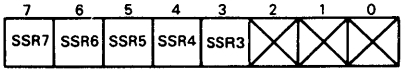
Bits 5 and 4 (SCR5, SCR4)
These bits are used to select a clock source. After reset, the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)
These bits are used to select a transfer clock rate. After reset, the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

•SCI Data Register (SDR; $0012)
A serial-parallel conversion register that is used for transfer of data.

•SCI Status Register (SSR; $0011)



Bit 7 (SSR7)
Bit 7 is the SCI interrupt request bit which is set upon completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5="1". The bit can also be cleared by writing "0" in it.

Bit 6 (SSR6)
Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is used commonly with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When reset, the bit is cleared. It also be cleared by writing "0" in it. (For details, see $TIMER_2$.)

Bit 5 (SSR5)
Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When reset, it is set to "1"

Bit 4 (SSR4)
Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When reset, it is set to "1".

Bit 3 (SSR3)
When "1" is written in this bit, the prescaler of the transfer clock generator is initialized. When read, the bit always is "0".

Bits 2 ~ 0
Not used.

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

• Data Transmission
By writing the desired control bits into the SCI control registers, a transfer rate and a source of transfer clock are determined and bits 7 and 5 of port C are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the C_7/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock. (See Fig. 16.) When 8 bit of

data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register with the rising edge of the last serial clock. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the C_7/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the C_5/$\overline{CK}$ terminal is set as input. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

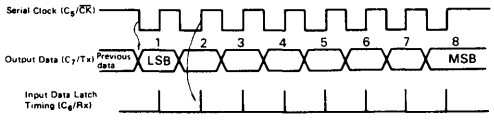


Figure 16 SCI Timing Chart

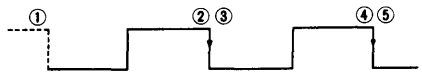
• Data Reception

By writing the desired control bits into the SCI control register, a transfer rate and a source of transfer clock are determined and bit 6 and 5 of port C are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed after reading subsequent received data. It must be taken after reset and after not reading subsequent received data.)

The data from the C_6/Rx terminal is input to the SCI data register synchronously with the rising edge of the serial clock (see Fig. 16). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored and the data is received synchronously with the clock from the C_5/$\overline{CK}$ terminal. If the internal clock has been selected, the C_5/$\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

• TIMER$_2$

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 ~ 0 of the SCI control register (4 μs ~ approx. 32 ms (for oscillation at 4 MHz)) is input to bit 6 of the SCI status register and the TIMER$_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, TIMER$_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : TIMER$_2$ interrupt request
③, ⑤ : TIMER$_2$ interrupt request bit cleared

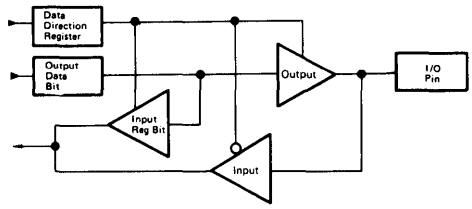
TIMER$_2$ is commonly used with the SCI transfer clock generator. If wanting to use TIMER$_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■ I/O PORTS

There are 24 input/output terminals (ports A, B, C). Each I/O terminal can be selected for either input or output by the data direction register. More specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B or C reads latched data if it has been programmed as output, even with the output level being fluctuated by the output load. (See Fig. 17.)

When reset, the data direction register and data register go to "0" and all the input/output terminals are used as input.



Bit of data direction register	Bit of output data	Status of output	Input to CPU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

Figure 17 Input/Output Port Diagram

Seven input-only terminals are available (port D). Writing to an input terminal is invalid.

All input/output terminals and input terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports or input ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite that they are not used.

■ RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 18.) On power up, the reset input must be held "Low" for at least t_{OSC} to assure that the internal oscillator is stabilized. A sufficient time of delay can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 19.

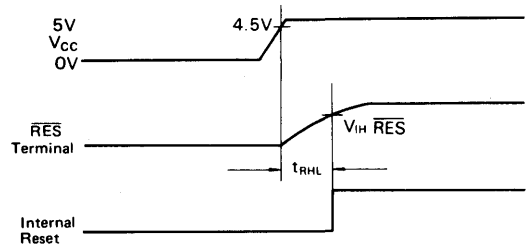


Figure 18 Power On and Reset Timing

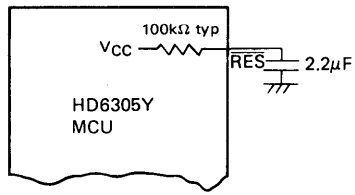


Figure 19 Input Reset Delay Circuit

■INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0MHz) or ceramic oscillator between pins 5 and 6 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 20. Figs. 21 and 22 illustrate the specifications and typical arrangement of the crystal, respectively.

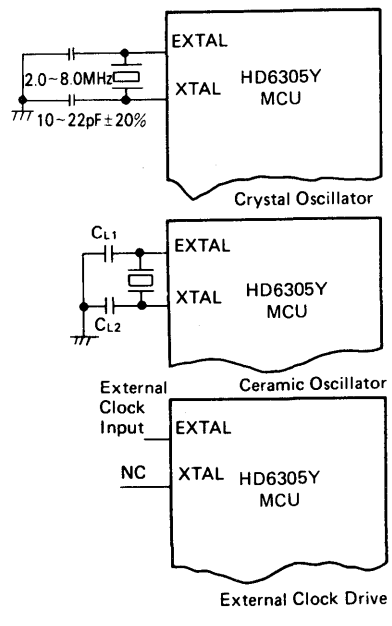


Figure 20 Internal Oscillator Circuit

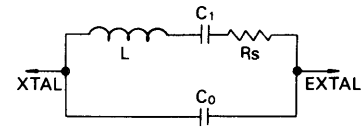
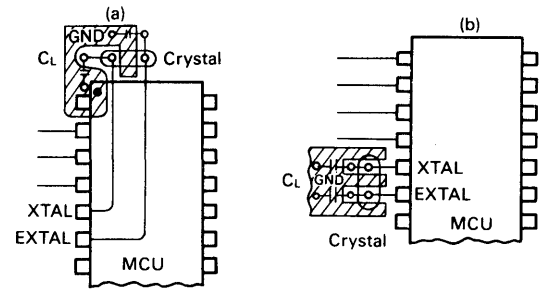


Figure 21 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 22 Typical Crystal Arrangement

■LOW POWER DISSIPATION MODE

The HD6305Y has three low power dissipation modes: wait, stop and standby.

• Wait Mode

When WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers, RAM and I/O terminals hold their condition just before entering into the wait mode.

The escape from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the wait mode the MCU executes the instruction next to the WAIT. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$) is masked by the timer control

register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 23 shows a flowchart for the wait function.

• Stop Mode

When STOP instruction being executed, MCU enters into the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive but the RAM, registers and I/O terminals hold their condition just before entering into the stop mode.

The escape from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings into the standby mode.

When interrupt is requested to the CPU and accepted, the stop mode escapes, then the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after releasing from the stop mode, the MCU executes the instruction next to the STOP. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 24 shows a flowchart for the stop function. Fig. 25 shows a timing chart of return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$="0" must exceed 30 ms to assure stabilized oscillation.

• Standby Mode

The MCU enters into the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are hold. The I/O terminals turn to high-impedance state. The standby mode should escape by bringing $\overline{STBY}$ "High". The CPU must be restarted by reset. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 26.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 27.

(Note)

When I bit of condition code register is "1" and interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$, SCI/TIMER$_2$) is held, MCU does not enter WAIT mode by the execution of WAIT instruction.

In that case, after the 4 dummy cycles MCU executes the next instruction.

In the same way, when external interrupts ($\overline{INT}$, $\overline{INT_2}$) are held at the bit I set, MCU does not enter STOP mode by the execution of STOP instruction. In that case, also, MCU executes the next instruction after the 4 dummy cycles.

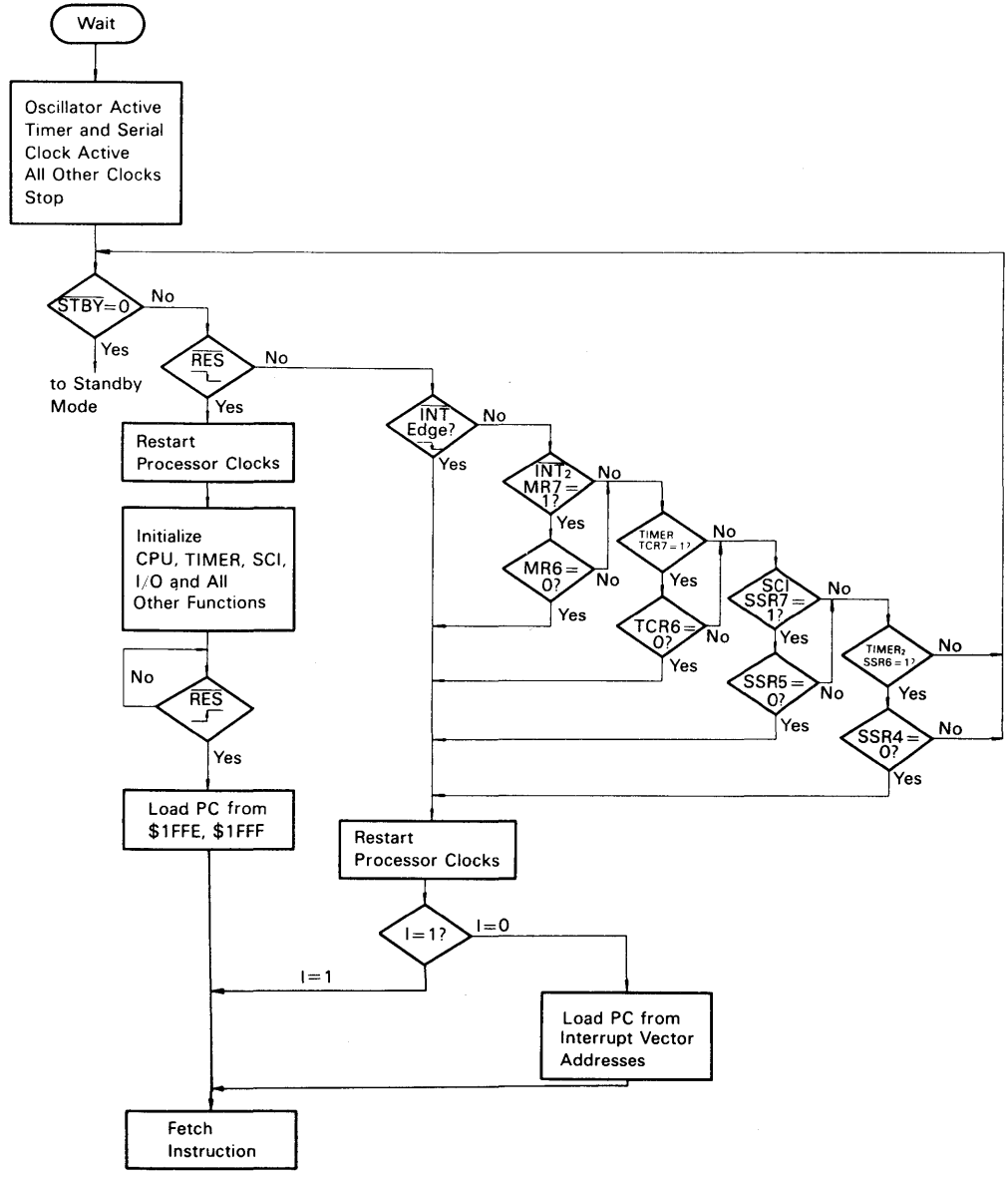


Figure 23 Wait Mode Flow Chart

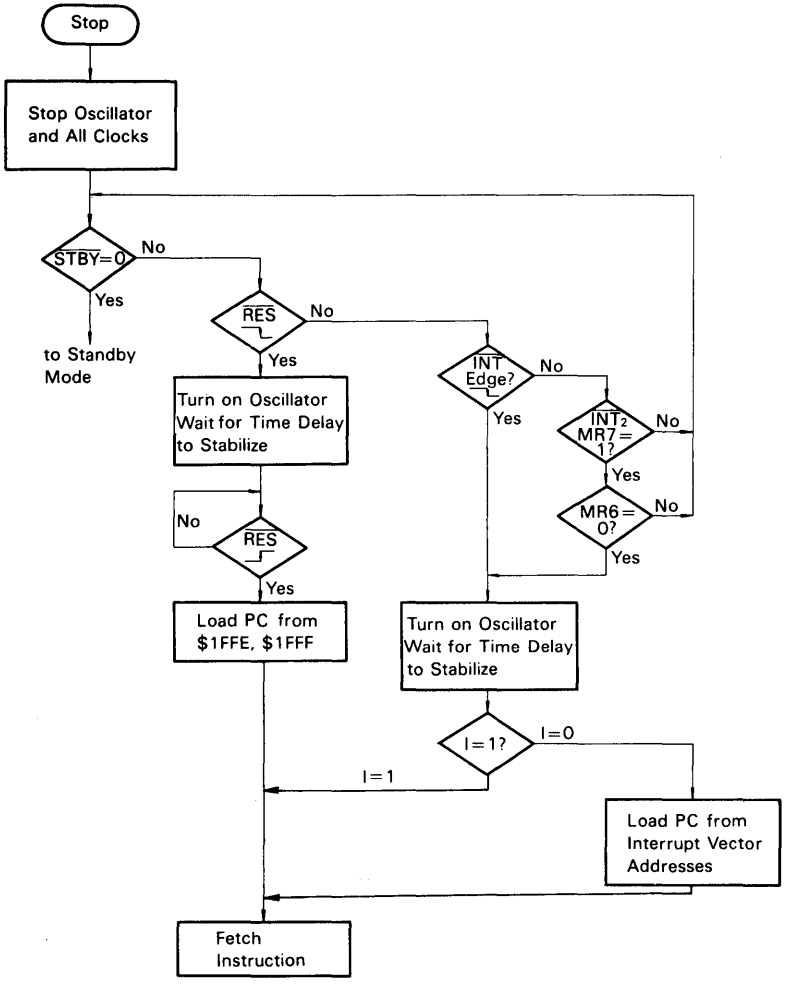
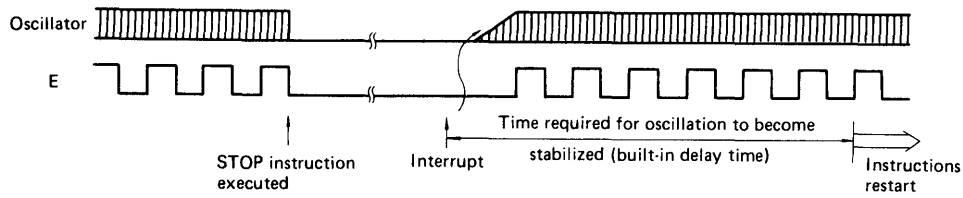
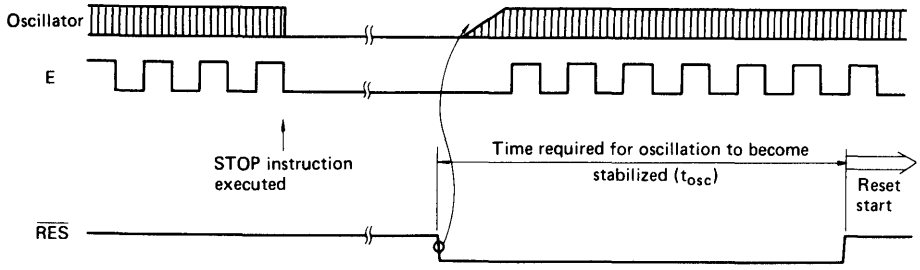


Figure 24 Stop Mode Flow Chart



(a) Restart by Interrupt



(b) Restart by Reset

Figure 25 Timing Chart of Releasing from Stop Mode

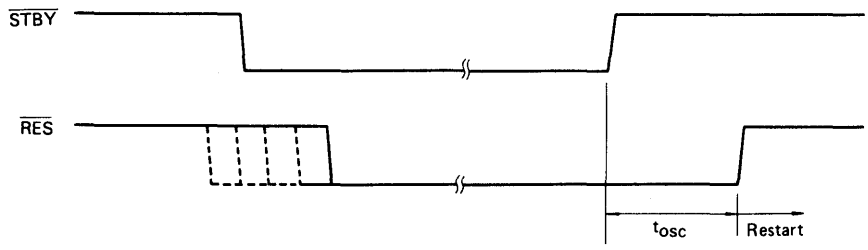


Figure 26 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Keep	Keep	Keep	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Keep	High impedance	$\overline{STBY}$="High"

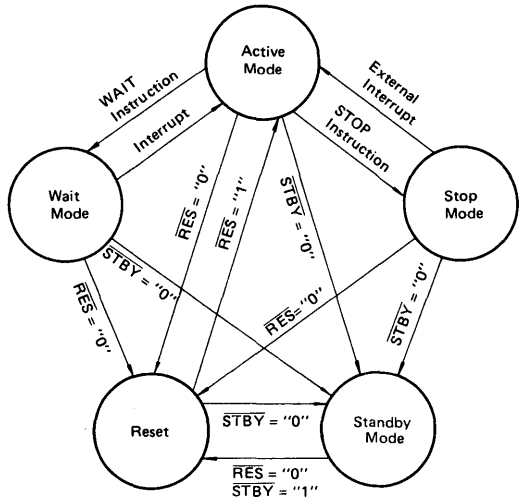


Figure 27 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■BIT MANIPULATION

The MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM within page 0 or an I/O port (except the write-only registers such as the data direction register). Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM on page 0, or I/O can be manipulated, the user may use a bit within the RAM on page 0 as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 28 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the ROM. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
          :
SELF 1.   BRCLR 0, PORT A, SELF 1
          BSET 1, PORT A
          BCLR 1, PORT A
          :
```

Figure 28 Example of Bit Manipulation

■ADDRESSING MODES

Ten different addressing modes are available to the MCU.

• Immediate

See Fig. 29. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 30. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. 192 byte RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 31. The extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 32. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 33. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 34. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 35. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 36. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 37. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The value of the test bit is written in the carry bit of the condition code register.

• Implied

See Fig. 38. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

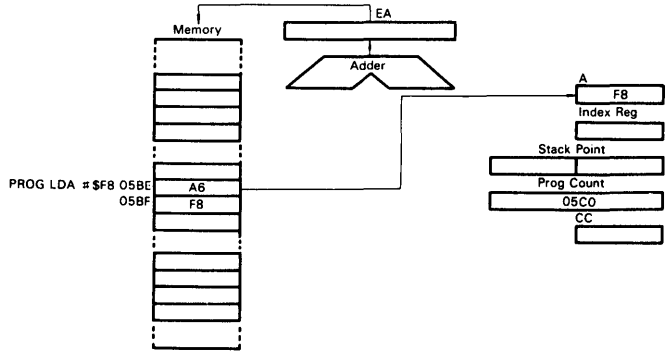


Figure 29 Example of Immediate Addressing

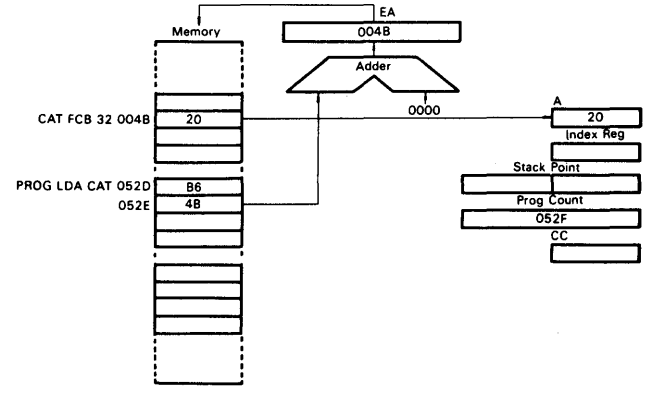


Figure 30 Example of Direct Addressing

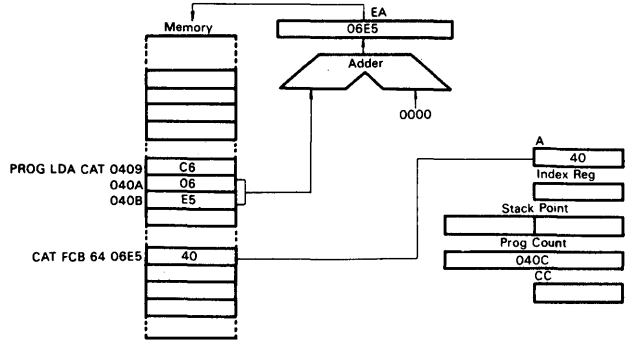


Figure 31 Example of Extended Addressing

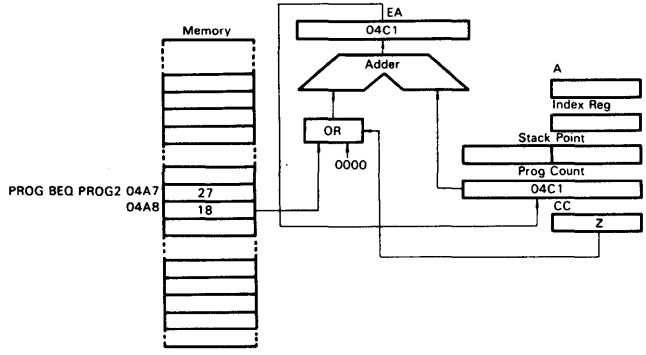


Figure 32 Example of Relative Addressing

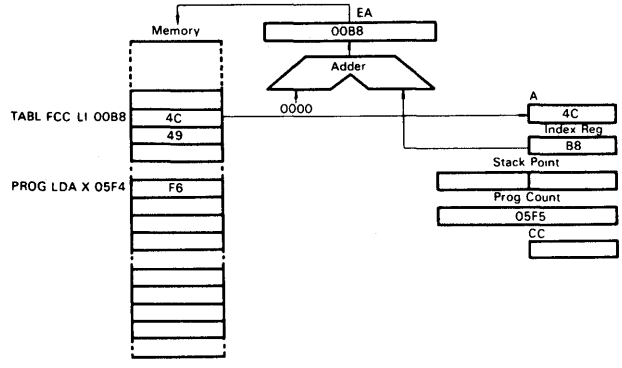


Figure 33 Example of Indexed (No Offset) Addressing

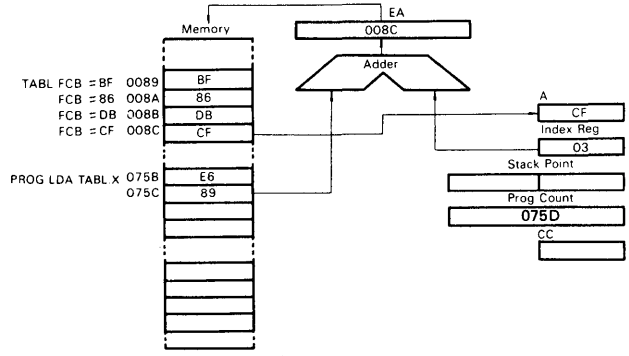


Figure 34 Example of Index (8-bit Offset) Addressing

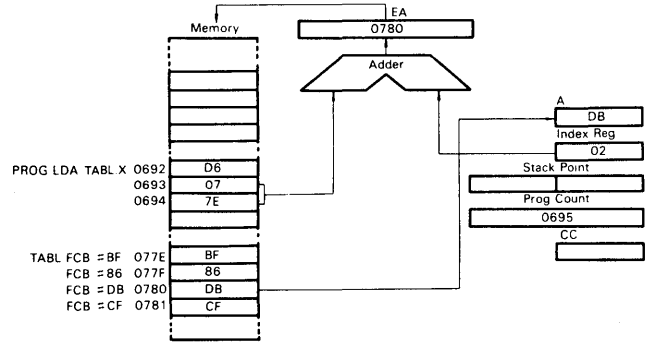


Figure 35 Example of Index (16-bit Offset) Addressing

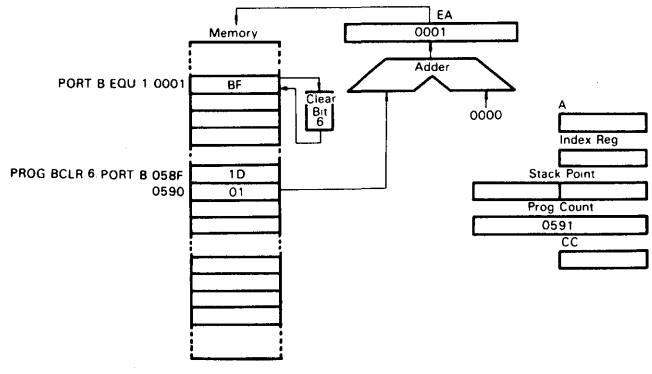


Figure 36 Example of Bit Set/Clear Addressing

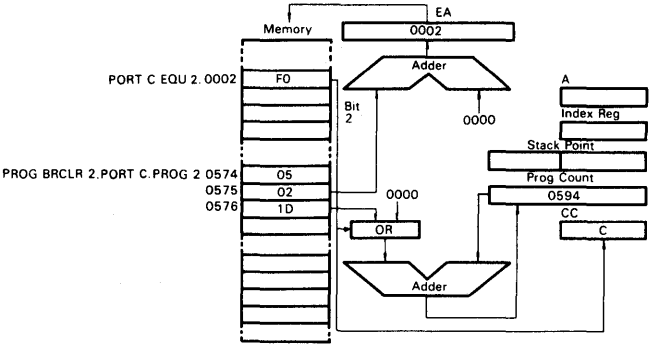


Figure 37 Example of Bit Test and Branch Addressing

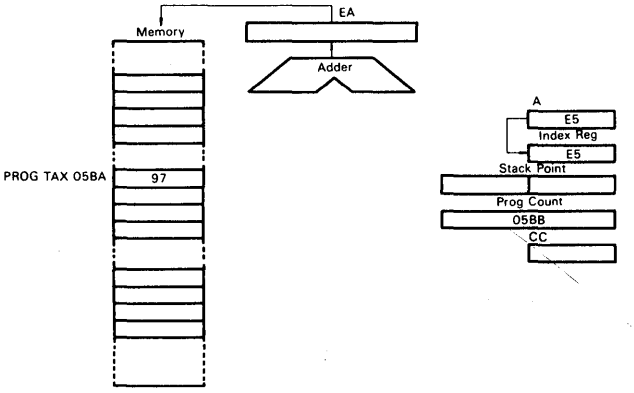


Figure 38 Example of Implied Addressing

■ INSTRUCTION SET

There are 62 basic instructions available to the HD6305Y MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the HD6305Y MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the HD6305Y MCU in the alphabetical order.

• Operation Code Map

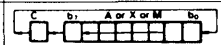
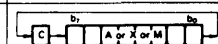
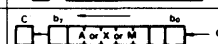
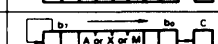
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate OP	#	~	Direct OP	#	~	Extended OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Indexed (16-Bit Offset) OP	#	~	Boolean/ Arithmetic Operation	Condition Code H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	^	^	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	^	^	•
Store A in Memory	STA	–	–	–	B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	^	^	•
Store X in Memory	STX	–	–	–	BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	^	^	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	^	•	^	^	^
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	^	•	^	^	^
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	^	^	^
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	^	^	^
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	^	^	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	^	^	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	^	^	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	^	^	^
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	^	^	^
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	^	^	•
Jump Unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A) OP	#	~	Implied(X) OP	#	~	Direct OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Boolean/Arithmetic Operation	Condition Code H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	^	^	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	^	^	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	^	^	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	^	^	^
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	^	^	^
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	^	^	^
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	^	^	^
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	^	^
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	^	^	^
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	^	^	^
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	^	^	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	•	•	•	•	•
Branch Never	BRN	21	2	3	None	•	•	•	•	•
Branch IF Higher	BHI	22	2	3	C+Z=0	•	•	•	•	•
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	•	•	•	•	•
Branch IF Carry Clear	BCC	24	2	3	C=0	•	•	•	•	•
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	•	•	•	•	•
Branch IF Carry Set	BCS	25	2	3	C=1	•	•	•	•	•
(Branch IF Lower)	(BLO)	25	2	3	C=1	•	•	•	•	•
Branch IF Not Equal	BNE	26	2	3	Z=0	•	•	•	•	•
Branch IF Equal	BEQ	27	2	3	Z=1	•	•	•	•	•
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	•	•	•	•	•
Branch IF Half Carry Set	BHCS	29	2	3	H=1	•	•	•	•	•
Branch IF Plus	BPL	2A	2	3	N=0	•	•	•	•	•
Branch IF Minus	BMI	2B	2	3	N=1	•	•	•	•	•
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	•	•	•	•	•
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	•	•	•	•	•
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	•	•	•	•	•
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	•	•	•	•	•
Branch to Subroutine	BSR	AD	2	5	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	•	•	•	•	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	•	•	•	•	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	•	•	•	•	•
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes: Implied OP	#	~	Boolean Operation	Condition Code: H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	●	●	●	●	●
Transfer X to A	TXA	9F	1	2	X→A	●	●	●	●	●
Set Carry Bit	SEC	99	1	1	1→C	●	●	●	●	1
Clear Carry Bit	CLC	98	1	1	0→C	●	●	●	●	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	●	1	●	●	●
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	●	0	●	●	●
Software Interrupt	SWI	83	1	10		●	1	●	●	●
Return from Subroutine	RTS	81	1	5		●	●	●	●	●
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	●	●	●	●	●
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	●	●	●	●	●
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	●	●	^	^	^*
Stop	STOP	8E	1	4		●	●	●	●	●
Wait	WAIT	8F	1	4		●	●	●	●	●

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes: Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	Condition Code: H	I	N	Z	C
ADC		×	×	×		×	×	×			^	●	^	^	^
ADD		×	×	×		×	×	×			^	●	^	^	^
AND		×	×	×		×	×	×			●	●	^	^	●
ASL	×		×			×	×				●	●	^	^	^
ASR	×		×			×	×				●	●	^	^	^
BCC					×						●	●	●	●	●
BCLR									×		●	●	●	●	●
BCS					×						●	●	●	●	●
BEQ					×						●	●	●	●	●
BHCC					×						●	●	●	●	●
BHCS					×						●	●	●	●	●
BHI					×						●	●	●	●	●
(BHS)					×						●	●	●	●	●
BIH					×						●	●	●	●	●
BIL					×						●	●	●	●	●
BIT		×	×	×		×	×	×			●	●	^	^	●
(BLO)					×						●	●	●	●	●
BLS					×						●	●	●	●	●
BMC					×						●	●	●	●	●
BMI					×						●	●	●	●	●
BMS					×						●	●	●	●	●
BNE					×						●	●	●	●	●
BPL					×						●	●	●	●	●
BRA					×						●	●	●	●	●

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						●	●	●	●	●
BRCLR										×	●	●	●	●	∧
BRSET										×	●	●	●	●	∧
BSET									×		●	●	●	●	●
BSR					×						●	●	●	●	●
CLC	×										●	●	●	●	0
CLI	×										●	0	●	●	●
CLR	×		×			×	×				●	●	0	1	●
CMP		×	×	×		×	×	×			●	●	∧	∧	∧
COM	×		×			×	×				●	●	∧	∧	1
CPX		×	×	×		×	×	×			●	●	∧	∧	∧
DAA	×										●	●	∧	∧	∧
DEC	×		×			×	×				●	●	∧	∧	●
EOR		×	×	×		×	×	×			●	●	∧	∧	●
INC	×		×			×	×				●	●	∧	∧	●
JMP			×	×		×	×	×			●	●	●	●	●
JSR			×	×		×	×	×			●	●	●	●	●
LDA		×	×	×		×	×	×			●	●	∧	∧	●
LDX		×	×	×		×	×	×			●	●	∧	∧	●
LSL	×		×			×	×				●	●	∧	∧	∧
LSR	×		×			×	×				●	●	0	∧	∧
NEG	×		×			×	×				●	●	∧	∧	∧
NOP	×										●	●	●	●	●
ORA		×	×	×		×	×	×			●	●	∧	∧	●
ROL	×		×			×	×				●	●	∧	∧	∧
ROR	×		×			×	×				●	●	∧	∧	∧
RSP	×										●	●	●	●	●
RTI	×										?	?	?	?	?
RTS	×										●	●	●	●	●
SBC		×	×	×		×	×	×			●	●	∧	∧	∧
SEC	×										●	●	●	●	1
SEI	×										●	1	●	●	●
STA			×	×		×	×	×			●	●	∧	∧	●
STOP	×										●	●	●	●	●
STX			×	×		×	×	×			●	●	∧	∧	●
SUB		×	×	×		×	×	×			●	●	∧	∧	∧
SWI	×										●	1	●	●	●
TAX	×										●	●	●	●	●
TST	×		×			×	×				●	●	∧	∧	●
TXA	×										●	●	●	●	●
WAIT	×										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	—	SUB						0
1	BRCLR0	BCLR0	BRN	—					RTS*	—	CMP						1
2	BRSET1	BSET1	BHI	—					—	—	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	—	CPX						3
4	BRSET2	BSET2	BCC	LSR					—	—	AND						4
5	BRCLR2	BCLR2	BCS	—					—	—	BIT						5
6	BRSET3	BSET3	BNE	ROR					—	—	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					—	TAX*	—	STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL					—	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					—	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					—	CLI*	ORA						A
B	BRCLR5	BCLR5	BMI	—					—	SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC					—	RSP*	—	JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL	—					STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*	—	STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

↓ LOW

(NOTES) 1. "—" is an undefined operation code.
2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD6305Y:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

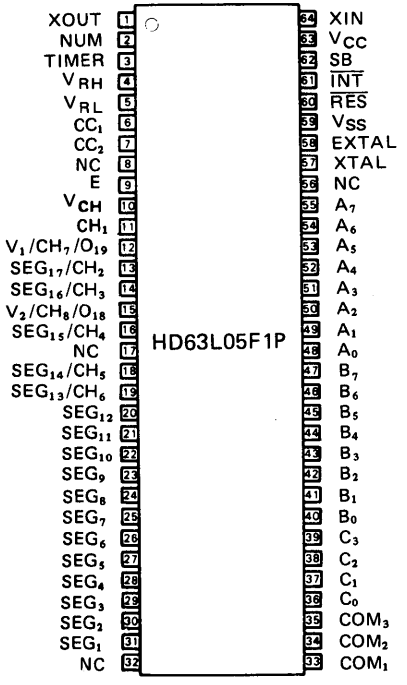
HD63L05F1
CMOS MCU (Microcomputer Unit)

The HD63L05F1 is a CMOS single-chip microcomputer suitable for low-voltage and low-current operation. Having CPU functions similar to those of the HMCS6800 family, the HD63L05F1 is equipped with a 4k bytes ROM, 96 bytes RAM, I/O, timer, 8 bits A/D, and LCD (7 x 7 segments max.) drivers, all on one chip.

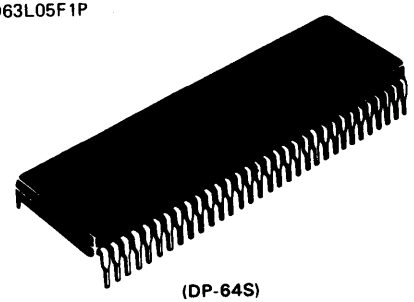
- **HARDWARE FEATURES**
- 3V Power Supply
- 8-Bit Architecture
- Built-in 4k Bytes ROM (Mask ROM)
- Built-in 96 Bytes RAM
- 20 Parallel I/O Ports
- Built-in 7 x 7 Segments LCD Driver Capability
- Built-in 8-Bit Timer
- Built-in 8-Bit A/D Converter
- Program Halt Function for Low Power Dissipation
- Stand-by Input Terminal for Data Holding

- **SOFTWARE FEATURES**
- An Instruction Set Similar to That of The HMCS6800 Family (Compatible with The HD6805S)
- HMCS6800 Family Software Development System is Applicable

- **PIN ARRANGEMENT (Top View)**

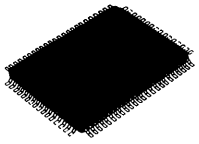


HD63L05F1P

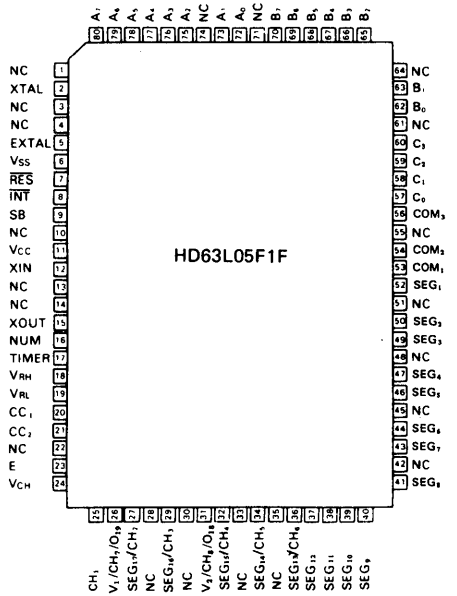


(DP-64S)

HD63L05F1F

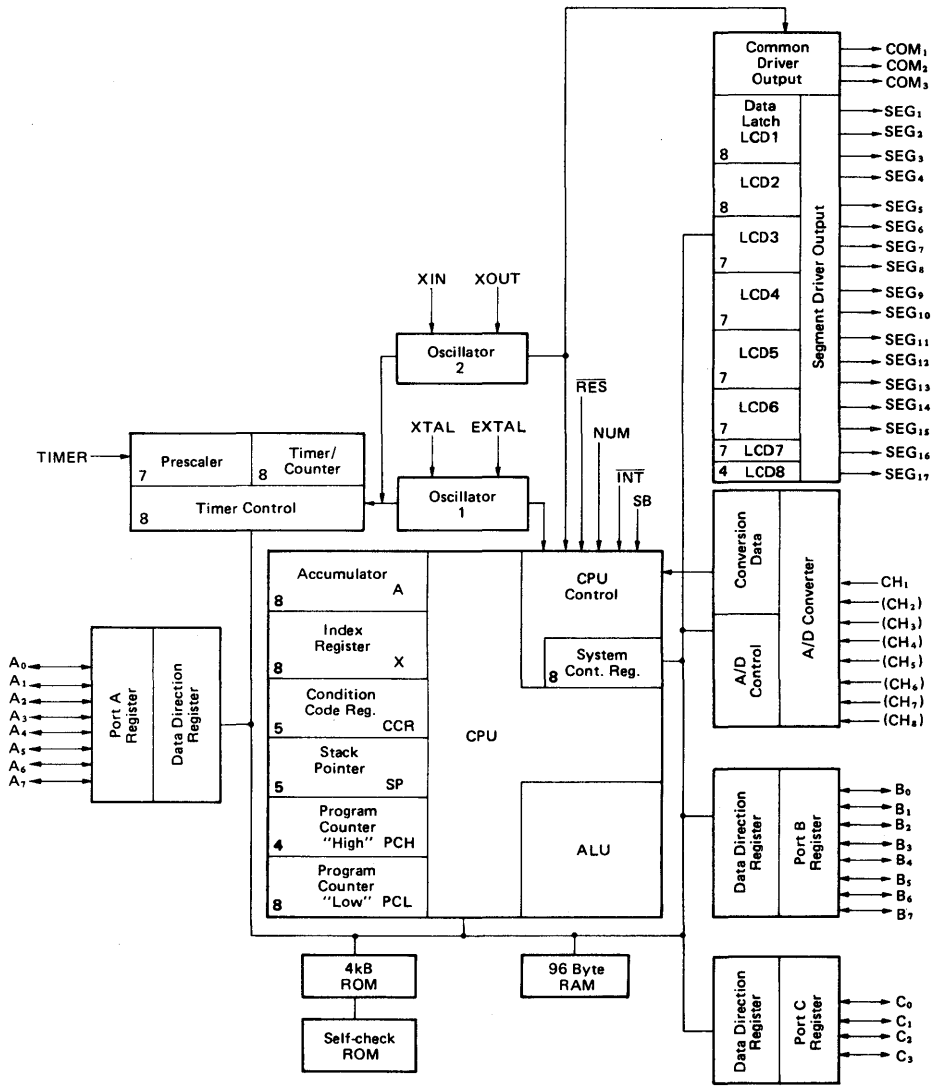


(FP-80)



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +5.5	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Output Voltage	V_{out}	-0.3 ~ V_{CC}+0.3	V
Operating Temparature	T_{opr}	-20 ~ +75	°C
Storage Temparature	T_{stg}	-55 ~ +125	°C

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded.
Normal operation should be under recommended operating conditions.
If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS (V_{CC} = 3.0V ±0.8V, V_{SS} = 0V, T_a = -20 ~ +75°C, typ means typical value at V_{CC} = 3.0V, unless otherwise noted.)

• DC CHARACTERISTICS

Item			Symbol	Test Condition	min	typ	max	Unit
Input "High" Level Voltage	XTAL, XIN		V_{IH}	Connect C_L = 0.5μF to V_{CH}	V_{CC}-0.3	–	V_{CC}	V
	$\overline{RES}$, $\overline{INT}$, SB				0.5V_{CC}+0.9	–	V_{CC}	V
	TIMER				0.8V_{CC}	–	V_{CC}	V
	NUM (Normal Mode)				V_{CC}-0.2	–	V_{CC}	V
Input "Low" Level Voltage	XTAL, XIN		V_{IL}	Connect C_L = 0.5μF to V_{CH}	V_{CC}-2.1	–	V_{CC}-1.8	V
	$\overline{RES}$, $\overline{INT}$, SB				V_{SS}	–	0.2V_{CC}	V
	TIMER				V_{SS}	–	0.2V_{CC}	V
	NUM (Test Mode)				V_{SS}	–	0.2	V
Self Check Input Voltage	NUM (Self Check Mode)		V_{IM}		0.5V_{CC}-0.2	–	0.5V_{CC}+0.2	V
Input Pull-Up Current	$\overline{RES}$ ($\overline{INT}$: Mask Option) NUM		$-I_{R1}$	V_{CC} = 3.0V, V_{in} = 0V	3	15	30	μA
Input Leakage Current	TIMER, SB		$\|I_{IN}\|$	V_{in} = 0V ~ V_{CC}	–	–	1.0	μA
Current Dissipation	Crystal* Oscillation	During System Operation	I_{CC1}	f = 400kHz No load. Tested after setting up the internal status by self check.	–	100	200	μA
		At Halt			–	40	80	μA
		At Standby			–	2	5	μA
		At A/D Operation			–	200	600	μA
	RC* Oscillation	During System Operation	I_{CC2}	R = 100kΩ No load. Tested after setting up the internal status by self check.	–	120	200	μA
		At Halt			–	60**	100**	μA
		At Standby			–	2	5	μA
		At A/D Operation			–	220	600	μA
Output "Low" Level Voltage	E		V_{OL}	I_{OL} = 30μA	–	–	0.3	V

* Depends on the mask-option.
** 60μA → 30μA and 100μA → 60μA when OSC1 is stopped by Halt.

• AC CHARACTERISTICS

Item		Symbol	Test Condition	min	typ	max	Unit
Operating Clock Frequency		f_{cl}		100	400	500	kHz
Cycle Time		t_{cyc}		8	10	40	μs
Oscillation Frequency * (Resistor Option)		f_{OSCR}	R = 100kΩ ±1%	300	400	500	kHz
External Clock Duty		Duty		45	50	55	%
Oscillation Start Time * (Crystal Option)		t_{OSCf}	C_D = 10pF ±20%, R_S = 1kΩ max	–	–	150	ms
Oscillation Start Time * (Resistor Option)		t_{OSCR}	R = 100kΩ ±1%, Connect C_L = 0.5μF to V_{CH}	–	–	2	ms
Oscillation Start Time (32kHz) *		t_{OSC1}	C_G = 10pF ±20%, R_S = 20kΩ max	–	–	1	s
Internal Capacitance of Oscillator	EXTAL	C_D		–	10	–	pF
	XOUT			–	10	–	pF
Delay Time of Oscillation Delay Time *		t_{DLY}	Selected by mask option	0	–	1	s
Reset Delay Time		t_{RLH}	External Capacitance = 2.2μF	200	–	–	ms
$\overline{RES}$ Pulse Width*		t_{RWL}	With 32kHz OSC	48	–	–	μs
			Without 32kHz OSC	$1.5t_{cyc}$ + 1	–	–	μs
$\overline{INT}$ Pulse Width *		t_{IWL}	When OSC1 is not stopped by Halt	t_{cyc}+ 1	–	–	μs
			When OSC1 is stopped by Halt	32	–	–	μs
TIMER Pulse Width		t_{TWL}	In the case of counter	t_{cyc}+ 1	–	–	μs

* Depends on mask-option.

• PORT CHARACTERISTICS

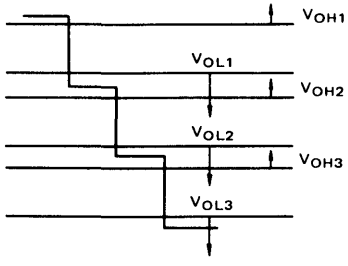
Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Level Voltage*	Port A, B, C	V_{OH}	CMOS Output, I_{OH} = -100μA	V_{CC}-0.3	–	–	V
	Port A, B, C		Key Load CMOS Output I_{OH} = -10μA	V_{CC}-0.3	–	–	V
Output "Low" Level Voltage	Port A, B, C	V_{OL}	I_{OL} = 100μA	–	–	0.3	V
Input "High" Level Voltage	Port A, B, C	V_{IH}		0.8V_{CC}	–	V_{CC}	V
Input "Low" Level Voltage	Port A, B, C	V_{IL}		V_{SS}	–	0.2V_{CC}	V
Input Leakage Current	Port A, B, C	$\|I_{IN}\|$	V_{in} = 0V ~ V_{CC}	–	–	1.0	μA
Input Pull-Up Current*	Port A, B, C	$-I_{R2}$	V_{CC} = 3.0V, V_{in} = 0V	4	20	40	μA

* Depends on mask-option.

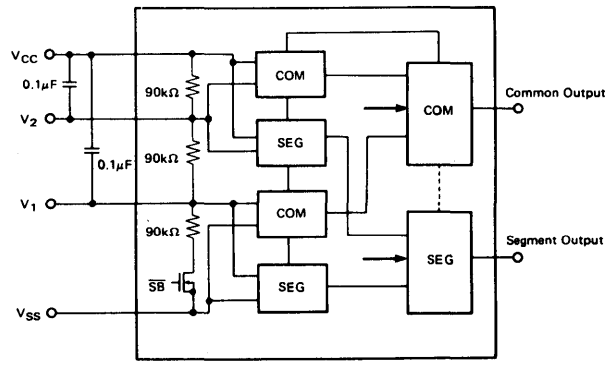
- **LCD DRIVER OUTPUT CHARACTERISTICS (V_{CC} = 3.0V, V_{SS} = 0V, T_a = -20 ~ +75°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Level Voltage	Segment	V_{OH1}	V_1 = 1.00V, V_2 = 2.00V I_{OH} = -1μA	2.8	–	–	V
		V_{OH2}		1.8	–	–	V
		V_{OH3}		0.8	–	–	V
Output "Low" Level Voltage	Segment	V_{OL1}	V_1 = 1.00V, V_2 = 2.00V I_{OL} = 1μA	–	–	2.2	V
		V_{OL2}		–	–	1.2	V
		V_{OL3}		–	–	0.2	V
Output "High" Level Voltage	Common	V_{OH1}	V_1 = 1.00V, V_2 = 2.00V I_{OH} = -5μA	2.8	–	–	V
		V_{OH2}		1.8	–	–	V
		V_{OH3}		0.8	–	–	V
Output "Low" Level Voltage	Common	V_{OL1}	V_1 = 1.00V, V_2 = 2.00V I_{OL} = 5μA	–	–	2.2	V
		V_{OL2}		–	–	1.2	V
		V_{OL3}		–	–	0.2	V
Dividing Resistor		R_{LCD}	Tested between V_1 and V_2	45	90	180	kΩ
Output "High" Level Voltage*	Segment	V_{OH}	In the case of Output Port, I_{OH} = -30μA	V_{CC}-0.3	–	–	V
Output "Low" Level Voltage*	Segment	V_{OL}	In the case of Output Port, I_{OL} = 30μA	–	–	0.3	V

* Depends on mask-option.



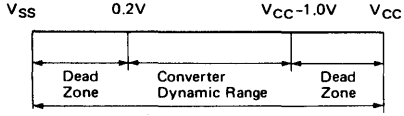
Output Level of SEG and COM



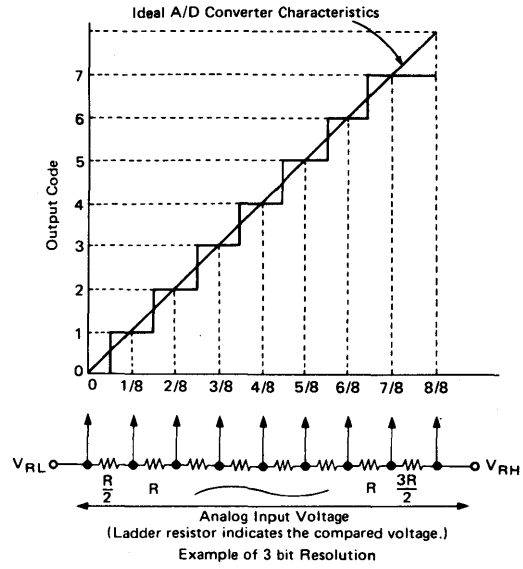
Power Supply Circuit for LCD Display

- **A/D CONVERTER CHARACTERISTICS** (V_{CC} = 3.0V, V_{SS} = 0V, T_a = -20°C ~ +75°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Conversion Accuracy	Resolution			–	–	8	bit
	Absolute Accuracy		$V_{RL} = 0.2V < V_{in} < V_{RH} = 2.0V$	-2	–	+2	LSB
Reference Voltage	"High" Side	V_{RH}		–	–	V_{CC}	V
	"Low" Side	V_{RL}		V_{SS}	–	–	V
	$V_{RH} - V_{RL}$	ΔV_{REF}		1.8	–	–	V
Input Voltage Range	Input Range	V_{IN}		V_{RL}	–	V_{RH}	V
	Input Dynamic Range	V_{DYN}		0.2	–	V_{CC}-1.0	V
Ladder Resistor ($V_{RH} - V_{RL}$)		R_{HL}		40	80	160	kΩ
Conversion Time		t_{CNV}		2	–	4	ms
Programmable Voltage Comparison	Judge Error		$V_{RL} = 0.2V < V_{in} < V_{RH} = 2.0V$	-4	–	+4	LSB
	Judge Time	t_{CMP}		–	–	60	μs



Dynamic Range of the Comparator



Example of 3 bit Resolution

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ SIGNALS

The input and output signals of the MCU are described in the following:

- **V_{CC}, V_{SS}**
  Power is applied to the MCU at these two terminals. V_{CC} is a positive power input port and V_{SS} is grounded.

- **$\overline{INT}$**
  This terminal is used to envoke an external interruption to the MCU. For details, see the information given under the title, "Interruptions" (⌐└ Negative going edge type).

- **XTAL, EXTAL**
  These are control input ports to the built-in clock circuit. A crystal or a resistor is connected to each of them depending on the degree of stability of the internal oscillation. For the method of using the input terminals, see the information, "Internal Oscillator Option"

- **XIN, XOUT**
  These terminals are connected to a crystal for the oscillator on the time base. A clock operation is possible by using a 32.768kHz crystal. For details, see "Internal Oscillator Option".

- **TIMER**
  An external input terminal at which the internal timer is counted down. For details, see the information, "Timer".

- **$\overline{RES}$**
  Used to reset the MCU. For details, see "Reset".

- **STANDBY (SB)**
  An external input terminal used to stop the MCU and hold data. For details, see "Internal Oscillator Option".

- **A/D Input Terminals ($CH_1 \sim CH_8$)**
  Input terminals for analog voltages needed for A/D conversion. These may also be used as level check inputs under program control. For details, see the information, "A/D Converter".

- **V_{RH}, V_{RL}**
  Reference voltages for A/D conversion are applied to these two terminals. For details, see "A/D Converter"

- **CC_1, CC_2**
  These are not intended for user applications. Open them.

- **NUM**
  This is not intended for user applications. Connect it to V_{CC}.

- **Input/Output Terminals ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_3$)**
  Each of these 20 terminals consists of two 8 bits ports and one 4 bits ports. It may be used as an input or output under program control of the data direction register. For details, see "Input/Output".

- **Liquid Crystal Driver Terminals ($COM_1 \sim COM_3$, $SEG_1 \sim SEG_{17}$)**
  $COM_1 \sim COM_3$ are for driving common electrodes, while $SEG_1 \sim SEG_{17}$ are for driving segments. $SEG_1 \sim SEG_{17}$ can be used as outputs by mask-option and $SEG_{13} \sim SEG_{17}$ can be used as analog inputs for A/D converter by mask-option.

Mixing segment driver with output port is not available in mask-option.

- **V_1, V_2**
  These are terminals for LCD driver. V_1 and V_2 are connected to V_{CC} via capacitors (0.1μF each). These two terminals can be used as output or analog inputs by mask-option when segments are used as output ports.

- **V_{CH}**
  Output terminal from internal voltage regulator. A capacitor (0.5μF) is connected between V_{CH} and V_{CC}. Don't draw current from this terminal.

- **E**
  System clock output (cycle clock 100kHz typ.)
  This NMOS open-drain output stays at "Low" level when the MCU is in halt mode, standby mode or reset.

■ MEMORY

The MCU memory is configured as shown in Figure 1. During the processing of an interrupt, the contents of the MCU resisters are pushed onto the stack in the order shown in Figure 2. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order four bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will cause only the program counter (PCH, PCL) contents to be pushed onto the stack.

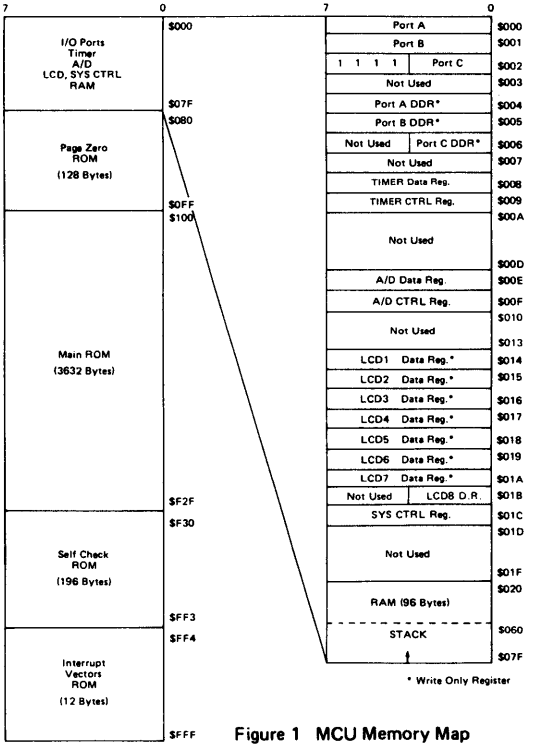


Figure 1 MCU Memory Map

(Cautions)

It is not possible to change the contents of the Write Only Register (For example, the Data Direction Register of the I/O port) of the HD63L05F1 by applying the Read/Modify/Write instructions, BSET, or BCLR.

For preventing the system from wild running, don't read the Not Used area of the memory map.

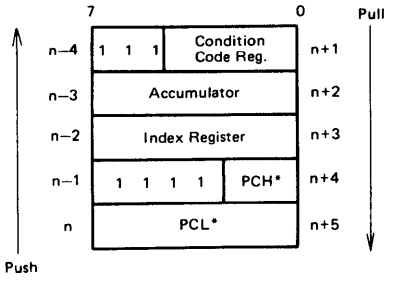


Figure 2 Interruption Stack Sequence

■ REGISTER

The CPU has five registers that can be operated by the programmer. They are shown in Figure 3.

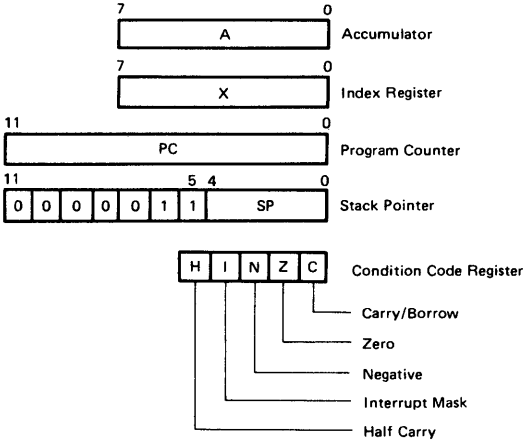


Figure 3 Programming Model

• Accumulator (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

• Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage register.

• Program Counter (PC)

The program counter is a 12-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 12-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $07F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The most significant bits of the stack pointer are permanently set to 0000011. During a MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $07F. Subroutines and interrupts may be nested down to location $061 which allow the programmer to use 15 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bit 3 and bit 4.

Interrupt (I)

This bit is set to mask the internal interrupts and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set, it is latched and will be processed as soon as the interrupt bit is reset.

(Note)

CLI (clear interrupt mask bit) is used to allow the interruption from the instruction after next. SEI (set interrupt mask bit) masks the interruption from next instruction.

Negative (N)

Used to indicate that the result of the last arithmetic, logical of data manipulation was negative (bit 7 in result equal to logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical of data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instruction, shifts, and rotates.

■ SYSTEM CONTROL REGISTER

Apart from the registers for program operation explained above, there is a register that controls system operation. Its configuration is shown in Figure 4.

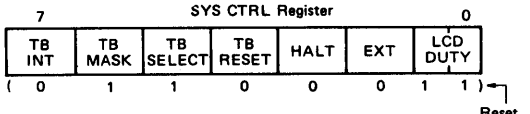


Figure 4 System Control Register Configuration

A Time Base counter is built in the MCU to generate two kinds of time base interrupts (1 second cycle and 1/16 second cycle). Clock signal to this counter is provided from the OSC1 or OSC2 depending on the mask-option. This counter is a frequency divider behind the 32 kHz oscillator.

- **Time Base Interrupt Request Flag (TB INT)**

Stores an interrupt request from the time base which is selected by the TB select bit and is cleared by system reset or by software. If the TB MASK bit or I (Interrupt bit in the CCR) is set, the interrupt request is not acknowledged. Only logical "0" can be written into this bit by software.

- **Time Base Interrupt Mask (TB MASK)**

If this bit is set, an interrupt request from the time base is not acknowledged.

- **Time Base Select Bit (TB SELECT)**

This bit selects the time base. In logical "1", an interrupt from the 1-second cycle time base is acknowledged. In logical "0", 1/16 second cycle time base is acknowledged.

- **Time Base Reset Bit (TB RESET)**

This bit resets the frequency divider behind the 32kHz oscillator. When this bit is set, one shot reset pulse is generated by the hardware. Then it resets the frequency divider and after that, the frequency divider restarts. The CPU always reads this bit as logical "0"

Since the frequency divider also provides the system clocks to the A/D converter and LCD drivers etc., writing "1" to "TB RESET" bit during execution of A/D converter and TIMER (when ϕ_{32k} is selected) causes different data from the correct result and writing "1" to this bit causes flicker of the LCD display.

- **Halt (HALT)**

Used to halt the CPU. When this bit is set, the registers are saved onto the stack in the same sequence as interrupt processing. After all registers have been saved, the CPU halts and is wait-for-interrupt state.

If this bit is reset by an external interrupt or an internal interrupt, the CPU restarts operating. By using the Halt function with Time Base Interrupt, the CPU can operate intermittently itself.

- **EXT**

When the form of output port is selected by DUTY selecting bit and the mask-option, ϕWRITE is available at the specified terminal (SEG1 to O19) according to the designation of pin location. ϕWRITE clock can be got on every writing data into LCD register 1 and be used as the write clock in the case of transferring data of LCD register 1 to the outside. Normally, EXT must be reset.

- **Duty Select Bit (LCD DUTY)**

The LCD drive signal is based on 1/3 bias – 1/3 duty. However, there are switching circuits built in for static drive signal and output ports. For details, see the information given in "LCD Circuit".

(Note)

The EXT bit and the LCD DUTY bits have to be initialized in 1 milli second from the start of CPU operation when the static drive signal or output port is selected.

■ TIMER

The MCU timer circuitry is shown in Figure 5. The 8-bit counter is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero the timer interrupt request bit (bit 7) in the timer control register is set. The CPU responds to this interrupt by saving the present CPU state in the stack, fetching the timer interrupt vector from locations $FF8 and $FF9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The interrupt bit (I bit) in the condition code register will also prevent a timer interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input terminal (active at negative edge) or it can be the internal signal (ϕ_2 or ϕ_{32k}). When the internal clock signal is used as the source, the clock input is gated by the input applied to the TIMER input terminal; this permits easy measurement of its pulse width. ϕ_2 is provided from OSC1 and the frequency is 1/4 of OSC1. ϕ_{32k} is provided from OSC1 (the frequency is 1/12 of OSC1) or OSC2 (32.768 kHz crystal) depending on the mask-option. If the OSC1 continues to oscillate during the halt mode, 32.768 kHz crystal is selected as the clock source or external clock is applied, the timer can be active in the halt mode. Note that the timer operation is asynchronous to the CPU when the mask-option which the OSC1 stops oscillating in the halt mode is selected.

A 7-bit prescaler is provided to extend the timing interval up to a maximum of 128 counts before being applied to the timer. The number of prescaling counts can be program controlled by the lower 3 bits within the TIMER CTRL register. The timer continues to count past zero and its present count can be monitored at any time by monitoring the TIMER Data register. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At the time of resetting, the prescaler and the counter are all initialized to logical "1". The timer interrupt request bit is cleared and the timer interrupt mask bit is set. The timer interrupt request bit (bit 7 of TIMER CTRL Register) is set to logical "1" when timer count reaches zero, and is cleared by program or by system reset. Only logical "0" can be written into this bit by program. The bit 6 of Timer Control Register is writable by program. Both of these bits can be read by CPU.

■ RESETS

The MCU can be reset either by initial power-up or by the external reset input ($\overline{RES}$). All the I/O ports are initialized to Input mode (DDRs are cleared) during reset.

Upon power-up, a minimum of 150 milliseconds is needed before allowing the reset input to go "High". This time allows the internal oscillator (OSC1) to stabilize. Connecting a capacitor to the $\overline{RES}$ input as shown in Figure 8 will provide sufficient delay.

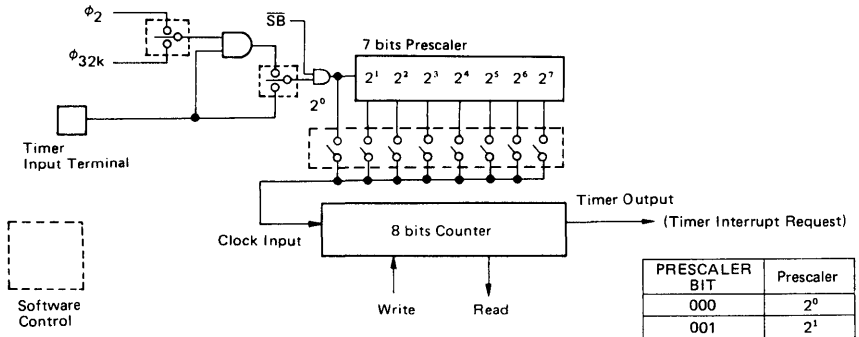


Figure 5 Timer Block Diagram

PRESCALER BIT	Prescaler
000	2^0
001	2^1
010	2^2
011	2^3
100	2^4
101	2^5
110	2^6
111	2^7

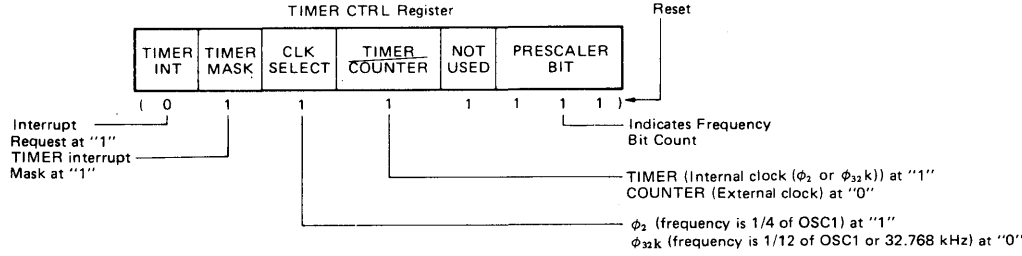


Figure 6 Timer Control Register Configuration

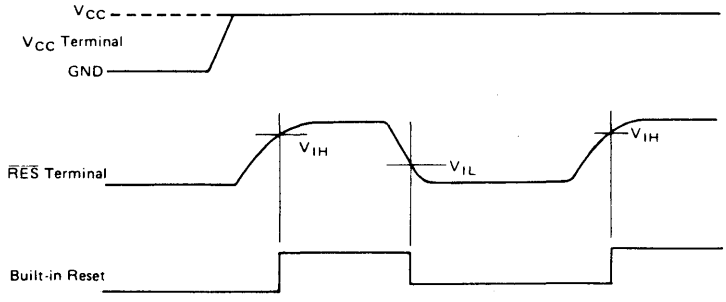


Figure 7 Application of Power and Reset Timing

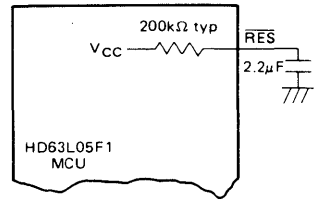
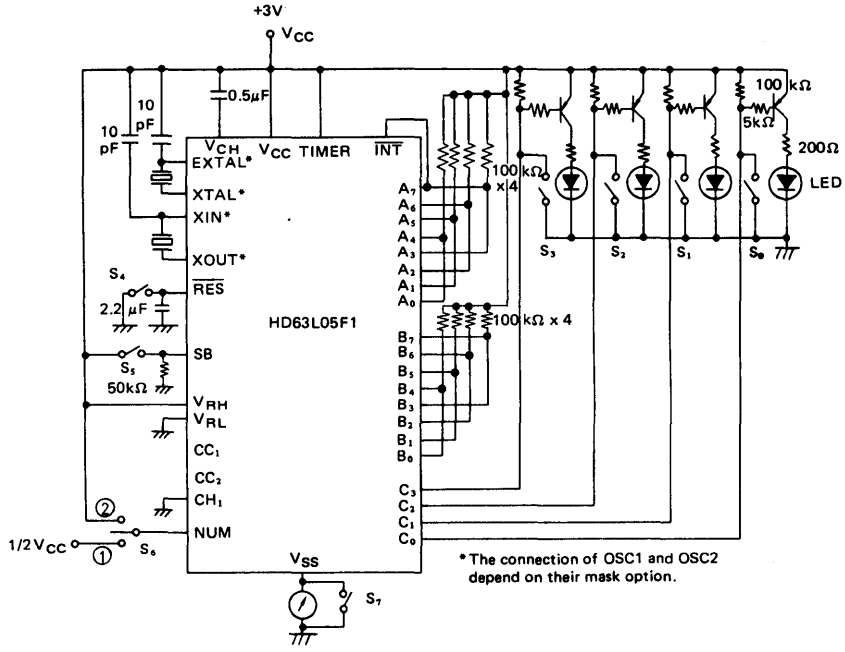


Figure 8 Input Reset Delay Circuit

■ SELF CHECK

The self check capability of the MCU provides an internal check to determine if the port is functional. Connect the MCU as shown in Figure 9 and monitor the output of port C bit 3 for an oscillation of approximately 0.5Hz. This self check capability also provides the internal state of the MCU to measure the LSI current. After a system reset, the MCU goes into each current measurement mode by the combination of the control switches. The LSI current can be measured when the NUM is returned to V_{CC} after setting of the current mode.



		Selection of Switch							
		S_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7
LSI Function		X	X	X	X	X	X	①	O
LSI Current	During operation	O	X	X	X	O→X	X	①→②	X
	Halt	O	O	O	X	O→X	X	①→②	X
	A/D	O	O	X	X	O→X	X	①→②	X
	Standby	O	O	O	X	O→X	X→O	①→②	X

X : OFF O : ON →: Change the state

Figure 9 Self Check Connections

■ INTERNAL OSCILLATOR OPTIONS

The MCU incorporates two oscillators: Oscillator 1 for system clock supply and Oscillator 2 for peripheral modules such as time base, A/D converter, LCD drivers, etc..

- **Oscillator 1 (OSC1; XTAL, EXTAL)**

The internal oscillator circuit can be driven by an external crystal or resistor depending on the stability. A manufacturing mask option is available to provide better matching between the external components and the internal oscillator. The oscillator 1 can stop when power is applied in Standby mode. Figure 10 shows the connection. A resistor selection graph is given in Figure 11.

- **Oscillator 2 (OSC2; XIN, XOUT)**

Clocks for time base, LCD drivers, an A/D converter and a timer are supplied by the OSC2 (32.768kHz crystal). In Halt mode, OSC2 and frequency divider operate and permit the operation of the peripheral modules with low power consumption. In Standby mode, the frequency divider is in reset state but only OSC2 keeps on running to control the delay time required when the MCU is released from Standby mode. Figure 12 shows the connection and the relation between oscillator 1 and oscillator 2 is shown Figure 13 and Table 1.

When OSC2 is not available in an user system, clocks for time base, LCD drivers, a A/D converter and a timer are supplied by the OSC1 through frequency divider. When OSC2 is not available or crystal option is selected for OSC1, OSC1 can not be stopped in Halt mode.

Only when CR option is selected for OSC1 and OSC2 is available in an user system, OSC1 can be stopped in Halt mode. (Note)

The accuracy of the time base (1 sec, 1/16 sec) is kept only when OSC2 is 32.768kHz crystal oscillator.

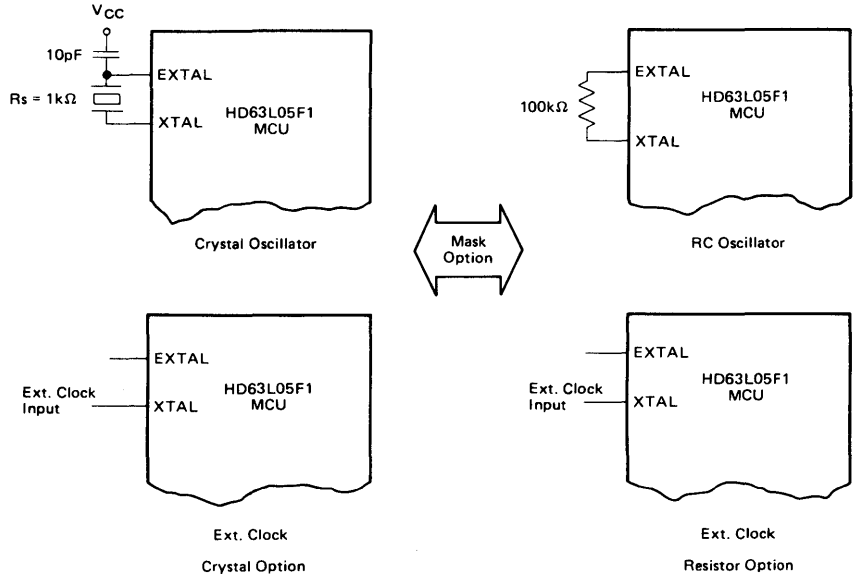


Figure 10 Mask Option for Oscillator 1

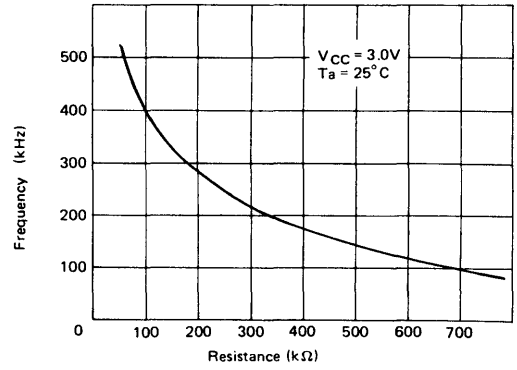


Figure 11 Typical Resistor Selection Graph

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

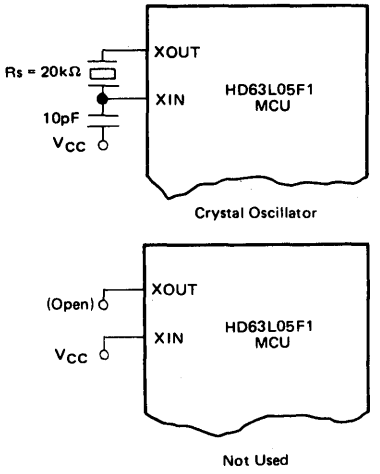


Figure 12 Connection of Oscillator 2

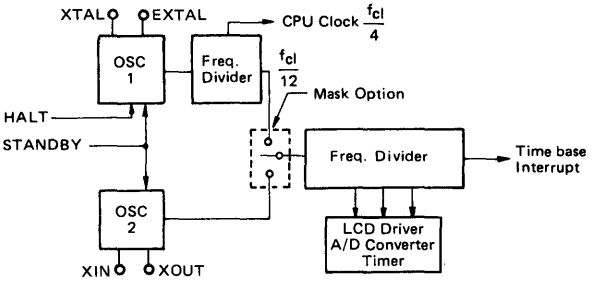


Figure 13 Relation between Oscillator 1 and Oscillator 2

Table 1 Oscillator 2 Mask-option and System Operation

Mask Option	When OSC1 is Crystal						When OSC1 is RC					
	OSC2 Not Available			OSC2 Available			OSC2 Not Available			OSC2 Available		
System / State	OSC1	CPU	Peripheral	OSC1	CPU	Peripheral	OSC1	CPU	Peripheral	OSC1	CPU	Peripheral
During System Operation	O	O	O	O	O	O	O	O	O	O	O	O
At Halt	O	X	O	O	X	O	O	X	O	O or X (mask option)	X	O
At Standby	X	X	X	X	X	X	X	X	X	X	X	X

(NOTE) O . . . . . run X . . . . . stop

Table 2 Mask-options of Oscillation Circuits and the Delay Time

Type of OSC1	Use of OSC2	Condition		Delay Time of Restart (second)			
				0	1/16	1/2	1
Crystal Option	Used	Standby mode	Used	X	X	O	O
			Not used	O	O	O	O
	Not used	Standby mode	Used	X	X	O	O
			Not used	O	O	O	O
CR Option	Used	Oscillation of OSC1 at HALT	Stop	O	X	X	X
			Continue	O	O	O	O
	Not used	Oscillation of OSC1 at HALT	Stop	X	X	X	X
			Continue	O	O	O	O

Note) Combinations of the mask-option indicated X is not available.

■ **STANDBY**

When the STANDBY (SB) terminal becomes "High" level, the MCU goes into standby mode at its instruction fetch cycle. On standby mode, only 32 kHz oscillator (OSC2) keeps on running while the others are stopped with holding the current data except A/D converter, timer, and time base. Restarting of the MCU from standby mode is controlled by the Delay Time which is available by counting the OSC2 oscillation or 1/12 frequency of the OSC1 in frequency divider after the STAND-BY terminal turned to "Low" level. Therefore, the CPU restarts operation from the previous state after the Delay Time (0 sec, 1/16 sec, 1/2 sec, or 1 sec), and the accuracy of the Delay Time

is kept when OSC2 is 32.768 kHz crystal oscillator. When 1/12 frequency of OSC1 is provided to the frequency divider, the Delay Time depends on the stability of OSC1 after restarting from standby mode and is not acculate.

■ **Delay Time**

Since OSC1 stops in standby mode, it is needed to inhibit restarting of CPU untill the OSC1 oscillation is stabilized after the STANDBY terminal has turned to "Low" level. To take this stabilizing time of OSC1, user can select the Delay Time out of 0 sec, 1/16 sec, 1/2 sec or 1 sec by mask-option depending on a combination in the Table 2. STANDBY terminal has to be kept at "Low" when resetting the MCU and has to be kept at "Low" during the Delay Time. Starting of the MCU by reset is also controlled by the Delay Time.

■ **INTERRUPTS**

There are six different interrupts to the MCU: **external** interrupt via external interrupt terminal ($\overline{INT}$), internal timer interrupt, interrupt by termination of A/D conversion, time base interrupt, and software interrupt by an instruction **(SWI)**.

When any interrupt occurs, processing is suspended, the present MCU state is pushed onto the stack, the interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. The interrupt service routines normally end with a return from interrupt instruction (RTI) which allows the MCU to resume processing of the program prior to the interrupt. Table 3 provides a listing of the interrupts, their priority, and the vector address that contains the starting address of the appropriate interrupt routine.

Figure 14 shows the system operation flow, in which the portion surrounded with dot-dash lined contains interruption execution sequence.

(Note)

A clear interrupt bit instruction (CLI) allows to suspend the processing of the program by an interruption after execution of the next instruction while a set interrupt bit instruction (SEI) inhibits any interrupts before execution of the next instruction. When a mask bit of a control register is cleared by an instruction, interruption is allowed before execution of the next instruction.

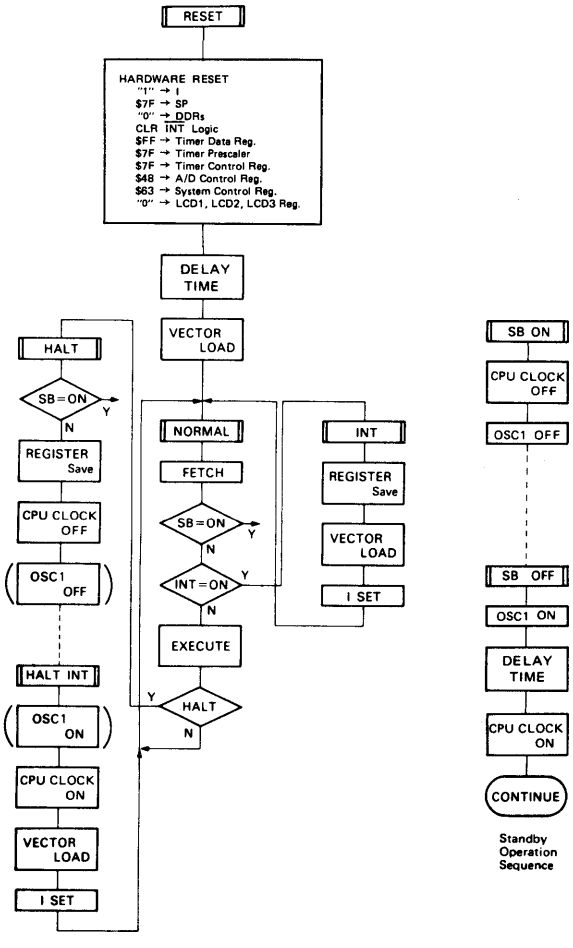


Figure 14 System Operation Flowchart

Table 3 Interruption Priority

Interruption	Priority	Vector Address
$\overline{RES}$	1	$FFE, $FFF
SWI	2	$FFC, $FFD
$\overline{INT}$	3	$FFA, $FFB
TIMER	4	$FF8, $FF9
A/D	5	$FF6, $FF7
TIME BASE	6	$FF4, $FF5

- **Acknowledging an INT in Halt mode**

In HALT mode, the CPU is not operating but the peripherals are operating. When an interruption is acknowledged, the CPU is activated and executes interruption service matching the interruption condition by means of vectoring.

- **Acknowledging an INT in Standby mode**

In Standby mode, the system is not operating with power supplied to it, therefore, any interruption request (including $\overline{RES}$) is not acknowledged.

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ INPUT/OUTPUT

There are 20 input/output terminals, which are program controlled by data direction registers for use as either input or output. If an I/O port has been programmed as an output and is read, then the latched logical level data is read even though the output level changes due to the output load.

If a port is to be used as an input terminal, the user must specify whether or not it will be equipped with a pull-up PMOS. Figure 15 shows the port I/O circuit.

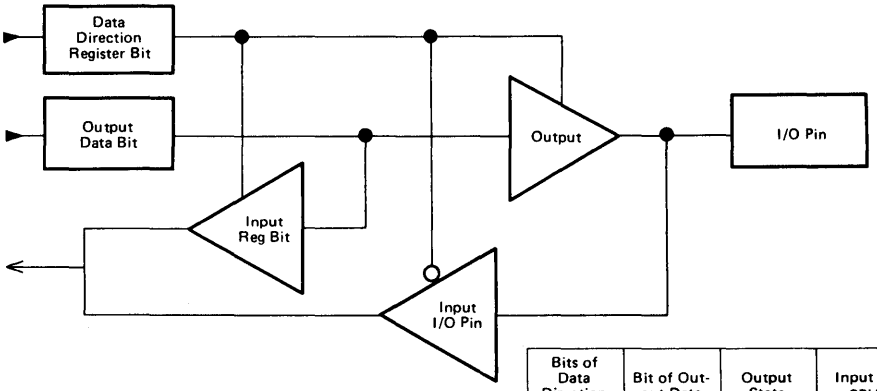


Figure 15 Port I/O Circuit

Bits of Data Direction Register	Bit of Output Data	Output State	Input to CPU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

• Configuration of Port

Figure 16 shows the configuration of I/O ports. As the output is on/off controlled by a data direction register, an I/O port may directly be applied as an input terminal. No problem is involved with the input if both "High" and "Low" levels are applied. For only one level, the user must specify the use of a pull-up PMOS for "Open/Low" input application.

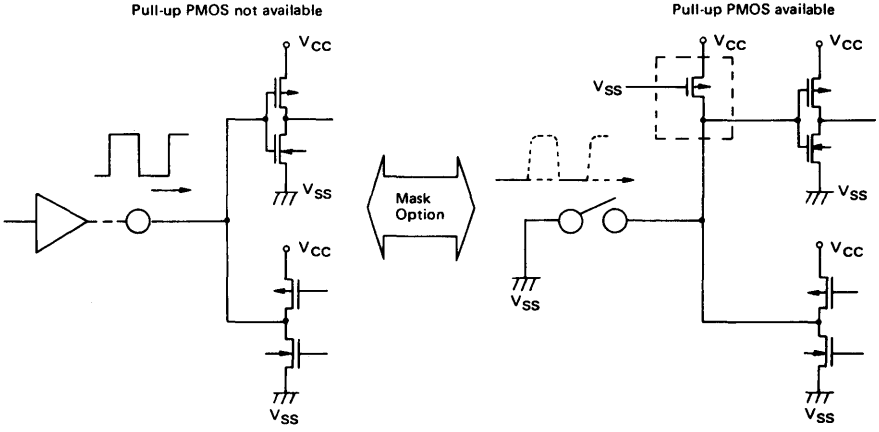


Figure 16 Selection of Input Configuration for I/O Port

■ A/D CONVERTER

The MCU incorporates an 8 bits A/D converter based on the resistor ladder system. Figure 17 shows its block diagram.

The "High" side of reference voltage is applied to V_{RH}, while the "Low" side of reference voltage is applied to V_{RL}. The reference voltage is divided by resistors into voltages matching each bit, which is compared with analog input voltage for A/D conversion. As the analog input voltage is applied to the MOS gate of the comparator through the analog multiplexer, this voltage comparison system achieves high input impedance.

The A/D Data Register stores the results of an A/D conversion or can be set 8 bit data for programmed comparator. These functions are controlled by software-controlled A/D CTRL Register. The result of A/D conversion is not assured if the conversion is interrupted by STANDBY. Figure 18 shows the configuration of the A/D control register.

• A/D Interrupt Request Flag (A/D INT)

The A/D INT bit is set to logical "1" after completion of A/D conversion and is cleared by program or by system reset. Only logical "0" can be written into this bit by software.

• A/D Interrupt Mask (A/D MASK)

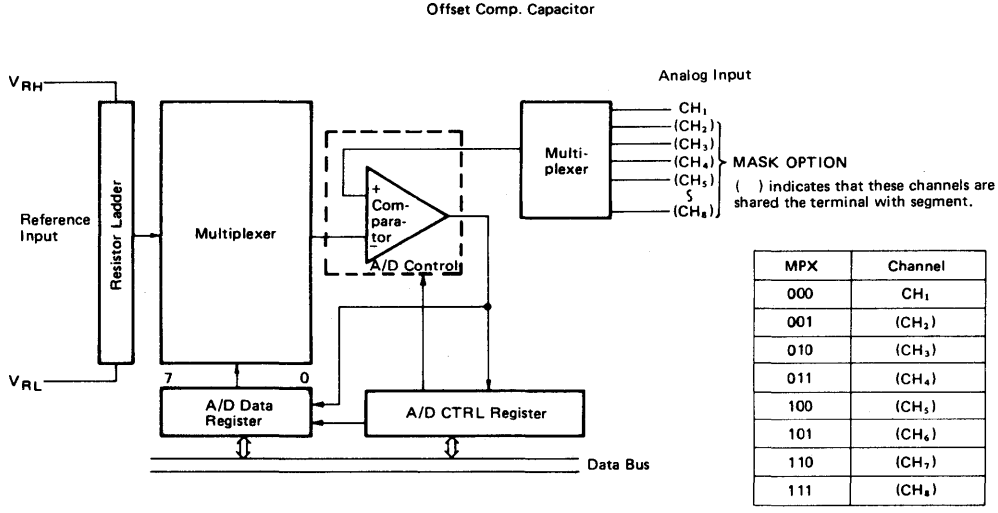
If this bit is set, interrupt from the A/D converter is not acknowledged. This bit can be written by program.

• A/D Conversion Flag (CNV)

To start auto A/D conversion, set this bit to logical "1". During conversion, data of this bit stays at "1". The bit is automatically reset to "0" when the auto A/D conversion ends. In auto A/D conversion, supply voltage is applied to the comparator only when CNV = "1". The digital data which is obtained by the A/D conversion is held in the A/D Data Register. This data is reset when the CNV is set to "1" again.

• A/D Operation Mode Select Bit (Auto/Program)

Used to select either auto-run 8 bits A/D conversion or 8 bit programmed comparator operation (Auto 8 bits A/D conversion at "0").



MPX	Channel
000	CH_1
001	(CH_2)
010	(CH_3)
011	(CH_4)
100	(CH_5)
101	(CH_6)
110	(CH_7)
111	(CH_8)

Figure 17 8 Bits A/D Converter Block Diagram

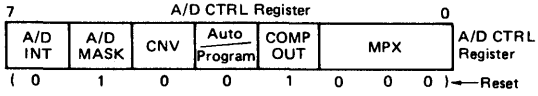


Figure 18 A/D Control Register Configuration

- **Comparator Output (COMP OUT)**

The result of comparator operation under program control can be read from this bit (Logical "1" means that input voltage is higher than programmed reference voltage).

- **Analog Input Channel Select Bits (MPX)**

Used to select 8-channel analog inputs. The multiplexer is an analog switch based on CMOS. Note that the analog inputs from CH_2 to CH_8 are mask option while CH_1 is exclusive.

When 1/3 bias – 1/3 duty or static LCD is used, CH_7 and CH_8 are not available because these two terminals are used for LCD power supply as V_1 and V_2.

■ LCD CIRCUIT

The system configuration of the LCD circuits is shown in Figure 19. Segment data for display are stored in data registers LCD1 to LCD8. Since the circuits are connected to the output terminals via pin location block, the user may specify a combination of data to be multiplexed to the segment output terminals.

The bit data of the LCD register is combined with the timing clock (ϕ_1, ϕ_2 or ϕ_3) and three combined bit data are gathered to make a segment output data for 1/3 bias – 1/3 duty driving in the pin location block. In case of static LCD drive or output port, timing is always fixed at ϕ_1 (always "High") and one bit data of the LCD register is transferred for an output terminal.

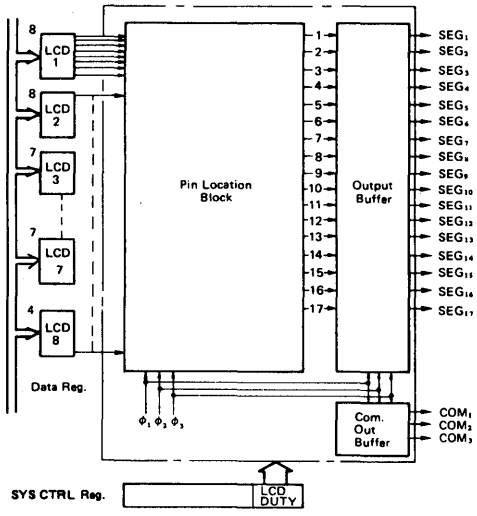
Note that the output terminals from SEG_{13} to SEG_{17} are mask option while the others (SEG_1 to SEG_{12}) are always available when the Duty bits are "01" or "11".

When the form of output port is selected by Duty bit ("00"), ϕWRITE can be got every time data is written into LCD1 register in the case that EXT bit is "1". As LCD1 register has 8 bits latches, it is easy to transfer the internal 8 bits data to external devices via output ports, with automatically generated write clock ϕWRITE. The cycle clock pulse can be also available as an internal data source for the output terminal when output port is selected as 1/4 OSC1.

Assignment of segment terminals to the bits of the LCD data register, including the case where they are used as output terminals, is to be specified by the user when he orders masks. In case of static LCD or output ports, only LCD1, LCD2, and LCD3 are allowed to be used. These registers are initialized at "0" by system resetting.

■ LIQUID CRYSTAL DRIVER WAVEFORMS

The LCD circuit is based on 1/3 bias – 1/3 duty driving. Figure 20 shows the common electrode output signal waveforms (COM_1, COM_2, COM_3), segment signal waveforms (SEG_1 to SEG_{17}) and LCD bias waveforms (between COM and SEGMENT).



System Cont. Duty	Contents of $SEG_1 \sim SEG_{17}$
00	OUTPUT PORT
01	STATIC LCD
10	———
11	1/3 Bias 1/3 Duty LCD

Note)
Both of mask-option and software control are needed to specify the contents of $SEG_1 \sim SEG_{17}$.

Figure 19 LCD Circuit System Configuration

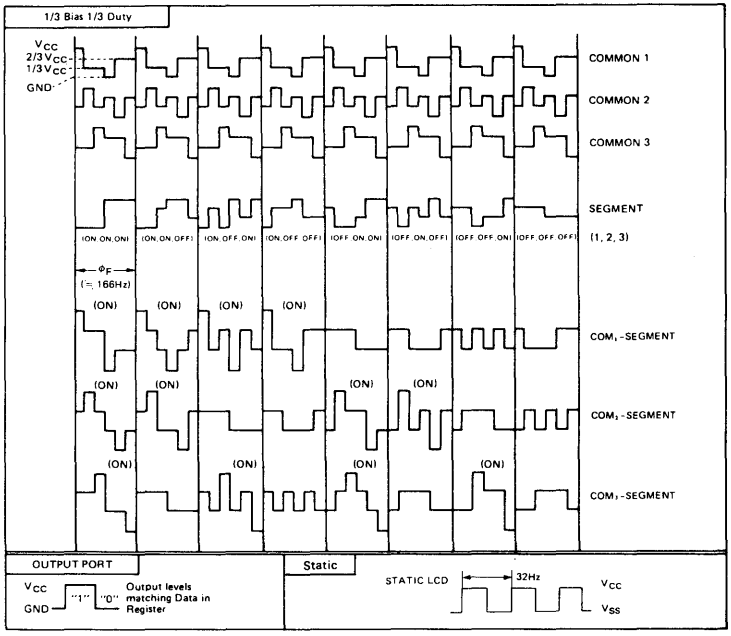


Figure 20 LCD Driving Waveforms

■ BIT MANIPULATION

The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines.

(Note)

It is needed to pay attention to the system control register, the timer control register, and A/D control register when BSET, BCLR, or Read/Modify/Write instructions are applied to them. If own interrupt request occured onto the interrupt request bit (bit 7) of the control register between read cycle and write cycle of these instructions, the bit 7 might be cleared in the write cycle and not acknowledged by CPU.

■ ADDRESSING MODE

There are 10 addressing modes available to the MCU for programming. Familiarize yourself with these modes by reading the information and referring to the diagrams that follow.

• Immediate

See Figure 21. In immediate addressing mode, constants that will not change during execution of a program are accessed. The instruction used for that purpose has a length of 2 bytes. The effective address (EA) is PC. The operand is fetched from the byte that follows the OP code.

• Direct

See Figure 22. In direct addressing mode, the address of the operand is contained in the second byte of the instruction. The user can gain direct access to the LSB 256 of memory. All RAM bytes, I/O registers, and 128 bytes of ROM are located on page 0 in order to utilize this useful addressing mode.

• Extended

See Figure 23. The extended addressing mode is used for referencing to all addresses of memory. The EA consists of the contents of the two bytes that follow the OP code. The instruction used for extended addressing has a length of 3 bytes.

• Relative

See Figure 24. Only Branch instructions are used in relative addressing mode. When a branching takes place, the contents of the byte next to the OP code are added to the program counter. EA = (PC) + 2 + Rel., where Rel. indicates signed 8 bits data at the address following the OP code. When no branching takes place, Rel. = 0. When a branching occurs, the program jumps to any byte of +129 to -127 of the current instruction. The length of the Branch instruction is 2 bytes.

- **Indexed (without Offset)**

See Figure 25. In this addressing mode, the lower 256 bytes of memory are accessed. The length of the instruction used for this mode is one byte. The EA consists of the contents of the index register.

- **Indexed (8 Bits Offset)**

See Figure 26. The EA consists of the contents of the byte following the OP code, and the contents of the index register. In this mode, the lower addresses of memory up to 511 can be accessed. Two bytes are required for the instruction.

- **Indexed (16 Bits Offset)**

See Figure 27. The EA consists of the contents of the two bytes following the OP code, and the contents of the index register. In this mode, the whole of the memory can be accessed. The instruction using this addressing mode has a length of 3 bytes.

- **Bit Set/Clear**

See Figure 28. This addressing mode can be applied to any instruction that permits any bit on page 0 to be set or cleared. The byte following the OP code indicates an address within page 0.

- **Bit Test, Branch**

See Figure 29. This addressing mode can be applied to instructions that test bits at the first 256 addresses ($00 to $FF) and are branched by relative qualification. The byte to be tested is addressed by the contents of the address next to the OP code. The individual bits of the byte to be tested are designated by the lower 3 bits of the OP code. The third byte indicates a relative value that is to be added to the program counter when a branch condition is satisfied. The instruction has a length of 3 bytes. The value of the bit that has been tested is written at the carry bit of the condition code register.

- **Implied**

See Figure 30. There is no EA for this mode. All information needed for execution of instructions is contained in the OP code. Operations that are carried out directly on the accumulator and index register are included in the implied addressing mode. In addition, the SWI and RTI instructions are also included in the group of this operation. The instruction using this addressing has a length of one byte.

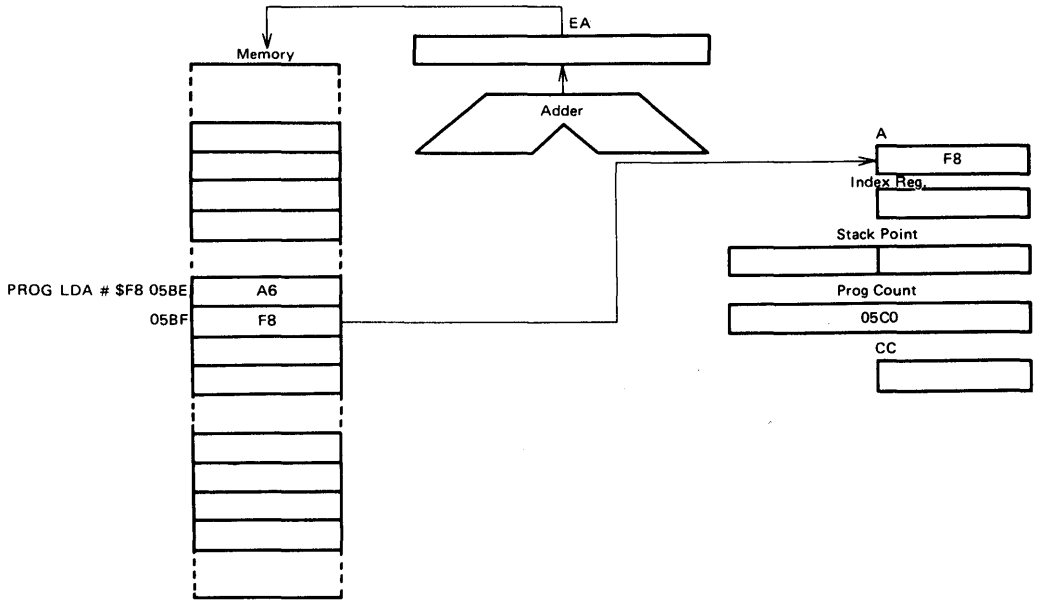


Figure 21 Example of Immediate Addressing

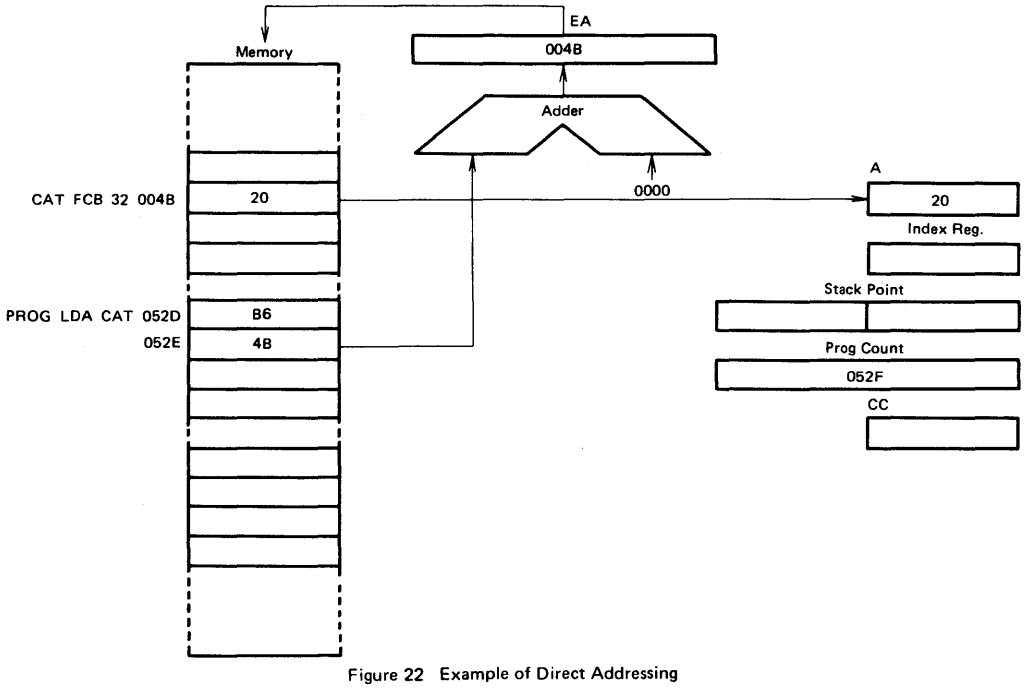


Figure 22 Example of Direct Addressing

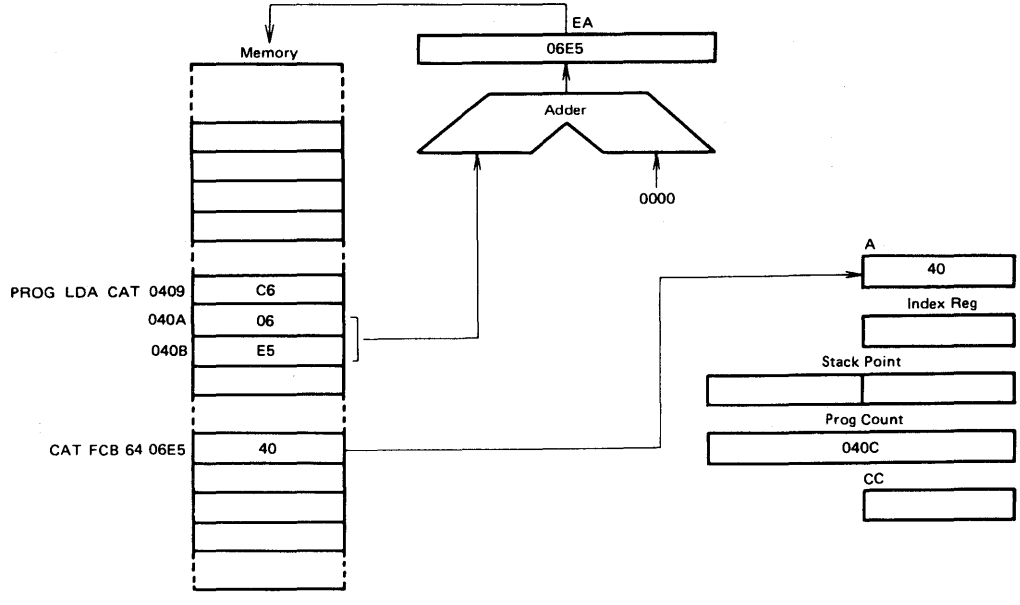


Figure 23 Example of Extended Addressing

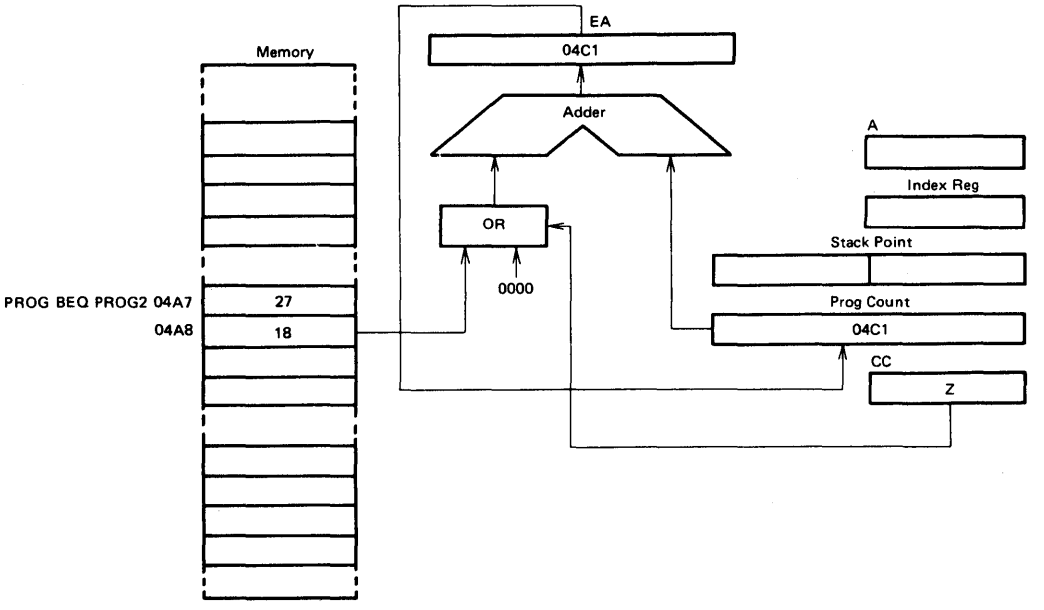


Figure 24 Example of Relative Addressing

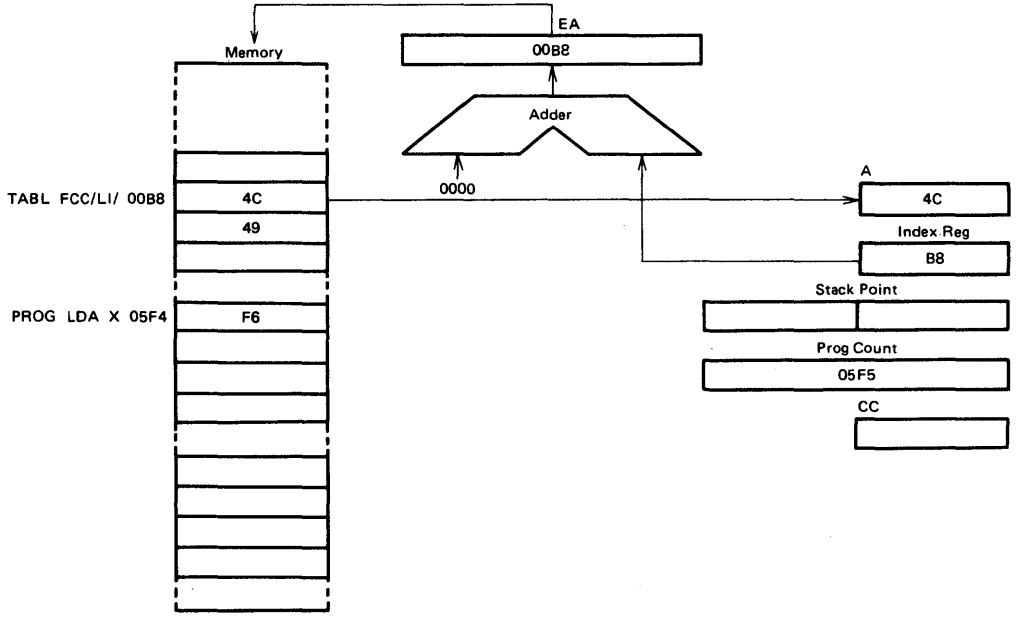


Figure 25 Example of Indexed (without Offset) Addressing

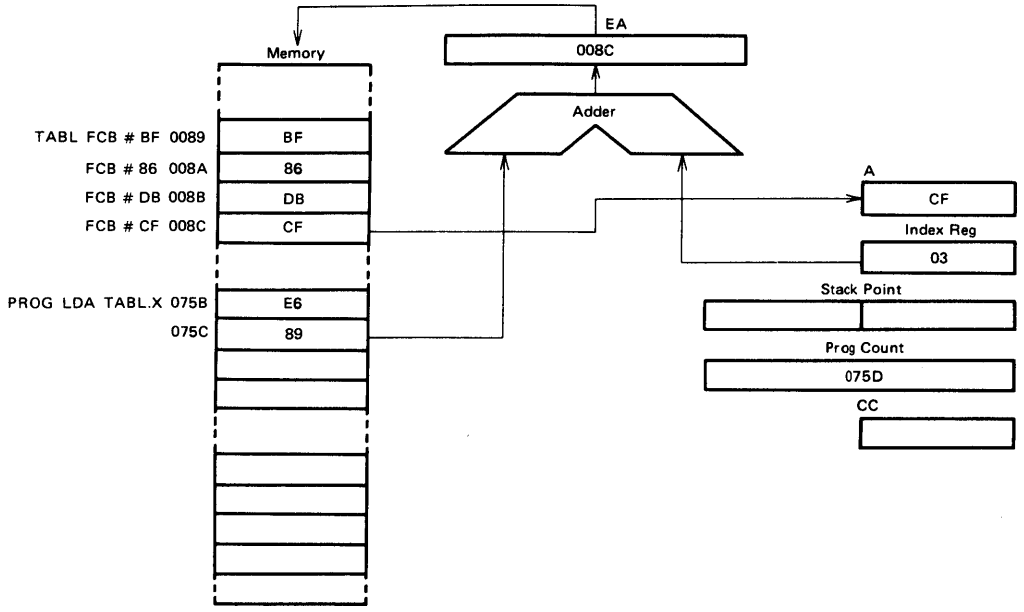


Figure 26 Example of Indexed (8 Bits Offset) Addressing

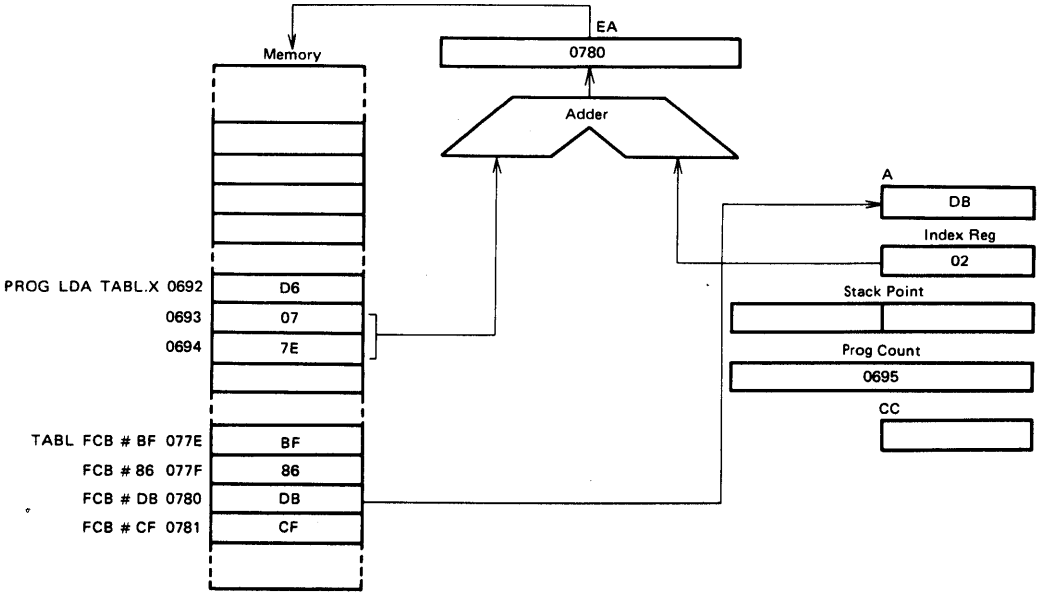


Figure 27 Example of Indexed (16 Bits Offset) Addressing

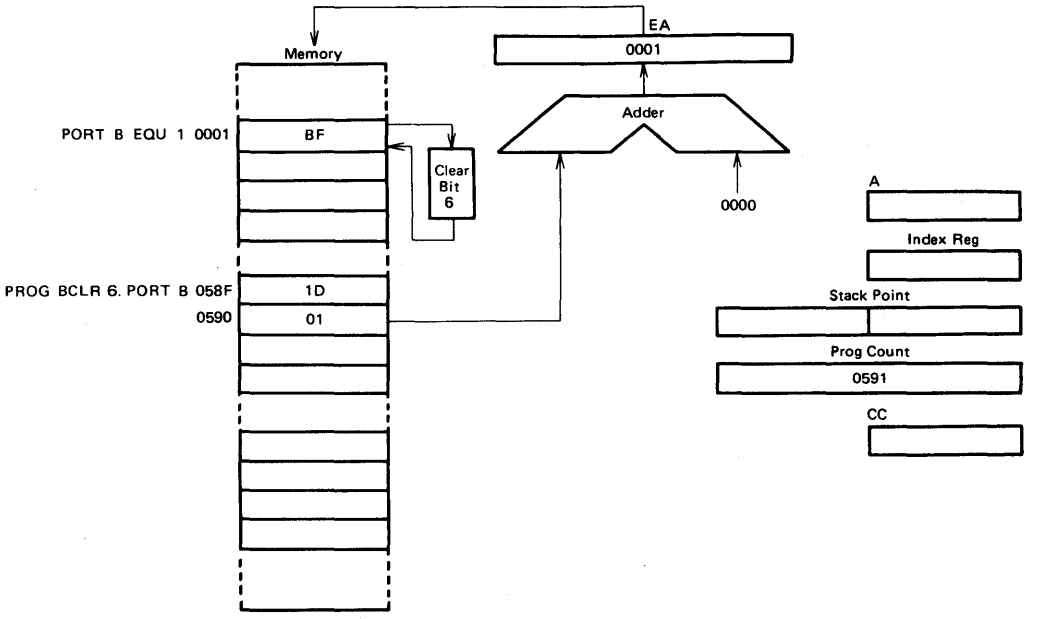


Figure 28 Example of Bit Set/Clear Addressing

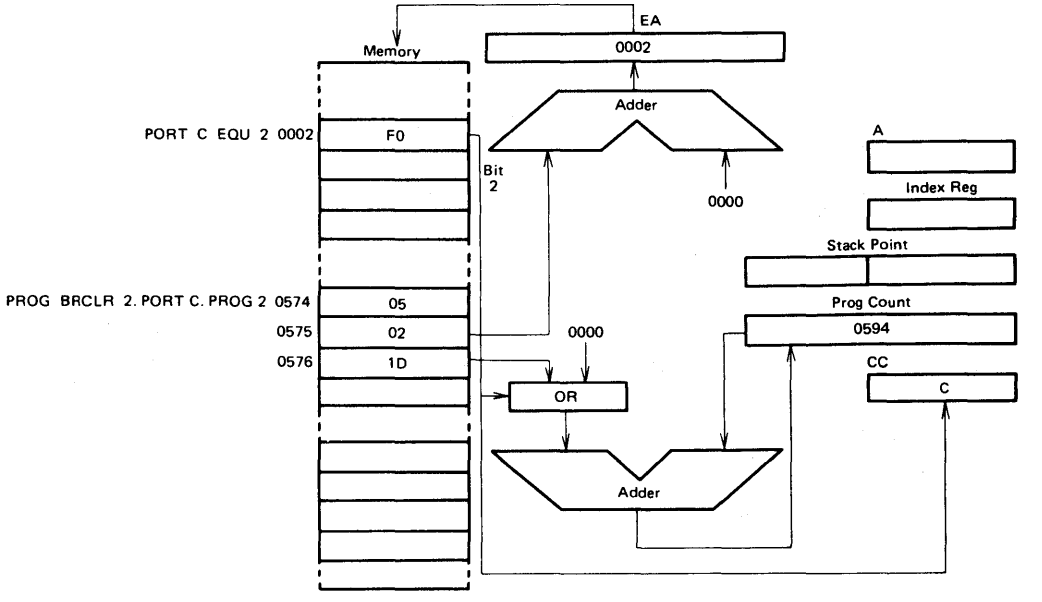


Figure 29 Example of Bit Test and Branch Addressing

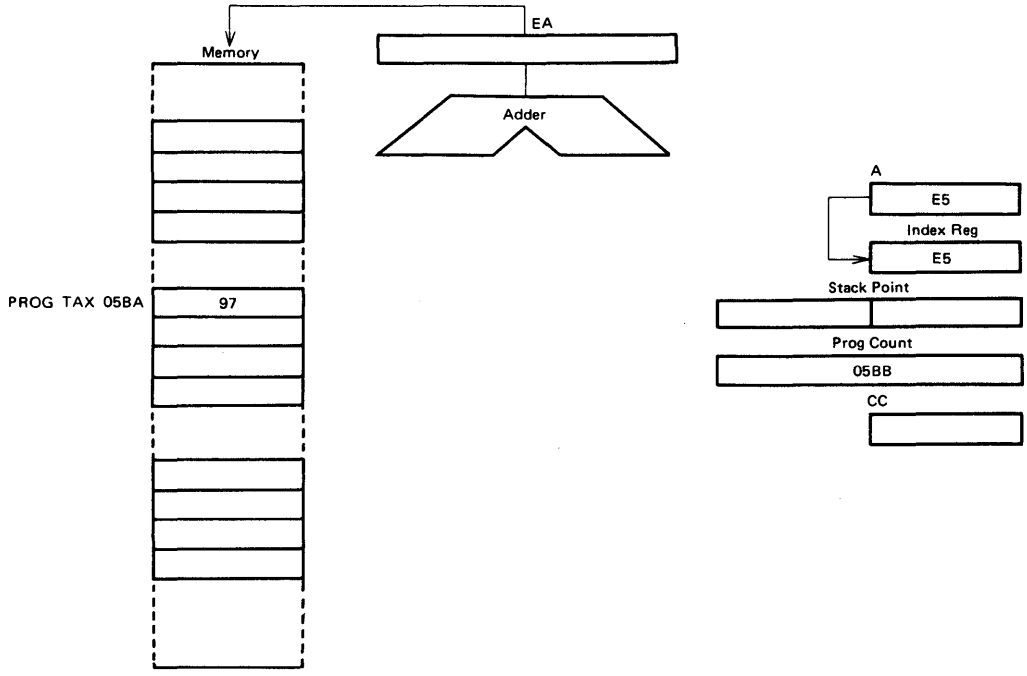


Figure 30 Example of Implied Addressing

■ INSTRUCTION SET

There are 59 instructions available to the MCU. They can be divided into five groups: Register/Memory, Read/Modify/Write, Branch, Bit Processing, and Control. All of these instructions are explained below according to the groups, and are summarized in individual tables.

• Register/Memory

Most of these instructions use two operands. One operand is either the accumulator or index register, while the other is acquired from memory using one of the addressing modes. No operand of register is available in the unconditional Jump (JMP) and Subroutine Jump (JSR) instructions. See Table 4.

• Read/Modify/Write

These instructions read a memory address or register, modify or test its contents, and writes a new value into the memory or register. Negative or Zero instructions (TST) do not provide writing, and are exceptions for the Read/Modify/Write. See Table 5.

• Branch

A Branch instruction will branch from the program sequence in progress if the specific branch condition is satisfied. See Table 6.

• Bit Processing

This instruction can be used for any bit of the first 256 bytes of memory. One group is used for setting or clearing, while the other is used for bit testing and branching. See Table 7.

• Control

The Control instruction controls the operation of the MCU for which a program is being executed. See Table 8.

• A List of Instructions Arranged in Alphabetical Order

All instructions are listed in Table 9 in the alphabetical order.

• OP Code Map

Table 10 shows an OP code map of the instructions used with the MCU.

Table 4 Register/Memory Instructions

		Addressing Mode																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
Operation	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	2	E6	2	4	D6	3	5
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	2	EE	2	4	DE	3	5
Store A in Memory	STA	–	–	–	B7	2	4	C7	3	5	F7	1	3	E7	2	5	D7	3	6
Store X in Memory	STX	–	–	–	BF	2	4	CF	3	5	FF	1	3	EF	2	5	DF	3	6
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	2	EB	2	4	DB	3	5
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	2	E9	2	4	D9	3	5
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	2	E0	2	4	D0	3	5
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	2	E2	2	4	D2	3	5
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	2	E4	2	4	D4	3	5
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	2	EA	2	4	DA	3	5
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	2	E8	2	4	D8	3	5
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	2	E1	2	4	D1	3	5
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	2	E3	2	4	D3	3	5
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	2	E5	2	4	D5	3	5
Jump Unconditional	JMP	–	–	–	BC	2	2	CC	3	3	FC	1	1	EC	2	3	DC	3	4
Jump to Subroutine	JSR	–	–	–	BD	2	4	CD	3	5	FD	1	3	ED	2	4	DD	3	5

Symbols: Op = Operation # = Instruction

Table 5 Read/Modify/Write Instructions

		Addressing Mode														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
Operation	Mnemonic	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	1	5C	1	1	3C	2	4	7C	1	3	6C	2	5
Decrement	DEC	4A	1	1	5A	1	1	3A	2	4	7A	1	3	6A	2	5
Clear	CLR	4F	1	1	5F	1	1	3F	2	4	7F	1	3	6F	2	5
Complement	COM	43	1	1	53	1	1	33	2	4	73	1	3	63	2	5
Negate (2's Complement)	NEG	40	1	1	50	1	1	30	2	4	70	1	3	60	2	5
Rotate Left Thru Carry	ROL	49	1	1	59	1	1	39	2	4	79	1	3	69	2	5
Rotate Right Thru Carry	ROR	46	1	1	56	1	1	36	2	4	76	1	3	66	2	5
Logical Shift Left	LSL	48	1	1	58	1	1	38	2	4	78	1	3	68	2	5
Logical Shift Right	LSR	44	1	1	54	1	1	34	2	4	74	1	3	64	2	5
Arithmetic Shift Right	ASR	47	1	1	57	1	1	37	2	4	77	1	3	67	2	5
Arithmetic Shift Left	ASL	48	1	1	58	1	1	38	2	4	78	1	3	68	2	5
Test for Negative or Zero	TST	4D	1	1	5D	1	1	3D	2	4	7D	1	3	6D	2	5

Symbols: Op = Operation # = Instruction

Table 6 Branch Instructions

Operation	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	3
Branch Never	BRN	21	2	2 or 3 *
Branch IF Higher	BHI	22	2	2 or 3 *
Branch IF Lower or Same	BLS	23	2	2 or 3 *
Branch IF Carry Clear	BCC	24	2	2 or 3 *
(Branch IF Higher or Same)	(BHS)	24	2	2 or 3 *
Branch IF Carry Set	BCS	25	2	2 or 3 *
(Branch IF Lower)	(BLO)	25	2	2 or 3 *
Branch IF Not Equal	BNE	26	2	2 or 3 *
Branch IF Equal	BEQ	27	2	2 or 3 *
Branch IF Half Carry Clear	BHCC	28	2	2 or 3 *
Branch IF Half Carry Set	BHCS	29	2	2 or 3 *
Branch IF Plus	BPL	2A	2	2 or 3 *
Branch IF Minus	BMI	2B	2	2 or 3 *
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	2 or 3 *
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	2 or 3 *
Branch IF Interrupt Line is Low	BIL	2E	2	2 or 3 *
Branch IF Interrupt Line is High	BIH	2F	2	2 or 3 *
Branch to Subroutine	BSR	AD	2	4

Symbol: Op = Operation # = Instruction
* If branched, each instruction will be a 3-cycle instruction.

Table 7 Bit Processing Instructions

Operations	Mnemonic	Addressing Mode					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is Set	BRSET n (n = 0.....7)	—	—	—	2 • n	3	4 or 5 *
Branch IF Bit n is Clear	BRCLR n (n = 0.....7)	—	—	—	01 + 2 • n	3	4 or 5 *
Set Bit n	BSET n (n = 0.....7)	10 + 2 • n	2	4	—	—	—
Clear Bit n	BCLR n (n = 0.....7)	11 + 2 • n	2	4	—	—	—

Symbol: Op = Operation # = Instruction
* If Branched, each instruction will be a 5-cycle instruction.

Table 8 Control Instructions

Operation	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	1
Transfer X to A	TXA	9F	1	1
Set Carry Bit	SEC	99	1	1
Clear Carry Bit	CLC	98	1	1
Set Interrupt Mask Bit	SEI	9B	1	1
Clear Interrupt Mask Bit	CLI	9A	1	1
Software Interrupt	SWI	83	1	9
Return from Subroutine	RTS	81	1	4
Return from Interrupt	RTI	80	1	7
Reset Stack Pointer	RSP	9C	1	1
No-Operation	NOP	9D	1	1

Symbol: Op = Operation # = Instruction

Table 9 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•

(Continued)

Symbols for condition code:
H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
∧ Test and Set if True, Cleared Otherwise
• Not Affected

Table 9 Instruction Set (Continued)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Symbols for condition code:

H Half Carry (From Bit 3)
I Interrupt Mask
N Negative (Sign Bit)
Z Zero
C Carry/Borrow
∧ Test and Set if True, Cleared Otherwise
• Not Affected
? Load CC Register From Stack

Table 10 OP Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	←HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA (+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+1)		JSR		JSR(+1)	D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/4 or 5	2/4	2/2 or 3	2/4	1/1	1/1	2/5	1/3	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/2	

(NOTES) 1. "–" is an undefined operation code.
2. The figure in the lowest row of each column gives the number of bytes and the cycles needed for the instruction.
The number of cycles for the asterisked (*) mnemonics is a follows:

RTI	7
RTS	4
SWI	9
BSR	4

3. The parenthesized figure must be added to the cycle count of the associated instruction.
4. If the instruction is branched, the cycle count is the larger figure.

HD63L05F
MASK OPTION LIST

DATE OF ORDER	
CUSTOMER	
DEPT.	
ACCEPTED BY	
ROM CODE ID.	
LSI TYPE NO.	HD63L05F

* Select one type for each item and check ●.

(1) OSC OPTION

Type of OSC1	Use of OSC2	Condition		Delay Time of Restart (sec.) 0	1/16	1/2	1
XTAL Option	Used	STANDBY mode	Used	***	***		
			Not Used				
	Not used	STANDBY mode	Used	***	***		
			Not Used				
CR Option	Used	Oscillation of OSC1 at HALT	Stop		***	***	***
			Continue				
	Not Used	Oscillation of OSC1 at HALT	Stop	***	***	***	***
			Continue				

* Specify a type of OSC option.
* Crystal option of OSC1 is not allowed to stop at HALT.
* If OSC2 is not used, the Delay Time is not acculate.

(2) I/O OPTION

Port	Mask Option A	B	C	D
A_0				
A_1				
A_2				
A_3				
A_4				
A_5				
A_6				
A_7				
B_0				
B_1				
B_2				
B_3				
B_4				
B_5				
B_6				
B_7				
C_0				
C_1				
C_2				
C_3				

Pin	Mask Option E	F
$\overline{INT}$		

Pin	Mask option G	H	K
SEG_{13}/CH_6			***
SEG_{14}/CH_5			***
SEG_{15}/CH_4			***
SEG_{16}/CH_3			***
SEG_{17}/CH_2			***
$O_{18}/CH_8/V_2$			
$O_{19}/CH_7/V_1$			

A : CMOS output without input pull-up PMOS
B : CMOS output with input pull-up PMOS
C : CMOS output for key scanning
D : NMOS open–drain output
E : Input without pull-up PMOS
F : Input with pull-up PMOS
G : A/D Input
H : Segment output
K : Terminals for LCD display

* Specify an I/O option for each terminal.

(3) LCD DRIVER

	Mask Option L	S	P
Segment			

Mask options indicated as *** are not available.

L : 1/3 bias-1/3 duty LCD
S : Static LCD
P : Output port

* Specify a type of LCD driver.

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

(4) LCD PIN LOCATION

LCD Register	Bit	Timing			Segment Output Terminal																	Output	
		COM_1	COM_2	COM_3	SEG_1	SEG_2	SEG_3	SEG_4	SEG_5	SEG_6	SEG_7	SEG_8	SEG_9	SEG_{10}	SEG_{11}	SEG_{12}	SEG_{13}	SEG_{14}	SEG_{15}	SEG_{16}	SEG_{17}	O_{18}	O_{19}
LCD1	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
	7																						
LCD2	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
	7																						
LCD3	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
LCD4	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
LCD5	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
LCD6	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
LCD7	0																						
	1																						
	2																						
	3																						
	4																						
	5																						
	6																						
LCD8	0																						
	1																						
	2																						
	3																						
ϕWRITE		○																					
1/4 OSC1		○																					

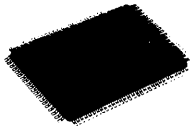
* Specify the multiplex timing and segment terminal for each bit of LCD1 to LCD8.
* When static or output port is selected, the Multiplex timing is fixed at COM_1.
* If there are unspecified bits, Hitachi specifies them as dummy.
* ϕWRITE is generated when data is written into the LCD1.
* 1/4 OSC1 is a quarter of the OSC1 clock speed. When the MCU is in standby mode, it becomes "Low".

HD63L05E0
Evaluation Chip for HD63L05F1

HD63L05E is a CMOS evaluation chip for the HD63L05F. Connecting an external EPROM (HN462732) to the chip, it can be operated as a single chip microcomputer HD63L05F. Interface signals are 12 bit Address Bus ($E_0 \sim E_7$, $F_0 \sim F_3$), 8 bit Data Bus ($D_0 \sim D_7$) and Chip Enable ($\overline{CE}$).

It is easy to debug the HD63L05F user program with this evaluation chip.

HD63L05E0



(FP-100)

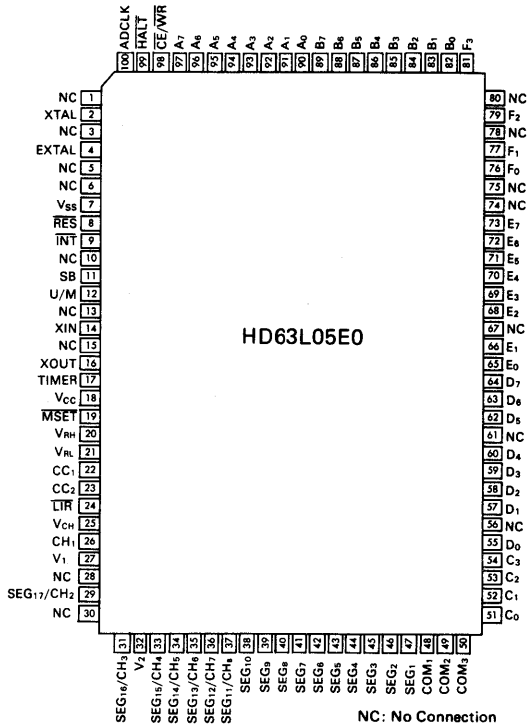
■ FEATURES

- 3V Power Supply
- 96 Bytes RAM
- EPROM (HN462732) Interface
- LCD Driver
- 8-bit Programmable Timer with 7-bit Prescaler
- 8-bit A/D Converter
- 20 parallel I/O Port
- Same Instruction Set as HD63L05F
- NMOS Open-drain Output
- 100 Pin Flat Package (FP-100)

■ TERMINALS

$A_0 \sim A_7$	:	I/O Port
$B_0 \sim B_7$	:	I/O Port
$C_0 \sim C_3$	:	I/O Port
$D_0 \sim D_7$	:	Data Bus (Input)
$E_0 \sim E_7$	:	Lower 8 bit Address Bus (Output)
$F_0 \sim F_3$	:	Upper 4 bit Address Bus (Output)
U/M	:	Test Terminal
$\overline{CE/WR}$	:	Chip Enable, Read/Write
$\overline{LIR}$	:	Instruction Fetch Signal
ADCLK	:	E Clock
$\overline{HALT}$	:	External clock control signal
$\overline{MSET}$	:	Connected to V_{CC}

■ PIN ARRANGEMENT

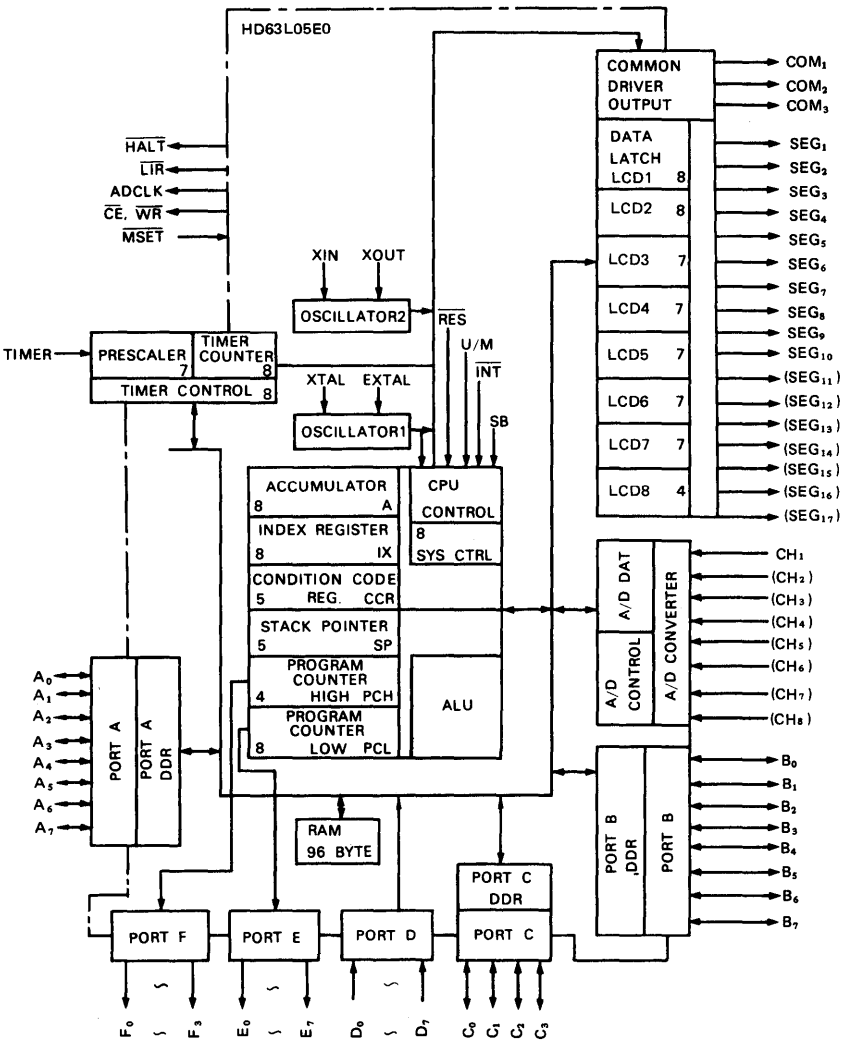


NC: No Connection

(Top View)

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM

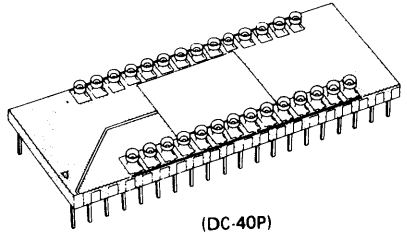


HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

HD68P01V07, HD68P01V07-1
HD68P01M0, HD68P01M0-1
MCU (Microcomputer Unit)

The HD68P01 is an 8-bit single chip microcomputer unit (MCU) which significantly enhances the capabilities of the HMCS6800 family of parts. It can be used in production systems to allow for easy firmware changes with minimum delay or it can be used to emulate the HD6801 for software development. It includes 128 bytes of RAM, Serial Communications Interface (SCI), parallel I/O and a three function Programmable Timer on chip, and 2048 bytes, 4096 bytes or 8192 bytes of EPROM on package. It includes an upgrade HD6800 microprocessing unit (MPU) while retaining upward source and object code compatibility. Execution times of key instructions have been improved and several new instructions have been added including an unsigned 8 by 8 multiply with 16-bit result. The HD68P01 can function as a monolithic microcomputer or can be expanded to a 65k byte address space. It is TTL compatible and requires one +5 volt power supply. A summary of HD68P01 features includes:

HD68P01V07, HD68P01V07-1, HD68P01M0, HD68P01M0-1



(DC-40P)

■ FEATURES

- Expanded HMCS6800 Instruction Set
- 8 x 8 Multiply Instruction
- Serial Communications Interface (SCI)
- Upward Source and Object Code Compatible with HD6800
- 16-bit Three-function Programmable Timer
- Applicable to All Type of EPROM
  - 4096 bytes; HN482732A
  - 8192 bytes; HN482764
- 128 Bytes of RAM (64 bytes Retainable on Powerdown)
- 29 Parallel I/O and Two Handshake Control Line
- Internal Clock Generator with Divide-by-Four Output
- Full TTL Compatibility
- Full Interrupt Capability
- Single-Chip or Expandable to 65k Bytes Address Space
- Bus compatible with HMCS6800 Family

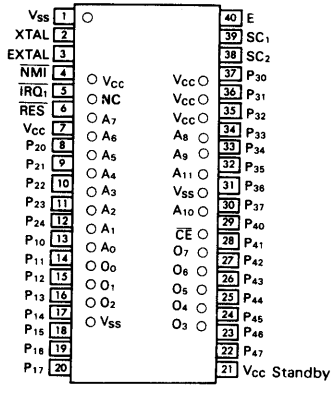
■ TYPE OF PRODUCTS

Type No.	Bus Timing	EPROM Type No.
HD68P01V07	1 MHz	HN482732A-30
HD68P01V07-1	1.25MHz	HN482732A-30
HD68P01M0	1 MHz	HN482764-3
HD68P01M0-1	1.25MHz	HN482764-3

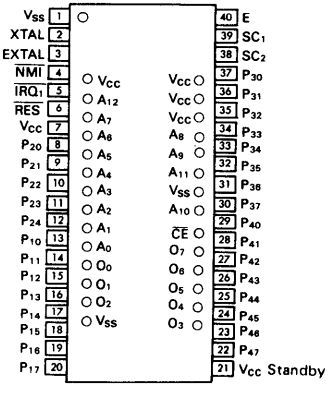
Note) EPROM is not attached to the MCU.

■ PIN ARRANGEMENT (Top View)

HD68P01V07, HD68P01V07-1

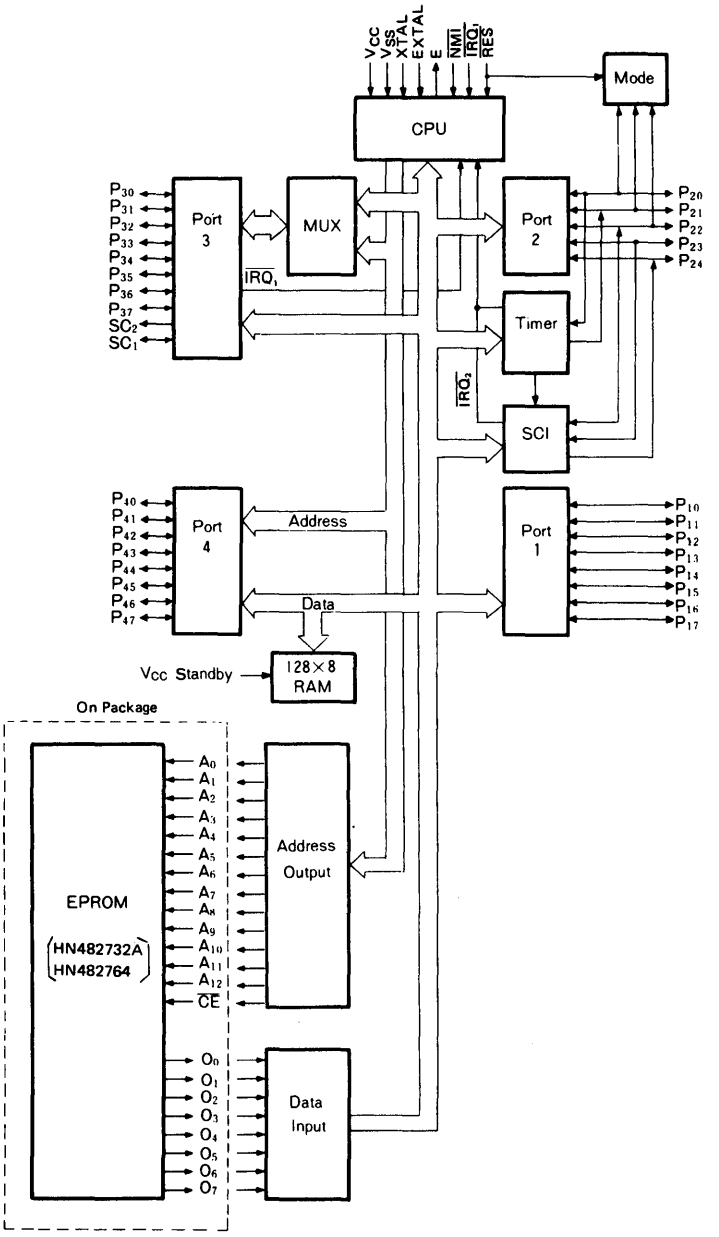


HD68P01M0, HD68P01M0-1



Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BLOCK DIAGRAM**



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage	V_{in} *	-0.3 ~ +7.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

[NOTE] Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} =5.0V±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	Other Inputs*			2.0	–	V_{CC}	
Input "Low" Voltage	All Inputs*	V_{IL}		-0.3	–	0.8	V
Input Load Current	P_{40} ~ P_{47}	$\|I_{in}\|$	V_{in} = 0 ~ 2.4V	–	–	0.5	mA
	SC_1			–	–	0.8	
	EXTAL		V_{in} = 0 ~ V_{CC}	–	–	1.2	
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$	$\|I_{in}\|$	V_{in} = 0 ~ 5.25V	–	–	2.5	μA
Three State (Offset) Leakage Current	P_{10} ~ P_{17}, P_{30} ~ P_{37}	$\|I_{TSI}\|$	V_{in} = 0.5 ~ 2.4V	–	–	10	μA
	P_{20} ~ P_{24}			–	–	100	
Output "High" Voltage	P_{30} ~ P_{37}	V_{OH}	I_{LOAD} = -205 μA	2.4	–	–	V
	P_{40} ~ P_{47}, E, SC_1, SC_2		I_{LOAD} = -145 μA	2.4	–	–	
	Other Outputs		I_{LOAD} = -100 μA	2.4	–	–	
Output "Low" Voltage	All Outputs	V_{OL}	I_{LOAD} = 1.6 mA	–	–	0.5	V
Darlington Drive Current	P_{10} ~ P_{17}	$-I_{OH}$	V_{out} = 1.5V	1.0	–	10.0	mA
Power Dissipation		P_D		–	–	1200	mW
Input Capacitance	P_{30} ~ P_{37}, P_{40} ~ P_{47}, SC_1	C_{in}	V_{in} = 0V, Ta = 25°C, f = 1.0 MHz	–	–	12.5	pF
	Other Inputs			–	–	12.5	
V_{CC} Standby	Powerdown	V_{SBB}		4.0	–	5.25	V
	Operating	V_{SB}		4.75	–	5.25	
Standby Current	Powerdown	I_{SBB}	V_{SBB} = 4.0V	–	–	8.0	mA

*Except Mode Programming Levels: See Figure 8.

• AC CHARACTERISTICS

BUS TIMING (V_{CC} = 5.0V±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	HD68P01V07/M0			HD68P01V07-1/M0-1			Unit
				min	typ	max	min	typ	max	
Cycle Time		t_{cyc}		1	–	10	0.8	–	10	μs
Address Strobe Pulse width "High"*		PW_{ASH}		200	–	–	150	–	–	ns
Address Strobe Rise Time		t_{ASr}		5	–	50	5	–	50	ns
Address Strobe Fall Time		t_{ASf}		5	–	50	5	–	50	ns
Address Strobe Delay Time*		t_{ASD}		60	–	–	30	–	–	ns
Enable Rise Time		t_{Er}		5	–	50	5	–	50	ns
Enable Fall Time		t_{Ef}		5	–	50	5	–	50	ns
Enable Pulse Width "High" Time*		PW_{EH}		450	–	–	340	–	–	ns
Enable Pulse Width "Low" Time*		PW_{EL}		450	–	–	350	–	–	ns
Address Strobe to Enable Delay Time*		t_{ASED}		60	–	–	30	–	–	ns
Address Delay Time		t_{AD}	Fig. 1	–	–	260	–	–	260	ns
Address Delay Time for Latch (f = 1.0MHz)*		t_{ADL}	Fig. 2	–	–	270	–	–	260	ns
Data Set-up Write Time		t_{DSW}		225	–	–	115	–	–	ns
Data Set-up Read Time		t_{DSR}		80	–	–	70	–	–	ns
Data Hold Time	Read	t_{HR}		10	–	–	10	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	
Address Set-up Time for Latch*		t_{ASL}		60	–	–	50	–	–	ns
Address Hold Time for Latch		t_{AHL}		20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	ns
Peripheral Read Access Time	Non-Multiplexed Bus*	(t_{ACCN})		–	–	(610)	–	–	(420)	ns
	Multiplexed Bus*	(t_{ACCM})		–	–	(600)	–	–	(420)	
Oscillator stabilization Time		t_{RC}	Fig. 11	100	–	–	100	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 12	200	–	–	200	–	–	ns

* These timings change in approximate proportion to tcyc. The figures in this characteristics represent those when tcyc is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Peripheral Data Setup Time	Port 1, 2, 3, 4	t_{PDSU}	Fig. 3	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		t_{OSD1}	Fig. 5	–	–	350	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		t_{OSD2}	Fig. 5	–	–	350	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4	t_{PWD}	Fig. 4	–	–	400	ns
Delay Time, Enable Negative Transition to Peripheral CMOS Data Valid	Port 2**, 4	t_{CMOS}	Fig. 4	–	–	2.0	μs
Input Strobe Pulse Width		t_{PWIS}	Fig. 6	200	–	–	ns
Input Data Hold Time	port 3	t_{IH}	Fig. 6	50	–	–	ns
Input Data Set-up Time	Port 3	t_{IS}	Fig. 6	20	–	–	ns

*Except P_{21} **10kΩ pull up register required for Port 2

TIMER, SCI TIMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Timer Input Pulse Width	t_{PWT}		$2\,t_{cyc}$+200	–	–	ns
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 7	–	–	600	ns
SCI Input Clock Cycle	t_{Scyc}		1	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING (V_{CC} = 5.0V ±5%, V_{SS} = 0V, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Mode Programming Input "Low" Voltage		V_{MPL}	Fig. 8	–	–	1.7	V
Mode Programming Input "High" Voltage		V_{MPH}		4.0	–	–	V
$\overline{RES}$ "Low" Pulse Width		PW_{RSTL}		3.0	–	–	t_{cyc}
Mode Programming Set-up Time		t_{MPS}		2.0	–	–	t_{cyc}
Mode Programming Hold Time	$\overline{RES}$ Rise Time $\geq 1\mu s$	t_{MPH}		0	–	–	ns
	$\overline{RES}$ Rise Time $< 1\mu s$			100	–	–	

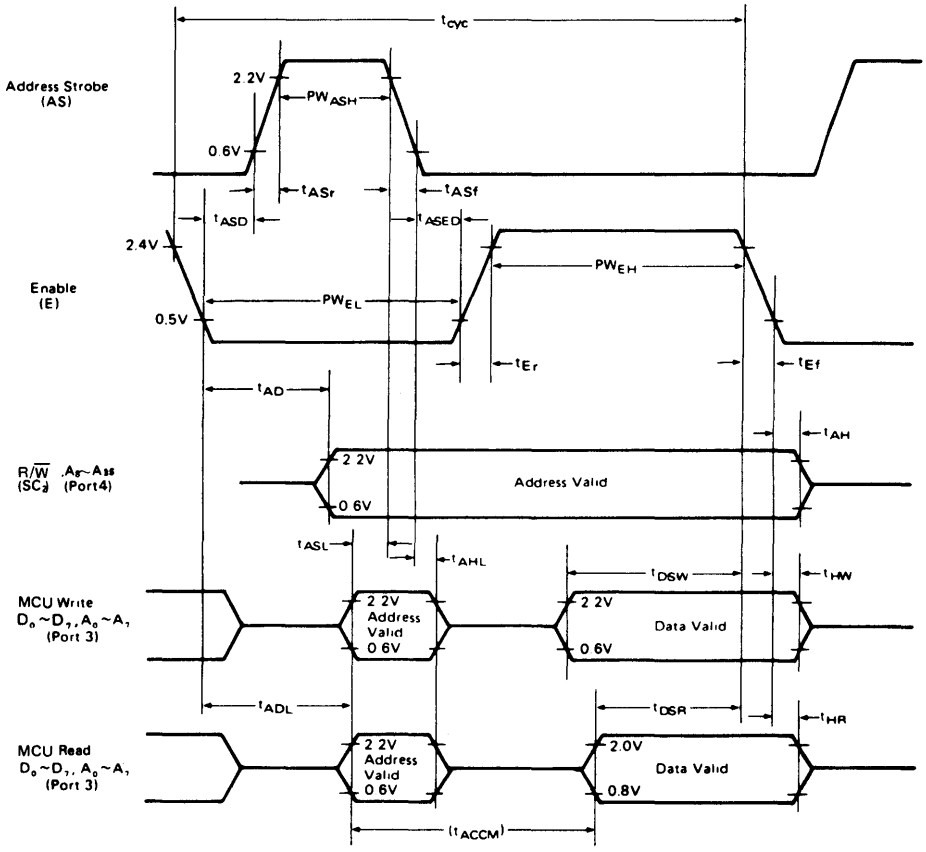


Figure 1 Expanded Multiplexed Bus Timing

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

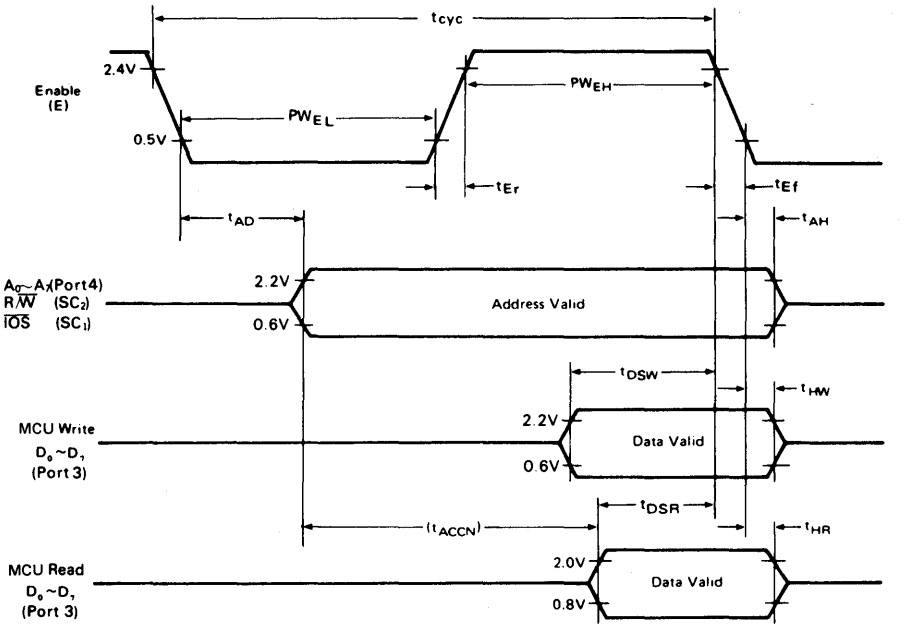


Figure 2 Expanded Non-Multiplexed Bus Timing

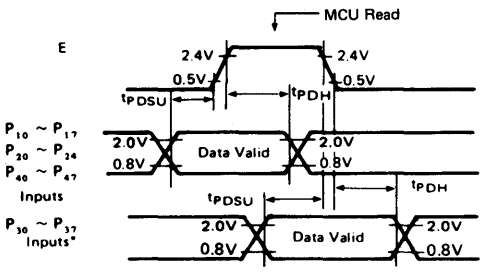
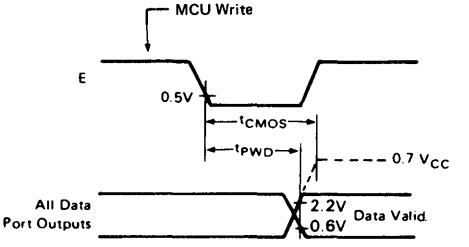


Figure 3 Data Set-up and Hold Times (MCU Read)



(NOTE) 1. 10 kΩ Pullup resistor required for Port 2 to reach 0.7 V_{CC}
2. Not applicable to P_{21}
3. Port 4 cannot be pulled above V_{CC}

Figure 4 Port Data Delay Timing (MCU Write)

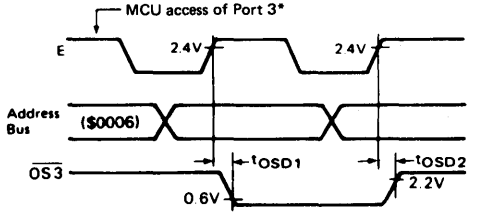


Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

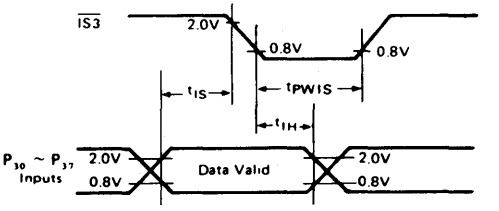


Figure 6 Port 3 Latch Timing (Single Chip Mode)

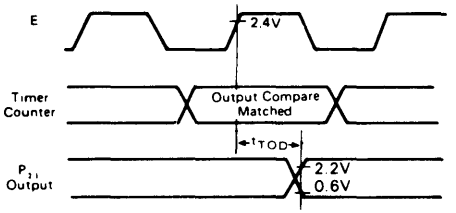


Figure 7 Timer Output Timing

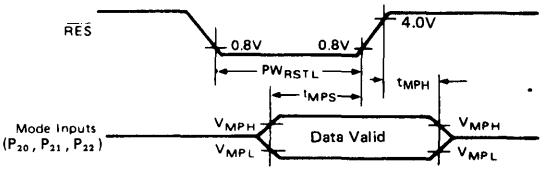
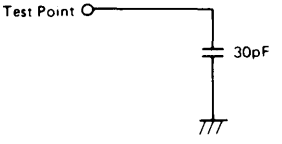
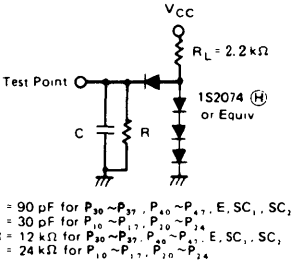


Figure 8 Mode Programming Timing



(a) CMOS Load



(b) TTL Load

Figure 9 Bus Timing Test Loads

■ INTRODUCTION

The HD68P01 is an 8-bit monolithic microcomputer which can be configured to function in a wide variety of applications. The facility which provides this extraordinary flexibility is its ability to be hardware programmed into eight different operating modes. The operating mode controls the configuration of 18 of the MCU's 40 pins, available on-chip resources, memory map, location (internal or external) of interrupt vectors, and type of external bus. The configuration of the remaining 22 pins is not dependent on the operating mode.

Twenty-nine pins are organized as three 8-bit ports and one 5-bit port. Each port consists of at least a Data Register and a write-only Data Direction Register. The Data Direction Register is used to define whether corresponding bits in the Data Register are configured as an input (clear) or output (set).

The term "port", by itself, refers to all of its associated hardware. When the port is used as a "data port" or "I/O port", it is controlled by its Data Direction Register and the programmer has direct access to its pins using the port's Data Register. Port pins are labled as Pij where i identifies one of four ports and j indicates the particular bit.

The Microprocessor Unit (MPU) is an enhanced HD6800 MPU with additional capabilities and greater throughput. It is upward source and object code compatible with the HD6800. The programming model is depicted in Figure 10 where Accumulator D is a concatenation of Accumulators A and B. A list of new operations added to the HMCS6800 instruction set are shown in Table 8.

The basic difference between the HD6801 and the HD68P01 is that the HD6801 has an on-chip ROM while the HD68P01 has an on the package EPROM. The HD68P01 is pin and code compatible with the HD6801 and can be used to emulate the HD6801, allowing easy software development using the on-package EPROM. Software developed using the HD68P01 can then be masked into the HD6801 ROM.

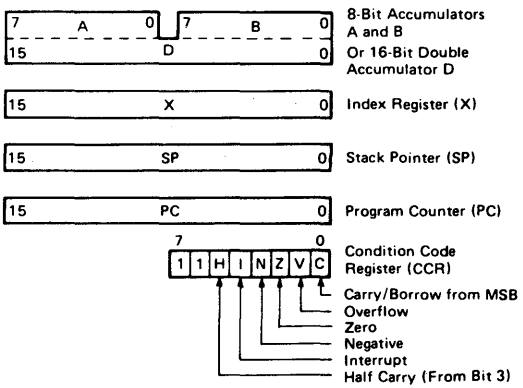


Figure 10 HD68P01 Programming Model

■ INTERRUPTS

The MCU supports two types of interrupt requests: maskable and non-maskable. A Non-Maskable Interrupt ($\overline{NMI}$) is always recognized and acted upon at the completion of the current instruction. Maskable interrupts are controlled by the Condition Code Register's I-bit and by individual enable bits. The I-bit controls all maskable interrupts. Of the maskable interrupts, there are two types: $\overline{IRQ_1}$ and $\overline{IRQ_2}$. The Programmable Timer and Serial Communications Interface use an internal $\overline{IRQ_2}$ interrupt line, as shown in BLOCK DIAGRAM. External devices (and $\overline{IS3}$) use $\overline{IRQ_1}$. An $\overline{IRQ_1}$ interrupt is serviced before $\overline{IRQ_2}$ if both are pending.

All $\overline{IRQ_2}$ interrupts use hardware prioritized vectors. The single SCI interrupt and three timer interrupts are serviced in a prioritized order where each is vectored to a separate location. All MCU interrupt vector locations are shown in Table 1.

The Interrupt flowchart is depicted in Figure 13 and is common to every MCU interrupt excluding Reset. The Program Counter, Index Register, A Accumulator, B Accumulator, and Condition Code Register are pushed to the stack. The I-bit is set to inhibit maskable interrupts and a vector is fetched corresponding to the current highest priority interrupt. The vector is transferred to the Program Counter and instruction execution is resumed. Interrupt and $\overline{RES}$ timing is illustrated in Figure 11 and 12.

Table 1 Interrupt Vector Locations

MSB	LSB	Interrupt
FFFE	FFFF	$\overline{RES}$
FFFC	FFFD	$\overline{NMI}$
FFFA	FFFB	Software Interrupt (SWI)
FFF8	FFF9	$\overline{IRQ_1}$ (or IS3)
FFF6	FFF7	ICF (Input Capture)
FFF4	FFF5	OCF (Output Compare)
FFF2	FFF3	TOF (Timer Overflow)
FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

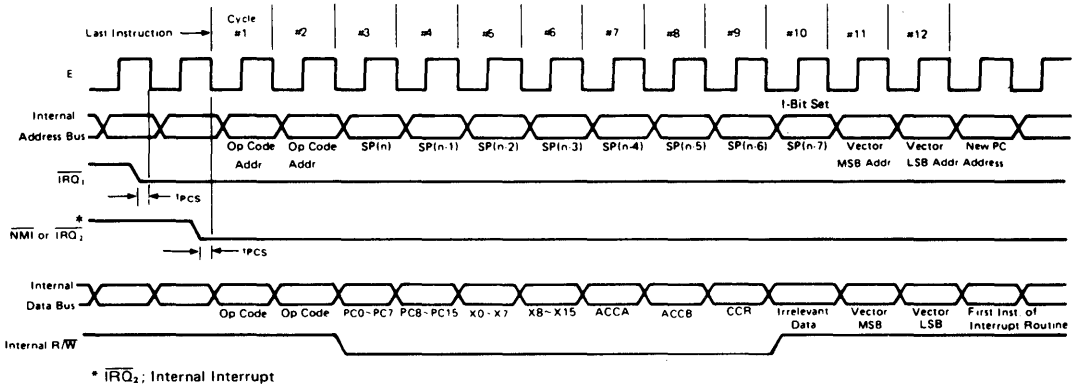


Figure 11 Interrupt Sequence

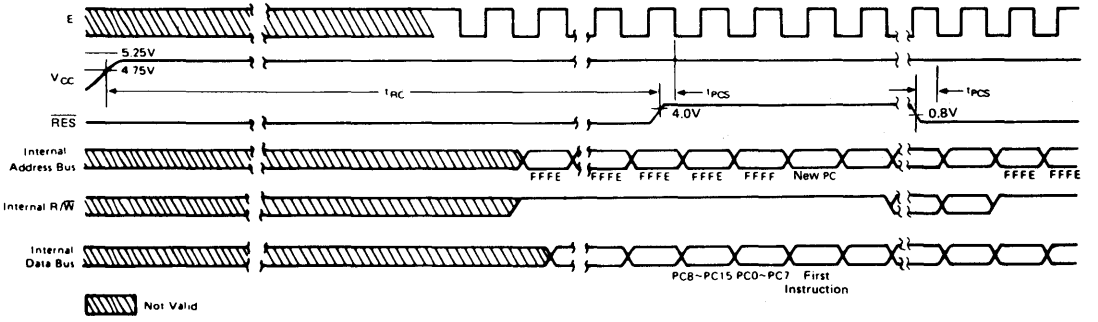


Figure 12 Reset Timing

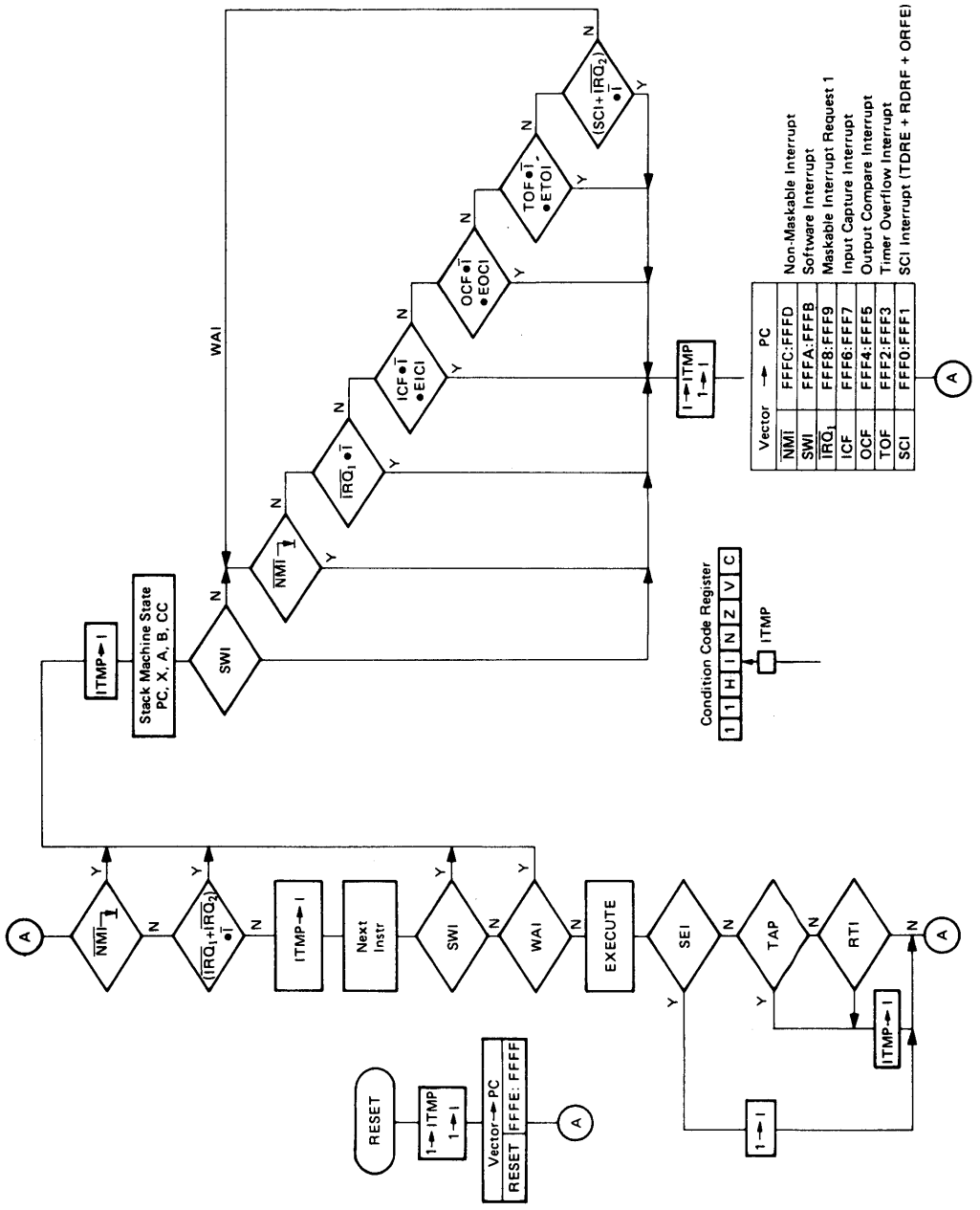


Figure 13 Interrupt Flowchart

■ FUNCTIONAL PIN DESCRIPTIONS

• V_{CC} and V_{SS}

V_{CC} and V_{SS} provide power to a large portion of the MCU. The power supply should provide +5 volts (±5%) to V_{CC}, and V_{SS} should be tied to ground. Total power dissipation (including V_{CC} Standby), will not exceed P_D milliwatts.

• V_{CC} Standby

V_{CC} Standby provides power to the standby portion ($80 through $BF) of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the MCU is in a powerup or powerdown state. In the powerup state, the power supply should provide +5 volts (±5%) and must reach V_{SB} volts before $\overline{RES}$ reaches 4.0 volts. During powerdown, V_{CC} Standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} Standby from the same source during normal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation. V_{CC} Standby should be tied to either ground or V_{CC} in Mode 3.

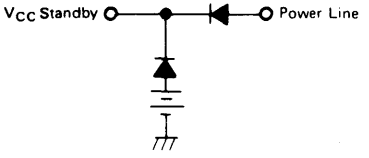


Figure 14 Battery Backup for V_{CC} Standby

• RAM Control Register ($14)

The RAM Control Register includes two bits which can be used to control RAM accesses and determine the adequacy of the standby power source during powerdown operation. It is intended that RAME be cleared and STBY PWR be set as part of a powerdown procedure.

RAM Control Register

7	6	5	4	3	2	1	0
STBY PWR	RAME	X	X	X	X	X	X

Bit 0~5 Not Used

Bit 6 RAME — RAM Enable. This Read/Write bit can be used to remove the entire RAM from the internal memory map. RAME is set (enabled) during Reset provided standby power is available on the positive edge of $\overline{RES}$. If RAME is clear, any access to a RAM address is external. If RAME is set and not in Mode 3, the RAM is included in the internal map.

Bit 7 STBY PWR — Standby Power. This bit is a Read/Write status bit which is cleared whenever V_{CC} Standby decreases below V_{SBB} (min). It can be set only by software and is not affected by $\overline{RES}$.

• XTAL and EXTAL

These two input pins interface either a crystal or TTL compatible clock to the MCU's internal clock generator. Divide-by-four circuitry is included which allows use of the inexpensive 3.58 MHz Color Burst TV crystals. A 22 pF capacitor is required from each crystal pin to ground to ensure reliable startup and operation. Alternatively, EXTAL may be driven with an external TTL compatible clock with a duty cycle of 45% ~ 55% with XTAL connected to ground.

The internal oscillator is designed to interface with an AT-cut quartz crystal resonator or a ceramic resonator operated in parallel resonance mode in the frequency range specified for 3.2 ~ 4 MHz. The crystal should be mounted as close as possible to the input pins to minimize output distortion and startup stabilization time. The MCU is compatible with most commercially available crystals and ceramic resonators and nominal crystal parameters are shown in Figure 15.

• $\overline{RES}$

This input is used to reset the MCU's internal state and provide an orderly startup procedure. During powerup, $\overline{RES}$ must be held below 0.8 volts: (1) at least t_{RC} after V_{CC} reaches 4.75 volts in order to provide sufficient time for the clock generator to stabilize, and (2) until V_{CC} Standby reaches 4.75 volts. $\overline{RES}$ must be held low at least three E-cycles if asserted during powerup operation.

When a "High" level is detected, the MCU does the following:

1) All the higher order address lines will be forced "High".
2) I/O Port 2 bits, 2, 1, and 0 are latched into programmed control bits PC2, PC1 and PC0.
3) The last two ($FFFE, $FFFF) locations in memory will be used to load the program addressed by the program counter.
4) The interrupt mask bit is set; must be cleared before the CPU can recognize maskable interrupts.

• E (Enable)

This is an output clock used primarily for bus synchronization. It is TTL compatible and is the slightly skewed divide-by-four result of the MCU input frequency. It will drive one Schottky TTL load and 90 pF, and all data given in cycles is referenced to this clock unless otherwise noted.

• $\overline{NMI}$ (Non-Maskable Interrupt)

An $\overline{NMI}$ negative edge request an CPU interrupt sequence, but the current instruction will be completed before it responds to the request. The CPU will then begin an interrupt sequence. Finally, a vector is fetched from $FFFC and $FFFD, transferred to the Program Counter and instruction execution resumes. $\overline{NMI}$ typically requires a 3.3 kΩ (nominal) resistor to V_{CC}. There is no internal $\overline{NMI}$ pullup resistor. $\overline{NMI}$ must be held low for at least one E-cycle to be recognized under all conditions.

• $\overline{IRQ_1}$ (Maskable Interrupt Request 1)

$\overline{IRQ_1}$ is a level-sensitive input which can be used to request an interrupt sequence. The CPU will complete the current instruction before it responds to the request. If the interrupt mask bit (I-bit) in the Condition Code Register is clear, the CPU will begin an interrupt sequence. Finally, a vector is fetched from $FFF8 and $FFF9, transferred to the Program Counter, and instruction execution is resumed.

$\overline{IRQ_1}$ typically requires an external 3.3 kΩ (nominal) resistor to V_{CC} for wire-OR application. $\overline{IRQ_1}$ has no internal pullup resistor.

- **SC_1 and SC_2 (Strobe Control 1 and 2)**

The function of SC_1 and SC_2 depends on the operating mode. SC_1 is configured as an output in all modes except single chip mode, whereas SC_2 is always an output. SC_1 and SC_2 can drive one Schottky load and 90 pF.

SC_1 and SC_2 in Single Chip Mode

In Single Chip Modes, SC_1 and SC_2 are configured as an input and output, respectively, and both function as Port 3 control lines. SC_1 functions as $\overline{IS3}$ and can be used to indicate that Port 3 input data is ready or output data has been accepted. Three options associated with $\overline{IS3}$ are controlled by Port 3's Control and Status Register and are discussed in Port 3's description. If unused, $\overline{IS3}$ can remain unconnected.

SC_2 is configured as $\overline{OS3}$ and can be used to strobe output data or acknowledge input data. It is controlled by Output Strobe Select (OSS) in Port 3's Control and Status Register. The strobe is generated by a read (OSS= 0) or write (OSS = 1) to Port 3's Data Register. $\overline{OS3}$ timing is shown in Figure 5.

SC_1 and SC_2 in Expanded Non-Multiplexed Mode

In the Expanded Non-Multiplexed Mode, both SC_1 and SC_2 are configured as outputs. SC_1 functions as Input/Output Select ($\overline{IOS}$) and is asserted only when $0100 through $01FF is sensed on the internal address bus.

SC_2 is configured as Read/Write and is used to control the direction of data bus transfers. An CPU read is enabled when Read/Write and E are high.

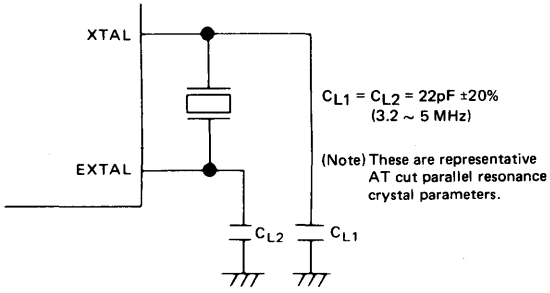
SC_1 and SC_2 in Expanded Multiplexed Mode

In the Expanded Multiplexed Modes, both SC_1 and SC_2 are configured as outputs. SC_1 functions as Address Strobe and can be used to demultiplex the eight least significant addresses and the data bus. A latch controlled by Address Strobe captures address on the negative edge, as shown in Figure 20.

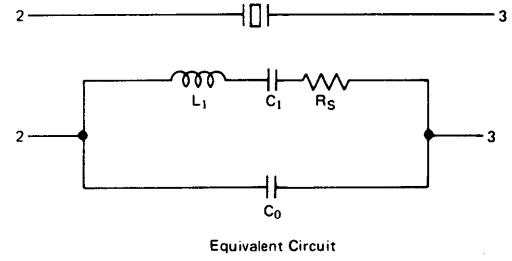
SC_2 is configured as Read/Write and is used to control the direction of data bus transfers. An CPU. read is enabled when Read/Write and E are high.

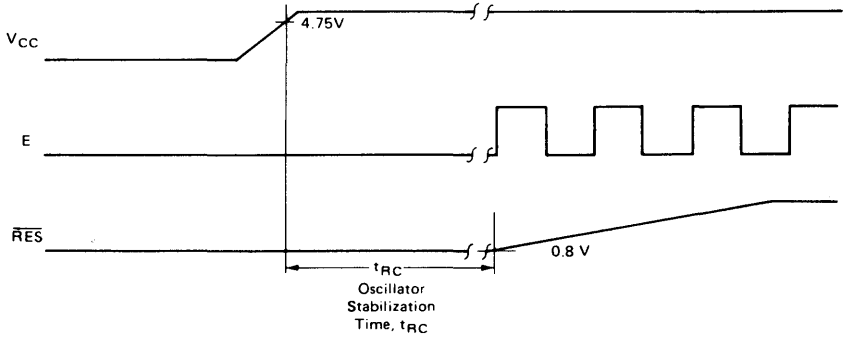
Nominal Crystal Parameter

Item \ Crystal	4 MHz	5 MHz
Co	7 pF max.	4.7 pF max.
R_S	60Ω max.	30Ω typ.



(a) Nominal Recommended Crystal Parameters





(b) Oscillator Stabilization Time (t_{RC})

Figure 15 Oscillator Characteristics

■ PORTS

There are four I/O ports on the MCU; three 8-bit ports and one 5-bit port. There are two control lines associated with one of the 8-bit ports. Each port has an associated write only Data Direction Register which allows each I/O line to be programmed to act as an input or an output. A "1" in the corresponding Data Direction Register bit will cause that I/O line to be an output. A "0" in the corresponding Data Direction Register bit will cause the I/O line to be an input. There are four ports: Port 1, Port 2, Port 3, and Port 4. Their addresses and the addresses of their Data Direction registers are given in Table 2.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

• P_{10}~P_{17} (Port 1)

Port 1 is a mode independent 8-bit I/O port where each line is an input or output as defined by its Data Direction Register. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF, Darlington transistors, or CMOS devices using external pullup resistors. It is configured as a data input port by $\overline{RES}$. Unused lines can remain unconnected.

• P_{20}~P_{24} (Port 2)

Port 2 is a mode independent 5-bit I/O port where each line is configured by its Data Direction Register. During $\overline{RES}$, all lines are configured as inputs. The TTL compatible three-state output buffers can drive one Schottky TTL load and 30 pF or CMOS devices using external pullup resistors. P_{20}, P_{21} and P_{22} must always be connected to provide the operating mode. If lines P_{23} and P_{24} are unused, they can remain unconnected.

P_{20}, P_{21}, and P_{22} provide the operating mode which is latched into the Program Control Register on the positive edge of $\overline{RES}$. The mode may be read from Port 2 Data Register as shown where PC2 is latched from pin 10.

Port 2 also provides an interface for the Serial Communications Interface and Timer. Bit 1, if configured as an output, is dedicated to the timer's Output Compare function and cannot be used to provide output from Port 2 Data Register.

Port 2 Data Register

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$0003

• P_{30}~P_{37} (Port 3)

Port 3 can be configured as an I/O port, a bidirectional 8-bit data bus, or a multiplexed address/data bus depending on the operating mode. The TTL compatible three-state output buffers can drive one Schottky TTL load and 90 pF. Unused lines can remain unconnected.

Port 3 in Single-Chip Mode

Port 3 is an 8-bit I/O port in Single-Chip Mode where each line is configured by its Data Direction Register. There are also two lines, $\overline{IS3}$ and $\overline{OS3}$, which can be used to control Port 3 data transfers.

Three Port 3 options are controlled by the Port 3 Control and Status Register and available only in Single-Chip Mode: (1) Port 3 input data can be latched using $\overline{IS3}$ as a control signal, (2) $\overline{OS3}$ can be generated by either an CPU read or write to Port 3's Data Register, and (3) an $\overline{IRQ_1}$ interrupt can be enabled by an $\overline{IS3}$ negative edge. Port 3 latch timing is shown in Figure 6.

Port 3 Control and Status Register

7	6	5	4	3	2	1	0	
$\overline{IS3}$ Flag	$\overline{IS3}$ $\overline{IRQ_1}$ Enable	X	OSS	Latch Enable	X	X	X	$000F

Bit 0~2 Not used.

Bit 3 LATCH ENABLE. This bit controls the input latch for Port 3. If set, input data is latched by an $\overline{IS3}$ negative edge. The latch is transparent after a read of Port 3's Data Register. LATCH ENABLE is cleared by $\overline{RES}$.

Bit 4 OSS (Output Strobe Select). This bit determines whether $\overline{OS3}$ will be generated by a read or write of Port 3's Data Register. When clear, the strobe is generated by a read; when set, it is generated by a write. OSS is cleared by $\overline{RES}$.

Bit 5 Not used.

Bit 6 $\overline{IS3}$ $\overline{IRQ_1}$ ENABLE. When set, an $\overline{IRQ_1}$ interrupt will be enabled whenever $\overline{IS3}$ FLAG is set; when clear, the interrupt is inhibited. This bit is cleared by $\overline{RES}$.

Bit 7 $\overline{IS3}$ FLAG. This read-only status bit is set by an $\overline{IS3}$ negative edge. It is cleared by a read of the Port 3 Control and Status Register (with $\overline{IS3}$ FLAG set) followed by a read or write to Port 3's Data Register or by $\overline{RES}$.

Port 3 in Expanded Non-Multiplexed Mode

Port 3 is configured as a bidirectional data bus (D_0~D_7) in the Expanded Non-Multiplexed Mode. The direction of data transfers is controlled by Read/Write (SC_2) and clocked by E (Enable).

Port 3 in Expanded Multiplexed Mode

Port 3 is configured as a time multiplexed address (A_0~A_7) and data bus (D_0~D_7) in Expanded Multiplexed Mode where Address Strobe (AS) can be used to demultiplex the two buses. Port 3 is held in a high impedance state between valid address and data to prevent potential bus conflicts.

• P_{40}~P_{47} (Port 4)

Port 4 is configured as an 8-bit I/O port, address outputs, or data inputs depending on the operating mode. Port 4 can drive one Schottky TTL load and 90 pF and is the only port with internal pullup resistors. Unused lines can remain unconnected.

Port 4 in Single Chip Mode

In Single Chip Mode, Port 4 functions as an 8-bit I/O port where each line is configured by its Data Direction Register.

Internal pullup resistors allow the port to directly interface with CMOS at 5 volt levels. External pullup resistors to more than 5 volts, however, cannot be used.

Port 4 in Expanded Non-Multiplexed Mode

Port 4 is configured from $\overline{RES}$ as an 8-bit input port where its Data Direction Register can be written to provide any or all of address lines, A_0 to A_7. Internal pullup resistors are intended to pull the lines high until its Data Direction Register is configured.

Port 4 in Expanded Multiplexed Mode

In all Expanded Multiplexed modes except Mode 6, Port 4 functions as half of the address bus and provides A_8 to A_{15}. In Mode 6, the port is configured from $\overline{RES}$ as an 8-bit parallel input port where its Data Direction Register can be written to provide any or all of address lines, A_8 to A_{15}. Internal pullup resistors are intended to pull the lines high until its Data Direction Register is configured where bit 0 controls A_8.

■ OPERATING MODES

The MCU provides eight different operating modes which are selectable by hardware programming and referred to as Mode 0 through Mode 7. The operating mode controls the memory map, configuration of Port 3, Port 4, SC_1, SC_2, and the physical location of interrupt vectors.

• Fundamental Modes

The MCU's eight modes can be grouped into three fundamental modes which refer to the type of bus it supports: Single Chip, Expanded Non-Multiplexed, and Expanded Multiplexed. Single chip modes include 4 and 7, Expanded Non-Multiplexed is Mode 5 and the remaining five are Expanded Multiplexed modes. Table 3 summarizes the characteristics of the operating modes.

Single Chip Modes (4, 7)

In Single-Chip Mode, the MCU's four ports are configured as parallel input/output data ports, as shown in Figure 16. The MCU functions as a monolithic microcomputer in these two modes without external address or data buses. A maximum of 29 I/O lines and two Port 3 control lines are provided. In addition to other peripherals, another MCU can be interfaced to Port 3 in a loosely coupled dual processor configuration, as shown in Figure 17.

In Single-Chip Test Mode (4), the RAM responds to $XX80 through $XXFF and the ROM is removed from the internal address map. A test program must first be loaded into the RAM using modes 0, 1, 2, or 6. If the MCU is Reset and then programmed into Mode 4, execution will begin at $XXFE: XXFF. Mode 5 can be irreversibly entered from Mode 4 without going through Reset by setting bit 5 of Port 2's Data Register. This mode is used primarily to test Ports 3 and 4 in the Single-Chip and Non-Multiplexed Modes.

Expanded Non-Multiplexed Mode (5)

A modest amount of external memory space is provided in the Expanded Non-Multiplexed Mode while retaining significant on-chip resources. Port 3 functions as an 8-bit bidirectional data bus and Port 4 is configured as an input data port. Any combination of the eight least-significant address lines may be obtained by writing to Port 4's Data Direction Register. Stated alternatively, any combination of A_0 to A_7 may be provided while retaining the remainder as input data lines. Internal pullup resistors are intended to pull Port 4's lines high until it is configured.

Figure 18 illustrates a typical system configuration in the Expanded Non-Multiplexed Mode. The MCU interfaces directly with HMCS6800 family parts and can access 256 bytes of external address space at $100 through $1FF. $\overline{IOS}$ provides an address decode of external memory ($100–$1FF) and can be used similarly to an address or chip select line.

Table 3 Summary of HD6800 Operating Modes

Common to all Modes:
Reserved Register Area
Port 1
Port 2
Programmable Timer
Serial Communication Interface
Single Chip Mode 7
128 bytes of RAM; 2048 bytes of ROM
Port 3 is a parallel I/O port with two control lines
Port 4 is a parallel I/O port
SC_1 is Input Strobe 3 ($\overline{IS3}$)
SC_2 is Output Strobe 3 ($\overline{OS3}$)
Expanded Non-Multiplexed Mode 5
128 bytes of RAM; 2048 bytes of ROM
256 bytes of external memory space
Port 3 is an 8-bit data bus
Port 4 is an input port/address bus
SC_1 is Input/Output Select ($\overline{IOS}$)
SC_2 is read/write (R/$\overline{W}$)
Expanded Multiplexed Modes 1, 2, 3, 6
Four memory space options (65k address space):
(1) No internal RAM or ROM (Mode 3)
(2) Internal RAM, no ROM (Mode 2)
(3) Internal RAM and ROM (Mode 1)
(4) Internal RAM, ROM with partial address bus (Mode 6)
Port 3 is a multiplexed address/data bus
Port 4 is an address bus (inputs/address in Mode 6)
SC_1 is Address Strobe (AS)
SC_2 is Read/Write (R/$\overline{W}$)
Test Modes 0 and 4
Expanded Multiplexed Test Mode 0
May be used to test RAM and ROM
Single Chip and Non-Multiplexed Test Mode 4
(1) May be changed to Mode 5 without going through Reset
(2) May be used to test Ports 3 and 4 as I/O ports

Expanded-Multiplexed Modes (0, 1, 2, 3, 6)

In the Expanded-Multiplexed Modes, the MCU has the ability to access a 65k bytes memory space. Port 3 functions as a time multiplexed address/data bus with address valid on the negative edge of Address Strobe (AS) and the data bus valid while E is high. In Modes 0 to 3, Port 4 provides address lines A_8 to A_{15}. In Mode 6, however, Port 4 is configured during $\overline{RES}$ as data port inputs and the Data Direction Register can be changed to provide any combination of address lines, A_8 to A_{15}. Stated alternatively, any subset of A_8 to A_{15} can be provided while retaining the remainder as input data lines. Internal pullup resistors are intended to pull Port 4's lines high until software configures the port.

Figure 19 depicts a typical configuration for the Expanded-Multiplexed Modes. Address Strobe can be used to control a transparent D-type latch to capture addresses A_0 to A_7, as shown in Figure 20. This allows Port 3 to function as a Data Bus when E is high.

In Mode 0, the Reset vector is external for the first two E-cycles after the positive edge of $\overline{RES}$ and internal thereafter. In

addition, the internal and external data buses are connected and there must be no memory map overlap to avoid potential bus conflicts. Mode 0 is used primarily to verify the ROM pattern and monitor the internal data bus with the automated test equipment.

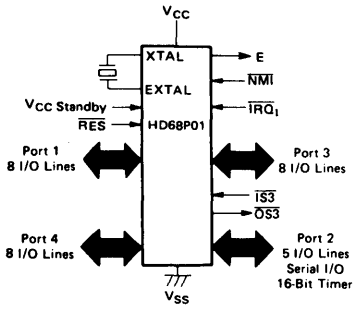


Figure 16 Single Chip Mode

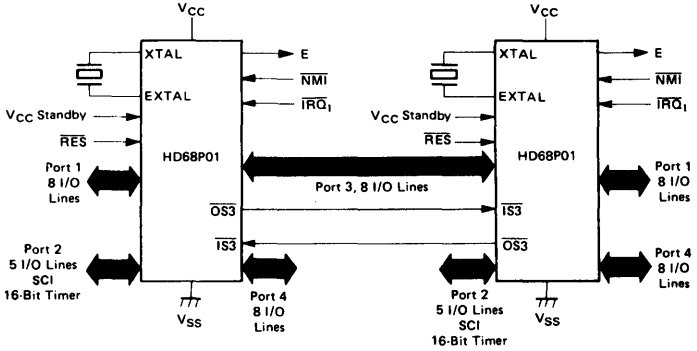
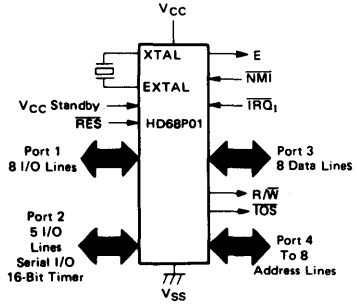


Figure 17 Single Chip Dual Processor Configuration



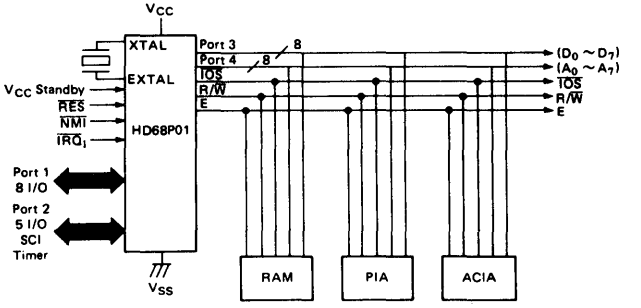
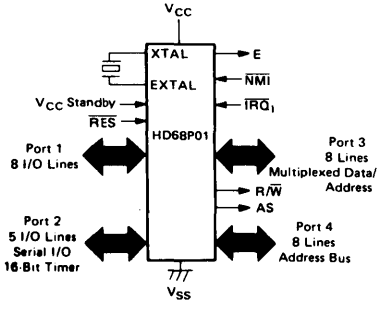


Figure 18 Expanded Non-Multiplexed Configuration



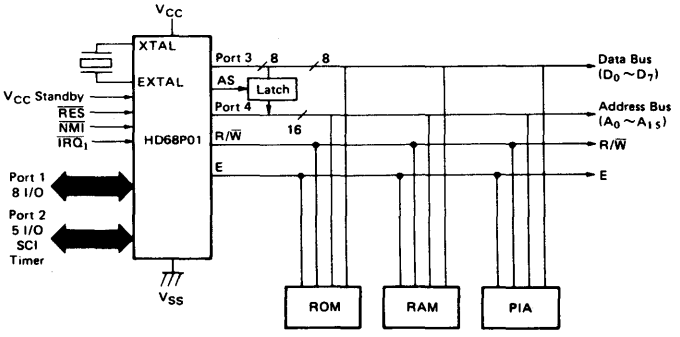
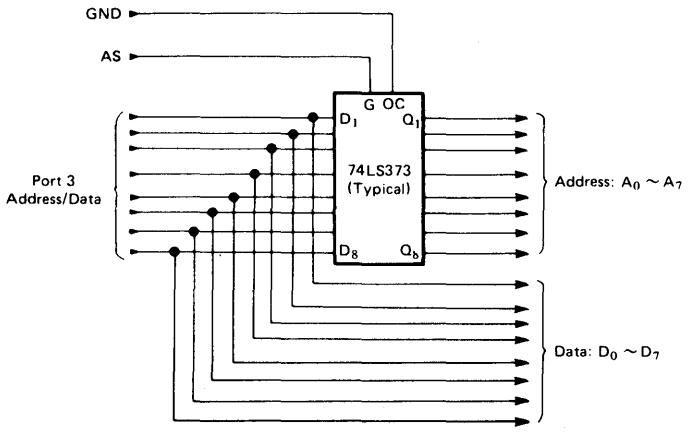


Figure 19 Expanded Multiplexed Configuration



Function Table

Output Control	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 20 Typical Latch Arrangement

- **Programming The Mode**

The operating mode is programmed by the levels asserted on P_{22}, P_{21}, and P_{20} which are latched into PC2, PC1, and PC0 of the program control register on the positive edge of $\overline{RES}$. The operating mode may be read from Port 2 Data Register as shown below, and programming levels and timing must be met as shown in Figure 8. A brief outline of the operating modes is shown in Table 4.

Port 2 Data Register

7	6	5	4	3	2	1	0	
PC2	PC1	PC0	P24	P23	P22	P21	P20	$0003

Circuitry to provide the programming levels is dependent primarily on the normal system usage of the three pins. If configured as outputs, the circuit shown in Figure 21 may be used; otherwise, three-state buffers can be used to provide isolation while programming the mode.

Table 4 Mode Selection Summary

Mode	P_{22} (PC2)	P_{21} (PC1)	P_{20} (PC0)	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(5, 6)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(5, 6)	Non-Multiplexed/Partial Decode
4	H	L	L	I(2)	I(1)	I	I	Single Chip Test
3	L	H	H	E	E	E	MUX(4)	Multiplexed /No RAM or ROM
2	L	H	L	E	I	E	MUX(4)	Multiplexed /RAM
1	L	L	H	I	I	E	MUX(4)	Multiplexed/RAM & ROM
0	L	L	L	I	I	I(3)	MUX(4)	Multiplexed Test

Legend:
I – Internal
E – External
MUX – Multiplexed
NMUX – Non-Multiplexed
L – Logic "0"
H – Logic "1"

Notes:
(1) Internal RAM is addressed at $XX80
(2) Internal ROM is disabled
(3) $\overline{RES}$ vector is external for 2 cycles after $\overline{RES}$ goes high
(4) Addresses associated with Ports 3 and 4 are considered external in Modes 0, 1, 2, and 3
(5) Addresses associated with Port 3 are considered external in Modes 5 and 6
(6) Port 4 default is user data input; address output is optional by writing to Port 4 Data Direction Register

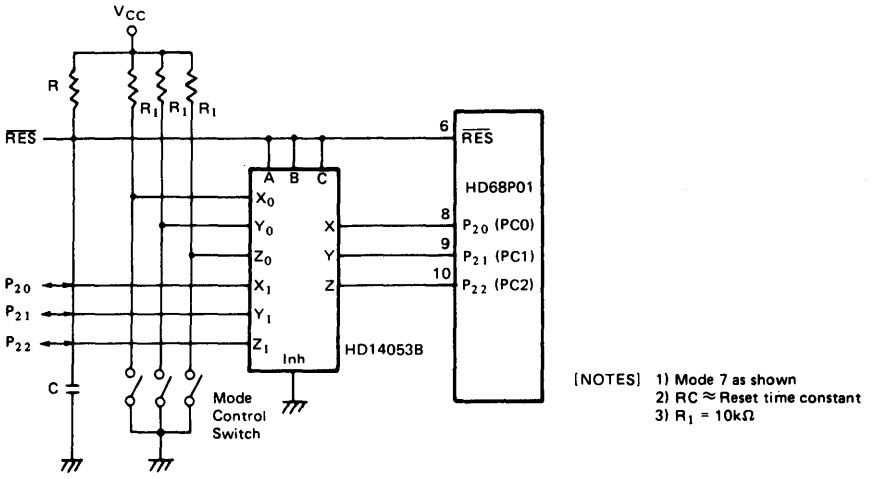
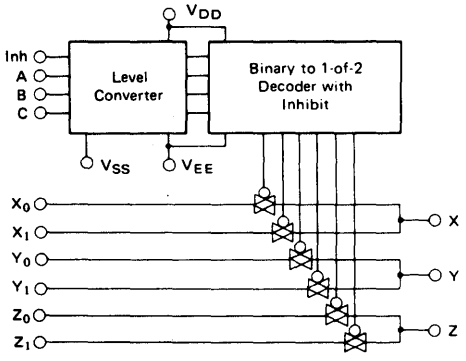


Figure 21 Recommended Circuit for Mode Selection



Truth Table

Control Input				On Switch		
Inhibit	Select					
	C	B	A	HD14053B		
0	0	0	0	Z_0	Y_0	X_0
0	0	0	1	Z_0	Y_0	X_1
0	0	1	0	Z_0	Y_1	X_0
0	0	1	1	Z_0	Y_1	X_1
0	1	0	0	Z_1	Y_0	X_0
0	1	0	1	Z_1	Y_0	X_1
0	1	1	0	Z_1	Y_1	X_0
0	1	1	1	Z_1	Y_1	X_1
1	X	X	X	–		

Figure 22 HD14053B Multiplexers/Demultiplexers

■ MEMORY MAPS

The MCU can provide up to 65k bytes address space depending on the operating mode. The HD68P01 provides 8k bytes address space for EPROM, but the maps differ in EPROM types as follows.

1) HN482732A (a 4k-byte EPROM)
   In order to support the HD6801V0, EPROM of the HD68P01V07/HD68P01V07-1 must be located at $F000-$FFFF.
2) HN482764 (a 8k-byte EPROM)
   The HD68P01M0/HD68P01M0-1 can provide up to 8k bytes address space using HN482764 instead of HN482732A. In this case, EPROM of the HD68P01M0/HD68P01M0-1 is located at $E000-$FFFF.

A memory map for each operating mode is shown in Figure 23. The first 32 locations of each map are reserved for the MCU's internal register area, as shown in Table 5, with exceptions as indicated.

Refer to "Precaution when emulating the HD6801 Family".

Table 5 Internal Register Area

Register	Address
Port 1 Data Direction Register***	00
Port 2 Data Direction Register***	01
Port 1 Data Register	02
Port 2 Data Register	03
Port 3 Data Direction Register***	04*
Port 4 Data Direction Register***	05**
Port 3 Data Register	06*
Port 4 Data Register	07**
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 control and Status Register	0F*
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External address in Modes 0, 1, 2, 3, 5, 6; cannot be accessed in Mode 5 (No $\overline{OS}$)
** External addresses in Modes 0, 1, 2, 3
*** 1 = Output, 0 = Input

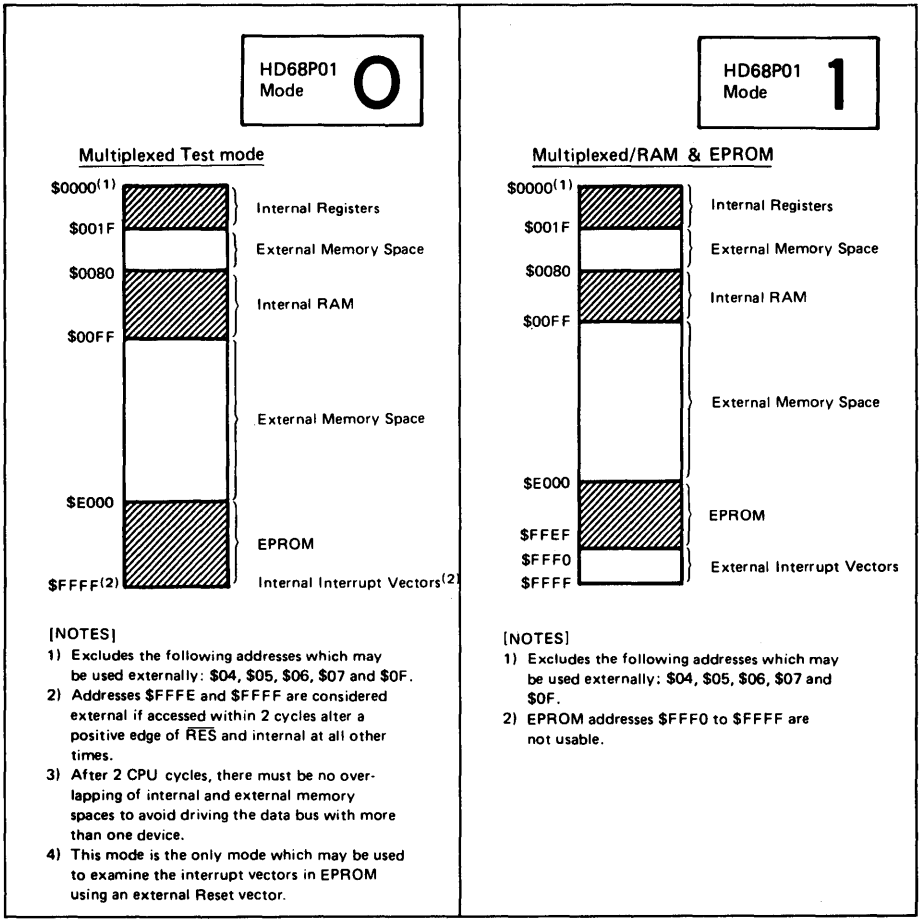


Figure 23 HD68P01 Memory Maps

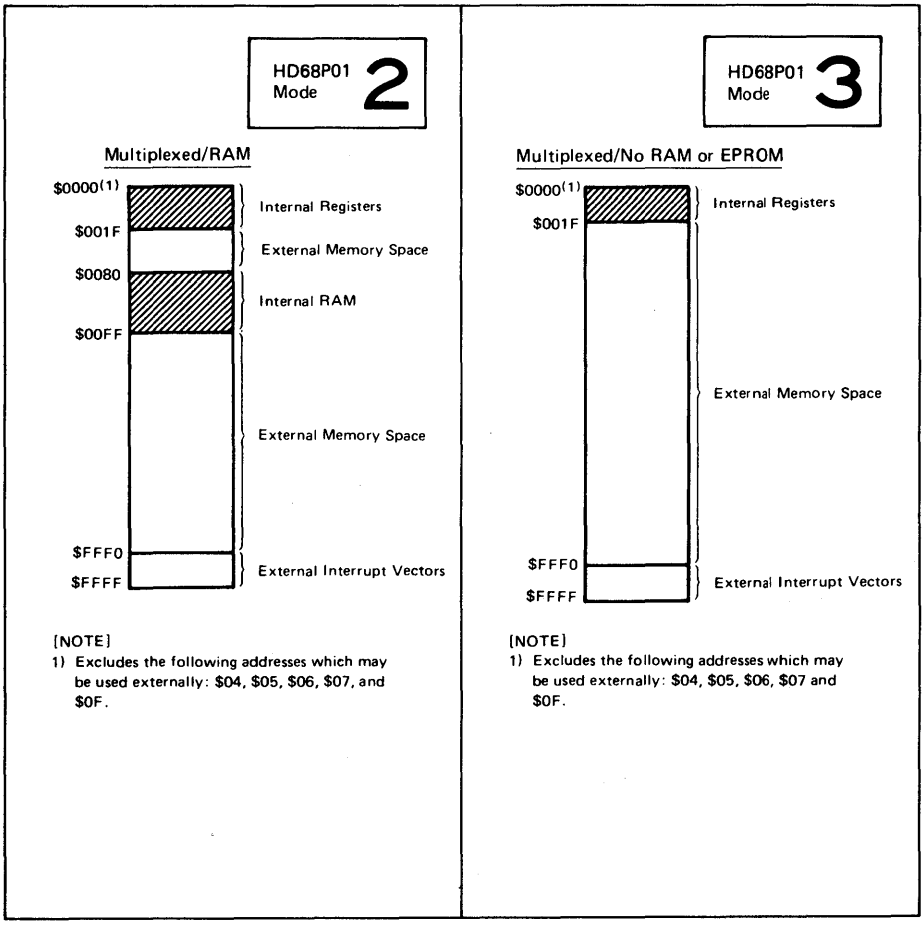


Figure 23 HD68P01 Memory Maps (Continued)

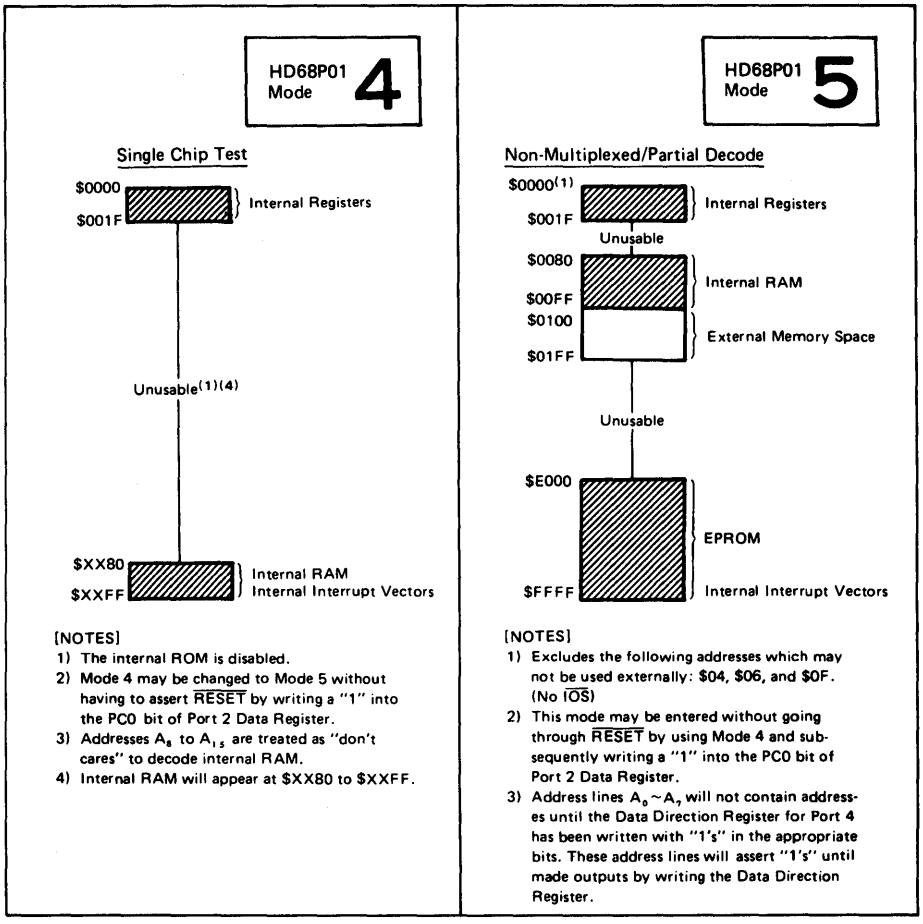


Figure 23 HD68P01 Memory Maps (Continued)

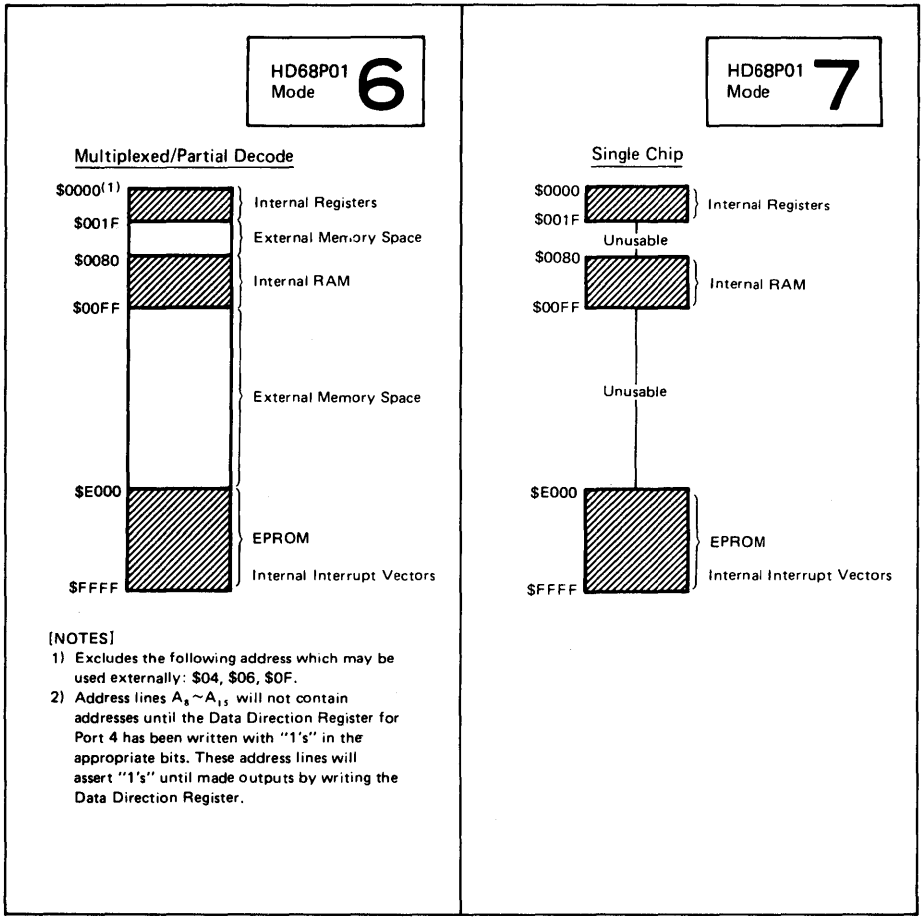


Figure 23 HD68P01 Memory Maps (Continued)

■ PROGRAMMABLE TIME

The Programmable Timer can be used to perform input waveform measurements while independently generating an output waveform. Pulse widths can vary from several microseconds to many seconds. A block diagram of the Timer is shown in Figure 24.

• Counter ($09:0A)

The key timer element is a 16-bit free-running counter which is incremented by E (Enable). It is cleared during $\overline{RES}$ and is read-only with one exception: a write to the counter ($09) will preset it to $FFF8. This feature, intended for testing, can disturb serial operations because the counter provides the SCI's internal bit rate clock. TOF is set whenever the counter contains all 1's.

• Output Compare Register ($0B:0C)

The Output Compare Register is a 16-bit Read/Write register used to control an output waveform or provide an arbitrary timeout flag. It is compared with the free-running counter on each E-cycle. When a match is found, OCF is set and OLVL is clocked to an output level register. If Port 2, bit 1, is configured as an output, OLVL will appear at P_{21} and the Output Compare Register and OLVL can then be changed for the next compare. The function is inhibited for one cycle after a write to its high byte of the Compare Resister ($0B) to ensure a valid compare. The Output Compare Register is set to $FFFF by $\overline{RES}$.

• Input Capture Register ($0D: 0E)

The Input Capture Register is a 16-bit read-only register used to store the free-running counter when a "proper" input transition occurs as defined by IEDG. Port 2, bit 0 should be configured as an input, but the edge detect circuit always senses P_{20} even when configured as an output. An input capture can occur independently of ICF: the register always contains the most current value. Counter transfer is inhibited, however, between accesses of a double byte CPU read. The input pulse width must be at least two E-cycles to ensure an input capture under all conditions.

• Timer Control and Status Register ($08)

The Timer Control and Status Register (TCSR) is an 8-bit register of which all bits are readable while bits 0~4 can be written. The three most significant bits provide the timer's status and indicate if:

- a proper level transition has been dtected,
- a match has been found between the free-running counter and the output compare register, and
- the free-running counter has overflowed.

Each of the three events can generate an $\overline{IRQ_2}$ interrupt and is controlled by an individual enable bit in the TCSR.

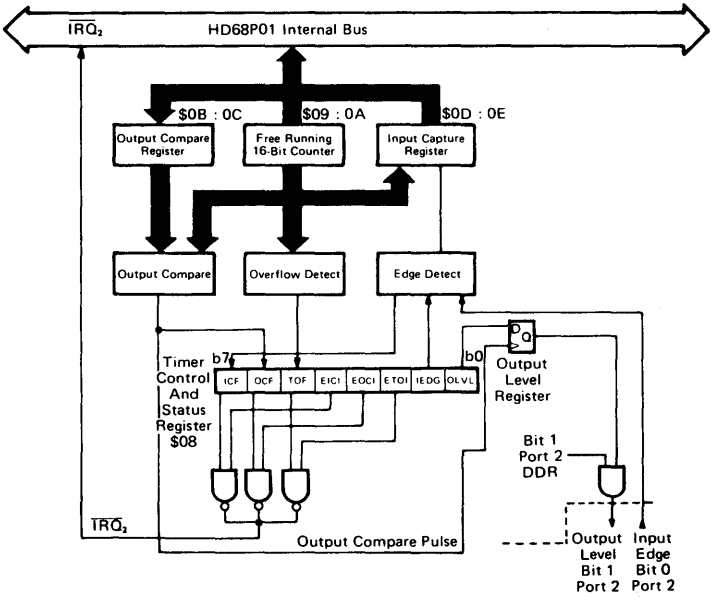


Figure 24 Block Diagram of Programmable Timer

Timer Control and Status Register (TCSR)

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

Bit 0 OLVL Output level. OLVL is clocked to the output level register by a successful output compare and will appear at P_{21} if Bit 1 of Port 2's Data Direction Register is set. It is cleared by $\overline{RES}$.

Bit 1 IEDG Input Edge. IEDG is cleared by $\overline{RES}$ and controls which level transition will trigger a counter transfer to the Input Capture Register:
IEDG = 0 Transfer on a negative-edge
IEDG = 1 Transfer on a positive-edge.

Bit 2 ETOI Enable Timer Overflow Interrupt. When set, an $\overline{IRQ_2}$ interrupt is enabled for a timer overflow; when clear, the interrupt is inhibited. It is cleared by $\overline{RES}$.

Bit 3 EOCI Enable Output Compare Interrupt. When set, an $\overline{IRQ_2}$ interrupt is enabled for an output compare; when clear, the interrupt is inhibited. It is cleared by $\overline{RES}$.

Bit 4 EICI Enable Input Capture Interrupt. When set, an $\overline{IRQ_2}$ interrupt is enabled for an input capture; when clear, the interrupt is inhibited. It is cleared by $\overline{RES}$.

Bit 5 TOF Timer Overflow Flag. TOF is set when the counter contains $FFFF. It is cleared by reading the TCSR (with TOF set) followed by the counter's high byte ($09), or by $\overline{RES}$.

Bit 6 OFC Output Compare Flag. OCF is set when the Output Compare Register matches the free-running counter. It is cleared by reading the TCSR (with OCF set) and then writing to the Output Compare Register ($0B or $0C), or by $\overline{RES}$.

Bit 7 ICF Input Capture Flag. ICF is set to indicate a proper level transition; it is cleared by reading the TCSR (with ICF set) and then the Input Capture Register High Byte ($0D), or by $\overline{RES}$.

■ SERIAL COMMUNICATIONS INTERFACE (SCI)

A full-duplex asynchronous Serial Communications Interface (SCI) is provided with a data format and a variety of rates. The SCI transmitter and receiver are functionally independent, but use the same data format and bit rate. Serial data format is standard mark/space (NRZ) and provides one start bit, eight data bits, and one stop bit. "Baud" and "bit rate" are used synonymously in the following description.

• Wake-Up Feature

In a typical serial loop multi-processor configuration, the software protocol will usually identify the addresse(s) at the beginning of the message. In order to permit uninterested MCU's to ignore the remainder of the message, a wake-up feature is included whereby all further SCI receiver flag (and interrupt) processing can be inhibited until its data line goes idle. An SCI receiver is re-enabled by an idle string of ten consecutive 1's or by $\overline{RES}$. Software must provide for the required idle string between consecutive messages and prevent it within messages.

• Programmable Options

The following features of the SCI are programmable:
- format: Standard mark/space (NRZ)
- clock: external or internal bit rate clock
- Baud (or bit rate): one of 4 per E-clock frequency, or external bit rate (X8) input
- wake-up feature: enabled or disabled
- interrupt requests: enabled individually for transmitter and receiver
- clock output: internal bit rate clock enabled or disabled to P_{22}
- Port 2 (bit 3, 4): dedicated or not dedicated to serial I/O individually for transmitter and receiver.

• Serial Communications Registers

The Serial Communications Interface includes four addressable registers as depicted in Figure 25. It is controlled by the Rate and Mode Control Register and the Transmit/Receive Control and Status Register. Data is transmitted and received utilizing a write-only Transmit Register and a read-only Receive Register. The shift registers are not accessible to software.

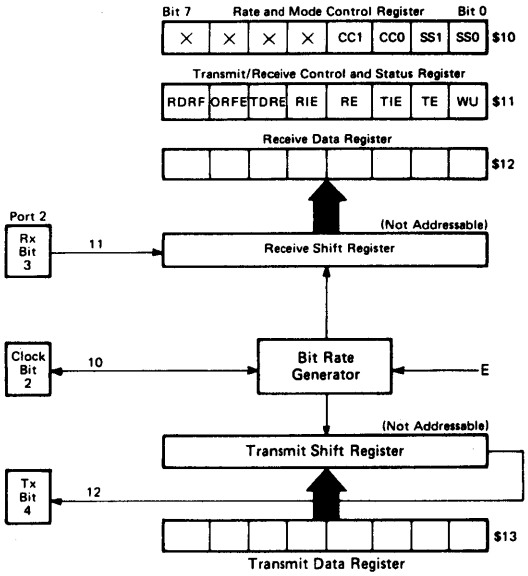


Figure 25 SCI Registers

Rate and Mode Control Register (RMCR) ($10)

The Rate and Mode Control Register controls the SCI bit rate, format, clock source, and under certain conditions, the configuration of P_{22}. The register consists of four write-only bits which are cleared by $\overline{RES}$. The two least significant bits control the bit rate of the internal clock and the remaining two bits control the format and clock source.

Rate and Mode Control Register (RMCR)

7	6	5	4	3	2	1	0	
X	X	X	X	CC1	CC0	SS1	SS0	$0010

Bit 1: Bit 0 SS1: SS0 Speed Select. These two bits select the Baud when using the internal clock. Four rates may be selected which are a function of the MCU input frequency. Table 6 lists bit time and rates for three selected MCU frequencies.

Bit 3: Bit 2 CC1:CC0 Clock Control Select. These two bits select the serial clock source. If CC1 is set, the DDR value for P_{22} is forced to the complement of CC0 and cannot be altered until CC1 is cleared. If CC1 is cleared after having been set, its DDR value is unchanged. Table 7 defines the clock source, and use of P_{22}.

If both CC1 and CC0 are set, an external TTL compatible clock must be connected to P_{22} at eight times (8X) the desired bit rate, but not greater than E, with a duty cycle of 50% (± 10%). If CC1:CC0 = 10, the internal bit rate clock is provided at P_{22} regardless of the values for TE or RE.

(Note) The source of SCI internal bit rate clock is the timer's free running counter. An CPU write to the counter can disturb serial operations.

Transmit/Receive Control and Status Register (TRCSR) ($11)

The Transmit/Receive Control and Status Register controls the transmitter, receiver, wake-up feature, and two individual interrupts and monitors the status of serial operations. All eight bits are readable while bits 0 to 4 are also writable. The register is initialized to $20 by $\overline{RES}$.

Transmit/Receive Control and Status Register (TRCSR)

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU "Wake-up" on Idle Line. When set, WU enables the wake-up function; it is cleared by ten consecutive 1's or by $\overline{RES}$. WU will not set if the line is idle.

Bit 1 TE Transmit Enable. When set, P_{24} DDR bit is set, cannot be changed, and will remain set if TE is subsequently cleared. When TE is changed from clear to set, the transmitter is connected to P_{24} and a preamble of nine consecutive 1's is transmitted. TE is cleared by $\overline{RES}$.

Bit 2 TIE Transmit Interrupt Enable. When set, an $\overline{IRQ_2}$ interrupt is enabled when TDRE is set; when clear, the interrupt is inhibited. TE is cleared by $\overline{RES}$.

Bit 3 RE Receive Enable. When set, P_{23}'s DDR bit is cleared, cannot be changed, and will remain clear if RE is subsequently cleared. While RE is set, the SCI receiver is enabled. RE is cleared by $\overline{RES}$.

Bit 4 RIE Receiver Interrupt Enable. When set, an $\overline{IRQ_2}$ interrupt is enabled when RDRF and/or ORFE is set; when clear, the interrupt is inhibited. RIE is cleared by $\overline{RES}$.

Bit 5 TDRE Transmit Data Register Empty. TDRE is set when the Transmit Data Register is transferred to the output serial shift register or by $\overline{RES}$. It is cleared by reading the TRCSR (with TDRE set) and then writing to the Transmit Data Register. Additional data will be transmitted only if TDRE has been cleared.

Bit 6 ORFE Overrun Framing Error. If set, ORFE indicates either an overrun or framing error. An overrun is a new byte ready to transfer to the Receiver Data Register with RDRF still set. A receiver framing error has occurred when the byte boundaries of the bit stream are not synchronized to the bit counter. An overrun can be distinguished from a framing error by the value of RDRF: if RDRF is set, then an overrun has occurred; otherwise a framing error has been detected. Data is not transferred to the Receive Data Register in an overrun or framing error condition. ORFE is cleared by reading the TRCSR (with ORFE set) then the Receive Data Register, or by $\overline{RES}$.

Bit 7 RDRF Receive Data Register Full. RDRF is set when the input serial shift register is transferred to the Receive Data Register. It is cleared by reading the TRCSR (with RDRF set), and then the Receive Data Register, or by $\overline{RES}$.

Table 6 SCI Bit Times and Rates

SS1 : SS0	XTAL E	2.4576 MHz 614.4 kHz	4.0 MHz 1.0 MHz	4.9152 MHz* 1.2288 MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800 Baud
0 1	E ÷ 128	208μs/4,800 Baud	128μs/7812.5 Baud	104.2 μs/9,600 Baud
1 0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud	833.3 μs/1,200 Baud
1 1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.33 ms/300 Baud

* HD68P01V07-1, HD68P01M0-1 only

Table 7 SCI Format and Clock Source Control

CC1 : CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used	**	**
1 0	NRZ	Internal	Output*	**	**
1 1	NRZ	External	Input	**	**

* Clock output is available regardless of values for bits RE and TE.

** Bit 3 is used for serial input if RE = "1" in TRCS; bit 4 is used for serial output if TE = "1" in TRCS.

- **Internally Generated Clcok**

If the user wishes for the serial I/O to furnish a clock, the following requirements are applicable:

- the values of RE and TE are immaterial.
- CC1, CC0 must be set to 10
- the maximum clock rate will be E ÷ 16.
- the clock will be at 1X the bit rate and will have a rising edge at mid-bit.

- **Externally Generated Clock**

If the user wishes to provide an external clock for the serial I/O, the following requirements are applicable:

- the CC1, CC0, field in the Rate and Mode Control Register must be set to 11,
- the external clock must be set to 8 times (X8) the desired baud rate and
- the maximum external clock frequency is 1.0 MHz.

- **Serial Operations**

The SCI is initialized by writing control bytes first to the Rate and Mode Control Register and then to the Transmit/Receive Control and Status Register.

The Transmitter Enable (TE) and Receiver Enable (RE) bits may be left set for dedicated operations.

Transmit operations

The transmit operation is enabled by TE in the Transmit/Receive Control and Status Register. When TE is set, the output of the transmit serial shift register is connected to P_{24} and the serial output by first transmitting to a ten-bit preamble of 1's. Following the preamble, internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) if the Transmit Data Register is empty (TDRE = 1), a continuous string of ones will be sent indicating an idle line, or,
2) if a byte has been written to the Transmit Data Register (TDRE = 0), it is transferred to the output serial shift register and transmission will begin.

During the transfer itself, the start bit (0) is first transmitted. Then the 8 data bits (beginning with bit 0) followed by the stop bit (1), are transmitted. When the Transmitter Data Register has been emptied, the TDRE flag bit is set.

If the MCU fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the parallel data register to the serial output register should occur) then a 1 will be sent (instead of a 0) at "Start" bit time, followed by more 1's until more data is supplied to the data register. No 0's will be sent while TDRE remains a 1.

Receive Operations

The receive operation is enabled by RE which configures P_{23}. The receive operation is controled by the contents of the Transmit/Receive Control and Status Register and the Rate and Mode Control Register.

The receiver bit interval is divided into 8 sub-intervals for internal synchronization. In the NRZ Mode, the received bit stream is synchronized by the first 0 (space) encountered.

The approximate center of each bit time is strobed during the next 10 bits. If the tenth bit is not a 1 (stop bit) a framing error is assumed, and ORFE is set. If the tenth bit is a 1, the data is transferred to the Receive Data Register, and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the MCU responds to either flag (RDRF or ORFE) by reading the status register followed by reading the Data Register, RDRF (or ORFE) will be cleared.

■ INSTRUCTION SET

The HD68P01 is upward source and object code compatible with the HD6800. Execution times of key instructions have been reduced and several new instructions have been added, including hardware multiply. A list of new operations added to the HD6800 instruction set is shown in Table 8.

In addition, two new special opcodes, 4E and 5E, are provided for test purposes. These opcodes force the Program Counter to increment like a 16-bit counter, causing address lines used in the expanded modes to increment until the device is reset. These opcodes have no mnemonics.

Table 8 New Instructions

Instruction	Description
ABX	Unsigned addition of Accumulator B to Index Register
ADDD	Adds (without carry) the double accumulator to memory and leaves the sum in the double accumulator
ASLD	Shifts the double accumulator left (towards MSB) one bit; the LSB is cleared and the MSB is shifted into the C-bit
BRN	Branch Never
LDD	Loads double accumulator from memory
LSRD	Shifts the double accumulator right (towards LSB) one bit; the MSB is cleared and the LSB is shifted into the C-bit
MUL	Unsigned multiply; multiplies the two accumulators and leaves the product in the double accumulator
PSHX	Pushes the Index Register to stack
PULX	Pulls the Index Register from stack
STD	Stores the double accumulator to memory
SUBD	Subtracts memory from the double accumulator and leaves the difference in the double accumulator

- **Programming Model**

A programming model for the HD68P01 is shown in Figure 10. Accumulator A can be concatenated with accumulator B and jointly referred to as accumulator D where A is the most significant byte. Any operation which modifies the double accumulator will also modify accumulator A and/or B. Other registers are defined as follows:

Program Counter

The program counter is a 16-bit register which always points to the next instruction.

Stack Pointer

The stack pointer is a 16-bit register which contains the address of the next available location in a pushdown/pullup (LIFO) queue. The stack resides in random access memory at a location defined by the programmer.

Index Register

The Index Register is a 16-bit register which can be used to store data or provide an address for the indexed mode of addressing.

Accumulators

The CPU contains two 8-bit accumulators, A and B, which are used to store operands and results from the arithmetic logic unit (ALU). They can also be concatenated and referred to as the D (double) accumulator.

Condition Code Registers

The condition code register indicates the results of an instruction and includes the following five condition bits: Negative (N), Zero (Z), Overflow (V), Carry/Borrow from MSB (C), and Half Carry from bit 3 (H). These bits are testable by the conditional branch instruction. Bit 4 is the interrupt mask (I-bit) and inhibits all maskable interrupts when set. The two unused bits, b6 and b7 are read as ones.

- **Addressing Modes**

The CPU provides six addressing modes which can be used to reference memory. A summary of addressing modes for all instructions is presented in Table 9, 10, 11, and 12 where execution times are provided in E-cycles. Instruction execution times are summarized in Table 13. With an input frequency of 4 MHz, E-cycles are equivalent to microseconds. A cycle-by-cycle description of bus activity for each instruction is provided in Table 14 and a description of selected instructions is shown in Figure 26.

Immediate Addressing

The operand or "immediate byte(s)" is contained in the following byte(s) of the instruction where the number of bytes matches the size of the register. These are two or three byte instructions.

Direct Addressing

The least significant byte of the operand address is contained in the second byte of the instruction and the most significant byte is assumed to be $00. Direct addressing allows the user to access $00 through $FF using two-byte instructions and execution time is reduced by eliminating the additional memory access. In most applications, the 256-byte area is reserved for frequently referenced data.

Extended Addressing

The second and third bytes of the instruction contain the absolute address of the operand. These are three byte instructions.

Indexed Addressing

The unsigned offset contained in the second byte of the instruction is added with carry to the Index Register and used to reference memory without changing the Index Register. These are two byte instructions.

Table 9 Index Register and Stack Manipulation Instructions

Pointer Operations	Mnemonic	Immed			Direct			Index			Extend			Implied			Boolean/ Arithmetic Operation	Cond. Code Reg.					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5 H	4 I	3 N	2 Z	1 V	0 C
Compare Index Reg	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X − M: M + 1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	3	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	3	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	3	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	↕	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	↕	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	↕	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M+1)	•	•	↕	↕	R	•
Index Reg → Stack Pntr	TXS													35	3	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	3	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	4	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	5	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•

The Condition Code Register notes are listed after Table 12.

Implied Addressing

The operand(s) are registers and no memory reference is required. These are single byte instructions.

Relative Addressing

Relative addressing is used only for branch instructions. If the branch condition is true, the Program Counter is overwritten with the sum of a signed single byte displacement in the second byte of the instruction and the current Program Counter. This provides a branch range of −126 to 129 bytes from the first byte of the instruction. These are two byte instructions.

Table 10 Accumulator and Memory Instructions

Accumulator and Memory Operations	Mnemonic	Immed			Direct			Index			Extend			Implied			Boolean Expression	Cond. Code Reg.					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	ASLA													48	2	1		•	•	↕	↕	↕	↕
	ASLB													58	2	1		•	•	↕	↕	↕	↕
Shift Left Dbl	ASLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•	↕	↕	↕	↕
	ASRA													47	2	1		•	•	↕	↕	↕	↕
	ASRB													57	2	1		•	•	↕	↕	↕	↕
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	•	•	↕	↕	R	•
Compare Acmltrs	CBA													11	2	1	A - B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Decimal Adj, A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A - 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B - 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•	↕	↕	↕	↕
	LSLA													48	2	1		•	•	↕	↕	↕	↕
	LSLB													58	2	1		•	•	↕	↕	↕	↕
	LSLD													05	3	1		•	•	↕	↕	↕	↕
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R	↕	↕	↕
	LSRA													44	2	1		•	•	R	↕	↕	↕
	LSRB													54	2	1		•	•	R	↕	↕	↕
	LSRD													04	3	1		•	•	R	↕	↕	↕

(Continued)

Table 10 Accumulator and Memory Instructions (Continued)

Accumulator and Memory Operations	Mnemonic	Immed			Direct			Index			Extend			Implied			Boolean Expression	Cond. Code Reg.				
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Multiply	MUL													3D	10	1	A X B → D	•	•	•	•	•	‡
2's Complement (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	‡	‡	‡	‡
	NEGA													40	2	1	00 - A → A	•	•	‡	‡	‡	‡
	NEGB													50	2	1	00 - B → B	•	•	‡	‡	‡	‡
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	‡	‡	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	‡	‡	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	‡	‡	‡	‡
	ROLA													49	2	1		•	•	‡	‡	‡	‡
	ROLB													59	2	1		•	•	‡	‡	‡	‡
Rotate Right	ROR							66	6	2	76	6	3					•	•	‡	‡	‡	‡
	RORA													46	2	1		•	•	‡	‡	‡	‡
	RORB													56	2	1		•	•	‡	‡	‡	‡
Subtract Acmltr	SBA													10	2	1	A - B → A	•	•	‡	‡	‡	‡
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	‡	‡	‡	‡
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	‡	‡	‡	‡
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	‡	‡	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	‡	‡	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	•	•	‡	‡	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	‡	‡	‡	‡
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	‡	‡	‡	‡
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	•	•	‡	‡	‡	‡
Transfer Acmltr	TAB													16	2	1	A → B	•	•	‡	‡	R	•
	TBA													17	2	1	B → A	•	•	‡	‡	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	•	•	‡	‡	R	R
	TSTA													4D	2	1	A - 00	•	•	‡	‡	R	R
	TSTB													5D	2	1	B - 00	•	•	‡	‡	R	R

The Condition Code Register notes are listed after Table 12.

Table 11 Jump and Branch Instructions

Operations	Mnemonic	Direct			Relative			Index			Extend			Implied			Branch Test	Cond. Code Reg.					
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		5	4	3	2	1	0
																		H	I	N	Z	V	C
Branch Always	BRA				20	3	2										None	•	•	•	•	•	•
Branch Never	BRN				21	3	2										None	•	•	•	•	•	•
Branch If Carry Clear	BCC				24	3	2										C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS				25	3	2										C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ				27	3	2										Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE				2C	3	2										N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT				2E	3	2										Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI				22	3	2										C + Z = 0	•	•	•	•	•	•
Branch If Higher or Same	BHS				24	3	2										C = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE				2F	3	2										Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Carry Set	BLO				25	3	2										C = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS				23	3	2										C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT				2D	3	2										N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI				2B	3	2										N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE				26	3	2										N = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC				28	3	2										V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS				29	3	2										V = 1	•	•	•	•	•	•
Branch If Plus	BPL				2A	3	2										N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR				8D	6	2											•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR	9D	5	2				AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		‡	‡	‡	‡	‡	‡
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait For Interrupt	WAI													3E	9	1		•	•	•	•	•	•

The Condition Code Register notes are listed after Table 12.

Table 12 Condition Code Register Manipulation Instructions

Operations	Implied				Boolean Operation	Cond. Code Reg.					
	Mnemonic	OP	~	#		5	4	3	2	1	0
						H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	2	1	A → CCR	‡	‡	‡	‡	‡	‡
CCR → Accumulator A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MPU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
- Arithmetic Minus
• Boolean AND
X Arithmetic Multiply
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer Into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from MSB
R Reset Always
S Set Always
‡ Affected
• Not Affected

Table 13 Instruction Execution Times in E-Cycles

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Implied	Relative
ABA	•	•	•	•	2	•
ABX	•	•	•	•	3	•
ADC	2	3	4	4	•	•
ADD	2	3	4	4	•	•
ADDD	4	5	6	6	•	•
AND	2	3	4	4	•	•
ASL	•	•	6	6	•	•
ASLD	•	•	•	•	3	•
ASR	•	•	6	6	•	•
BCC	•	•	•	•	•	3
BCS	•	•	•	•	•	3
BEQ	•	•	•	•	•	3
BGE	•	•	•	•	•	3
BGT	•	•	•	•	•	3
BHI	•	•	•	•	•	3
BHS	•	•	•	•	•	3
BIT	2	3	4	4	•	•
BLE	•	•	•	•	•	3
BLO	•	•	•	•	•	3
BLS	•	•	•	•	•	3
BLT	•	•	•	•	•	3
BMI	•	•	•	•	•	3
BNE	•	•	•	•	•	3
BPL	•	•	•	•	•	3
BRA	•	•	•	•	•	3
BRN	•	•	•	•	•	3
BSR	•	•	•	•	•	6
BVC	•	•	•	•	•	3
BVS	•	•	•	•	•	3
CBA	•	•	•	•	2	•
CLC	•	•	•	•	2	•
CLI	•	•	•	•	2	•
CLR	•	•	6	6	•	•
CLV	•	•	•	•	2	•
CMP	2	3	4	4	•	•
COM	•	•	6	6	2	•
CPX	4	5	6	6	•	•
DAA	•	•	•	•	2	•
DEC	•	•	6	6	•	•
DES	•	•	•	•	3	•
DEX	•	•	•	•	3	•
EOR	2	3	4	4	•	•
INC	•	•	6	6	•	•
INS	•	•	•	•	3	•

	Addressing Mode					
	Immediate	Direct	Extended	Indexed	Implied	Relative
INX	•	•	•	•	3	•
JMP	•	•	3	3	•	•
JSR	•	5	6	6	•	•
LDA	2	3	4	4	•	•
LDD	3	4	5	5	•	•
LDS	3	4	5	5	•	•
LDX	3	4	5	5	•	•
LSR	•	•	6	6	•	•
LSRD	•	•	•	•	3	•
MUL	•	•	•	•	10	•
NEG	•	•	6	6	•	•
NOP	•	•	•	•	2	•
ORA	2	3	4	4	•	•
PSH	•	•	•	•	•	•
PSHX	•	•	•	•	4	•
PUL	•	•	•	•	•	•
PULX	•	•	•	•	5	•
ROL	•	•	6	6	•	•
ROR	•	•	6	6	•	•
RTI	•	•	•	•	10	•
RTS	•	•	•	•	5	•
SBA	•	•	•	•	2	•
SBC	2	3	4	4	•	•
SEC	•	•	•	•	2	•
SEI	•	•	•	•	2	•
SEV	•	•	•	•	2	•
STA	•	3	4	4	•	•
STD	•	4	5	5	•	•
STS	•	4	5	5	•	•
STX	•	4	5	5	•	•
SUB	2	3	4	4	•	•
SUBD	4	5	6	6	•	•
SWI	•	•	•	•	12	•
TAB	•	•	•	•	2	•
TAP	•	•	•	•	2	•
TBA	•	•	•	•	2	•
TPA	•	•	•	•	2	•
TST	•	•	6	6	•	•
TSX	•	•	•	•	3	•
TXS	•	•	•	•	3	•
WAI	•	•	•	•	9	•

■ SUMMARY OF CYCLE BY CYCLE OPERATION

Table 14 provides a detailed description of the information present on the Address Bus, Data Bus, and the Read/Write (R/$\overline{W}$) line during each cycle of each instruction.

The information is useful in comparing actual with expected results during debug to both software and hardware as the program is executed. The information is categorized in groups according to addressing mode and number of cycles per instruction. In general, instructions with the same addressing mode and number of cycles execute in the same manner. Exceptions are indicated in the table.

Note that during MCU reads of internal locations, the resultant value will not appear on the external Data Bus except in Mode 0. "High order" byte refers to the most significant byte of a 16-bit value.

Table 14 Cycle by Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMMEDIATE					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Operand Data
LDS LDX LDD	3	1 2 3	Op Code Address Op Code Address + 1 Op Code Address + 2	1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte)
CPX SUBD ADDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Op Code Address + 2 Address Bus FFFF	1 1 1 1	Op Code Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
DIRECT					
ADC EOR ADD LDA AND ORA BIT SBC CMP SUB	3	1 2 3	Op Code Address Op Code Address + 1 Address of Operand	1 1 1	Op Code Address of Operand Operand Data
STA	3	1 2 3	Op Code Address Op Code Address + 1 Destination Address	1 1 0	Op Code Destination Address Data from Accumulator
LDS LDX LDD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Operand Address + 1	1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte)
STS STX STD	4	1 2 3 4	Op Code Address Op Code Address + 1 Address of Operand Address of Operand + 1	1 1 0 0	Op Code Address of Operand Register Data (High Order Byte) Register Data (Low Order Byte)
CPX SUBD ADDD	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Operand Address Operand Address + 1 Address Bus FFFF	1 1 1 1 1	Op Code Address of Operand Operand Data (High Order Byte) Operand Data (Low Order Byte) Low Byte of Restart Vector
JSR	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Subroutine Address Stack Pointer Stack Pointer + 1	1 1 1 0 0	Op Code Irrelevant Data First Subroutine Op Code Return Address (Low Order Byte) Return Address (High Order Byte)

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
EXTENDED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Jump Address (High Order Byte)
		3	Op Code Address + 2	1	Jump Address (Low Order Byte)
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
AND ORA		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
BIT SBC		4	Address of Operand	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Destination Address (High Order Byte)
		3	Op Code Address + 2	1	Destination Address (Low Order Byte)
		4	Operand Destination Address	0	Data from Accumulator
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
LDD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	1	Operand Data (High Order Byte)
		5	Address of Operand + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
STD		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
		4	Address of Operand	0	Operand Data (High Order Byte)
		5	Address of Operand + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Address of Operand (High Order Byte)
CLR ROL		3	Op Code Address + 2	1	Address of Operand (Low Order Byte)
COM ROR		4	Address of Operand	1	Current Operand Data
DEC TST*		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Address of Operand	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Operand Address (High Order Byte)
ADDD		3	Op Code Address + 2	1	Operand Address (Low Order Byte)
		4	Operand Address	1	Operand Data (High Order Byte)
		5	Operand Address + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Address of Subroutine (High Order Byte)
		3	Op Code Address + 2	1	Address of Subroutine (Low Order Byte)
		4	Subroutine Starting Address	1	Op Code of Next Instruction
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

* In the TST instruction, the line condition of the sixth cycle does the following: R/$\overline{W}$ = "High", AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
INDEXED					
JMP	3	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
ADC EOR	4	1	Op Code Address	1	Op Code
ADD LDA		2	Op Code Address + 1	1	Offset
AND ORA		3	Address Bus FFFF	1	Low Byte of Restart Vector
BIT SBC		4	Index Register Plus Offset	1	Operand Data
CMP SUB					
STA	4	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data
LDS	5	1	Op Code Address	1	Op Code
LDX		2	Op Code Address + 1	1	Offset
LDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	1	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	1	Operand Data (Low Order Byte)
STS	5	1	Op Code Address	1	Op Code
STX		2	Op Code Address + 1	1	Offset
STD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register Plus Offset	0	Operand Data (High Order Byte)
		5	Index Register Plus Offset + 1	0	Operand Data (Low Order Byte)
ASL LSR	6	1	Op Code Address	1	Op Code
ASR NEG		2	Op Code Address + 1	1	Offset
CLR ROL		3	Address Bus FFFF	1	Low Byte of Restart Vector
COM ROR		4	Index Register Plus Offset	1	Current Operand Data
DEC TST *		5	Address Bus FFFF	1	Low Byte of Restart Vector
INC		6	Index Register Plus Offset	0	New Operand Data
CPX	6	1	Op Code Address	1	Op Code
SUBD		2	Op Code Address + 1	1	Offset
ADDD		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	Operand Data (High Order Byte)
		5	Index Register + Offset + 1	1	Operand Data (Low Order Byte)
		6	Address Bus FFFF	1	Low Byte of Restart Vector
JSR	6	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Offset
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Index Register + Offset	1	First Subroutine Op Code
		5	Stack Pointer	0	Return Address (Low Order Byte)
		6	Stack Pointer – 1	0	Return Address (High Order Byte)

* In the TST instruction, the line condition of the sixth cycle does the following: R/$\overline{W}$ = "High", AB = FFFF, DB = Low Byte of Reset Vector.

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
IMPLIED					
ABA DAA SEC ASL DEC SEI ASR INC SEV CBA LSR TAB CLC NEG TAP CLI NOP TBA CLR ROL TPA CLV ROR TST COM SBA	2	1 2	Op Code Address Op Code Address + 1	1 1	Op Code Op Code of Next Instruction
ABX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
ASLD LSRD	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Irrelevant Data Low Byte of Restart Vector
DES INS	3	1 2 3	Op Code Address Op Code Address + 1 Previous Register Contents	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
INX DEX	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PSHA PSHB	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 0	Op Code Op Code of Next Instruction Accumulator Data
TSX	3	1 2 3	Op Code Address Op Code Address + 1 Stack Pointer	1 1 1	Op Code Op Code of Next Instruction Irrelevant Data
TXS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Op Code of Next Instruction Low Byte of Restart Vector
PULA PULB	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 1 1	Op Code Op Code of Next Instruction Irrelevant Data Operand Data from Stack
PSHX	4	1 2 3 4	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1	1 1 0 0	Op Code Irrelevant Data Index Register (Low Order Byte) Index Register (High Order Byte)
PULX	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer +2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Index Register (High Order Byte) Index Register (Low Order Byte)
RTS	5	1 2 3 4 5	Op Code Address Op Code Address + 1 Stack Pointer Stack Pointer + 1 Stack Pointer + 2	1 1 1 1 1	Op Code Irrelevant Data Irrelevant Data Address of Next Instruction (High Order Byte) Address of Next Instruction (Low Order Byte)

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instruction	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
WAI **	9	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Op Code of Next Instruction
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
MUL	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Address Bus FFFF	1	Low Byte of Restart Vector
		4	Address Bus FFFF	1	Low Byte of Restart Vector
		5	Address Bus FFFF	1	Low Byte of Restart Vector
		6	Address Bus FFFF	1	Low Byte of Restart Vector
		7	Address Bus FFFF	1	Low Byte of Restart Vector
		8	Address Bus FFFF	1	Low Byte of Restart Vector
		9	Address Bus FFFF	1	Low Byte of Restart Vector
		10	Address Bus FFFF	1	Low Byte of Restart Vector
RTI	10	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	1	Irrelevant Data
		4	Stack Pointer + 1	1	Contents of Cond. Code Reg. from Stack
		5	Stack Pointer + 2	1	Contents of Accumulator B from Stack
		6	Stack Pointer + 3	1	Contents of Accumulator A from Stack
		7	Stack Pointer + 4	1	Index Register from Stack (High Order Byte)
		8	Stack Pointer + 5	1	Index Register from Stack (Low Order Byte)
		9	Stack Pointer + 6	1	Next Instruction Address from Stack (High Order Byte)
		10	Stack Pointer + 7	1	Next Instruction Address from Stack (Low Order Byte)
SWI	12	1	Op Code Address	1	Op Code
		2	Op Code Address + 1	1	Irrelevant Data
		3	Stack Pointer	0	Return Address (Low Order Byte)
		4	Stack Pointer – 1	0	Return Address (High Order Byte)
		5	Stack Pointer – 2	0	Index Register (Low Order Byte)
		6	Stack Pointer – 3	0	Index Register (High Order Byte)
		7	Stack Pointer – 4	0	Contents of Accumulator A
		8	Stack Pointer – 5	0	Contents of Accumulator B
		9	Stack Pointer – 6	0	Contents of Cond. Code Register
		10	Stack Pointer – 7	1	Irrelevant Data
		11	Vector Address FFFA (Hex)	1	Address of Subroutine (High Order Byte)
		12	Vector Address FFFB (Hex)	1	Address of Subroutine (Low Order Byte)

** While the MCU is in the "Wait" state, its bus state will appear as a series of the MCU reads of an address which is seven locations less than the original contents of the Stack Pointer. Contrary to the HD6800, none of the ports are driven to the high impedance state by a WAI instruction.

(Continued)

Table 14 Cycle by Cycle Operation (Continued)

Address Mode & Instruction	Cycles	Cycle #	Address Bus	R/$\overline{W}$ Line	Data Bus
RELATIVE					
BCC BHT BNE BCS BLE BPL BEQ BLS BRA BRN BGE BLT BVC BGT BMT BVS	3	1 2 3	Op Code Address Op Code Address + 1 Address Bus FFFF	1 1 1	Op Code Branch Offset Low Byte of Restart Vector
BSR	6	1 2 3 4 5 6	Op Code Address Op Code Address + 1 Address Bus FFFF Subroutine Starting Address Stack Pointer Stack Pointer – 1	1 1 1 1 0 0	Op Code Branch Offset Low Byte of Restart Vector Op Code of Next Instruction Return Address (Low Order Byte) Return Address (High Order Byte)

■ SUMMARY OF UNDEFINED INSTRUCTIONS OPERATION

The MCU has 36 undefined instructions. When these are carried out, the contents of Register and Memory in MCU change at random.

When the op codes (4E, 5E) are used to execute, the MCU continues to increase the program counter and it will not stop until the Reset signal enters. These op codes are used to test the LSI.

Table 15 Op Codes Map

HD68P01 MICROCOMPUTER INSTRUCTIONS																			
OP CODE						ACC A	ACC B	IND	EXT	ACCA or SP				ACCB or X					
										IMM	DIR	IND	EXT	IMM	DIR	IND	EXT		
HI / LO		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111		
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0000	0		SBA	BRA	TSX	NEG				SUB								0	
0001	1	NOP	CBA	BRN	INS					CMP								1	
0010	2			BHI	PULA (+1)					SBC								2	
0011	3			BLS	PULB (+1)	COM				*	SUBD (+2)			*	ADDD (+2)			3	
0100	4	LSRD (+1)		BCC	DES	LSR				AND								4	
0101	5	ASLD (+1)		BCS	TXS					BIT								5	
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6	
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7	
1000	8	INX (+1)		BVC	PULX (+2)	ASL				EOR								8	
1001	9	DEX (+1)	DAA	BVS	RTS (+2)	ROL				ADC								9	
1010	A	CLV		BPL	ABX	DEC				ORA								A	
1011	B	SEV	ABA	BMI	RTI (+7)					ADD								B	
1100	C	CLC		BGE	PSHX (+1)	INC				*	CPX (+2)			*	LDD (+1)			C	
1101	D	SEC		BLT	MUL (+7)	TST				BSR (+4)	JSR (+2)			* (+1)	STD (+1)			D	
1110	E	CLI		BGT	WAI (+6)	**		JMP (-3)		*	LDS (+1)			*	LDX (+1)			E	
1111	F	SEI		BLE	SWI (+9)	CLR				* (+1)	STS (+1)			* (+1)	STX (+1)			F	
BYTE/CYCLE		1/2	1/2	2/3	1/3	1/2	1/2	2/6	3/6	2/2	2/3	2/4	3/4	2/2	2/3	2/4	3/4		

(NOTES) 1. Undefined Op codes are marked with ▨.
2. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.
3. The instructions shown below are all 3 bytes and are marked with "*".
Immediate addressing mode of SUBD, CPX, LDS, ADDD, LDD and LDX instructions, and undefined op codes (8F, CD, CF).
4. The Op codes (4E, 5E) are 1 byte/∞ cycles instructions, and are marked with "**".

■ PRECAUTIONS WHEN EMULATING THE HD6801 FAMILY

The HD68P01 series has 8k-byte EPROM space internally in location $E000 to $FFFF. Note the following when emulating the HD6801S0 (2k-byte ROM on-chip) and the HD6801V0 (4k-byte ROM on-chip) with the HD68P01 series.

1) Mode 0, 1, 6

Table 16 shows the address which may be used for the internal ROM space.

Table 16

HD6801S0	$F800 to $FFFF (2k bytes)
HD6801V0	$F000 to $FFFF (4k bytes)

Mode 0, 1 and 6 are expanded modes. When emulating the HD6801S0 and the HD6801V0, the addresses shown in Table 17 should not be used externally because they are the internal space in the EPROM on the package type. (See Fig. 26)

Table 17

HD6801S0	$E000 to $F7FF (6k bytes)
HD6801V0	$E000 to $EFFF (4k bytes)

(Note 1) In Table 16, the following addresses are external like the ROM on-chip type:
$FFF0 to $FFFF in Mode 1
$FFFE and $FFFF (reset vector) just after releasing reset in Mode 0

(Note 2) In Mode 0, data will not appear at Port 3 if accessing the EPROM addresses. It is different from the ROM on-chip type.

2) Mode 5, 7

Table 18 shows the addresses which may be used for the internal ROM space without any limitations.

Table 18

HD6801S0	$F800 to $FFFF (2k bytes)
HD6801V0	$F000 to $FFFF (4k bytes)

3) Mode 2, 3, 4

In these modes, the internal ROM is disable. The EPROM on the package type may be used equivalently as the ROM on-chip type.

(Example)

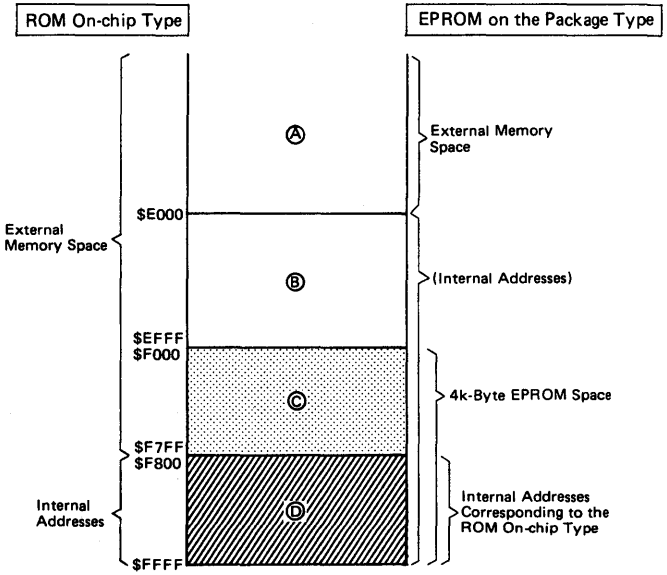


Figure 26 Memory Map Example when Emulating the HD6801S0 with the HD68P01M0 and the 4k-Byte EPROM

Figure 26 shows an address map example when emulating the HD6801S0 with the HD68P01M0 and the 4k-byte EPROM in mode 0, 1 and 6. In the emulation of expanded modes, the addresses for memories and peripherals may be used externally in space A, but not in space B and C which are internal addresses in the EPROM on the package type.

Figure 27 and 28 show the memory maps when emulating the HD6801S0 and HD6801V0 with the EPROM on the package type and the EPROM.

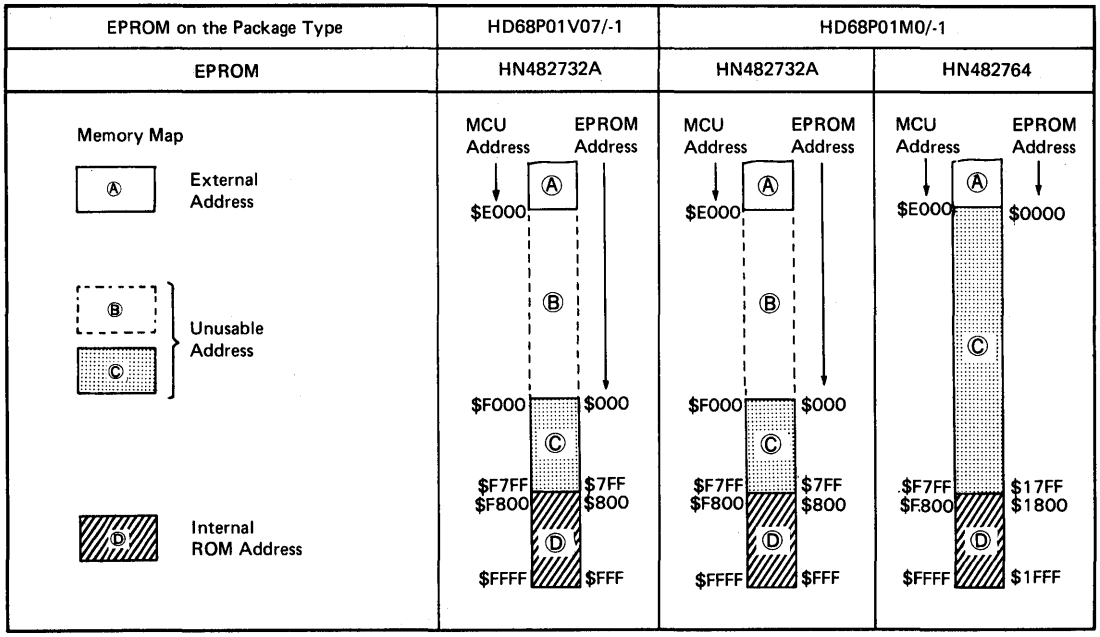


Figure 27 Memory Map When Emulating the HD6801S0

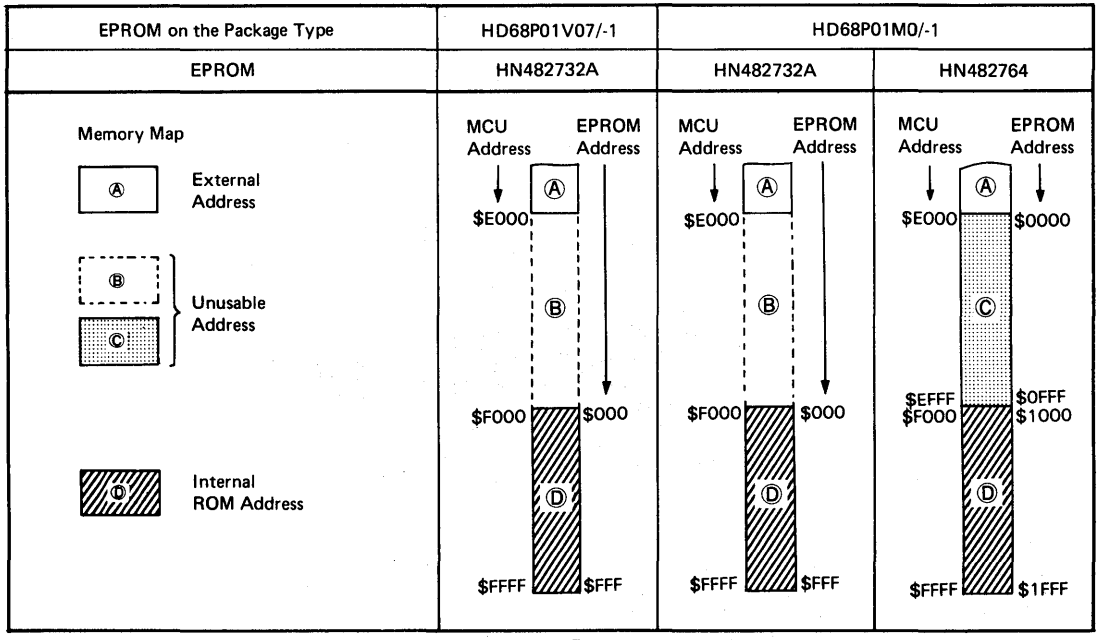


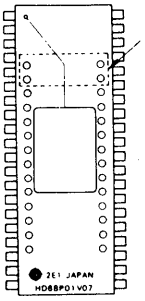
Figure 28 Memory Map When Emulating the HD6801V0

■ PRECAUTION TO USE EPROM ON THE PACKAGE 8-BIT SINGLE-CHIP MICROCOMPUTER

As this microcomputer takes a special packaging type with pin sockets on its surface, pay attention to the followings;

(1) Do not apply higher electro-static voltage or serge voltage etc. than maximum rating, or it may cause permanent damage to the device.

(2) There are 28 pin sockets on its surface. When using 32k



Let the index-side four pins open. When using 24 pin EPROM, match its index and insert it into lower 24 pin sockets.

EPROM (24 pins), let the index-side four pins open.

(3) When assembling this LSI into user's system products as well as the mask ROM type 8-bit single-chip microcomputer, pay attention to the followings to keep the good ohmic contact between EPROM pins and pin sockets.
   (a). When soldering on a printed circuit board, etc., keep its condition under 250°C within 10 seconds. Over-time/temperature may cause the bonding solder of socket pins to meet and the sockets may drop.
   (b) Keep out detergent or coater from the pin sockets at aft-solder flux removal or board coating. The flux or coater may make pin socket contactivity worse.
   (c) Avoid the permanent use of this LSI under the ever-vibratory place and system.
   (d) Repeating insertion/removal of EPROMs may damage the contactivity of the pin sockets, so it is recommended to assemble new ones to your system products.

(4) In order to perform the normal operation at 1.25 MHz, it is recommended to use the EPROM whose access time is less than 300 ns.

Ask our sales agent about anything unclear.

■ PRECAUTION FOR HD68P01 FAMILY SCI, TIMER STATUS FLAGS

The flags shown in Table 19 are cleared by read/write (flag reset condition 2) the data register corresponding to each flag after reading the status register (flag reset condition 1). To clear the flag correctly, take the following procedure:

1. Read the status register
2. Test the flag
3. Read/Write the data register

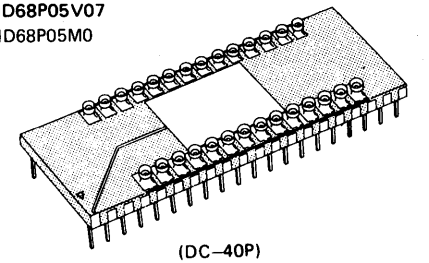
Table 19 Status Flag Reset Conditions

	Status Flag	Flag Reset Condition 1 (Status Register)	Flag Reset Condition 2 (Data Register)
TIMER	ICF	When each flag is "1"	ICR/Read
	OCF		OCR/Write
	TOF	TRCSR/Read	TC/Read
SCI	RDRF	When each flag is "1"	RDR/ Read
	ORFE		
	TDRE	TRCSR/Read	TDR/Write

HD68P05V07, HD68P05M0
MCU (Microcomputer Unit)

The HD68P05 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, RAM, I/O and timer. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD6800-based instruction set. Setting EPROM on the package, this MCU has the equivalent function as the HD6805U and HD6805V.

HD68P05V07
HD68P05M0



(DC–40P)

■ **HARDWARE FEATURES**
- 8-Bit Architecture
- 96 Bytes of RAM
- Memory Mapped I/O
- Internal 8-Bit Timer with 7-Bit Prescaler
- Vectored Interrupts – External, Timer and Software
- 24 I/O Ports + 8 Input Port
  (8 Lines Directly Drive LEDs; 7 Bits Comparator Inputs)
- On-Chip Clock Circuit
- Master Reset
- Easy for System Development and debugging
- 5 Vdc Single Supply

■ **SOFTWARE FEATURES**
- Similar to HD6800
- Byte Efficient Instruction Set
- Easy to Program
- True Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible Instruction Set with HD6805

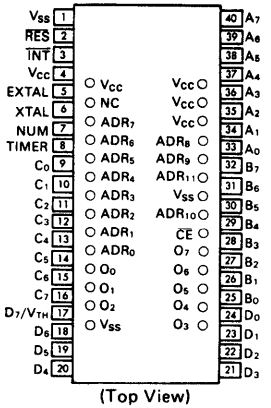
Note) EPROM is not attached to the MCU.

■ **TYPE OF PRODUCTS**

Type No.	Bus Timing	EPROM Type No.
HD68P05V07	1 MHz	HN482732A-30
HD68P05M0	1 MHz	HN482764-3

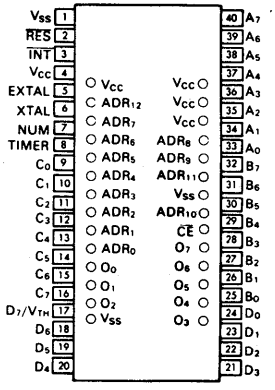
■ **PIN ARRANGEMENT**

HD68P05V07



(Top View)

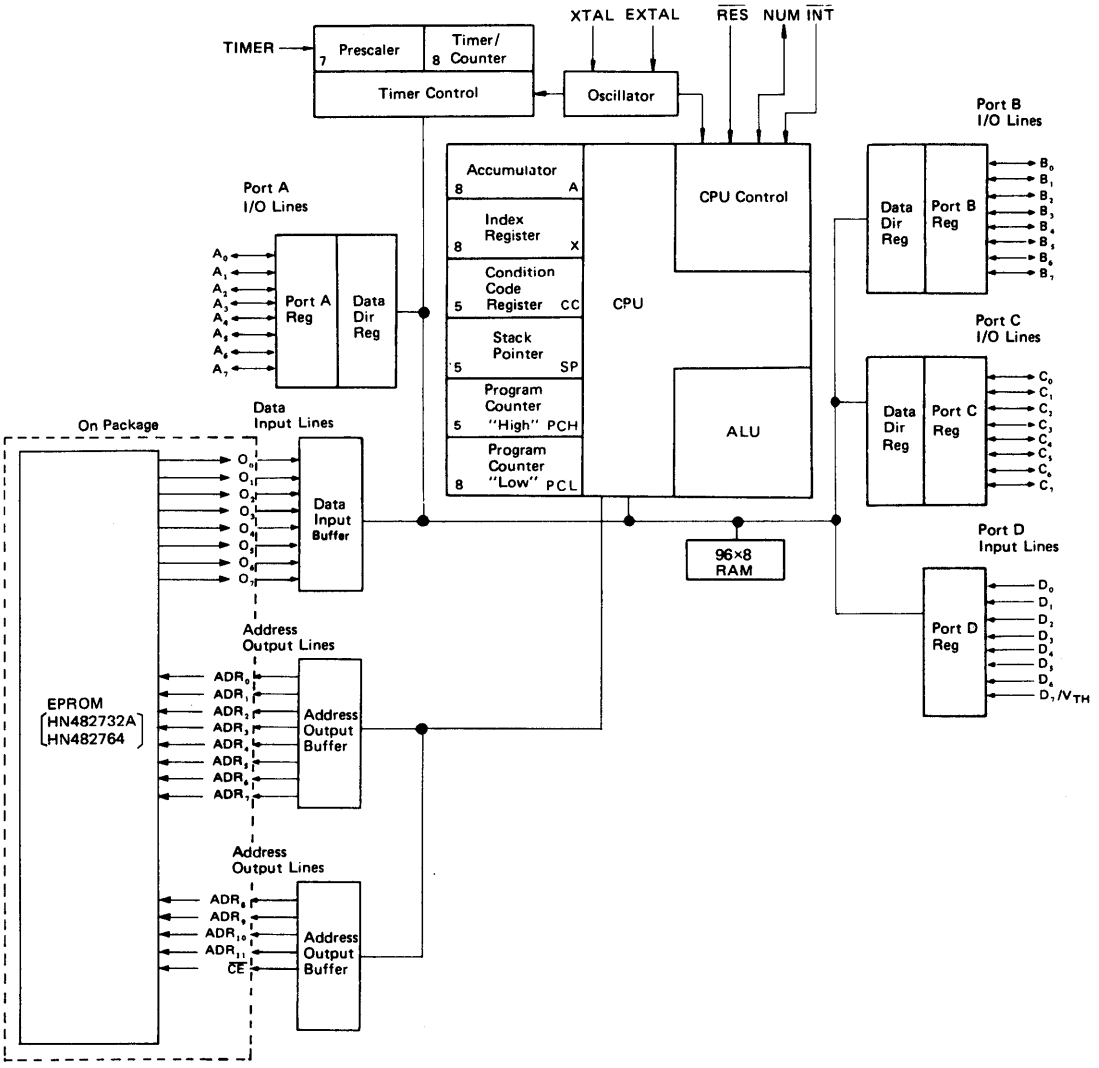
HD68P05M0



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC} *	-0.3 ~ +7.0	V
Input Voltage (EXCEPT. TIMER)	V_{in} *	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +12.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	- 55 ~ +150	°C

* With respect to V_{SS} (SYSTEM GND)

(NOTE) Permanent LSI damage may occur if maximum ratings are exceeded. Normal operation should be under recommended operating conditions. If these conditions are exceeded, it could affect reliability of LSI.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT}$			3.0	–	V_{CC}	V
	All Other			2.0	–	V_{CC}	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT}$			-0.3	–	0.8	V
	XTAL (Crystal Mode)			-0.3	–	0.6	V
	All Other			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	700	mW
Low Voltage Recover		LVR		–	–	4.75	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT}$			-50	–	50	μA
	XTAL (Crystal Mode)			-1200	–	0	μA

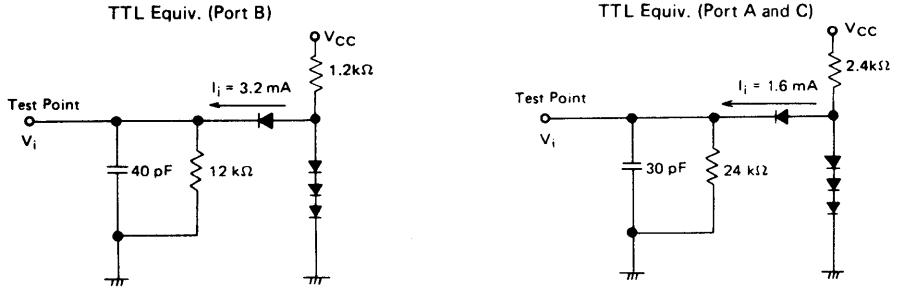
• AC CHARACTERISTICS (V_{CC}=5.25V ± 0.5V, V_{SS}=GND, Ta=0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
$\overline{INT}$ Pulse Width		t_{IWL}		$\frac{t_{cyc}}{250}$ +	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		$\frac{t_{cyc}}{250}$ +	–	–	ns
TIMER Pulse Width		t_{TWL}		$\frac{t_{cyc}}{250}$ +	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20%, R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2μF	100	–	–	ms
Input Capacitance	EXTAL	C_{in}	V_{in}=0V	–	–	35	pF
	All Other			–	–	10	pF

● PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ± 0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = −10 μA	3.5	–	–	V
			I_{OH} = −100 μA	2.4	–	–	V
	Port B		I_{OH} = −200 μA	2.4	–	–	V
			I_{OH} = −1 mA	1.5	–	–	V
	Port C		I_{OH} = −100 μA	2.4	–	–	V
Output "Low" Voltage	Port A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.4	V
	Port B		I_{OL} = 3.2 mA	–	–	0.4	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Port A, B, C, and D*	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		−0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	−500	–	–	μA
			V_{in} = 2V	−300	–	–	μA
	Port B, C, and D		V_{in} = 0.4V ~ V_{CC}	− 20	–	20	μA
Input "High" Voltage	Port D** (D_0 ~ D_6)	V_{IH}		–	V_{TH}+0.2	—	V
Input "Low" Voltage	Port D** (D_0 ~ D_6)	V_{IL}		–	V_{TH}-0.2	—	V
Threshold Voltage	Port D**(D_7)	V_{TH}		0	—	0.8×V_{CC}	V

* Port D as digital input
** Port D as analog input



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074 Ⓗ or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

● V_{CC} and V_{SS}

Power is supplied to the MCU using these two pins. V_{CC} is +5.25V ±0.5V. V_{SS} is the ground connection.

● $\overline{INT}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

● XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4 MHz maximum) can be connected to these pins to provide a system clock with various stability. Refer to INTERNAL OSCILLATOR for recommendations about these inputs.

● TIMER

This pin allows an external input to be used to decrement the internal timer circuitry. Refer to TIMER for additional information about the timer circuitry.

● $\overline{RES}$

This pin allows resetting of the MCU. Refer to RESETS for additional information.

● NUM

This pin is not for user application and should be connected to V_{SS}.

- **Input/Output Lines ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$)**

These 24 lines are arranged into three 8-bit ports (A, B and C). All lines are programmable as either inputs or outputs under software control of the Data Direction Register (DDR). Refer to INPUT/OUTPUT for additional information.

- **Input Lines ($D_0 \sim D_7$)**

These are 8-bit input lines, which has two functions. Firstly, these are TTL compatible inputs, in location $003. The other function of them is 7 bits comparator in location $007. Refer to INPUT for more detail.

■ REGISTERS

The MCU has five registers available to the programmer. They are shown in Figure 2 and are explained in the following paragraphs.

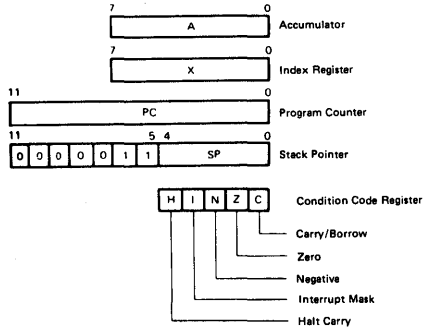


Figure 2 Programming Model

- **Accumulator (A)**

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

- **Index Register (X)**

The index register is an 8-bit register used for the indexed addressing mode. It contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations and data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

- **Program Counter (PC)**

The program counter is a 13-bit register that contains the address of the next instruction to be executed.

- **Stack Pointer (SP)**

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $007F and is decremented as data is being pushed onto the stack and incremented as data is being pulled from the stack. The six most significant bits of the stack pointer are permanently set to 00000011. During an MCU reset or the reset stack pointer (RSP) instruction, the stack pointer is set to location $007F. Subroutines and interrupts may be nested down to location $0061 which allows the programmer to use up to 15 levels of subroutine calls.

- **Condition Code Register (CC)**

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained in the following paragraphs.

Half Carry (H)

Used during arithmetic operations (ADD and ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask the timer and external interrupt ($\overline{INT}$). If an interrupt occurs while this bit is set it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

Used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in result equal to a logical one).

Zero (Z)

Used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts, and rotates.

■ TIMER

The MCU timer circuitry is shown in Figure 3. The 8-bit counter, the Timer Data Register (TDR), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the timer reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register (TCR) is set. The MCU responds to this interrpt by saving the present CPU state on the stack, fetching the timer interrupt vector from locations $0FF8 and $0FF9 and executing the interrupt routine. The timer interrupt can be masked by setting the timer

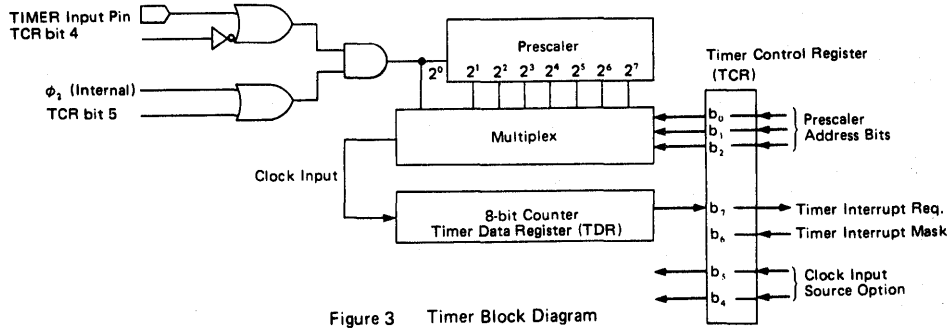


Figure 3 Timer Block Diagram

interrupt mask bit (bit 6) in the TCR. The interrupt bit (I bit) in the Condition Code Register also prevents a timer interrupt from being processed.

The clock input to the timer can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When the ϕ_2 signal is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. A prescaler option can be applied to the clock input that extends the timing interval up to a maximum of 128 counts before decrementing the counter (TDR). The timer continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter (TDR) can be read at any time by reading the TDR. This allows a program to determine the length of time since a time interrupt has occurred and not disturb the counting process.

The TDR is 8-bit read/write register in location $008. At power-up or reset, the TDR and the prescaler are initialize with all logical ones.

The timer interrupt request bit (bit 7 of the TCR) is set by hardware when timer count reaches zero, and is cleared by program or by hardware reset. The bit 6 of the TCR is writable by program. Both of those bits can be read by MPU.

The bit 5 and bit 4 of the TCR select a clock input source. The selections are shown in Table 1. Bit 3 is not used. Bit 2, bit 1 and bit 0 are used to select the prescaler dividing ratio, shown in Table 2. At reset, an internal clock by the TIMER input pin is selected as clock source and "÷ 1 mode" is selected as the prescaler dividing ratio.

(NOTE) If the MCU Timer is not used, the TIMER input pin must be grounded.

Table 1 Selection of Clock Input Source

Timer Control Register (TCR)		Clock Input Source
Bit 5	Bit 4	
0	0	
0	1	ϕ_2 Controlled by TIMER Input (Note)
1	0	
1	1	Event Input from TIMER

(NOTE) 1. 0.0 and 1.0 are not usable in mask option of 6805
2. The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock.

Table 2 Selection of Prescaler Dividing Ratio

Timer Control Register (TCR)			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	Prescaler ÷ 1
0	0	1	Prescaler ÷ 2
0	1	0	Prescaler ÷ 4
0	1	1	Prescaler ÷ 8
1	0	0	Prescaler ÷ 16
1	0	1	Prescaler ÷ 32
1	1	0	Prescaler ÷ 64
1	1	1	Prescaler ÷ 128

■ RESETS

The MCU can be reset two ways; by initial power-up and by the external reset input ($\overline{RES}$), see Figure 4. All the I/O ports are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High".

This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 5, typically provides sufficient delay.

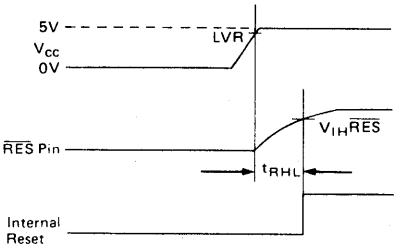


Figure 4 Power and $\overline{RES}$ Timing

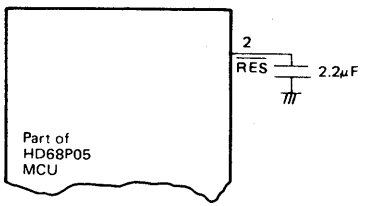


Figure 5 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR

The internal oscillator circuit is designed to require a minimum of external components. The use of a crystal (AT cut, 4 MHz max) is sufficient to drive the internal oscillator with various stability. The different connection methods are shown in Figure 6. Crystal specifications are given in Figure 7.

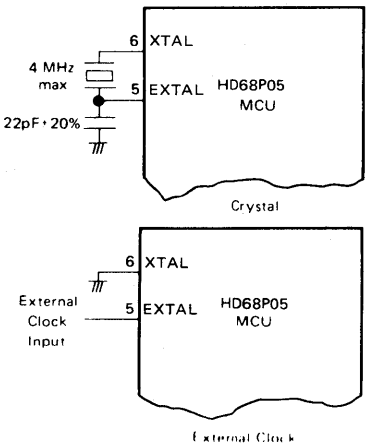
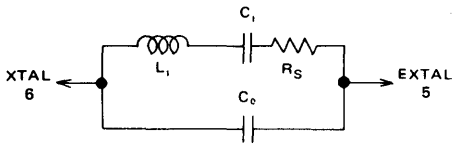


Figure 6 Internal Oscillator



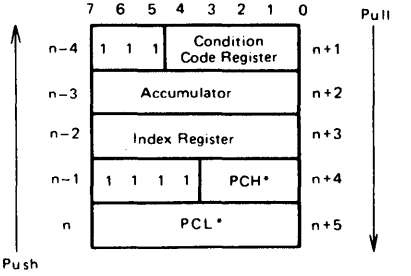
AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz
R_S = 60 Ω max.

Figure 7 Crystal Parameters

■ INTERRUPTS

The MCU can be interrupted three different ways: through the external interrupt ($\overline{INT}$) input pin, the internal timer interrupt request, and a software interrupt instruction (SWI). When any interrupt occurs, processing is suspended, the present CPU state is pushed onto the stack in the order shown in Fig. 8, the interrupt bit (I) in the Condition Code Register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Since the stack pointer decrements during pushes, the low order byte (PCL) of the program counter is stacked first; then the high order five bits (PCH) are stacked. This ensures that the program counter is loaded correctly as the stack pointer increments when it pulls data from the stack. A subroutine call will canse only the program counter (PCH, PCL) contents to be pushed onto the stack. This interrupt bit (I) in the condition code register is set, the address of the interrupt routine is obtained from the appropriate interrupt vector address, and the interrupt routine is executed. Stacking the CPU registers, setting the I bit, and vector fetching requires 11 cycles. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the MCU to resume processing of the program prior to the interrupt. Table 3 provides a listing of the interrupts, their priority, and the vector address that contain the starting address of the appropriate interrupt routine.

A flowchart of the interrupt processing sequence is given in Fig. 9.



* For subroutine calls, only PCH and PCL are stacked.

Figure 8 Interrupt Stacking Order

Table 3 Interrupt Priorities

Interrupt	Priority	Vector Address	
		HD68P05V07	HD68P05M0
$\overline{RES}$	1	$0FFE, $0FFF	$1FFE, $1FFF
SWI	2	$0FFC, $0FFD	$1FFC, $1FFD
$\overline{INT}$	3	$0FFA, $0FFB	$1FFA, $1FFB
TIMER	4	$0FF8, $0FF9	$1FF8, $1FF9

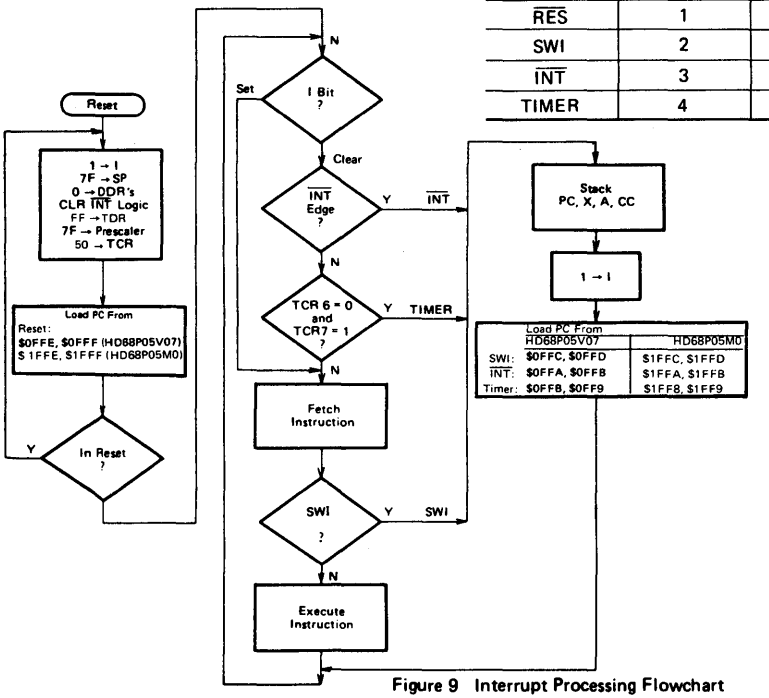
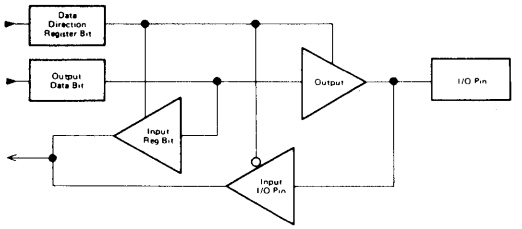


Figure 9 Interrupt Processing Flowchart

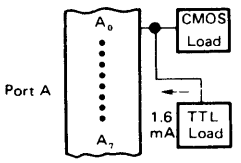
■ INPUT/OUTPUT

There are 24 input/output pins. All pins are programmable as either inputs or outputs under software control of the corresponding Data Direction Register (DDR). When programmed as outputs, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (see Fig. 10). When Port B is programmed for outputs, it is capable of sinking 10mA on each pin (V_{OL} = 1V max). All input/output lines are TTL compatible as both inputs and outputs. Port A is CMOS compatible as outputs, and Port B and C are CMOS compatible as inputs. Figure 11 provides some examples of port connections.

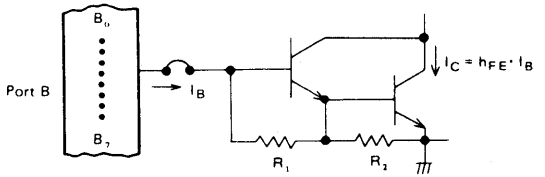


Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0		3-State	Pin

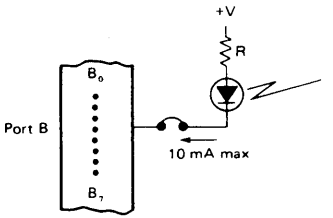
Figure 10 Typical Port I/O Circuitry



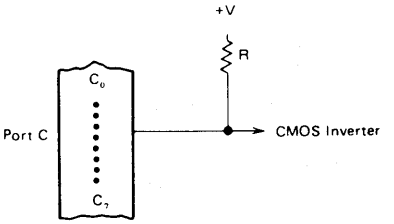
Port A Programmed as output(s), driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s), driving Darlington base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)



Port C Programmed as output(s), driving CMOS loads, using external pull-up resistors. (d)

Figure 11 Typical Port Connections

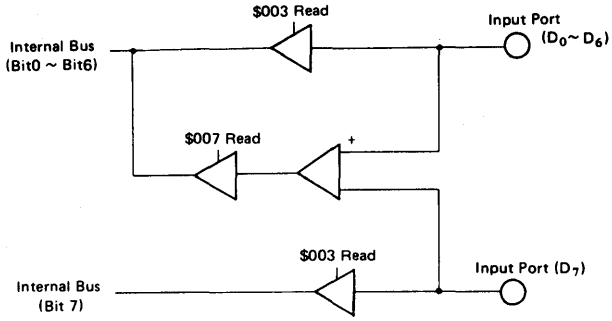
■ INPUT

Port D can be used as either 8 TTL compatible inputs or 1 threshold input and 7 analog inputs pins. Fig. 12 (a) shows the construction of port D. The Port D register at location S003 stores TTL compatible inputs, and those in location S007 store the result of comparison D_0 to D_6 inputs with D_7 threshold input. Port D has not only the conventional function as inputs but also voltage-comparison function. Applying the latter, can easily check that 7 analog input electric potential max. exceeds the limit with the construction shown in Fig. 12 (b). Also, using one output pin of MCU, after external capacity is discharged at the preset state, charge the CR circuit of long enough time constant, apply the charging curve to the D_7 pin. The construction described above is shown in Fig. 12 (c). The compared result of D_0 to D_6 is regularly monitored, which gives the analog input electric potential applied to D_0 to D_6 pins from inverted time. This method enables 7 inputs to be converted from analog to digital. Furthermore, combination of two functions gives 3 level voltages from D_0 to D_6. Fig. 12 (d) provides the example when V_{TH} is set to 3.5V.

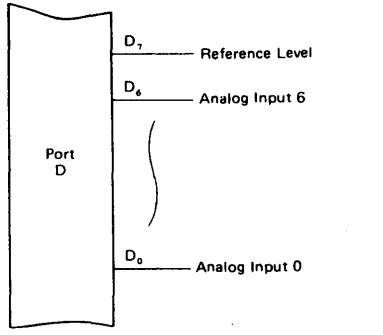
■ BIT MANIPULATION

The MCU has the ability to set or clear any single random access memory or input/output bit (except the data direction registers) with a single instruction (BSET, BCLR). Any bit in the page zero read only memory can be tested, using the BRSET and BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 13 illustrates the usefulness of the bit manipulation and test instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which power the controlled hardware.

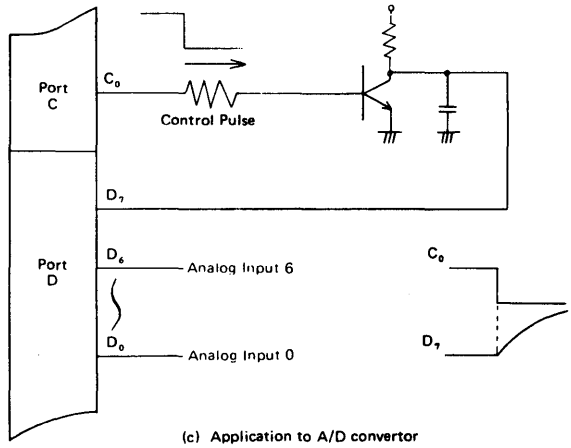
This program, which uses only seven ROM locations, provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer could also be incorporated to provide turn-on at some later time which would permit pulse-width modulation of the controlled power.



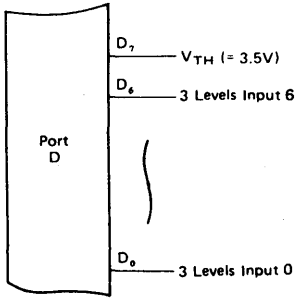
(a) The logic configuration of Port D



(b) Seven analog inputs and a reference level input of Port D



(c) Application to A/D convertor



(d) Application to 3 levels input

Input Voltage	($003)	($007)
0V ~ 0.8V	0	0
2.0V ~ 3.3V	1	0
3.7V ~ V_{CC}	1	1

Figure 12 Configuration and Application of Port D

```
        .
        .
        .
SELF 1  BRCLR 0, PORT A, SELF 1
        BSET 1, PORT A
        BCLR 1, PORT A
        .
        .
        .
```

Figure 13 Bit Manipulation Example

■ ADDRESSING MODES

The MCU has ten addressing modes available for use by the programmer. They are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 14. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 15. In direct addressing, the address of the operand is contained in the second byte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 16. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 17. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA=(PC)+2+Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken Rel=0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 18. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 19. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 20. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 21. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 22. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($00-$FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 23. The implied mode of addressing has no EA. All the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI, RTI belong to this group. All implied addressing instructions are one byte long.

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. They can be divided into five different types: register/memory, read/modify/write, branch, bit manipulation, and control. The following paragraphs briefly explain each type. All the instructions within a given type are presented in individual tables.

- **Register/Memory Instructions**

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 4.

- **Read/Modify/Write Instructions**

These instructions read a memory location or a register, modify or test its contents, and write the modified value back to memory or to the register. The test for negative or zero (TST) instruction is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 5.

- **Branch Instructions**

The branch instructions cause a branch from the program when a certain condition is met. Refer to Table 6.

- **Bit Manipulation Instructions**

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 7.

- **Control Instructions**

The control instructions control the MCU operations during program execution. Refer to Table 8.

- **Alphabetical Listing**

The complete instruction set is given in alphabetical order in Table 9.

- **Opcode Map**

Table 10 is an opcode map for the instructions used on the MCU.

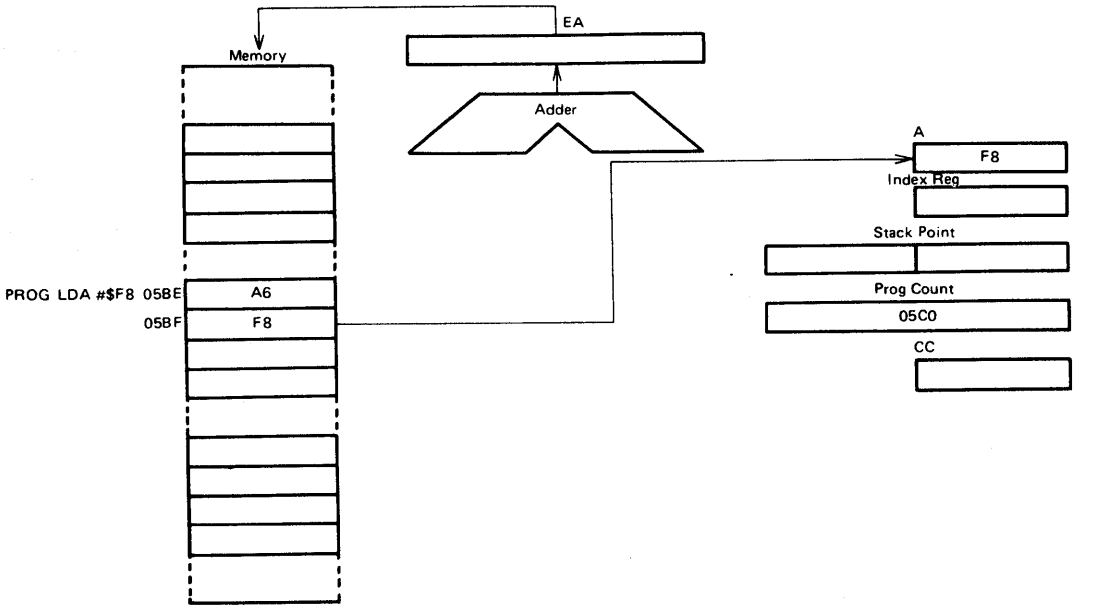


Figure 14 Immediate Addressing Example

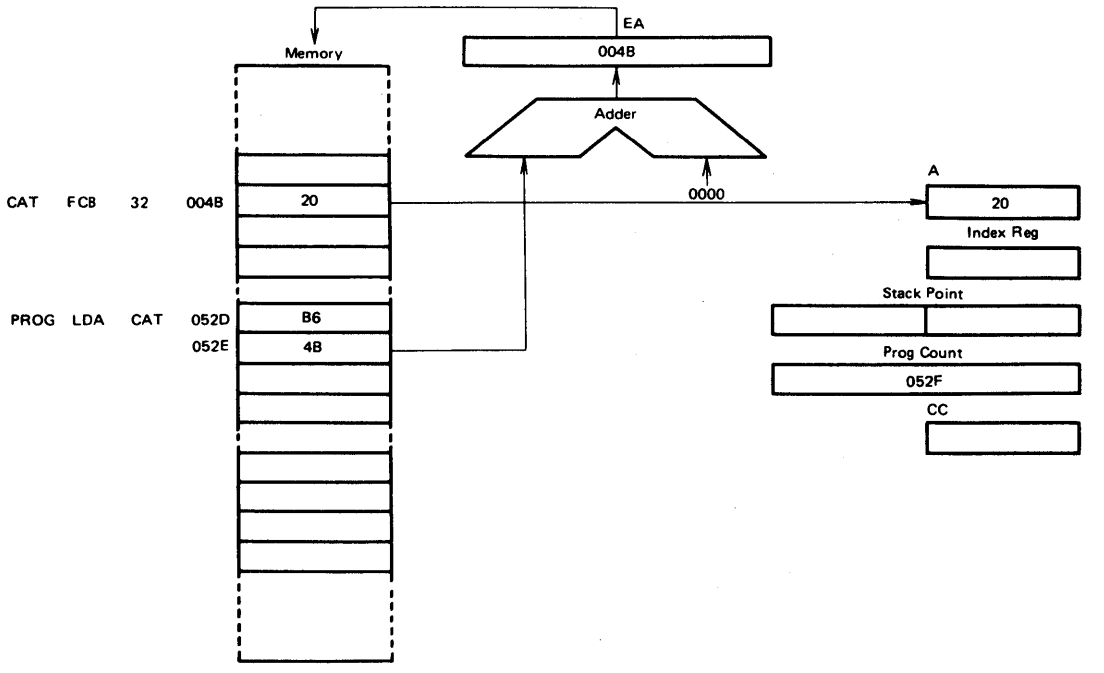


Figure 15 Direct Addressing Example

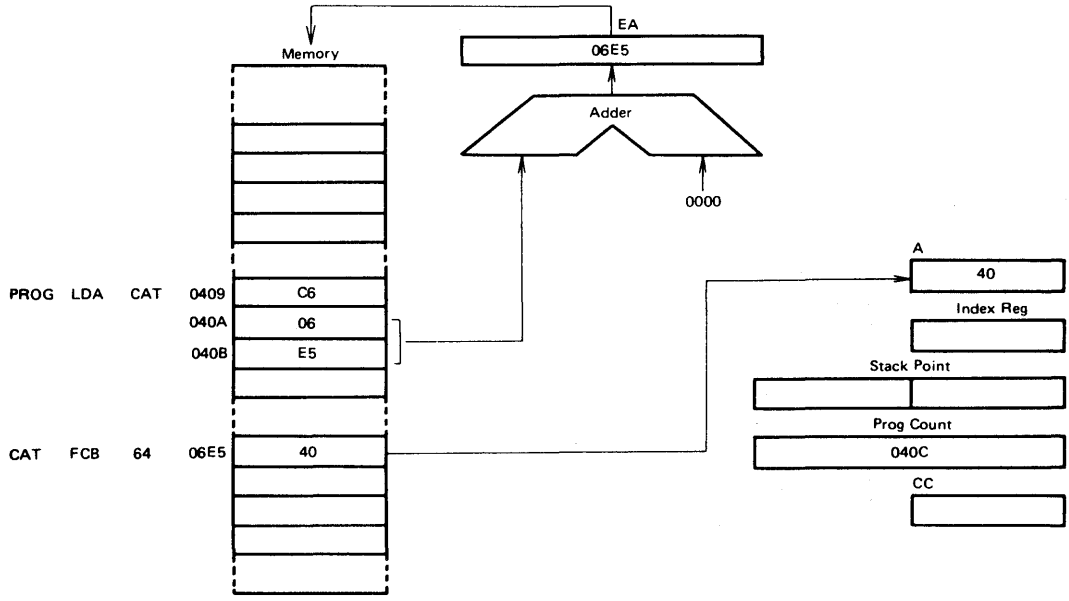


Figure 16 Extended Addressing Example

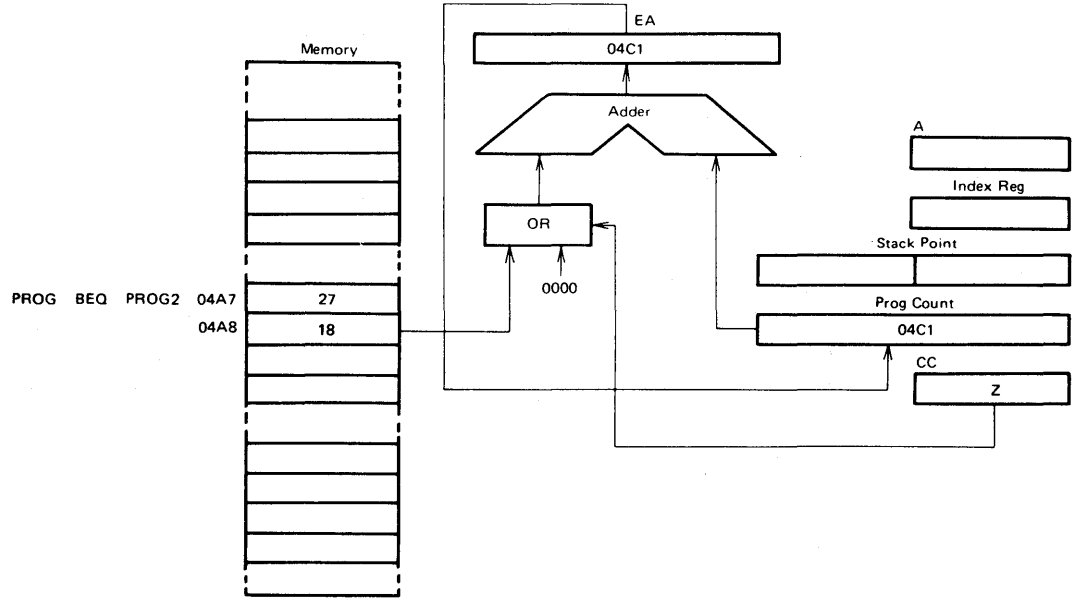


Figure 17 Relative Addressing Example

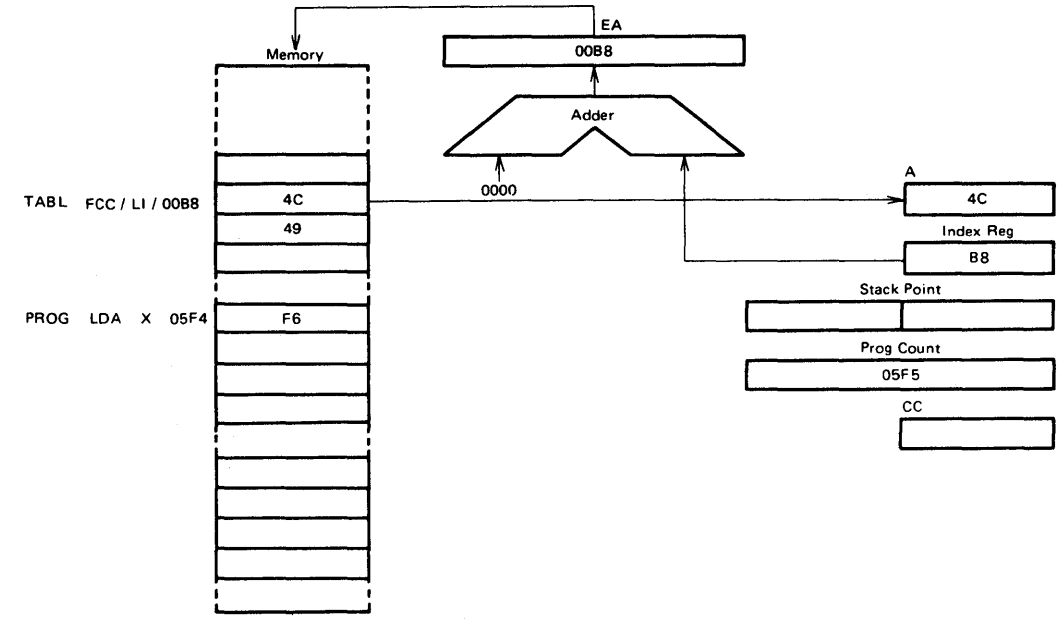


Figure 18 Indexed (No Offset) Addressing Example

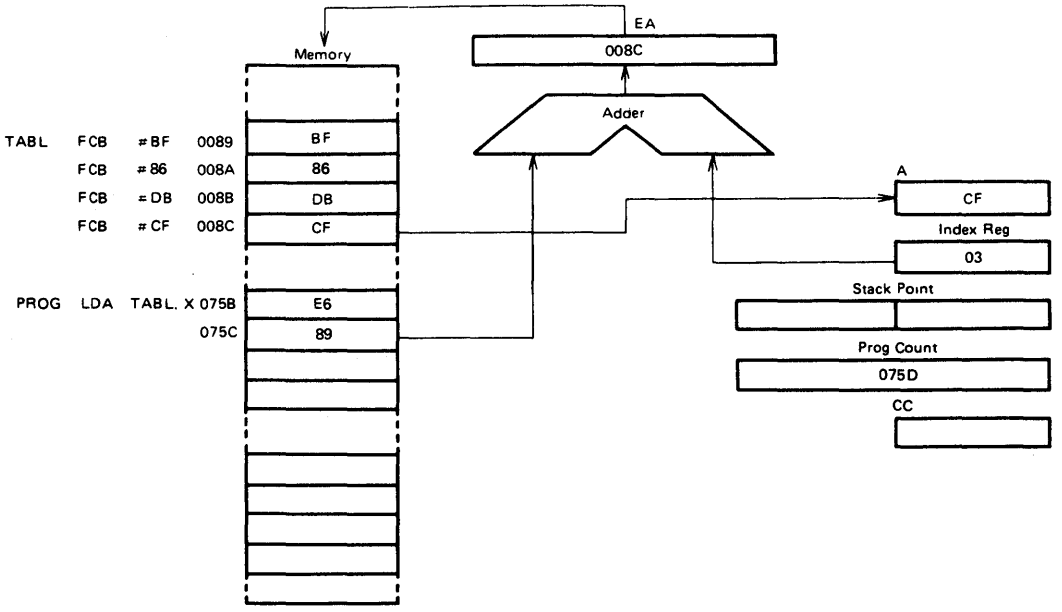


Figure 19 Indexed (8-Bit Offset) Addressing Example

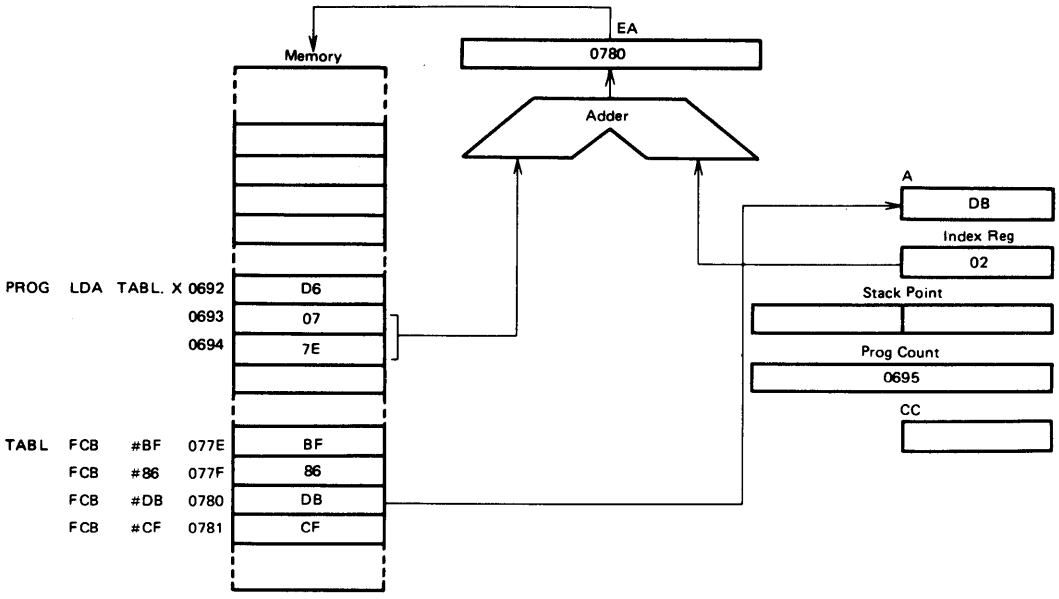


Figure 20 Indexed (16-Bit Offset) Addressing Example

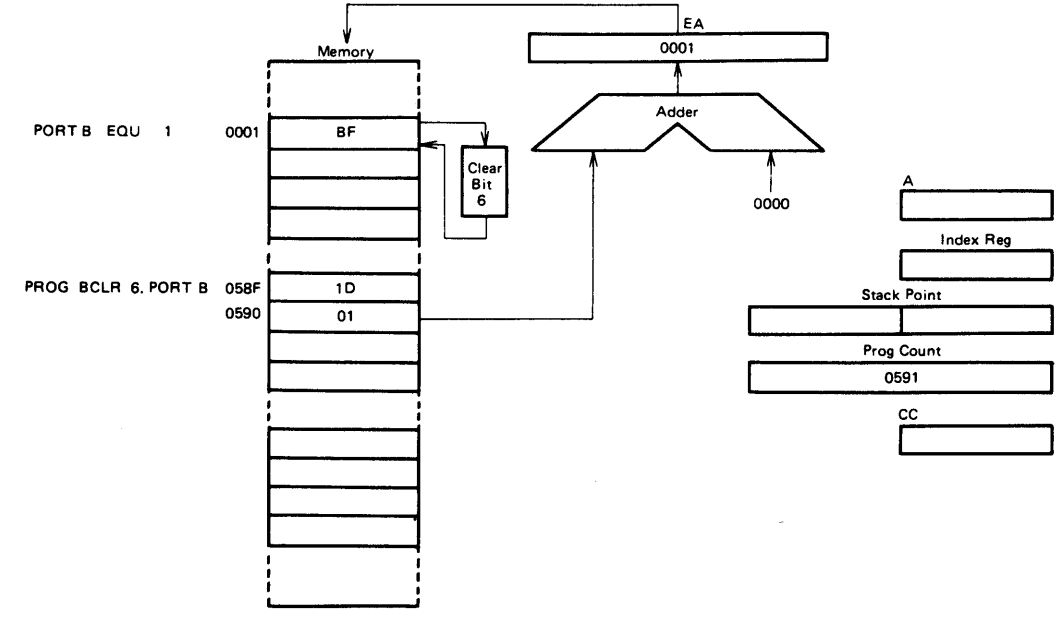


Figure 21 Bit Set/Clear Addressing Example

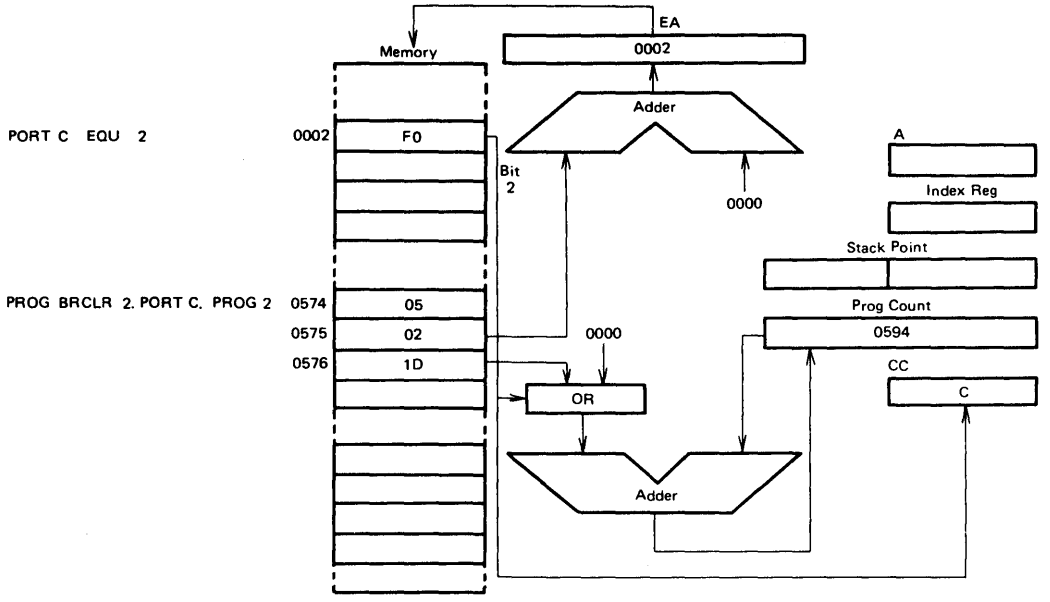


Figure 22 Bit Test and Branch Addressing Example

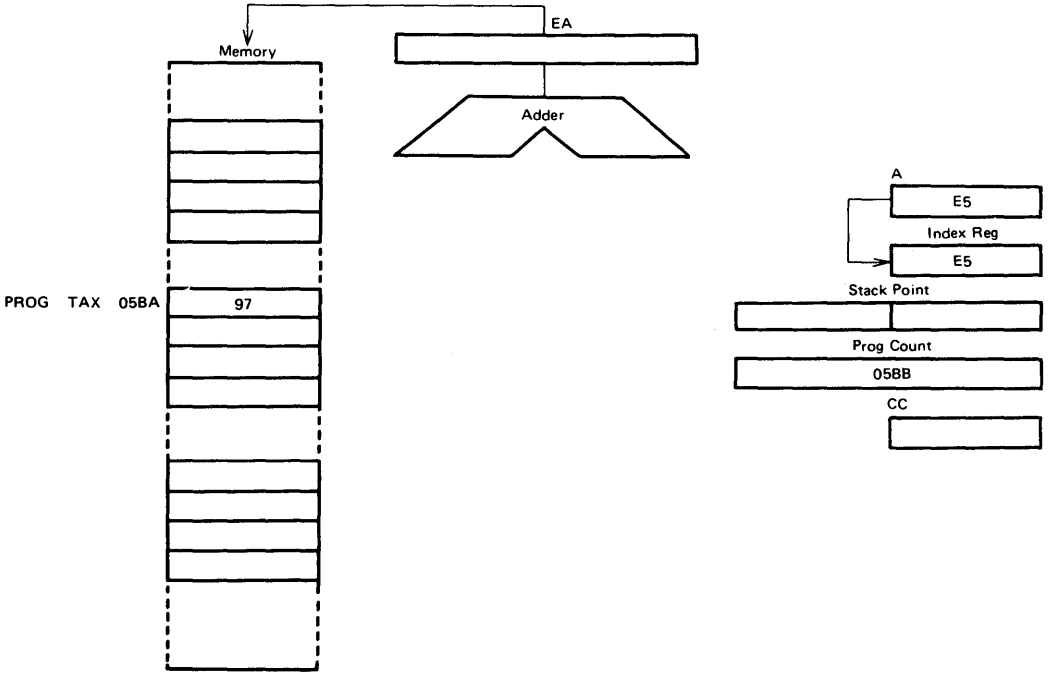


Figure 23 Implied Addressing Example

Table 4 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Table 5 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Table 6 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Table 7 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Table 8 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Table 9 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•

(to be continued)

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

Table 9 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Opcode Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

(NOTE) 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denote the number of bytes and the number of cycles required (Bytes/Cycles). Mnemonics followed by a "*" require a different number of cycles as follows:

RTI	9
RTS	6
SWI	11
BSR	8

3. ( ) indicate that the number in parenthesis must be added to the cycle count for that instruction.

■ HD68P05 USED AS ROM-ON-CHIP HD6805U/V

When using the HD68P05 for the HD6805U (2k ROM) or the HD6805V (4k ROM), take the memory configuration shown in Figure 25 (a) or (b). "Not Used" or "Self Test" ($F80 to $FF7) locations can be used in the HD68P05. Note that these locations cannot be used for a user program when making the program in mask ROM version. The HD6805U or HD6805V takes mask option method for internal oscillation, low voltage inhibit circuit or timer. The HD68P05 takes crystal option for oscillation without low voltage inhibit circuits. The HD68P05 should specify timer part by software, so it is required to set bit 0 to bit 5 of the Timer Control Register after reset and select the prescaler dividing ratio and the clock input source. Figure 24 shows a program example where external clock is selected as an input source at 128 dividing ratio.

When the program emulated by the EPROM on package type (the HD68P05) is built in the ROM-on-chip type, the instructions operating these bits are ignored because the HD6805U/V doesn't have the TCR bit 0 to 5.

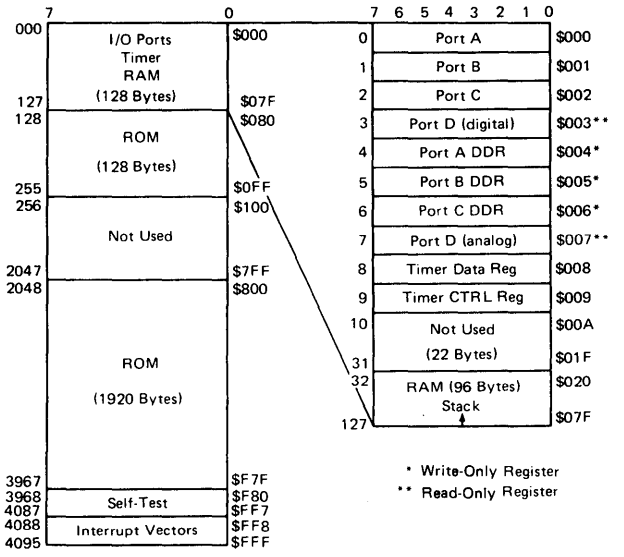
Bit 4 and 5 of the TCR should be specified to be "ϕ_2 controlled by TIMER Input" or "Event Input from TIMER".

The HD6805U/V has the self test program in locations $F80 to $FF7 as shown in Fig. 25. The HD68P05 can use this area. But a user program written in this area cannot be built in the ROM-on-chip type. (See Table 3.) The vector address of the HD68P05M0 is $1FF8 to $1FFF differently from the HD6805U/V.

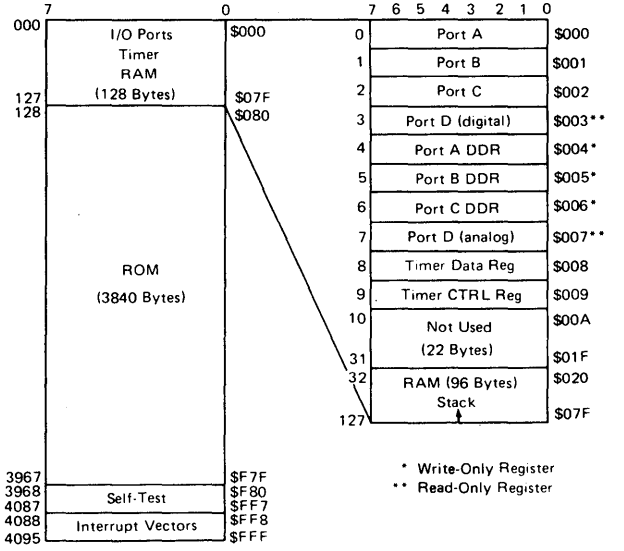
Pay attention to the above statements when debugging the program for the ROM-on-chip type.

```
LDA  #$77
STA  TCR ($009)
  •
  •
  •
```

Figure 24 Example to initialize timer control register (TCR)



(a) HD6805U Configuration



(b) HD6805V Configuration

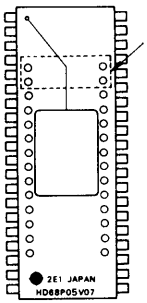
Figure 25 MCU Memory Configuration

■ PRECAUTION TO USE EPROM ON THE PACKAGE 8-BIT SINGLE-CHIP MICROCOMPUTER

As this microcomputer takes a special packaging type with pin sockets on its surface, pay attention to the followings;

(1) Do not apply higher electro-static voltage or serge voltage etc. than maximum rating, or it may cause permanent damage to the device.

(2) There are 28 pin sockets on its surface. When using 32k EPROM (24 pins), let the index-side four pins open.



Let the index-side four pins open. When using 24 pin EPROM, match its index and insert it into lower 24 pin sockets.

(3) When assembling this LSI into user's system products as well as the mask ROM type 8-bit single-chip microcomputer, pay attention to the followings to keep the good ohmic contact between EPROM pins and pin sockets.
   (a) When soldering on a printed circuit board, etc., keep its condition under 250°C within 10 seconds. Over-time/temperature may cause the bonding solder of socket pins to meet and the sockets may drop.
   (b) Keep out detergent or coater from the pin sockets at aft-solder flux removal or board coating. The flux or coater may make pin socket contactivity worse.
   (c) Avoid the permanent use of this LSI under the ever-vibratory place and system.
   (d) Repeating insertion/removal of EPROMs may damage the contactivity of the pin sockets, so it is recommended to assemble new ones to your system products.

Ask our sales agent about anything unclear.

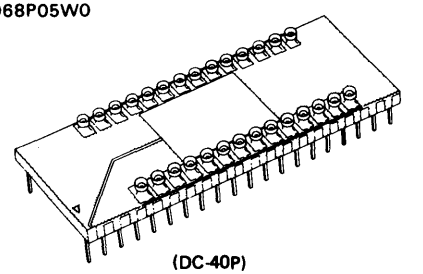
HD68P05W0
MCU (Microcomputer Unit)

The HD68P05W0 is the 8-bit Microcomputer Unit (MCU) which contains a CPU, on-chip clock, RAM, an A/D converter, I/O and two timers. It is designed for the user who needs an economical microcomputer with the proven capabilities of the HD6800-based instruction set. Setting EPROM on the package, this MCU has the same function as the HD6805W1 which has on-chip ROM. It is useful not only for a means of debugging and evaluating the HD6805W1 but also for small-scale-production.

The following EPROMs are available.

4k byte : HN482732A
8k byte : HN482764

HD68P05W0



(DC-40P)

■ HARDWARE FEATURES

- 8-Bit Architecture
- 96 Bytes of RAM
- Memory Mapped I/O
- Internal 8-Bit Timer with 7-Bit Prescaler
- Internal 8-bit Programmable Timer (Timer2) with 7-bit Prescaler
- Vectored interrupts : External, Timer and Software
- 23 I/O Ports + 6 Input Ports (8 Lines Directly Drive LEDs.)
- On-chip 8 bits A/D Converter
- On-chip Clock Generator
- Master Reset
- Easy for System Development and Debugging
- 5 Vdc Single Supply

■ SOFTWARE FEATURES

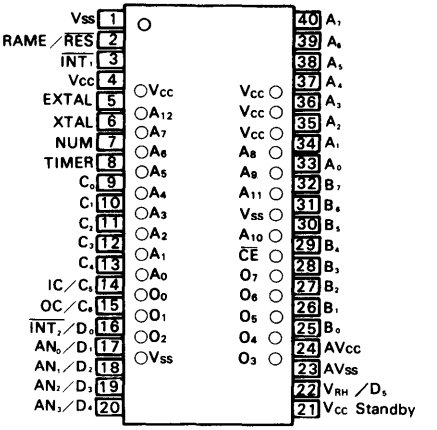
- Similar to HD6800 Family
- Byte Efficient Instruction Set
- Easy to Program
- Ture Bit Manipulation
- Bit Test and Branch Instructions
- Versatile Interrupt Function
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- Memory Usable as Registers/Flags
- Single Instruction Memory Examine/Change
- 10 Powerful Addressing Modes
- All Addressing Modes Apply to ROM, RAM and I/O
- Compatible Instruction Set with HD6805

■ TYPE OF PRODUCTS

Type No.	Bus Timing	EPROM Type No.
HD68P05W0	1 MHz	HN482732A-30
		HN482764-3

(NOTE) EPROM is not attached to the MCU.

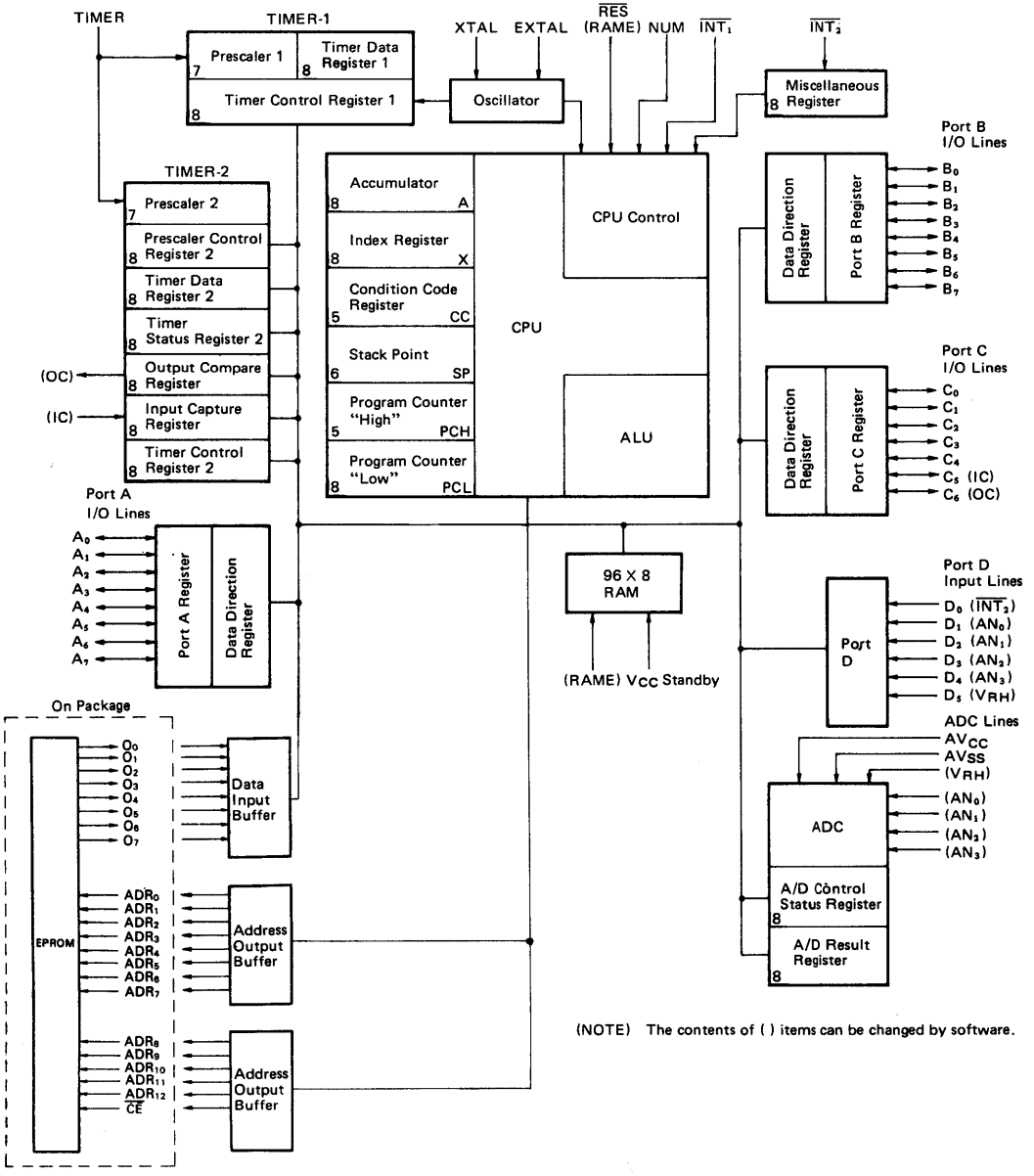
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



(NOTE) The contents of ( ) items can be changed by software.

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage (EXCEPT TIMER)	V_{in}	-0.3 ~ +7.0	V
Input Voltage (TIMER)		-0.3 ~ +15.0	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This device has an input protection circuit for high quiescent voltage and field, however, be careful not to impress a high input voltage than the insulation maximum value to the high input impedance circuit. To insure normal operation, the following are recommended for V_{in} and V_{out}:
$V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$	V_{IH}		4.0	–	V_{CC}	V
	$\overline{INT_1}$, $\overline{INT_2}$			3.0	–	V_{CC}	V
	All Others			2.0	–	V_{CC}	V
	Timer			2.0	–	V_{CC}	V
Input "Low" Voltage	$\overline{RES}$	V_{IL}		-0.3	–	0.8	V
	$\overline{INT_1}$, $\overline{INT_2}$			-0.3	–	0.8	V
	EXTAL			-0.3	–	0.6	V
	All Others			-0.3	–	0.8	V
Power Dissipation		P_D		–	–	750	mW
Low Voltage Recover		LVR		–	–	4.75	V
Input Leak Current	TIMER	I_{IL}	V_{in}=0.4V~V_{CC}	-20	–	20	μA
	$\overline{INT_1}$, $\overline{INT_2}$			-50	–	50	μA
	EXTAL			-1200	–	0	μA
Standby Voltage	Nonoperation Mode	V_{SBB}		4.0	–	V_{CC}	V
	Operation Mode	V_{SB}		4.75	–	V_{CC}	
Standby Current	Nonoperation Mode	I_{SBB}	V_{SBB}=4.0V	–	–	3	mA

• AC CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Clock Frequency		f_{cl}		0.4	–	4.0	MHz
Cycle Time		t_{cyc}		1.0	–	10	μs
$\overline{INT}$ Pulse Width		t_{IWL}		t_{cyc}+ 250	–	–	ns
$\overline{RES}$ Pulse Width		t_{RWL}		t_{cyc}+ 250	–	–	ns
TIMER Pulse Width		t_{TWL}		t_{cyc}+ 250	–	–	ns
Oscillation Start-up Time (Crystal Mode)		t_{OSC}	C_L=22pF±20% R_S=60Ω max.	–	–	100	ms
Delay Time Reset		t_{RHL}	External Cap. = 2.2μF	100	–	–	ms
Input Capacitance	XTAL, V_{RH}/Ds	C_{in}	V_{in}=0V	–	–	35	pF
	All Others			–	–	10	pF

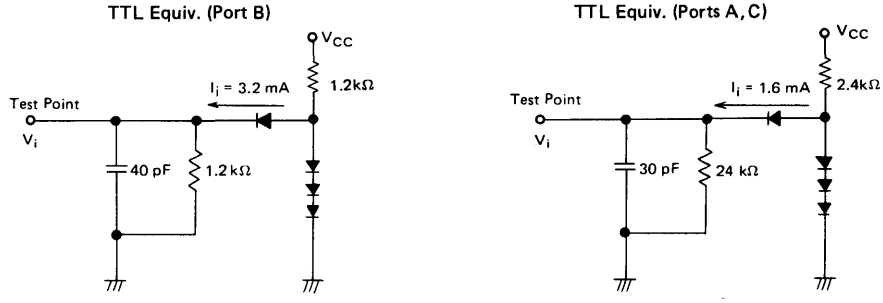
- **PORT ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V ±0.5V, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output "High" Voltage	Port A	V_{OH}	I_{OH} = -10 μA	3.5	–	–	V
			I_{OH} = -100 μA	2.4	–	–	V
	Port B		I_{OH} = -200 μA	2.4	–	–	V
			I_{OH} = -1 mA	1.5	–	–	V
	Port C		I_{OH} = -100 μA	2.4	–	–	V
Output "Low" Voltage	Ports A and C	V_{OL}	I_{OL} = 1.6 mA	–	–	0.5	V
	Port B		I_{OL} = 3.2 mA	–	–	0.5	V
			I_{OL} = 10 mA	–	–	1.0	V
Input "High" Voltage	Ports A, B, C and D*	V_{IH}		2.0	–	V_{CC}	V
Input "Low" Voltage		V_{IL}		-0.3	–	0.8	V
Input Leak Current	Port A	I_{IL}	V_{in} = 0.8V	-500	–	–	μA
			V_{in} = 2V	-300	–	–	μA
	Ports B, C and D*		V_{in} = 0.4V~V_{CC}	-20	–	20	μA

* Port D as digital input

- **A/D CONVERTER ELECTRICAL CHARACTERISTICS (V_{CC} = 5.25V±0.5V, V_{SS} = AV_{SS} = GMD, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	min	typ	max	Unit
Analog Power Supply Voltage	AV_{CC}		4.75	5.25	5.75	V
Analog Input Voltage	AV_{in}		0	–	V_{RH}	V
Reference "High" Voltage	V_{RH}	4.75V ≦ V_{CC} ≦ 5.25V	4.0	–	V_{CC}	V
		5.25V < V_{CC} ≦ 5.75V	4.0	–	5.25	V
Analog Multiplexer Input Capacitance			–	–	7.5	pF
Resolution Power			–	8	–	Bit
Conversion Time			76	76	76	t_{cyc}
Input Channels			4	4	4	Channel
Absolute Accuracy		Ta = 25°C	–	–	±1.5	LSB



(NOTE) 1. Load capacitance includes the floating capacitance of the probe and the jig etc.
2. All diodes are 1S2074Ⓗ or equivalent.

Figure 1 Bus Timing Test Loads

■ SIGNAL DESCRIPTION

The input and output signals for the MCU, shown in PIN ARRANGEMENT, are described in the following paragraphs.

• V_{CC} and V_{SS}

Voltage is supplied to the MCU using these two pins. V_{CC} is 5.25V ±0.5V. V_{SS} is the ground connection.

• $\overline{INT_1}/\overline{INT_2}$

This pin provides the capability for asynchronously applying an external interrupt to the MCU. Refer to INTERRUPTS for additional information.

• XTAL and EXTAL

These pins provide connections for the on-chip clock circuit. A crystal (AT cut, 4MHz maximum) or an external signal can be connected to these pins to provide the internal oscillator with varying degrees of stability. Refer to INTERNAL OSCILLATOR for recommendations about these inputs.

• TIMER

This pin allows an external input to be used to count for the internal timer circuitry. Refer to TIMER 1 and TIMER 2 for additional information about the timer circuitry.

• $\overline{RES}$

This pin allows resetting of the MCU. Refer to RESETS for additional information.

• NUM

This pin is not for user application and should be connected to V_{SS}.

• I/O Lines ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_6$)

These 23 lines are arranged into three ports (A, B and C). All lines are programmable as either inputs or outputs under software control of the Data Direction Registers. Refer to INPUT / OUTPUT for additional information.

• Input Lines ($D_0 \sim D_5$)

These are TTL compatible input lines, in location $0003. These also allow analog inputs to be used for an A/D converter. Refer to INPUT for additional information.

• V_{CC} Standby

V_{CC} Standby provides power to the standby portion of the RAM and the STBY PWR and RAME bits of the RAM Control Register. Voltage requirements depend on whether the MCU is in a powerup or powerdown state. In the powerup state, the power supply should provide V_{CC} and must reach V_{SB} before $\overline{RES}$ reaches 4.0V. During powerdown, V_{CC} Standby must remain above V_{SBB} (min) to sustain the standby RAM and STBY PWR bit. While in powerdown operation, the standby current will not exceed I_{SBB}.

It is typical to power both V_{CC} and V_{CC} Standby from the same source during nomal operation. A diode must be used between them to prevent supplying power to V_{CC} during powerdown operation shown Figure 2.

To sustain the standby RAM during powerdown, the following software or hardware are needed.

(1) Software

When clearing the RAM Enable bit (RAME) which is bit 6 of the RAM Control Register at location $001F, the RAM is disabled.

V_{CC} Standby must remain above V_{SBB} (min).

(2) Hardware

When RAME pin is "Low" before powerdown, the RAM is disabled. V_{CC} Standby must remain above V_{SBB} (min).

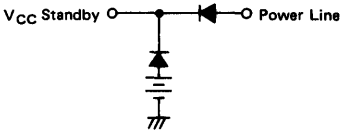


Figure 2 Battery Backup for V_{CC} Standby

• RAME

This pin is used for the external control of the RAM. When it is "Low" before powerdown, the RAM is disabled. If V_{CC} Standby remains above V_{SBB} (min), the standby RAM is sustained.

• AV_{CC}

This pin is used for the power supply of the A/D converter. When high accuracy is required, a different power source from V_{CC} should be impressed.

Connect to V_{CC} for all other cases. AV_{SS} corresponds to AV_{CC} as a GND terminal.

• $AN_0 \sim AN_3$

These pins allow analog inputs to be used for an A/D converter. These inputs are switched by the internal multiplexer and selected by bit 0 and 1 of the A/D Control Status Register (ADCSR: $000E).

• V_{RH} and AV_{SS}

The input terminal reference voltage for the A/D converter is "High" (V_{RH}) or "Low" (AV_{SS}). AV_{SS} is fixed at 0V.

• Input Capture (IC)

This pin is used for input of Timer2 control. in this case, Port C_5 should be configured as input. Refer to TIMER 2 for more details.

• Output Compare (OC)

This pin is used for output of Timer2 when the Output Compare Register is matched with the Timer Data Register 2. In this case, Port C_6 should be configured as an output. Refer to TIMER 2 for more details.

■ REGISTERS

The CPU has five registers available to the programmer, as shown in Figure 3 and explained below.

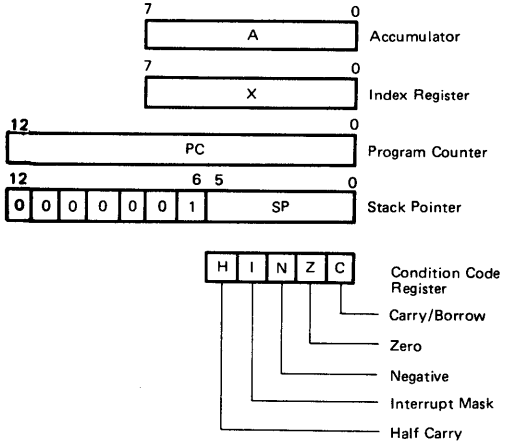


Figure 3 Programming Model

• Accumulator (A)

The accumulator is a general purpose 8-bit register used to hold operands and results of arithmetic calculations or data manipulations.

• Index Register (X)

The index register is an 8-bit register used for the indexed addressing mode and contains an 8-bit address that may be added to an offset value to create an effective address. The index register can also be used for limited calculations or data manipulations when using read/modify/write instructions. When not required by a code sequence being executed, the index register can be used as a temporary storage area.

• Program Counter (PC)

The program counter is a 13-bit register that contains the address of the next instruction to be executed.

• Stack Pointer (SP)

The stack pointer is a 13-bit register that contains the address of the next free location on the stack. Initially, the stack pointer is set to location $007F and is decremented as data is being pushed onto the stack and incremented while data is being pulled from the stack. The seven most significant bits of the stack pointer are permanently set to 0000001. During an MCU reset or reset stack pointer (RSP) instruction, the stack pointer is set to location $007F. Subroutines and interrupts may be nested down to location $0041 which allows the programmer to use up to 31 levels of subroutine calls.

• Condition Code Register (CC)

The condition code register is a 5-bit register in which each bit is used to indicate or flag the results of the instruction just executed. These bits can be individually tested by a program and specific action taken as a result of their state. Each individual condition code register bit is explained below.

Half Carry (H)

The half carry bit is used during arithmetic operations (ADD or ADC) to indicate that a carry occurred between bits 3 and 4.

Interrupt (I)

This bit is set to mask everything. If an interrupt occurs while this bit is set, it is latched and will be processed as soon as the interrupt bit is reset.

Negative (N)

The negative bit is used to indicate that the result of the last arithmetic, logical or data manipulation was negative (bit 7 in a result equal to a logical one).

Zero (Z)

Zero is used to indicate that the result of the last arithmetic, logical or data manipulation was zero.

Carry/Borrow (C)

Carry/borrow is used to indicate that a carry or borrow out of the arithmetic logic unit (ALU) occurred during the last arithmetic operation. This bit is also affected during bit test and branch instructions, shifts and rotates.

■ TIMER 1

The MCU timer circuitry is shown in Figure 4. The 8-bit counter, Timer Data Register 1 (TDR1), is loaded under program control and counts down toward zero as soon as the clock input is applied. When the TDR1 reaches zero, the timer interrupt request bit (bit 7) in the Timer Control Register 1 (TCR1) is set. The MCU responds to this interrupt by saving the present CPU state in the stack, fetching the timer 1 interrupt vector from locations $0FF8 and $0FF9 and executing the interrupt routine. The timer 1 interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the TCR 1. The interrupt bit (I bit) in the Condition Code Register also prevents a timer 1 interrupt from being processed.

The clock input to the timer 1 can be from an external source applied to the TIMER input pin or it can be the internal ϕ_2 signal. When ϕ_2 is used as the source, it can be gated by an input applied to the TIMER input pin allowing the user to easily perform pulse-width measurements. The timer 1 continues to count past zero, falling through to $FF from zero and then continuing the count. Thus, the counter (TDR1) can be read at any time by reading the TDR1. This allows a program to determine the length of time since a timer interrupt has occurred and not disturb the counting process.

At power-up or reset, the prescaler and counter are initialized with all logical ones; the timer 1 interrupt request bit (bit 7) is cleared and the timer 1 interrupt mask bit (bit 6) is set. In order to release the timer 1 interrupt, bit 7 of the TCR 1 must be cleared by software.

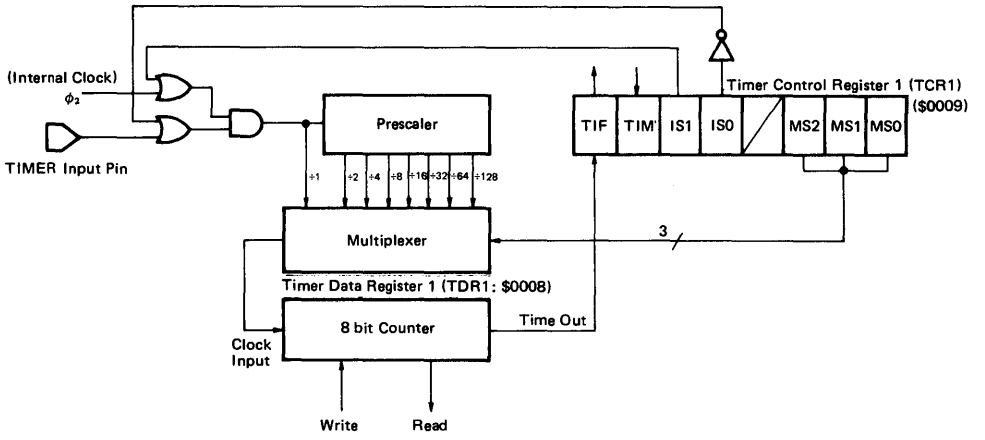
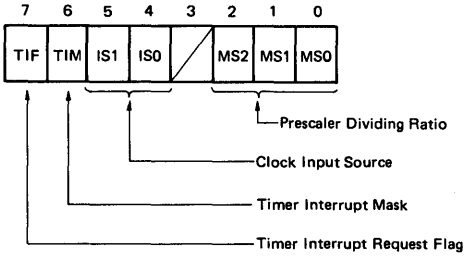


Figure 4 Timer Clock

- **Timer Control Register 1 (TCR1: $0009)**

The Timer Control Register 1 (TCR1: $0009) can control selection of clock input source and prescaler dividing ratio and timer interrupt.

Timer Control Register 1 (TCR1: $0009)



As shown in Table 1, the selection of the clock input source is IS0 and IS1 in the TCR1 (bit 4 and bit 5) and 3 kinds of input are selectable. At reset, internal clock ϕ_2 controlled by the TIMER input (bit 4 = 1, bit 5 = 0) is selected.

The prescaler dividing ratio is selected by MS0, MS1, and MS2 in the TCR1 (bit 0, bit 1, bit 2) as shown in Table 2. The dividing ratio is selectable from eight ways (÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128). At reset, ÷1 mode is selected. The prescaler is initialized by writing in the TDR1.

Timer 1 interrupt mask bit (TIM) allows the Timer 1 into interrupt at "0" and masks at "1". Timer 1 interrupt causes Timer 1 interrupt request bit (TIF) to be set. TIF must be cleared by software.

(NOTE) If the MCU Timer1 and Timer2 are not used, the TIMER input pin must be grounded.

Table 1 Selection of Clock Input Source

TCR1		Clock Input Source
Bit 5	Bit 4	
0	0	Internal Clock ϕ_2 *
0	1	ϕ_2 Controlled by TIMER Input
1	0	
1	1	Event Input From TIMER

* The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock input.

Table 2 Selection of Prescaler Dividing Ratio

TCR1			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

■ TIMER 2

The HD68P05W0 includes an 8-bit programmable timer (Timer 2) which can not only measure the input waveform but also generate the output waveform. The pulse width for both input and output waveform can be varied from several microseconds to several seconds.

(NOTE) If the MCU Timer1 and Timer2 are not used, the TIMER input pin must be grounded.

Timer 2 hardware consists of the followings.

- 8-bit Control Register 2
- 8-bit Status Register 2
- 8-bit Timer Data Register 2
- 8-bit Output Compare Register
- 8-bit Input Capture Register
- 5-bit Prescaler Control Register
- 7-bit Prescaler 2

Block Diagram of Timer 2 is shown in Fig. 5.

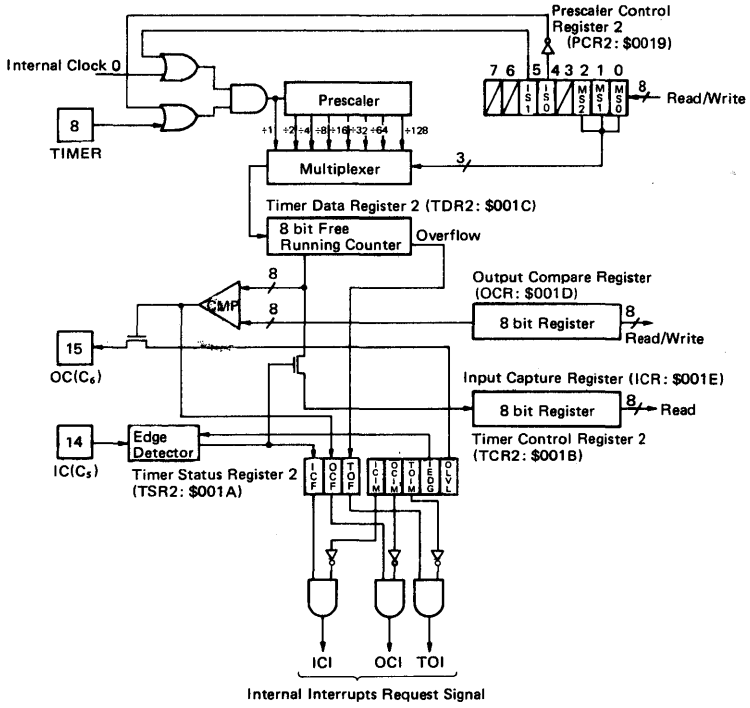


Figure 5 Block Diagram of Timer 2

• Timer Data Register 2 (TDR2; $001C)

The main part of the Timer 2 is the 8-bit Timer Data Register 2 (TDR2) as free-running counter, which is driven by internal clock ϕ_2 or the TIMER input and increments the value. The values in the counter is always readable by software.

The Timer Data Register 2 is Read/Write register and is cleared at reset.

• Output Compare Register (OCR; $001D)

The Output Compare Register (OCR) is an 8-bit read/write register used to control an output waveform. The contents of this register are always compared with those of the TDR2. When these two contents conform to each other, the flag (OCF) in the Timer Status Register 2 (TSR2) is set and the value of the output level bit (OLVL) in the TCR2 is transferred to Port C_6 (OC).

If Port C_6's Data Direction Register (DDR) is "1" (output), this value will appear at Port C_6 (OC). Then the values of OCF and OLVL can be changed for the next compare. The OCR is set to $FF at reset.

• Input Capture Register (ICR; $001E)

The Input Capture Register (ICR) is an 8-bit read-only register used to store the value of the TDR2 when Port C_5 (IC) input transition occurs as defined by the input edge bit (IEDG) of the TCR2.

In order to apply Port C_5 (IC) input to the edge detect circuit, the DDR of Port C_5 should be cleared ("0").*

To ensure an input capture under all condition, Port C_5 (IC) input pulse width should be 2 Enable-cycles at least.

*The edge detect circuit always senses Port C_5 (IC) even if the DDR is set with Port C_5 output.

- **Timer Control Register 2 (TCR2; $001B)**

The Timer Control Register 2 (TCR2) consists of an 5-bit register of which all bits can be read and written.

Timer Control Register 2 (TCR2: $001B)

7	6	5	4	3	2	1	0
			ICIM	OCIM	TOIM	IEDG	OLVL

Bit 0 OLVL Output Level

This bit will appear at Port C_6 when the value in the TDR2 equals the value in the OCR, if the DDR of Port C_6 is set. It is cleared by reset.

Bit 1 IEDG Input Edge

This bit determines which level transition of Port C_5 (IC) input will trigger a data store to ICR from the TDR2. When this function is used, it is necessary to clear DDR of Port C_5. When IEDG = 0, the negative edge triggers ("High" to "Low" transition). When IEDG = 1, the positive edge triggers ("Low" to "High" transition). It is cleared by reset.

Bit 2 TOIM Timer Overflow Interrupt Mask

When this bit is cleared, internal interrupt (TOI) is enabled by TOF interrupt but when set, interrupt is inhibited.

Bit 3 OCIM Output Compare Interrupt Mask

When this bit is cleared, internal interrupt (OCI) by OCF interrupt occurs. When set, interrupt is inhibited.

Bit 4 ICIM Input Capture Interrupt Mask

When this bit is cleared, internal interrupt (ICI) by ICF interrupt occurs. When set, interrupt is inhibited.

- **Timer Status Register 2 (TSR2: $001A)**

The Timer Status Register 2 (TSR2) is an 8-bit read-only register which indicates that:

(1) A proper leveltransition has been detected on the input pin with a subsequent transfer of the TDR2 value to the ICR (ICF).
(2) A match has been found between the TDR2 and the OCR (OCF).
(3) The TDR2 is zero (TOF).

Each of the event can generate 3 kinds of internal interrupt request and is controlled by an individual inhibit bits in the TCR2. If the I bit in the Condition Code Register is cleared, priority vectors are generated in response to clearing each interrupt mask bit. Each bit is described below.

Timer Status Register 2 (TSR2: $001A)

7	6	5	4	3	2	1	0
ICF	OCF	TOF					

Bit 5 TOF Timer Overflow Flag

This read-only bit is set when the TDR2 contains $00. It is cleared by reading the TSR2 followed by reading of the TDR2.

Bit 6 OCF Output Compare Flag

This read-only bit is set when a match is found between the OCR and the TDR2. It is cleared by reading the TSR2 and then writing to the OCR.

Bit 7 ICF Input Capture Flag

This read-only bit is set to indicate a proper level transition and cleared by reading the TSR2 and then reading the TCR2.

User can write into port C_6 by software.

Accordingly, after port C_6 has output by hardware and is immediately write into by software, simultaneous cyclic pulse control with a short width is easy.

- **Prescaler Control Register 2 (PCR2: $0019)**

The selections of clock input source and prescaler dividing ratio are performed by the Prescaler Control Register 2 (PCR2: $0019).

Prescaler Control Register 2 (PCR2: $0019)

7	6	5	4	3	2	1	0
		IS1	IS0		MS2	MS1	MS0

Prescaler Dividing Ratio

Clock Input Source

The selection of clock input source is performed in three different ways by bit 4 and bit 5 of the PCR2, as shown in Table 3. At reset, internal clock ϕ_2 controlled by the TIMER input (bit 4 = 1, bit 5 = 0) is selected.

The prescaler dividing ratio is selected by three bits in the PCR2 (bits 0, 1, 2), as shown in Table 4. The dividing ratio can be selected in 8 ways (÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64, ÷128). At reset, ÷1 (bit 0 = bit 1 = bit 2 = 0) is selected.

When writing into the PCR2, or when writing into the TDR2, prescaler is initialized to $FF.

Table 3 Selection of Clock Input Source

PCR2		Clock Input Source
Bit 5	Bit 4	
0	0	Internal Clock ϕ_2 *
0	1	ϕ_2 Controlled by TIMER Input
1	0	
1	1	Event Input from TIMER

* The TIMER input pin must be tied to V_{CC}, for uncontrolled ϕ_2 clock.

Table 4 Selection of Prescaler Dividing Ratio

PCR2			Prescaler Dividing Ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷ 1
0	0	1	÷ 2
0	1	0	÷ 4
0	1	1	÷ 8
1	0	0	÷ 16
1	0	1	÷ 32
1	1	0	÷ 64
1	1	1	÷ 128

CAUTION

(1) When executing Branch instructions* from address $117 to $11C, two flags (TOF and ICF) of the Timer Status Register 2 (TSR2) will be occasionally cleared.

Cause: These instructions have some dummy read cycles so the TSR2 can be read when executing the instructions.

Countermeasure : Don't program branch instructions shown in Table 1 and 3 at address $117 to $11C.

(2) When manipulating or testing the Timer Status Register 2 (TSR2) by Read/Modify/Write instructions**, two flags (TOF and ICF) of the TSR2 will be occasionally cleared.

Cause: These instructions have some dummy read cycles so the TSR2 can be read when executing the instructions.

Countermeasure: Don't use the instructions shown in Table 1, 2 and 4 for read/write/test operation of the TSR2 flags.

* Branch instructions

Bit Test & Branch	($00 ~ $0F)	in Table 5-(1)
Branch	($20 ~ $2F, $AD)	in Table 5-(4)

** Read/Modify/Write instructions

Bit Test & Branch	($00 ~ $0F)	in Table 5-(1)
Bit Set/Clear	($10 ~ $1F)	in Table 5-(2)
Memory Manipulation	($30 ~ $3F)	in Table 5-(3)

Table 5 Instruction Inhibited to Operate the TSR2

(1) Bit Test and Branch Instruction

Mnemonic	Op Code	# Bytes	# Cycles
BRSET n (n=0~7)	2 · n	3	10
BRCLR n (n=0~7)	01+2 · n	3	10

(2) Bit Set/Clear Instruction

Mnemonic	Op Code	# Bytes	# Cycles
BSET n (n=0~7)	10+2 · n	2	7
BCLR n (n=0~7)	11+2 · n	2	7

(3) Read/Modify/Write Instruction

Mnemonic	Op Code	# Bytes	# Cycles
INC	3C	2	6
DEC	3A	2	6
CLR	3F	2	6
COM	33	2	6
NEG	30	2	6
ROL	39	2	6
ROR	36	2	6
LSL	38	2	6
LSR	34	2	6
ASR	37	2	6
ASL	38	2	6
TST	3D	2	6

(4) Branch Instruction

Mnemonic	Op Code	# Bytes	# Cycles
BRA	20	2	4
BRN	21	2	4
BHI	22	2	4
BLS	23	2	4
BCC	24	2	4
(BHS)	24	2	4
BCS	25	2	4
(BLO)	25	2	4
BNE	26	2	4
BEQ	27	2	4
BHCC	28	2	4
BHCS	29	2	4
BPL	2A	2	4
BMI	2B	2	4
BMC	2C	2	4
BMS	2D	2	4
BIL	2E	2	4
BIH	2F	2	4
BSR	AD	2	8

■ RESETS

The MCU can be reset two ways; by initial power-up and by the external reset input ($\overline{RES}$), see Figure 6. All the I/O ports are initialized to input mode (DDRs are cleared) during reset.

During power-up, a minimum 100 milliseconds is needed before allowing the $\overline{RES}$ input to go "High". This time allows the internal crystal oscillator to stabilize. Connecting a capacitor to the $\overline{RES}$ input, as shown in Figure 7, typically provides

sufficient delay.

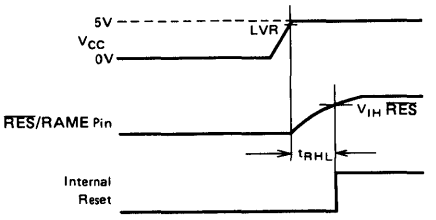


Figure 6 Power Up and Reset Timing

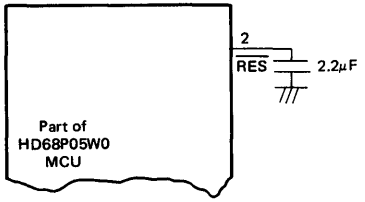


Figure 7 Power Up Reset Delay Circuit

■ INTERNAL OSCILLATOR

The internal oscillator circuit is designed to interface with a crystal (AT cut, 4 MHz max.) which is sufficient to drive it with various stability. As shown in Figure 8, a 22 pF capacitor is required from EXTAL to ground. Crystal specifications are given in Figure 9. Alternatively, EXTAL may be driven with a duty cycle of 50% with XTAL connected to ground.

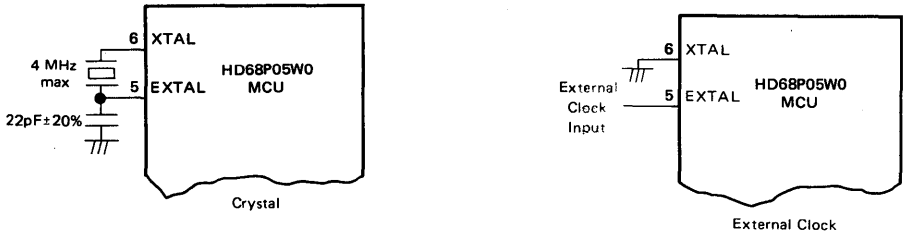
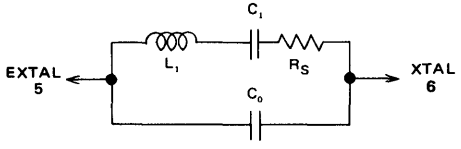


Figure 8 Internal Oscillator Options



AT – Cut Parallel Resonance Crystal
C_0 = 7 pF max.
f = 4 MHz (C_1 =22pF±20%)
R_S = 60 Ω max.

Figure 9 Crystal parameters

Table 6 Interrupt Priorities

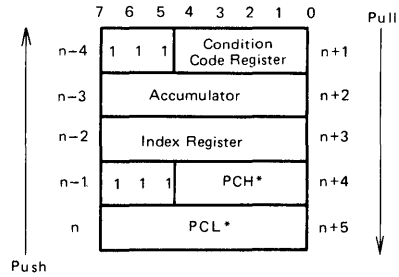
	Interrupt	Priority	Vector Address
4k bytes type	$\overline{RES}$	1	$0FFE, $0FFF
	SWI	2	$0FFC, $0FFD
	$\overline{INT_1}$	3	$0FFA, $0FFB
	Timer/$\overline{INT_2}$	4	$0FF8, $0FF9
	ICI	5	$0FF6, $0FF7
	OCI	6	$0FF4, $0FF5
	OFI	7	$0FF2, $0FF3

	Interrupt	Priority	Vector Address
8k bytes type	$\overline{RES}$	1	$1FFE, $1FFF
	SWI	2	$1FFC, $1FFD
	$\overline{INT_1}$	3	$1FFA, $1FFB
	Timer/$\overline{INT_2}$	4	$1FF8, $1FF9
	ICI	5	$1FF6, $1FF7
	OCI	6	$1FF4, $1FF5
	OFI	7	$1FF2, $1FF3

■ INTERRUPTS

The MCU can be interrupted in seven different ways: through external interrupt input pin ($\overline{INT_1}$ and $\overline{INT_2}$), internal timer interrupt request (Timer 1, ICI, OCI and OFI) and a software interrupt instruction (SWI). $\overline{INT_2}$ and Timer 1 are generated by the same vector address. When interrupt occurs, processing of the program is suspended, the present CPU state is pushed onto the stack. Figure 10 shows interrupt stacking order. Moreover, the interrupt mask bit (I) of the Condition Code Register is set and the external routine priority address is achieved from the special external vector address. After that, the external interrupt routine is executed. The interrupt service routines normally end with a return from interrupt (RTI) instruction which allows the CPU to resume processing of the program prior to the interrupt. The priority interrupts are shown in Table 6 with the vector address that contains the starting address of the appropriate interrupt routine. The interrupt sequence is shown as a flowchart in Figure 11.

Note that the Vector Address when using the 8k byte type EPROM is different from the 4k byte type EPROM.



* For subroutine calls, only PCH and PCL are stacked.

Figure 10 Interrupt Stacking Order

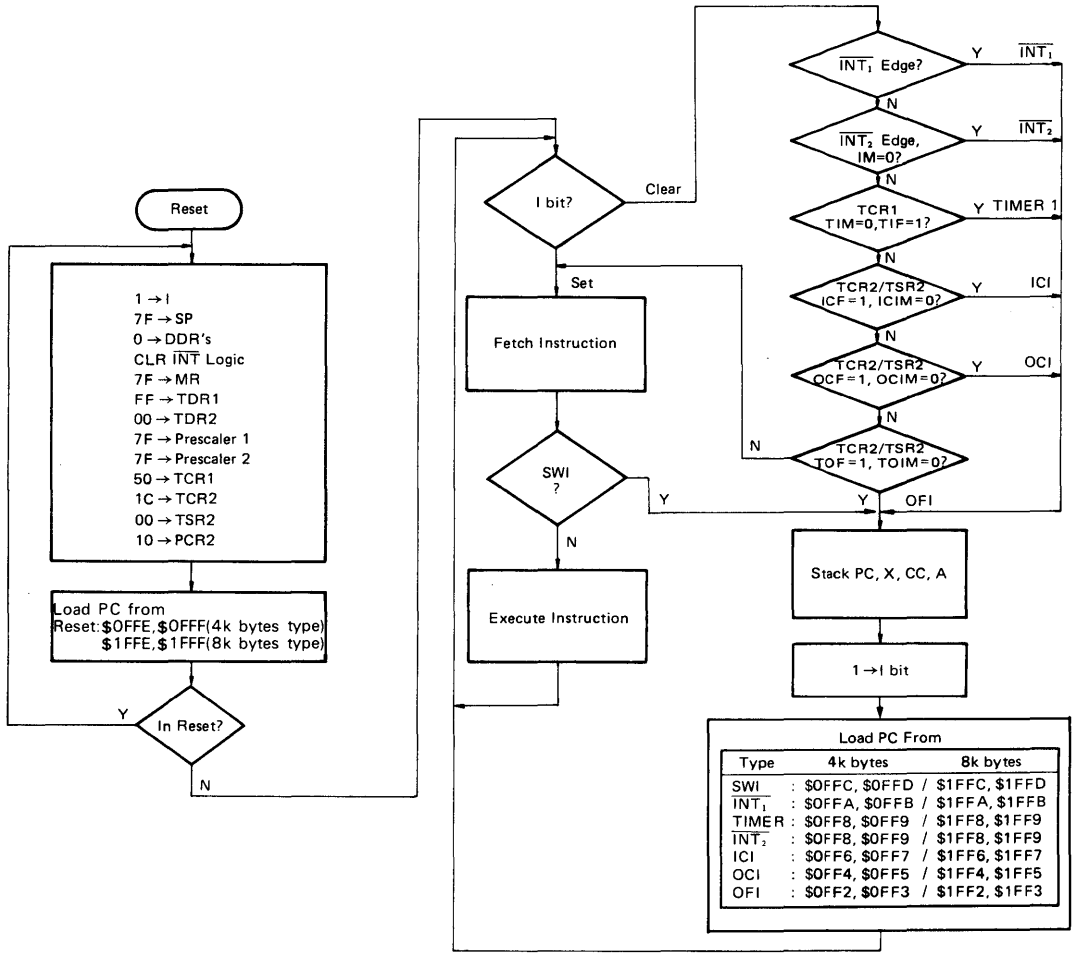
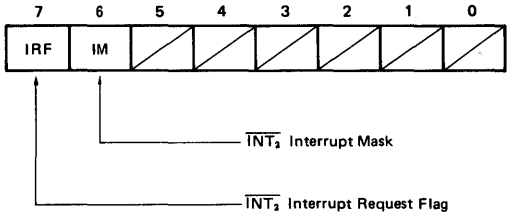


Figure 11 Interrupt Flowchart

- **Miscellaneous Register (MR: $000A)**

The vector address generated by the external interrupt ($\overline{INT_2}$) is the same as that of TIMER1 as shown in Table 6. The miscellaneous register (MR) controls the $\overline{INT_2}$ interrupt.

Miscellaneous Register (MR: $000A)



Bit 7 (IRF) of the MR is used as an $\overline{INT_2}$ interrupt request flag. $\overline{INT_2}$ interrupt occurs at the $\overline{INT_2}$ negative edge, and IRF is set. $\overline{INT_2}$ interrupt or not can be proved by checking IRF by software in the interrupt routine of the vector address ($FF8, $FF9). IRF should be reset by software (BCLR instruction).

Bit 6 (IM) of the MR is an $\overline{INT_2}$ interrupt mask bit. When IM is set, $\overline{INT_2}$ interrupt is disabled. $\overline{INT_2}$ interrupt is also disabled by bit (I) of the Condition Code Register (CC) like other interrupts.

IRF is available for both read and write. However, IRF is not writable by software. Therefore, $\overline{INT_2}$ interrupt cannot be requested by software. At reset, IRF is cleared and IM is set.

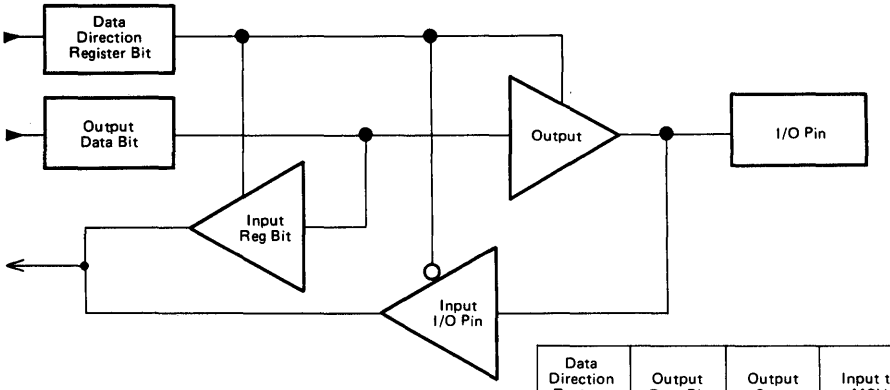
■ INPUT/OUTPUT

There are 23 input/output pins. All pins are controlled by the Data Direction Register and both input and output are programmable. When programmed as output, the latched output data is readable as input data, regardless of the logic levels at the output pin due to output loading (See Figure 12.) When Port B is programmed for output, it is capable of sinking 10 mA on each pin (V_{OL} max. = 1V). Furthermore, Port A is CMOS compatible as output. Ports B and C are CMOS compatible as inputs. Some examples of the Port connections are shown in Figure 13.

Port C_5 and C_6 are also used for Timer 2.

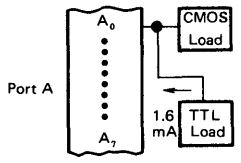
When Port C_5 is used as Timer 2 Input Capture (IC), Port C_5's DDR should be cleared (Port C_5 as input) and bit 4 (ICIM) in the Timer Control Register 2 (TCR2) should be cleared too. The Input Capture Register (ICR) stores the TDR2 when a Port C_5 input transition occures as defined by bit 1 (IDEG) of the TCR2.

When Port C_6 is used as Timer 2 Output Compare (OC), Port C_6's DDR should be set (Port C_6 as output). When the Output Compare Register (OCR) matches the TDR2, bit 0 (OLVL) in the TCR2 is set and OLVL will appear at Port C_6. Port C_6 is writable by software. But the writing by software is unavailable when a match between the TDR2 and the OCR is found at the same time.

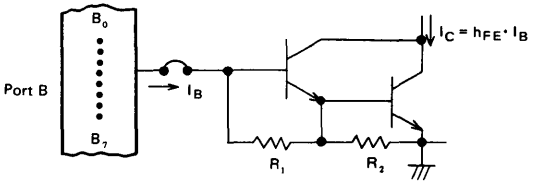


Data Direction Register Bit	Output Data Bit	Output State	Input to MCU
1	0	0	0
1	1	1	1
0	x	3-State	Pin

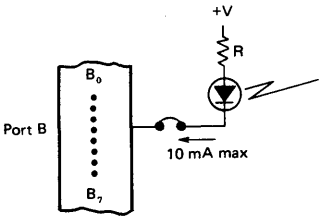
Figure 12 Typical Port I/O Circuitry



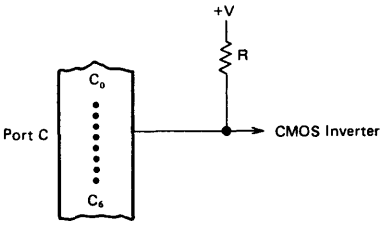
Port A Programmed as output(s), driving CMOS and TTL Load directly.
(a)



Port B Programmed as output(s), driving Darlington base directly.
(b)



Port B Programmed as output(s), driving LED(s) directly.
(c)



Port C Programmed as output(s), driving CMOS load(s), using external pull-up resistors.
(d)

Figure 13 Typical Port Connections

■ INPUT

Port D is usable as either TTL compatible inputs or a 4-channel input for an A/D converter. Fig. 14 shows port D logic configuration.

The Port D register at location $003 stores TTL compatible inputs. When using as analog inputs for an A/D converter, refer to "A/D CONVERTER".

■ A/D CONVERTER

The HD68P05W0 has an internal 8 bit A/D converter. The A/D converter, shown in Figure 15, includes 4 analog inputs (AN_0 to AN_3), the Result Register (ADRR) and the Control Status Register (ADCSR).

CAUTION

The MCU has circuitry to protect the inputs against damage due to high static voltages or electric field; however, the design of the input circuitry for the A/D converter, $AN_0 \sim AN_3$, V_{RH} and AV_{CC}, does not offer the same level of protection. Precautions should be taken to avoid applications of any voltage higher than maximum-rated voltage or handled in any environment producing high-static voltages.

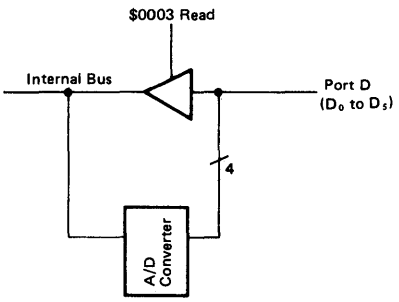


Figure 14 Port D

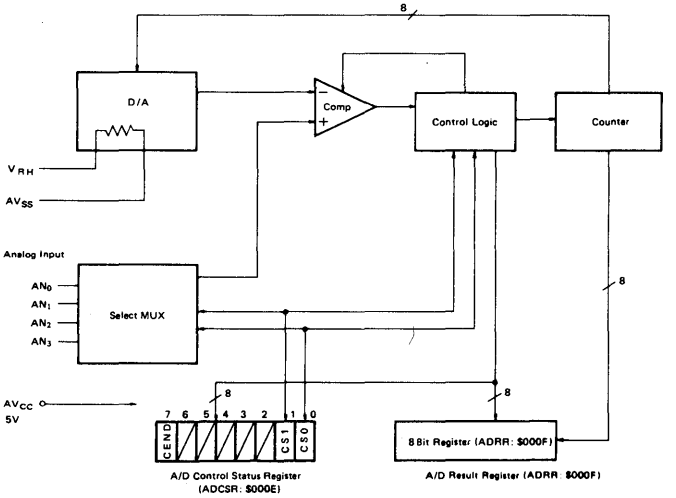


Figure 15 A/D Converter Block Diagram

- **Analog Input (AN_0 to AN_3)**

Analog inputs AN_0 to AN_3 accept analog voltages of 0V to 5V. The resolution is 8 bits (256 divisions) with a conversion time of 76 μs at 1 MHz. Analog conversion starts selecting analog inputs by bit 0 and bit 1 of the ADCSR analog input. Since the CPU is not required during conversion, other user programs can be executed.

Table 7 Analog Input Selection

ADCSR		Analog Input Signal
Bit 1	Bit 0	
0	0	AN_0
0	1	AN_1
1	0	AN_2
1	1	AN_3

- **A/D Control Status Register (ADCSR: $000E)**

The Control Status Register (ADCSR) is used to select analog input pin and confirm A/D conversion termination. An analog input pin is selected by bit 0 and bit 1 as shown in Table 7.

A/D conversion begins when the data is written into bit 0 and bit 1 of the ADCSR. When A/D conversion ends, bit 7 (CEND) is set. Bit 7 is reset after the ADRR is read. Even if bit 7 is set, A/D conversion execution still continues. To end the A/D conversion, the A/D Result Register (ADRR) stores the most current value. During A/D conversion execution, new data is written into the ADCSR selecting the input channel and the A/D conversion execution at that time is suspended. CEND is reset and new A/D conversion begins.

- **A/D Result Register (ADRR: $000F)**

When the A/D conversion ends, the result is set in the A/D Result Register ($000F). When CEND of the ADCSR is set, converted result is obtained by reading the ADRR. Furthermore, CEND is cleared.

■ STANDBY RAM

The portion from $020 to $027 of the RAM can be used for the standby RAM.

When using the standby RAM, V_{CC} Standby should remain above V_{SBB} (min) during powerdown. Consequently, power is provided only to the standby RAM and STBY PWR bit of the RAM Control Register. 8 byte RAM is sustained with small power dissipation. The RAM including the standby RAM is controlled by the RAM Control Register (RCR) or RAME pin.

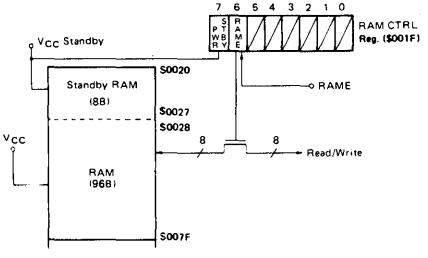
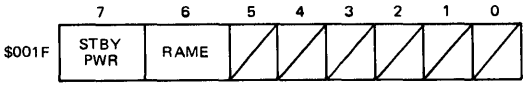


Figure 16 Standby RAM

- **RAM Control Register (RCR: $001F)**

This register at location $01F gives the status information about the RAM. When RAM Enable bit (RAME) is "0", the RAM is disabled. When V_{CC} Standby is greater than V_{SBB}, Standby Power bit (STBY PWR) is set and the standby RAM is sustained during powerdown.

RAM Control Register (RCR: $001F)



Bit 6 RAM Enable

RAME bit is set or cleared by either software or hardware. When the MCU is reset, RAME bit is set and the RAM is enabled. If RAME bit is cleared, the user can neither read nor write the RAM.

When the RAM is disabled (logic "0"), the RAM address is invalid.

Bit 7 Standby Power

STBY PWR bit is cleared whenever V_{CC} standby decreases below V_{SBB} (min). This bit is a read/write status bit that the user can read. When this bit is set, it indicates that the standby power is applied and data in the standby RAM is valid.

- **RAME Signal**

RAME bit in the RCR can be cleared when RAME pin goes "Low" by hardware (RAM is disabled). To make standby mode by hardware, set RAME pin "Low" during V_{CC} Standby remains above V_{SBB} (min) and powerdown sequence should be as shown in Fig. 17.

When RAME pin gets "Low" in the powerup state, RAME bit of the RCR is cleared and the RAM is disabled. During powerdown, RAME bit is sustained by V_{CC} Standby. When RAME pin gets "High" in the powerup state, RAME bit of the RCR is set and the RAM is enabled.

RAME pin can be used to control the RAM externally without software.

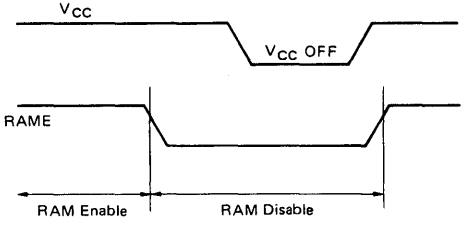


Figure 17 RAM Control Signal (RAME)

■ BIT MANIPULATION

The MCU has the ability to set or clear any single RAM or input/output port (except the data direction registers) with a single instruction (BSET and BCLR). Any bit in the page zero read only memory can be tested by using the BRSET and

BRCLR instructions, and the program branches as a result of its state. This capability to work with any bit in RAM, ROM or I/O allows the user to have individual flags in RAM or to handle single I/O bits as control lines. The example in Figure 18 shows the usefulness of the bit manipulation and test instructions. Assume that bit 0 of port A is connected to a zero crossing detector circuit and that bit 1 of port A is connected to the trigger of a TRIAC which powers the controlled hardware.

This program, which uses only seven bytes of ROM provides turn-on of the TRIAC within 14 microseconds of the zero crossing. The timer is also incorporated to provide turn-on at some later time which permits pulse-width modulation of the controlled power.

```
        .
        .
        .
        .
        .
SELF 1  BRCLR0, PORTA, SELF 1
        BSET 1, PORTA
        BCLR 1, PORTA
        .
        .
        .
        .
        .
```

Figure 18 Bit Manipulation Example

■ ADDRESSING MODES

The MCU has ten addressing modes available for use by the programmer. These modes are explained and illustrated briefly in the following paragraphs.

- **Immediate**

Refer to Figure 19. The immediate addressing mode accesses constants which do not change during program execution. Such instructions are two bytes long. The effective address (EA) is the PC and the operand is fetched from the byte following the opcode.

- **Direct**

Refer to Figure 20. In direct addressing, the address of the operand is contained in the secondbyte of the instruction. Direct addressing allows the user to directly address the lowest 256 bytes in memory. All RAM space, I/O registers and 128 bytes of ROM are located in page zero to take advantage of this efficient memory addressing mode.

- **Extended**

Refer to Figure 21. Extended addressing is used to reference any location in memory space. The EA is the contents of the two bytes following the opcode. Extended addressing instructions are three bytes long.

- **Relative**

Refer to Figure 22. The relative addressing mode applies only to the branch instructions. In this mode the contents of the byte following the opcode is added to the program counter when the branch is taken. EA = (PC) + 2 + Rel. Rel is the contents of the location following the instruction opcode with bit 7 being the sign bit. If the branch is not taken, Rel = 0, when a branch takes place, the program goes to somewhere within the range of +129 bytes to -127 bytes of the present instruction. These instructions are two bytes long.

- **Indexed (No Offset)**

Refer to Figure 23. This mode of addressing accesses the lowest 256 bytes of memory. These instructions are one byte long and their EA is the contents of the index register.

- **Indexed (8-bit Offset)**

Refer to Figure 24. The EA is calculated by adding the contents of the byte following the opcode to the contents of the index register. In this mode, 511 low memory locations are accessable. These instructions occupy two bytes.

- **Indexed (16-bit Offset)**

Refer to Figure 25. This addressing mode calculates the EA by adding the contents of the two bytes following the opcode to the index register. Thus, the entire memory space may be accessed. Instructions which use this addressing mode are three bytes long.

- **Bit Set/Clear**

Refer to Figure 26. This mode of addressing applies to instructions which can set or clear any bit on page zero. The lower three bits in the opcode specify the bit to be set or cleared while the byte following the opcode specifies the address in page zero.

- **Bit Test and Branch**

Refer to Figure 27. This mode of addressing applies to instructions which can test any bit in the first 256 locations ($0000 through $00FF) and branch to any location relative to the PC. The byte to be tested is addressed by the byte following the opcode. The individual bit within that byte to be tested is addressed by the lower three bits of the opcode. The third byte is the relative address to be added to the program counter if the branch condition is met. These instructions are three bytes long. The value of the bit to be tested is written to the carry bit in the condition code register.

- **Implied**

Refer to Figure 28. The implied mode of addressing has no EA. All of the information necessary to execute an instruction is contained in the opcode. Direct operations on the accumulator and the index register are included in this mode of addressing. In addition, control instructions such as SWI and RTI belong to this group. All implied addressing instructions are one byte long.

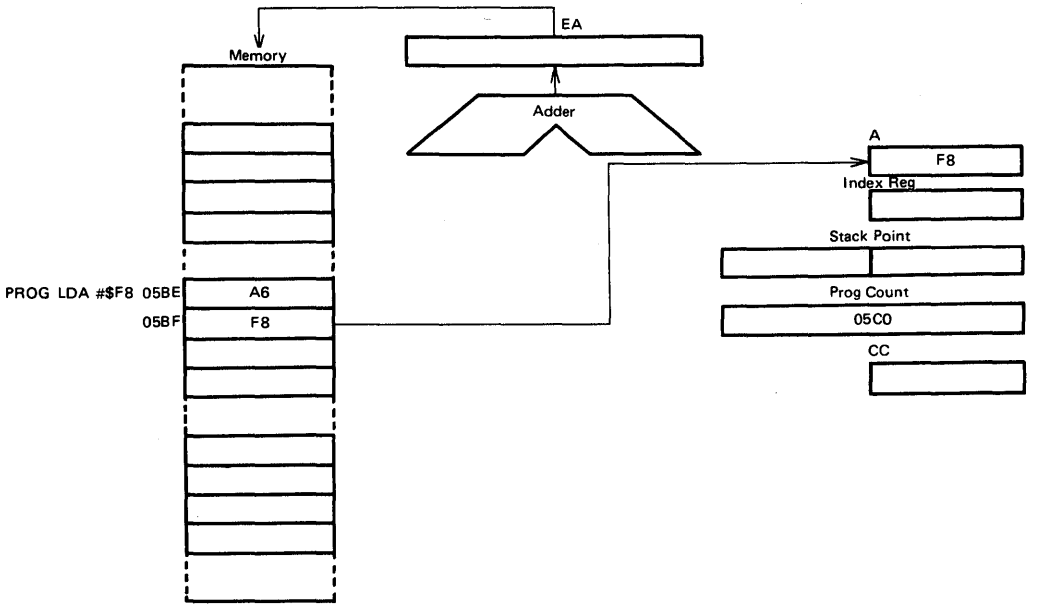


Figure 19 Immediate Addressing Example

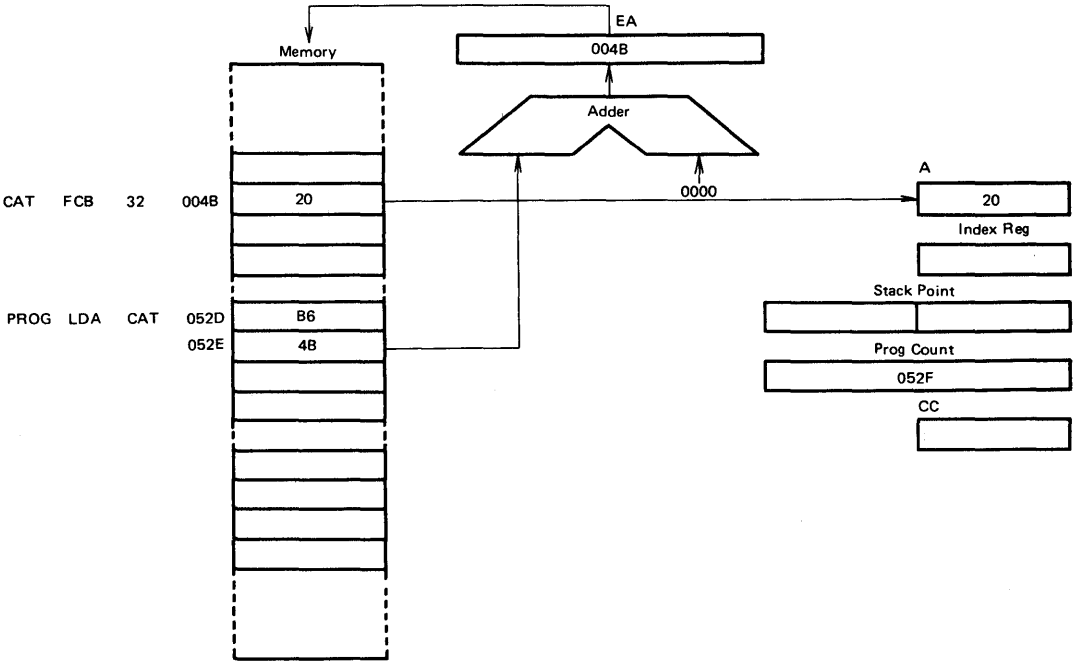


Figure 20 Direct Addressing Example

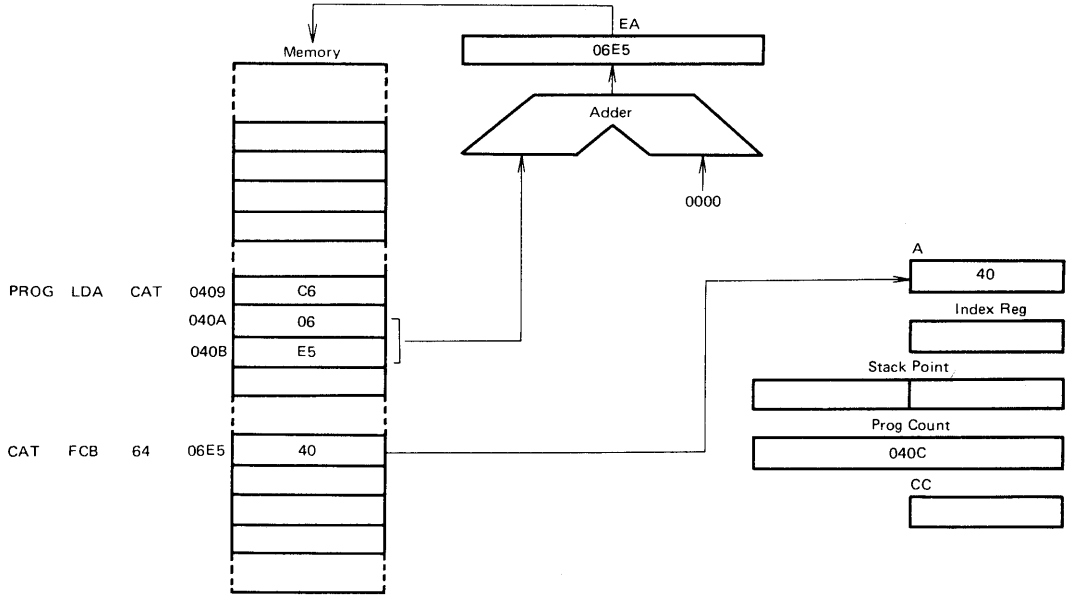


Figure 21 Extended Addressing Example

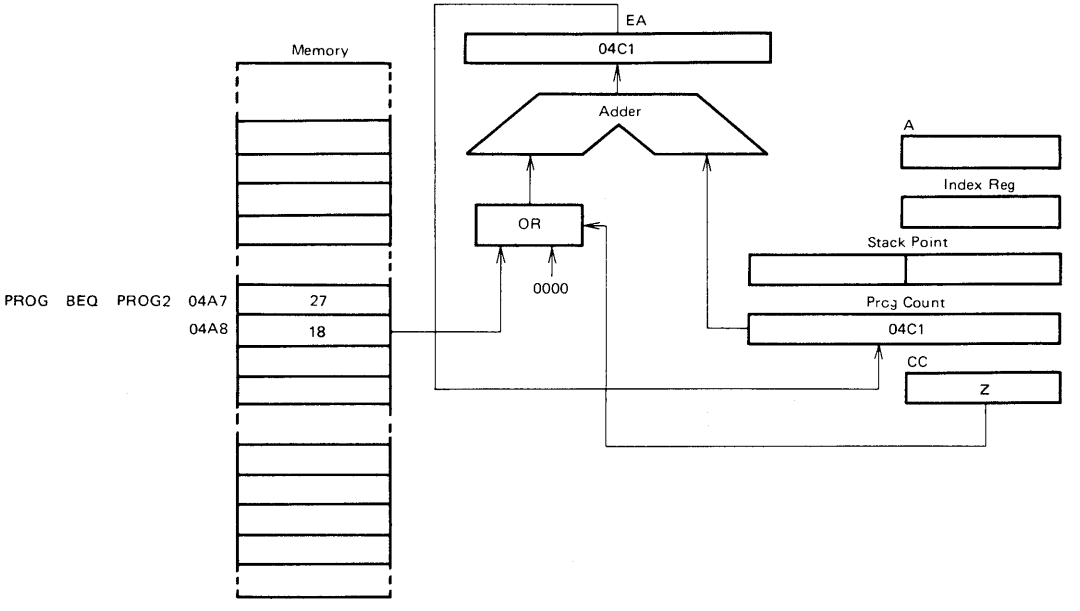


Figure 22 Relative Addressing Example

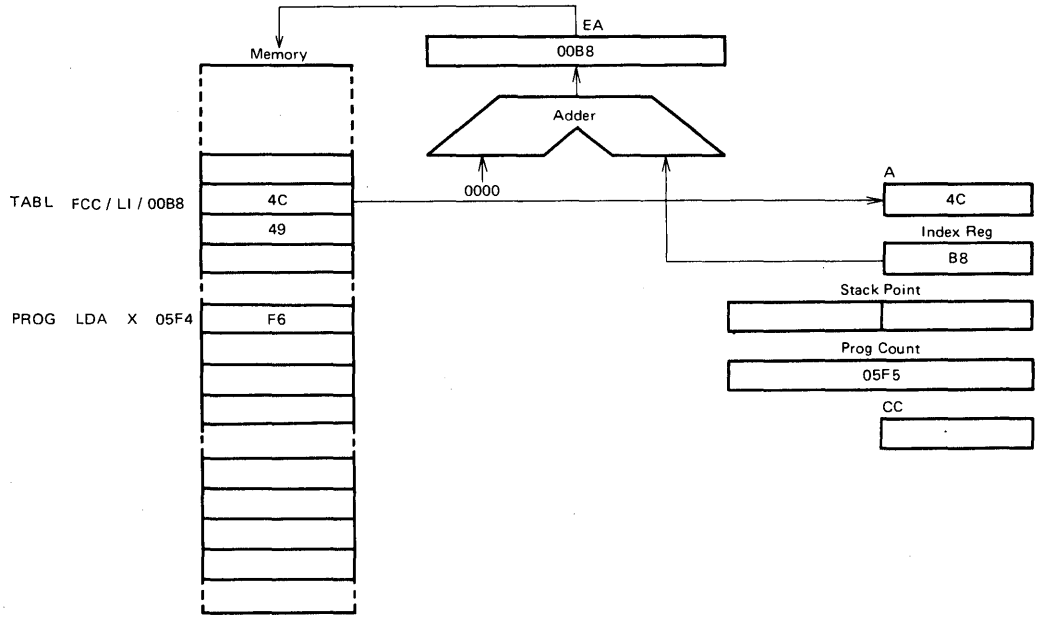


Figure 23 Indexed (No Offset) Addressing Example

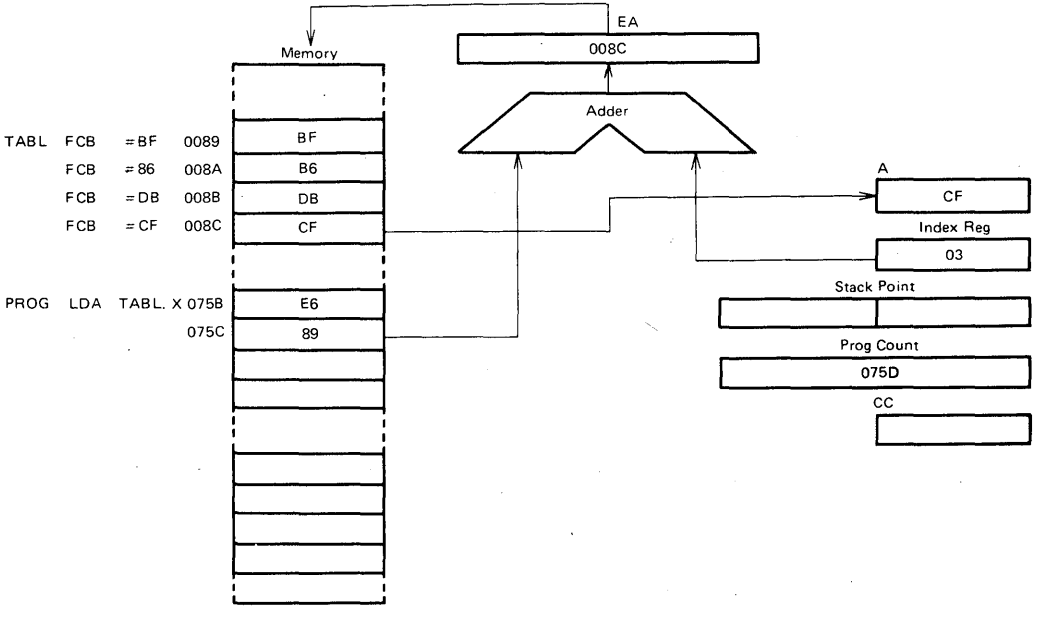


Figure 24 Indexed (8-Bit Offset) Addressing Example

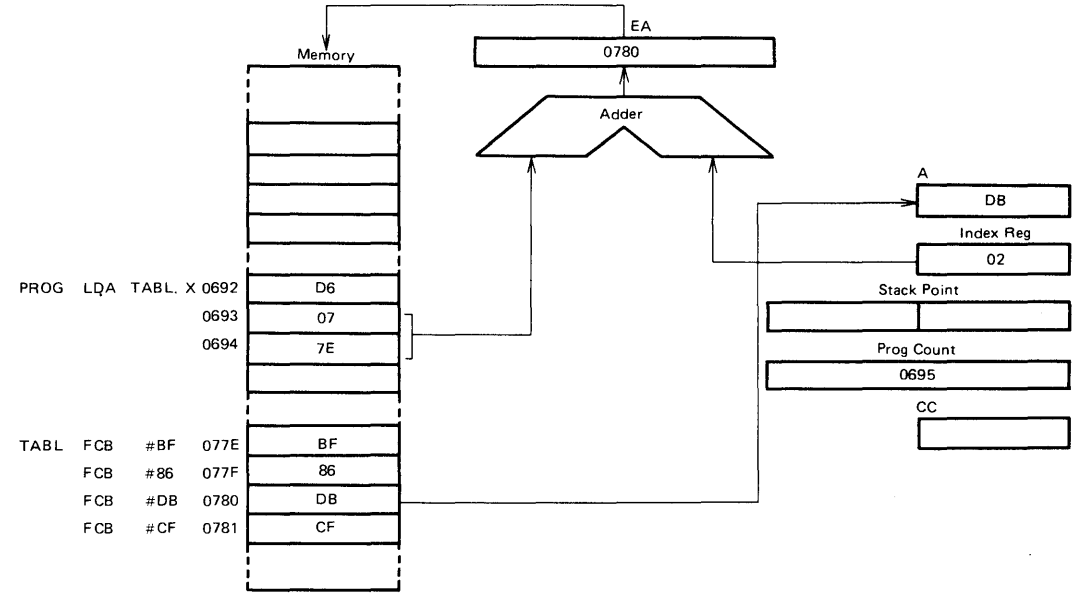


Figure 25 Indexed (16-Bit Offset) Addressing Example

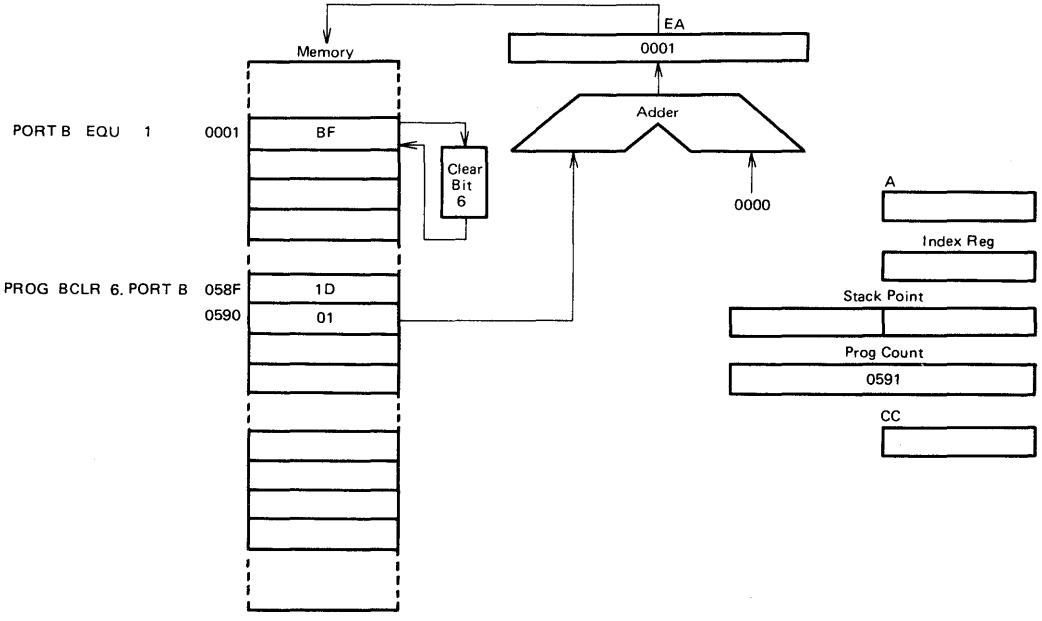


Figure 26 Bit Set/Clear Addressing Example

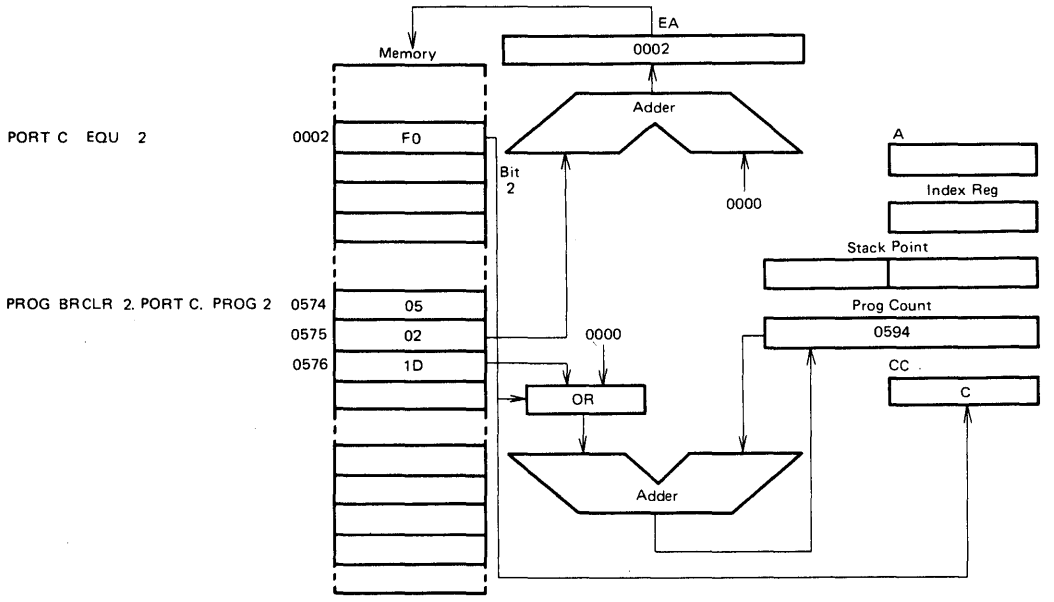


Figure 27 Bit Test and Branch Addressing Example

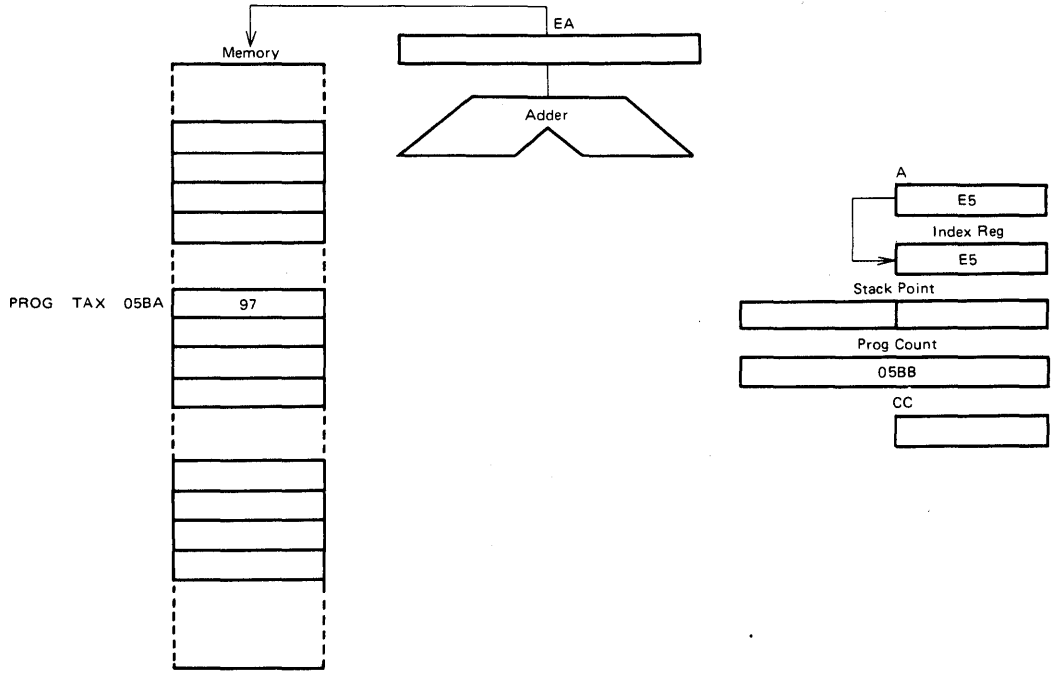


Figure 28 Implied Addressing Example

■ INSTRUCTION SET

The MCU has a set of 59 basic instructions. These instructions can be divided into five different types; register/memory, read/modify/write, branch, bit manipulation and control. Each instruction is briefly explained below. All of the instructions within a given type are presented in individual tables.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory by using one of the addressing modes. The jump unconditional (JMP) and jump to subroutine (JSR) instructions have no register operand. Refer to Table 8.

• Read/Modify/Write Instructions

These instructions read a memory location or a register, modify or test its contents and write the modified value back to the memory or register. The TST instruction for test of negative or zero is an exception to the read/modify/write instructions since it does not perform the write. Refer to Table 9.

• Branch Instructions

The branch instructions cause a branch from a program when a certain condition is met. Refer to Table 10.

• Bit Manipulation Instructions

These instructions are used on any bit in the first 256 bytes of the memory. One group either sets or clears. The other group performs the bit test and branch operations. Refer to Table 11.

• Control Instructions

The control instructions control the MCU operations during program execution. Refer to Table 12.

• Alphabetical Listing

The complete instruction set is given in alphabetical order in Table 13.

• Opcode Map

Table 14 is an opcode map for the instructions used on the MCU.

Table 8 Register/Memory Instructions

Function	Mnemonic	Addressing Modes																	
		Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Load A from Memory	LDA	A6	2	2	B6	2	4	C6	3	5	F6	1	4	E6	2	5	D6	3	6
Load X from Memory	LDX	AE	2	2	BE	2	4	CE	3	5	FE	1	4	EE	2	5	DE	3	6
Store A in Memory	STA	–	–	–	B7	2	5	C7	3	6	F7	1	5	E7	2	6	D7	3	7
Store X in Memory	STX	–	–	–	BF	2	5	CF	3	6	FF	1	5	EF	2	6	DF	3	7
Add Memory to A	ADD	AB	2	2	BB	2	4	CB	3	5	FB	1	4	EB	2	5	DB	3	6
Add Memory and Carry to A	ADC	A9	2	2	B9	2	4	C9	3	5	F9	1	4	E9	2	5	D9	3	6
Subtract Memory	SUB	A0	2	2	B0	2	4	C0	3	5	F0	1	4	E0	2	5	D0	3	6
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	4	C2	3	5	F2	1	4	E2	2	5	D2	3	6
AND Memory to A	AND	A4	2	2	B4	2	4	C4	3	5	F4	1	4	E4	2	5	D4	3	6
OR Memory with A	ORA	AA	2	2	BA	2	4	CA	3	5	FA	1	4	EA	2	5	DA	3	6
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	4	C8	3	5	F8	1	4	E8	2	5	D8	3	6
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	4	C1	3	5	F1	1	4	E1	2	5	D1	3	6
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	4	C3	3	5	F3	1	4	E3	2	5	D3	3	6
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	4	C5	3	5	F5	1	4	E5	2	5	D5	3	6
Jump Unconditional	JMP	–	–	–	BC	2	3	CC	3	4	FC	1	3	EC	2	4	DC	3	5
Jump to Subroutine	JSR	–	–	–	BD	2	7	CD	3	8	FD	1	7	ED	2	8	DD	3	9

Symbols:
Op: Operation Abbreviation
: Instruction Statement

Table 9 Read/Modify/Write Instructions

Function	Mnemonic	Addressing Modes														
		Implied (A)			Implied (X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Increment	INC	4C	1	4	5C	1	4	3C	2	6	7C	1	6	6C	2	7
Decrement	DEC	4A	1	4	5A	1	4	3A	2	6	7A	1	6	6A	2	7
Clear	CLR	4F	1	4	5F	1	4	3F	2	6	7F	1	6	6F	2	7
Complement	COM	43	1	4	53	1	4	33	2	6	73	1	6	63	2	7
Negate (2's Complement)	NEG	40	1	4	50	1	4	30	2	6	70	1	6	60	2	7
Rotate Left Thru Carry	ROL	49	1	4	59	1	4	39	2	6	79	1	6	69	2	7
Rotate Right Thru Carry	ROR	46	1	4	56	1	4	36	2	6	76	1	6	66	2	7
Logical Shift Left	LSL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Logical Shift Right	LSR	44	1	4	54	1	4	34	2	6	74	1	6	64	2	7
Arithmetic Shift Right	ASR	47	1	4	57	1	4	37	2	6	77	1	6	67	2	7
Arithmetic Shift Left	ASL	48	1	4	58	1	4	38	2	6	78	1	6	68	2	7
Test for Negative or Zero	TST	4D	1	4	5D	1	4	3D	2	6	7D	1	6	6D	2	7

Symbols:
Op: Operation Abbreviation
: Instruction Statement

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 10 Branch Instructions

Function	Mnemonic	Relative Addressing Mode		
		Op Code	# Bytes	# Cycles
Branch Always	BRA	20	2	4
Branch Never	BRN	21	2	4
Branch IF Higher	BHI	22	2	4
Branch IF Lower or Same	BLS	23	2	4
Branch IF Carry Clear	BCC	24	2	4
(Branch IF Higher or Same)	(BHS)	24	2	4
Branch IF Carry Set	BCS	25	2	4
(Branch IF Lower)	(BLO)	25	2	4
Branch IF Not Equal	BNE	26	2	4
Branch IF Equal	BEQ	27	2	4
Branch IF Half Carry Clear	BHCC	28	2	4
Branch IF Half Carry Set	BHCS	29	2	4
Branch IF Plus	BPL	2A	2	4
Branch IF Minus	BMI	2B	2	4
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	4
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	4
Branch IF Interrupt Line is Low	BIL	2E	2	4
Branch IF Interrupt Line is High	BIH	2F	2	4
Branch to Subroutine	BSR	AD	2	8

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 11 Bit Manipulation Instructions

Function	Mnemonic	Addressing Modes					
		Bit Set/Clear			Bit Test and Branch		
		Op Code	# Bytes	# Cycles	Op Code	# Bytes	# Cycles
Branch IF Bit n is set	BRSET n (n=0 ..... 7)	–	–	–	2•n	3	10
Branch IF Bit n is clear	BRCLR n (n=0 .....7)	–	–	–	01+2•n	3	10
Set Bit n	BSET n (n=0 ..... 7)	10+2•n	2	7	–	–	–
Clear bit n	BCLR n (n=0 ..... 7)	11+2•n	2	7	–	–	–

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 12 Control Instructions

Function	Mnemonic	Implied		
		Op Code	# Bytes	# Cycles
Transfer A to X	TAX	97	1	2
Transfer X to A	TXA	9F	1	2
Set Carry Bit	SEC	99	1	2
Clear Carry Bit	CLC	98	1	2
Set Interrupt Mask Bit	SEI	9B	1	2
Clear Interrupt Mask Bit	CLI	9A	1	2
Software Interrupt	SWI	83	1	11
Return from Subroutine	RTS	81	1	6
Return from Interrupt	RTI	80	1	9
Reset Stack Pointer	RSP	9C	1	2
No-Operation	NOP	9D	1	2

Symbols: Op: Operation Abbreviation #: Instruction Statement

Table 13 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
ADC		x	x	x		x	x	x			∧	•	∧	∧	∧
ADD		x	x	x		x	x	x			∧	•	∧	∧	∧
AND		x	x	x		x	x	x			•	•	∧	∧	•
ASL	x		x			x	x				•	•	∧	∧	∧
ASR	x		x			x	x				•	•	∧	∧	∧
BCC					x						•	•	•	•	•
BCLR									x		•	•	•	•	•
BCS					x						•	•	•	•	•
BEQ					x						•	•	•	•	•
BHCC					x						•	•	•	•	•
BHCS					x						•	•	•	•	•
BHI					x						•	•	•	•	•
BHS					x						•	•	•	•	•
BIH					x						•	•	•	•	•
BIL					x						•	•	•	•	•
BIT		x	x	x		x	x	x			•	•	∧	∧	•
BLO					x						•	•	•	•	•
BLS					x						•	•	•	•	•
BMC					x						•	•	•	•	•
BMI					x						•	•	•	•	•
BMS					x						•	•	•	•	•
BNE					x						•	•	•	•	•
BPL					x						•	•	•	•	•
BRA					x						•	•	•	•	•
BRN					x						•	•	•	•	•
BRCLR										x	•	•	•	•	∧
BRSET										x	•	•	•	•	∧
BSET									x		•	•	•	•	•
BSR					x						•	•	•	•	•
CLC	x										•	•	•	•	0
CLI	x										•	0	•	•	•
CLR	x		x			x	x				•	•	0	1	•
CMP		x	x	x		x	x	x			•	•	∧	∧	∧
COM	x		x			x	x				•	•	∧	∧	1
CPX		x	x	x		x	x	x			•	•	∧	∧	∧
DEC	x		x			x	x				•	•	∧	∧	•
EOR		x	x	x		x	x	x			•	•	∧	∧	•
INC	x		x			x	x				•	•	∧	∧	•
JMP			x	x		x	x	x			•	•	•	•	•
JSR			x	x		x	x	x			•	•	•	•	•
LDA		x	x	x		x	x	x			•	•	∧	∧	•
LDX		x	x	x		x	x	x			•	•	∧	∧	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected

(to be continued)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 13 Instruction Set

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8 Bits)	Indexed (16 Bits)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
LSL	x		x			x	x				•	•	∧	∧	∧
LSR	x		x			x	x				•	•	0	∧	∧
NEG	x		x			x	x				•	•	∧	∧	∧
NOP	x										•	•	•	•	•
ORA		x	x	x		x	x	x			•	•	∧	∧	•
ROL	x		x			x	x				•	•	∧	∧	∧
ROR	x		x			x	x				•	•	∧	∧	∧
RSP	x										•	•	•	•	•
RTI	x										?	?	?	?	?
RTS	x										•	•	•	•	•
SBC		x	x	x		x	x	x			•	•	∧	∧	∧
SEC	x										•	•	•	•	1
SEI	x										•	1	•	•	•
STA			x	x		x	x	x			•	•	∧	∧	•
STX			x	x		x	x	x			•	•	∧	∧	•
SUB		x	x	x		x	x	x			•	•	∧	∧	∧
SWI	x										•	1	•	•	•
TAX	x										•	•	•	•	•
TST	x		x			x	x				•	•	∧	∧	•
TXA	x										•	•	•	•	•

Condition Code Symbols:
- H Half Carry (From Bit 3)
- I Interrupt Mask
- N Negative (Sign Bit)
- Z Zero
- C Carry/Borrow
- ∧ Test and Set if True, Cleared Otherwise
- • Not Affected
- ? Load CC Register From Stack

Table 14 Opcode Map

	Bit Manipulation		Brnch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	←HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*	–	SUB						0
1	BRCLR0	BCLR0	BRN	–					RTS*	–	CMP						1
2	BRSET1	BSET1	BHI	–					–	–	SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*	–	CPX						3 L
4	BRSET2	BSET2	BCC	LSR					–	–	AND						4 O
5	BRCLR2	BCLR2	BCS	–					–	–	BIT						5 W
6	BRSET3	BSET3	BNE	ROR					–	–	LDA						6
7	BRCLR3	BCLR3	BEQ	ASR					–	TAX	–	STA(+1)					7
8	BRSET4	BSET4	BHCC	LSL/ASL					–	CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL					–	SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC					–	CLI	ORA						A
B	BRCLR5	BCLR5	BMI	–					–	SEI	ADD						B
C	BRSET6	BSET6	BMC	INC					–	RSP	–	JMP(–1)					C
D	BRCLR6	BCLR6	BMS	TST					–	NOP	BSR*	JSR(+3)					D
E	BRSET7	BSET7	BIL	–					–	–	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					–	TXA	–	STX(+1)					F
	3/10	2/7	2/4	2/6	1/4	1/4	2/7	1/6	1/*	1/2	2/2	2/4	3/5	3/6	2/5	1/4	

[NOTE] 1. Undefined opcodes are marked with "–".
2. The number at the bottom of each column denotes the number of bytes and the number of cycles required (Bytes/Cycles). Mnemonics followed by a "*" require a different number of cycles as follows:
RTI 9
RTS 6
SWI 11
BSR 8
3. ( ) indicates that the number in parenthesis must be added to the cycle count for that instruction.

■ HD68P05W0 USED FOR HD6805W1

The HD6805W1 provides mask option of the internal oscillator and low voltage inhibit, while the HD68P05W0 provides only crystal option and without low voltage inhibit function.

The address from $0F7A to $0FF1 cannot be used for user program because the self test program of the HD6805W1 (on-chip ROM version) is located at these addresses.

In order to be pin compatible with the HD6805W1, the address of the HD68P05W0's ROM must be located at $0080 – $0FFF. Memory addresses $1000 to $1FFF should not be usable.

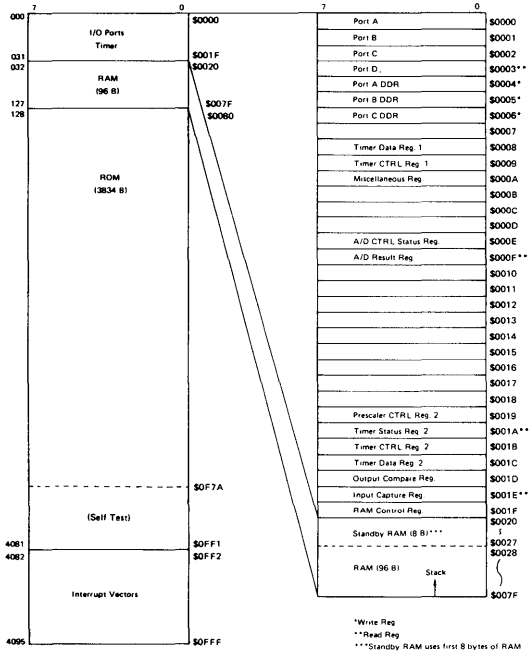


Figure 29 MCU Memory Structure (For 4k bytes)

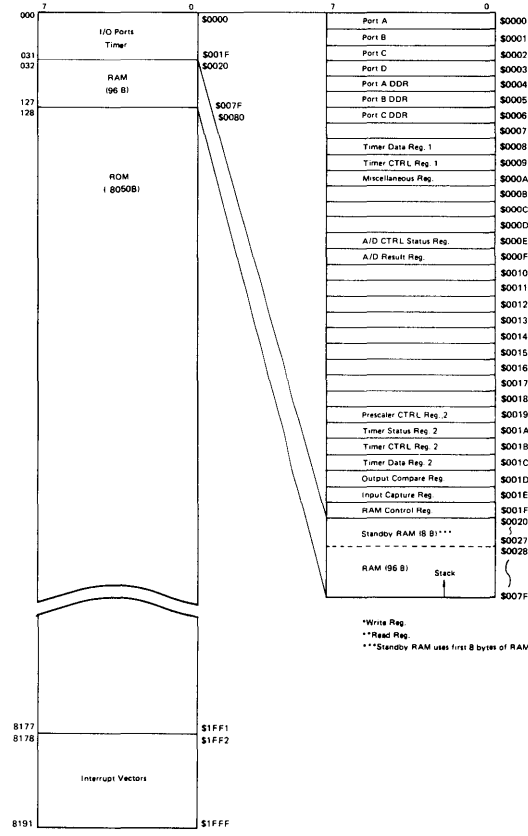


Figure 30 MCU Memory Structure (For 8k bytes)

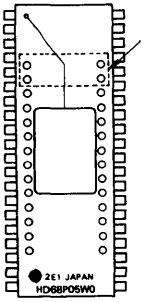
– CAUTION –
This 8k bytes type should not be used debugging on-chip ROM of the HD6805W1.

■ PRECAUTION TO USE EPROM ON THE PACKAGE 8-BIT SINGLE-CHIP MICROCOMPUTER

As this microcomputer takes a special packaging type with pin sockets on its surface, pay attention to the followings;

(1) Do not apply higher electro-static voltage or serge voltage etc. than maximum rating, or it may cause permanent damage to the device.

(2) There are 28 pin sockets on its surface. When using 32k EPROM (24 pins), let the index-side four pins open.



Let the index-side four pins open. When using 24 pin EPROM, match its index and insert it into lower 24 pin sockets.

(3) When assembling this LSI into user's system products as well as the mask ROM type 8-bit single-chip microcomputer, pay attention to the followings to keep the good ohmic contact between EPROM pins and pin sockets.

(a) When soldering on a printed circuit board, etc., keep its condition under 250°C within 10 seconds. Over-time/temperature may cause the bonding solder of socket pins to meet and the sockets may drop.

(b) Keep out detergent or coater from the pin sockets at aft-solder flux removal or board coating. The flux or coater may make pin socket contactivity worse.

(c) Avoid the permanent use of this LSI under the ever-vibratory place and system.

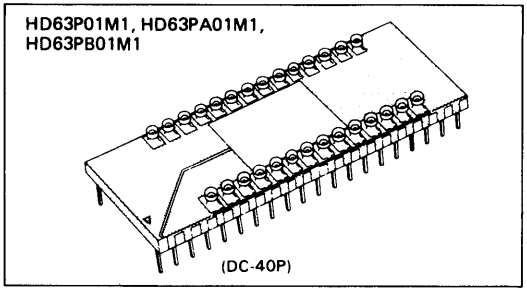
(d) Repeating insertion/removal of EPROMs may damage the contactivity of the pin sockets, so it is recommended to assemble new ones to your system products.

Ask our sales agent about anything unclear.

HD63P01M1, HD63PA01M1, HD63PB01M1
CMOS MCU (Microcomputer Unit)

The HD63P01M1 is an 8-bit single chip Microcomputer Unit (MCU) which has 4096 bytes or 8192 bytes of EPROM on the package. It is pin and function (except ROM) compatible with the HD6301V1. The HD63P01M1 can be used to emulate the HD6301V1 for software development or it can be used in production to allow for easy firmware changes with minimum delay.

– The specifications for HD63PA01M1 and HD63PB01M1 are preliminary. –



■ FEATURES

- Pin Compatible with HD6301V1
- On Chip Function Compatible with HD6301V1
  - 128 Bytes of RAM
  - 29 Parallel I/O
  - 16 Bit Programmable Timer
  - Serial Communication Interface
- Low Power Consumption Mode
  Sleep Mode, Standby Mode
- Minimum Instruction Cycle Time
  1μs (f = 1MHz), 0.67μs (f = 1.5MHz),
  0.5μs (f = 2MHz)
- Bit Manipulation, Bit Test Instruction
- Protection from System Upset
  Address Trap, Op-Code Trap
- Up to 65k Words Address Space
- Applicable to 4k or 8k Bytes of EPROM
  4096 Bytes: HN482732A
  8192 Bytes : HN482764, HN27C64

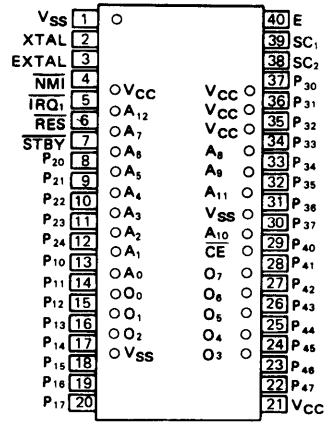
■ TYPE OF PRODUCTS

Type No.	Bus Timing	EPROM Type No.
HD63P01M1	1MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PA01M1*	1.5MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PB01M1*	2MHz	HN482732A-25, HN482764, HN27C64-25

* Preliminary

■ PIN ARRANGEMENT

HD63P01M1, HD63PA01M1, HD63PB01M1

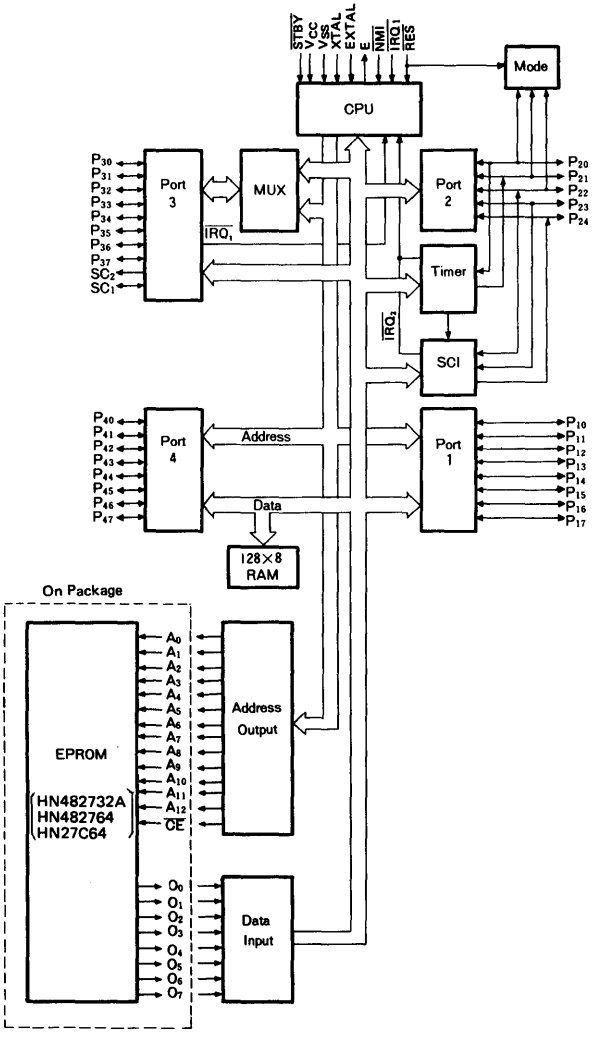


(Top View)

(NOTE) EPROM is not included.

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BLOCK DIAGRAM**



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +150	°C

(NOTE) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out} : $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC}-0.5	–	V_{CC} +0.3	V
	EXTAL			V_{CC}x0.7	–		
	Other Inputs			2.0	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{IRQ_1}$, $\overline{RES}$, $\overline{STBY}$	$\|I_{in}\|$	V_{in} =0.5~V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	P_{10}~P_{17}, P_{20}~P_{24}, P_{30}~P_{37}, P_{40}~P_{47}, $\overline{IS3}$	$\|I_{TSI}\|$	V_{in} =0.5~V_{CC} -0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	All Outputs	V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input Capacitance	All Inputs	C_{in}	V_{in}=0V, f=1.0MHz, Ta = 25°C	–	–	12.5	pF
Standby Current	Non Operation	I_{CC}		–	2.0	15.0	μA
Current Dissipation*		I_{CC}	Operating (f=1MHz**)	–	6.0	10.0	mA
			Sleeping (f=1MHz**)	–	1.0	2.0	
RAM Stand-By Voltage		V_{RAM}		2.0	–	–	V

* V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V, I_{CC} of EPROM is not included.

** Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at f = x MHz operation are decided according to the following formula;

typ. value (f = x MHz) = typ. value (f = 1MHz) x x
max. value (f = x MHz) = max. value (f = 1MHz) x x
(both the sleeping and operating)

- **AC CHARACTERISTICS (V_{CC} = 5.0V±10%, V_{SS} = 0V, Ta = 0~+70°C, unless otherwise noted.)**

BUS TIMING

Item		Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}		1	–	10	0.666	–	10	0.5	–	10	μs
Address Strobe Pulse Width "High" *		PW_{ASH}		220	–	–	150	–	–	110	–	–	ns
Address Strobe Rise Time		t_{ASr}		–	–	20	–	–	20	–	–	20	ns
Address Strobe Fall Time		t_{ASf}		–	–	20	–	–	20	–	–	20	ns
Address Strobe Delay Time*		t_{ASD}		60	–	–	40	–	–	20	–	–	ns
Enable Rise Time		t_{Er}		–	–	20	–	–	20	–	–	20	ns
Enable Fall Time		t_{Ef}		–	–	20	–	–	20	–	–	20	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address Strobe to Enable Delay* Time		t_{ASED}		60	–	–	40	–	–	20	–	–	ns
Address Delay Time		t_{AD1}		–	–	250	–	–	190	–	–	160	ns
		t_{AD2}	Fig. 1	–	–	250	–	–	190	–	–	160	ns
Address Delay Time for Latch*		t_{ADL}	Fig. 2	–	–	250	–	–	190	–	–	160	ns
Data Set-up Time	Write	t_{DSW}		230	–	–	150	–	–	100	–	–	ns
	Read	t_{DSR}		80	–	–	60	–	–	50	–	–	ns
Data Hold Time	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
	Write	t_{HW}		20	–	–	20	–	–	20	–	–	ns
Address Set-up Time for Latch*		t_{ASL}		60	–	–	40	–	–	20	–	–	ns
Address Hold Time for Latch		t_{AHL}		30	–	–	20	–	–	20	–	–	ns
Address Hold Time		t_{AH}		20	–	–	20	–	–	20	–	–	ns
A_0 ~ A_7 Set-up Time Before E*		t_{ASM}		200	–	–	110	–	–	60	–	–	ns
Peripheral Read Access Time	Non-Multiplexed Bus*	(t_{ACCN})		–	–	650	–	–	395	–	–	270	ns
	Multiplexed Bus*	(t_{ACCM})		–	–	650	–	–	395	–	–	270	ns
Oscillator stabilization Time		t_{RC}	Fig. 10	20	–	–	20	–	–	20	–	–	ms
Processor Control Set-up Time		t_{PCS}	Fig. 11	200	–	–	200	–	–	200	–	–	ns

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set-up Time	Port 1, 2, 3, 4	t_{PDSU}	Fig. 3	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Port 1, 2, 3, 4	t_{PDH}	Fig. 3	200	–	–	200	–	–	200	–	–	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Negative Transition		t_{OSD1}	Fig. 5	–	–	300	–	–	300	–	–	300	ns
Delay Time, Enable Positive Transition to $\overline{OS3}$ Positive Transition		t_{OSD2}	Fig. 5	–	–	300	–	–	300	–	–	300	ns
Delay Time, Enable Negative Transition to Peripheral Data Valid	Port 1, 2*, 3, 4	t_{PWD}	Fig. 4	–	–	300	–	–	300	–	–	300	ns
Input Strobe Pulse Width		t_{PWIS}	Fig. 6	200	–	–	200	–	–	200	–	–	ns
Input Data Hold Time	Port 3	t_{IH}	Fig. 6	150	–	–	150	–	–	150	–	–	ns
Input Data Setup Time	Port 3	t_{IS}	Fig. 6	0	–	–	0	–	–	0	–	–	ns

* Except P_{21}

TIMER, SCI TIMING

Item	Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Timer Input Pulse Width	t_{PWT}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time, Enable Positive Transition to Timer Out	t_{TOD}	Fig. 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	t_{Scyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Input Clock Pulse Width	t_{PWSCK}		0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}

MODE PROGRAMMING

Item	Symbol	Test Condition	HD63P01M1			HD63PA01M1			HD63PB01M1			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{RES}$ "Low" Pulse Width	PW_{RSTL}	Fig. 8	3	–	–	3	–	–	3	–	–	t_{cyc}
Mode Programming Set-up Time	t_{MPS}		2	–	–	2	–	–	2	–	–	t_{cyc}
Mode Programming Hold Time	t_{MPH}		150	–	–	150	–	–	150	–	–	ns

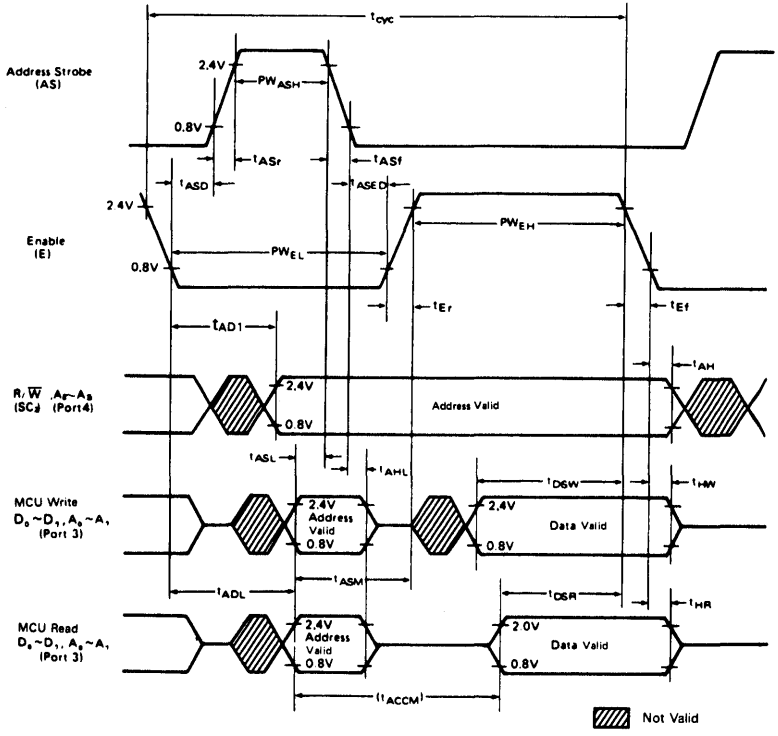


Figure 1 Expanded Multiplexed Bus Timing

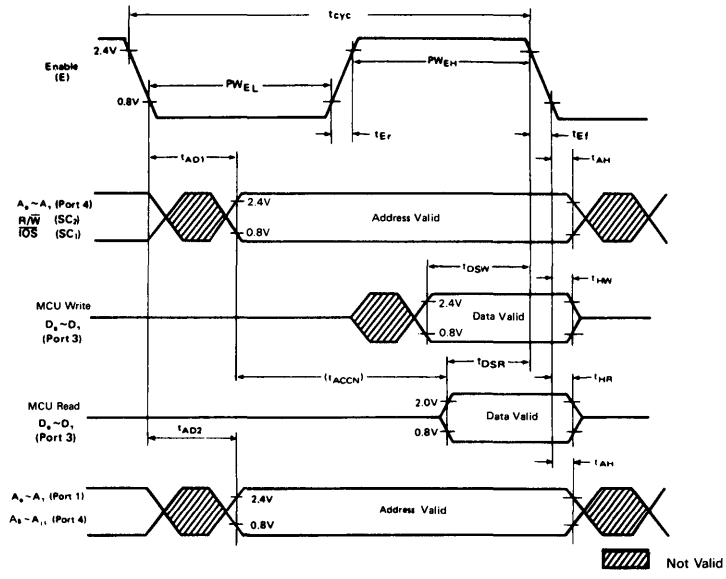
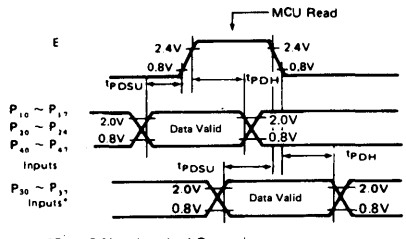
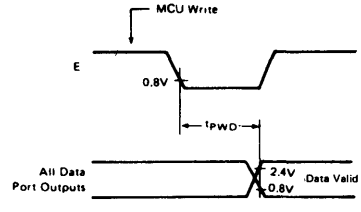


Figure 2 Expanded Non-Multiplexed Bus Timing



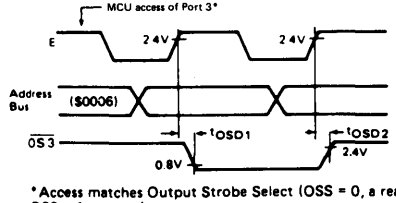
*Port 3 Non-Latched Operation

Figure 3 Port Data Set-up and Hold Times (MCU Read)



Note) Port 2: Except P21

Figure 4 Port Data Delay Times (MCU Write)



*Access matches Output Strobe Select (OSS = 0, a read; OSS = 1, a write)

Figure 5 Port 3 Output Strobe Timing (Single Chip Mode)

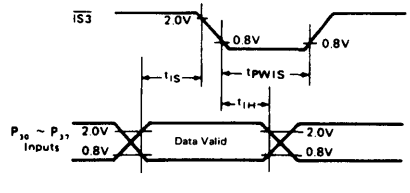


Figure 6 Port 3 Latch Timing (Single Chip Mode)

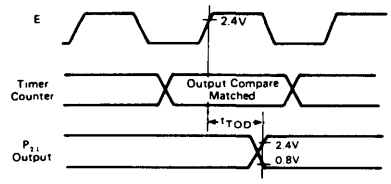


Figure 7 Timer Output Timing

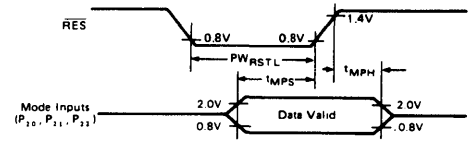
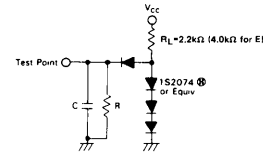


Figure 8 Mode Programming Timing



C = 90pF for P_{30} ~ P_{37}, P_{40} ~ P_{47}, SC_1, SC_2
= 30pF for P_{10} ~ P_{17}, P_{20} ~ P_{24}
= 40pF for E
R = 12kΩ

Figure 9 Bus Timing Test Loads (TTL Load)

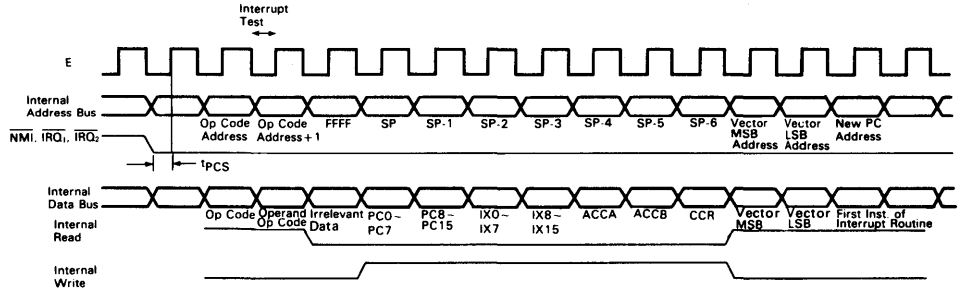


Figure 10 Interrupt Sequence

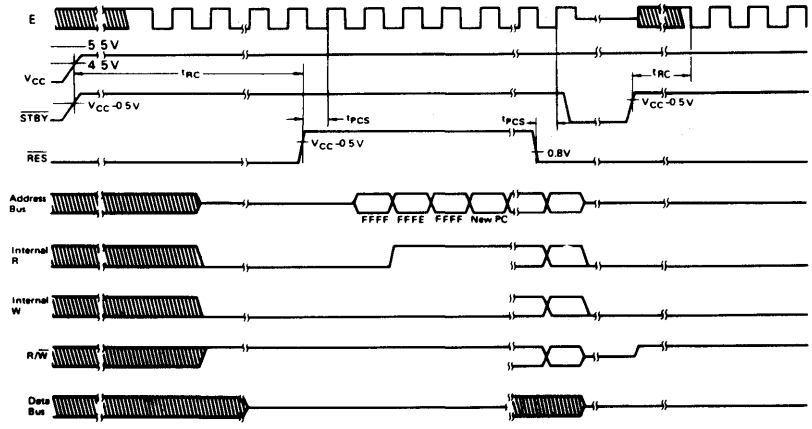


Figure 11 Reset Timing

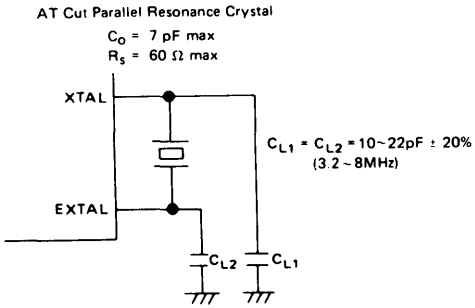
■ FUNCTIONAL PIN DESCRIPTION

• V_{CC}, V_{SS}

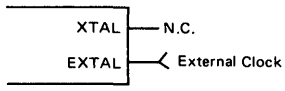
These two pins are used for power supply and GND. Recommended power supply voltage is 5V ±10%. If the operating voltage of the EPROM is 5V ±5%, 5V±5% should be used.

• XTAL, EXTAL

These two pins are connected with parallel resonant fundamental crystal, AT cut. For instance, in order to obtain the system clock 1MHz, a 4MHz resonant fundamental crystal is used because the devide-by-4 circuitry is included. An example of connection circuit is shown in Fig. 12. EXTAL accepts an external clock input of duty 45% to 55% to drive, then internal clock is a quarter the frequency of an external clock. External driving frequency will be less than 4 times as maximum internal clock. For external clock, XTAL pin should be open.



(a) Crystal Interface



(b) External Clock

Figure 12 Connection Circuit

• Standby ($\overline{STBY}$)

This pin is used to place the MCU in the Standby mode. If this goes to "Low" level, the oscillation stops, the internal clock is tied to V_{SS} or V_{CC} and the MCU is reset. In order to retain information in RAM during standby, write "0" into RAM enable bit (RAME). RAME is bit 6 of the RAM Control Register at address $0014. This disables the RAM, so the contents of RAM is guaranteed. For details of the standby mode, see the Standby section.

• Reset ($\overline{RES}$)

This input is used to reset the MCU. $\overline{RES}$ must be held "Low" for at least 20ms when the power starts up. It should be noted that, before clock generator stabilize, the internal state and I/O ports are uncertain, because MCU can not be reset without clock. To reset the MCU during system operation, it must be held "Low" for at least 3 system clock cycles. From the third cycle, all address buses become "High-impedance" and it continues while $\overline{RES}$ is "Low". If $\overline{RES}$ goes to "High", CPU does the following.

(1) I/O Port 2 bits, 2,1,0 are latched into bits PC2, PC1, PC0 of program control register.
(2) The contents of the two Start Addresses, $FFFE, $FFFF are brought to the program counter, from which program starts (see Table 1).
(3) The interrupt mask bit is set. In order to have the CPU recognize the maskable interrupts $\overline{IRQ_1}$ and $\overline{IRQ_2}$, clear it before those are used.

• Enable (E)

This output pin supplies system clock. Output is a single-phase, TTL compatible and 1/4 of the crystal oscillation frequency. It will drive two LS TTL load and 40pF capacitance.

• Non maskable Interrupt ($\overline{NMI}$)

When the falling edge of the input signal of this pin is recognized, NMI sequence starts. The current instruction is continued to complete, even if $\overline{NMI}$ signal is detected. Interrupt mask bit in Condition Code Register has no effect on NMI detection. In response to NMI interrupt, the information of Program Counter, Index Register, Accumulators, and Condition Code Register are stored on the stack. On completion of this sequence, vectoring address $FFFC and $FFFD are generated to load the contents to the program counter. Then the CPU branch to a non maskable interrupt service routine.

• Interrupt Request ($\overline{IRQ_1}$)

This level sensitive input requests maskable interrupt sequence. When $\overline{IRQ_1}$ goes to "Low", the CPU waits until it completes the current instruction that is being executed. Then, if the interrupt mask bit in Condition Code Register is not set, CPU begins interrupt sequence; otherwise, interrupt request is neglected.

Once the sequence has started, the information of Program Counter, Index Register, Accumulators, Condition Code Register are stored on the stack. Then the CPU sets the interrupt mask bit so that no further maskable interrupts may be responded.

Table 1 Interrupt Vectoring memory map

	Vector		Interrupt
	MSB	LSB	
Highest Priority	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	Software Interrupt (SWI)
	FFF8	FFF9	$\overline{IRQ_1}$ (or $\overline{IS3}$)
	FFF6	FFF7	ICF (Timer Input Capture)
	FFF4	FFF5	OCF (Timer Output Compare)
	FFF2	FFF3	TOF (Timer Overflow)
Lowest Priority	FFF0	FFF1	SCI (RDRF + ORFE + TDRE)

At the end of the cycle, the CPU generates 16 bit vectoring addresses indicating memory addresses $FFF8 and $FFF9, and load the contents to the Program Counter, then branch to an interrupt service routine.

The Internal Interrupt will generate signal ($\overline{IRQ_2}$) which is quite the same as $\overline{IRQ_1}$ except that it will use the vector address $FFF0 to $FFF7.

When $\overline{IRQ_1}$ and $\overline{IRQ_2}$ are generated at the same time, the former precede the latter. Interrupt Mask Bit in the condition code register, if being set, will keep the both interrupts off.

$\overline{IRQ_1}$ has no internal latch. Therefore, if $\overline{IRQ_1}$ is removed during suspension, that $\overline{IRQ_1}$ is ignored.

On occurrence of Address error or Op-code error, TRAP interrupt is invoked. This interrupt has priority next to $\overline{RES}$. Regardless of the Interrupt Mask Bit condition, the CPU will start an interrupt sequence. The vector for this interrupt will be $FFEE, $FFEF.

The following pins are available only in single chip mode.

- **Input Strobe ($\overline{IS3}$) (SC_1)**

This signal controls $\overline{IS3}$ interrupt and the latch of Port 3. When the falling edge of this signal is detected, the flag of Port 3 Control Status Register is set.

For detailed explanation of Port 3 Control Status Register, see the I/O PORT 3 CONTROL STATUS REGISTER section.

- **Output Strobe ($\overline{OS3}$) (SC_2)**

This signal is used to send a strobe to an external device, indicating effective data is on the I/O pins. The timing chart for Output Strobe are shown in Figure 5.

The following pins are available for Expanded Modes.

- **Read/Write (R/$\overline{W}$) (SC_2)**

This TTL compatible output signal indicates peripheral and memory devices whether CPU is in Read ("High"), or in Write ("Low"). The normal stand-by state is Read ("High"). Its output will drive one TTL load and 90pF capacitance.

- **I/O Strobe ($\overline{IOS}$) (SC_1)**

In expanded non multiplexed mode 5 of operation, $\overline{IOS}$ goes to "Low" only when A_9 through A_{15} are "0" and A_8 is "1" This allows external access up to 256 addresses from $0100 to $01FF in memory. The timing chart is shown in Figure 2.

- **Address Strobe (AS) (SC_1)**

In the expanded multiplexed mode, address strobe signal appears at this pin. It is used to latch the lower 8 bits addresses multiplexed with data at Port 3. The 8-bit latch is controlled by address strobe as shown in Figure 18. Thereby, I/O Port 3 can become data bus during E pulse. The timing chart of this signal is shown in Figure 1.

Address Strobe (AS) is sent out even if the internal address area is accessed.

■ PORTS

There are four I/O Ports on HD63P01M1 MCU (three 8-bit ports and one 5-bit port). 2 control pins are connected to one of the 8-bit port. Each port has an independent write-only data direction register to program individual I/O pins for input or output.*

When the bit of associated Data Direction Register is "1". I/O pin is programmed for output, if "0", then programmed for an input.

There are four ports: Port 1, Port 2, Port 3, and Port 4. Addresses of each port and associated Data Direction Registers are shown in Table 2.

* Only one exception is bit 1 of Port 2 which becomes either a data input or a timer output. It cannot be used as an output port.

$\overline{RES}$ does not affect I/O port Data Register. Therefore, just after $\overline{RES}$, Data Register is uncertain. Data Direction Registers are reset.

Table 2 Port and Data Direction Register Addresses

Ports	Port Address	Data Direction Register Address
I/O Port 1	$0002	$0000
I/O Port 2	$0003	$0001
I/O Port 3	$0006	$0004
I/O Port 4	$0007	$0005

- **I/O Port 1**

This is an 8-bit port, each bit being defined individually as input or outputs by associated Data Direction Register. The 8-bit output buffers have three-state capability, maintaining in high impedance state when they are used for input. In order to be read accurately, the voltage on the input lines must be more than 2.0V for logic "1" and less than 0.8 V for logic "0".

These are TTL compatible. After the MCU has been reset, all I/O lines of Port 1 are configured as inputs in all modes except mode 1. In all modes except expanded non multiplexed mode (Mode 1), Port 1 is always parallel I/O. In mode 1, Port 1 will be output line for lower order address lines (A_0 to A_7).

- **I/O Port 2**

This port has five lines, whose I/O direction depends on its data direction register. The 5-bit output buffers have three-state capability, going high impedance state when used as inputs. In order to be read accurately, the voltage on the input lines must be more than 2.0V for logic "1" and less than 0.8V for logic "0". After the MCU has been reset, I/O lines are configured as inputs. These pins on Port 2 (pins 10,9,8 of the chip) are used to program the mode of operation during reset. The values of these three pins during reset are latched into the upper 3 bits (bit 7, 6 and 5) of Port 2 Data Register, which is explained in the MODE SELECTION section.

In all modes, Port 2 can be configured as I/O lines. This port also provides access to the Serial I/O and the Timer. However, note that bit 1 (P_{21}) is the only pin restricted to data input or Timer output.

- **I/O Port 3**

This is an 8-bit port which can be configured as I/O lines, a data bus, or an address bus multiplexed with data bus. Its function depends on hardware operation mode programmed by the user using 3 bits of Port 2 during Reset. Port 3 as a data bus is bi-directional. For an input from peripherals, regular TTL level must be supplied, that is greater than 2.0V for a logic "1" and less than 0.8V for a logic "0". This TTL compatible three-state buffer can drive one TTL load and 90pF capacitance. In the expanded Modes, data direction register will be inhibited after Reset and data direction will depend on the state of the R/$\overline{W}$ line. Function of Port 3 is shown below.

Single Chip Mode (Mode 7)

Parallel Inputs/Outputs as programmed by its corresponding Data Direction Register.

There are two control lines associated with this port in this mode, an input strobe ($\overline{IS3}$) and an output strobe ($\overline{OS3}$), both being used for handshaking. They are controlled by I/O Port 3 Control/Status Register. Function of these two control lines of Port 3 are summarized as follows:

(1) Port 3 input data can be latched using $\overline{IS3}$ (SC_1) as a input strobe signal.
(2) $\overline{OS3}$ can be generated by CPU read or write to Port 3's data register.
(3) $\overline{IRQ_1}$ interrupt can be generated by an $\overline{IS3}$ falling edge.

Port 3 strobe and latch timing is shown in Figs. 5 and 6 respectively.

I/O Port 3 Control/Status Register is explained as follows:

I/O Port 3 Control/Status Register

	7	6	5	4	3	2	1	0
	$\overline{IS3}$	$\overline{IS3}$	X	OSS	LATCH	X	X	X
	$\overline{IRQ_1}$							
$000F	FLAG	ENABLE			ENABLE			

Bit 0 Not used.

Bit 1 Not used.

Bit 2 Not used.

Bit 3 LATCH ENABLE.

Bit 3 is used to control the input latch of Port 3. If the bit is set at "1", the input data on Port 3 is latched by the falling edge of $\overline{IS3}$. The latch is released by the MCU read to Port 3; now new data can be latched again by $\overline{IS3}$ falling edge. Bit 3 is cleared by a reset. If this bit is "0", $\overline{IS3}$ does not affect I/O Port 3 latch operation.

Bit 4 OSS (Output Strobe Select)

This bit identifies the cause of output strobe generation: a write operation or read operation to I/O Port 3. When the bit is cleared, the strobe will be generated by a read operation to Port 3. When the bit is not cleared, the strobe will be generated by a write operation. Bit 4 is cleared by a reset.

Bit 5 Not used.

Bit 6 $\overline{IS3}$ $\overline{IRQ_1}$ ENABLE.

If this bit is set, $\overline{IRQ_1}$ interrupt by $\overline{IS3}$ Flag is enabled. Otherwise the interrupt is disabled. The bit is cleared by a reset.

Bit 7 $\overline{IS3}$ FLAG.

Bit 7 is a read-only bit which is set by the falling edge of $\overline{IS3}$ (SC_1). It is cleared by a read of the Control/Status Register followed by a read/write of I/O Port 3. The bit is cleared by reset.

Expanded Non Multiplexed Mode (mode 1, 5)

In this mode, Port 3 becomes data bus. ($D_0 \sim D_7$)

Expanded Multiplexed Mode (mode 0, 2, 4, 6)

Port 3 becomes both the data bus ($D_0 \sim D_7$) and lower bits of the address bus ($A_0 \sim A_7$). An address strobe output is "High" while the address is on the port.

- **I/O Port 4**

This is an 8-bit port that becomes either I/O or address outputs depending on the selected operation mode. In order to be read accurately, the voltage at the input lines must be greater than 2.0V for a logic "1", and less than 0.8V for a logic "0". For outputs, each line is TTL compatible and can drive one TTL load and 90pF. Function of Port 4 for each mode is explained below.

Single Chip Mode (Mode 7): Parallel Inputs/Outputs as programmed by its associated data direction register.

Expanded Non Multiplexed Mode (Mode 5): In this mode, Port 4 becomes the lower address lines (A_0 to A_7) by writing "1"s on the data direction register. After reset, this port becomes inputs. In order to use these pins as addresses, they should be programmed as outputs.

When all of the eight bits are not required as addresses, the remaining lines can be used as I/O lines (Inputs only).

Expanded Non Multiplexed Mode (Mode 1): In this mode, Port 4 becomes output for upper order address lines (A_8 to A_{15}) regardless of the value of the direction register.

Expanded Multiplexed Mode (Mode 6): In this mode, Port 4 becomes the upper address lines (A_8 to A_{15}). After reset, this port becomes inputs. In order to use these pins as addresses, they should be programmed as outputs. When all of the eight bits are not required, the remaining lines can be used as I/O lines (input only).

Expanded Multiplexed Mode (Mode 0, 2, 4): In this mode, Port 4 becomes output for upper order address lines (A_8 to A_{15}) regardless of the value of data direction register.

The relation between each mode and I/O Port 1 to 4 is summarized in Table 3.

■ MODE SELECTION

The operation mode after the reset must be determined by the user wiring the 10, 9, and 8th pins externally. These three pins are lower order bits; I/O 0, I/O 1, I/O 2 of Port 2. They are latched into the control bits PC0, PC1, PC2 of I/O Port 2 register when reset goes "High". I/O Port 2 Register is shown below.

Port 2 DATA REGISTER

	7	6	5	4	3	2	1	0
$0003	PC2	PC1	PC0	I/O 4	I/O 3	I/O 2	I/O 1	I/O 0

An example of external hardware used for Mode Selection is shown in Fig. 13. The HD14053B is used to separate the peripheral device from the MCU during reset. It is necessary if the data may conflict between peripheral device and Mode generation circuit.

No mode can be changed through software because the bits 5, 6, and 7 of Port 2 Data Register are read-only. The mode selection of the HD63P01M1 is shown in Table 4.

The HD63P01M1 operates in three basic modes: (1) Single Chip Mode; (2) Expanded Multiplexed Mode (compatible with the HMCS6800 peripheral family), (3) Expanded Non Multiplexed Mode (compatible with HMCS6800 peripheral family).

- **Single Chip Mode (Mode 7)**

In the Single Chip Mode, all ports will become I/O. This is shown in Figure 15. In this mode, SC_1, SC_2 pins are configured for control lines of Port 3 and can be used as input strobe ($\overline{IS3}$) and output strobe ($\overline{OS3}$) for data handshaking.

- **Expanded Multiplexed Mode (Mode 0, 2, 4, 6)**

In this mode, Port 4 is configured for I/O (inputs only) or address lines. The data bus and the lower order address bus are multiplexed in Port 3 and can be separated by the Address Strobe.

Port 2 is configured for 5 parallel I/O or Serial I/O, or Timer, or any combination thereof. Port 1 is configured for 8 parallel I/O. In this mode, HD63P01M1 is expandable up to 65k words (See Fig. 16).

- **Expanded Non Multiplexed Mode (Mode 1, 5)**

In this mode, the HD63P01M1 can directly address HMCS6800 peripherals with no address latch. In mode 5, Port 3 becomes a data bus. Port 4 becomes A_0 to A_7 address bus or partial address bus and I/O (inputs only). Port 2 is configured for a parallel I/O, Serial I/O, Timer or any combination thereof. Port 1 is configured as a parallel I/O only. In this mode, HD63P01M1 is expandable to 256 locations. In mode 1, Port 3 becomes a data bus and Port 1 becomes A_0 to A_7 address bus, and Port 4 becomes A_8 to A_{15} address bus.

In this mode, the HD63P01M1 is expandable to 65k bytes with no address latch. (See Fig. 17)

- **Lower Order Address Bus Latch**
  Because the data bus is multiplexed with the lower order address bus in Port 3 in the expanded multiplexed mode, address bits must be latched. It requires the 74LS373 Transparent octal D-type to latch the **LSB**. Latch connection of the HD63P01M1 is shown in Figure 18.

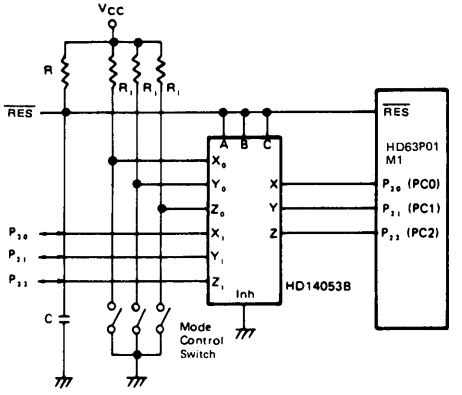
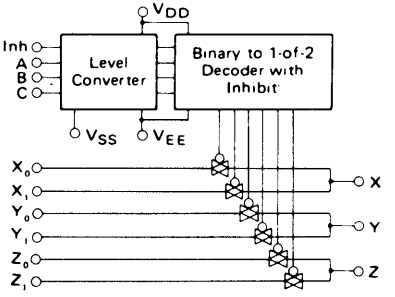


Figure 13 Recommended Circuit for Mode Selection



Truth Table

Control Input				On Switch		
Inhibit	Select					
	C	B	A	HD14053B		
0	0	0	0	Z_0	Y_0	X_0
0	0	0	1	Z_0	Y_0	X_1
0	0	1	0	Z_0	Y_1	X_0
0	0	1	1	Z_0	Y_1	X_1
0	1	0	0	Z_1	Y_0	X_0
0	1	0	1	Z_1	Y_0	X_1
0	1	1	0	Z_1	Y_1	X_0
0	1	1	1	Z_1	Y_1	X_1
1	X	X	X	—		

Figure 14 HD14053B Multiplexers/De-Multiplexers

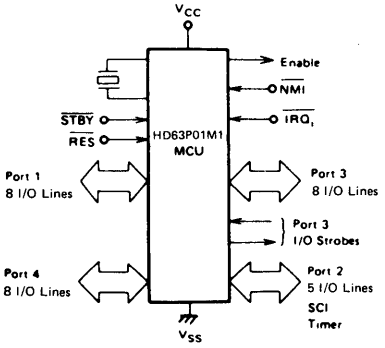


Figure 15 HD63P01M1 MCU Single-Chip Mode

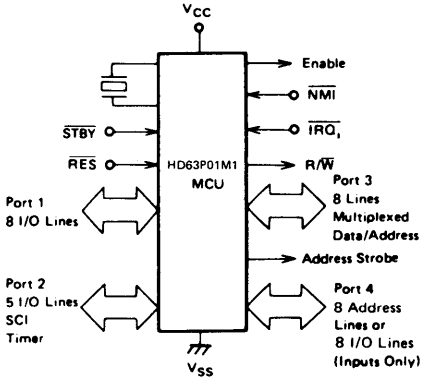


Figure 16 HD63P01M1 MCU Expanded Multiplexed Mode

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

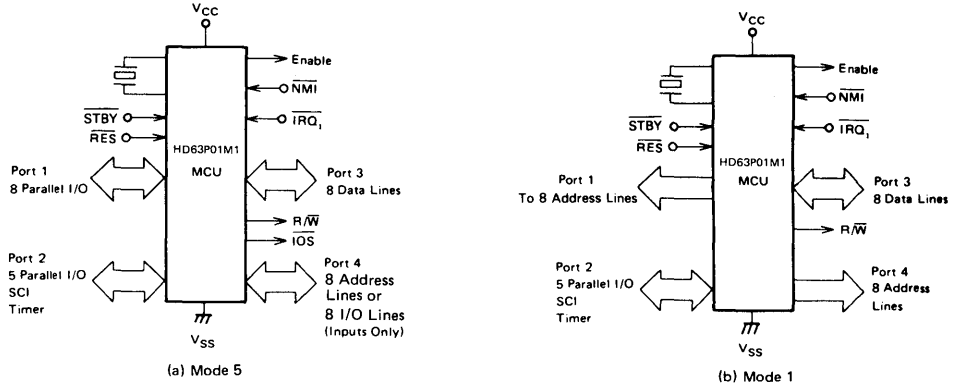
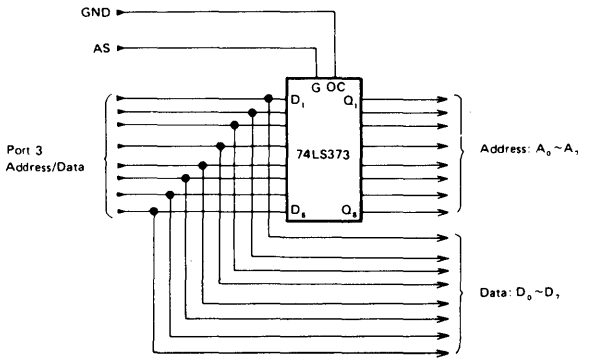


Figure 17 HD63P01M1 MCU Expanded Non Multiplexed Mode



Function Table

Output Control (OC)	Enable G	Enable D	Output Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

Figure 18 Latch Connection

- **Summary of Mode and MCU Signal**

This section gives a description of the MCU signals for the various modes. SC_1 and SC_2 are signals which vary with the mode.

Table 3 Feature of each mode and lines

MODE		PORT 1 Eight Lines	PORT 2 Five Lines	PORT 3 Eight Lines	PORT 4 Eight Lines	SC_1	SC_2
SINGLE CHIP (Mode 7)		I/O	I/O	I/O	I/O	$\overline{IS3}$ (I)	$\overline{OS3}$ (O)
EXPANDED MUX (Mode 0, 2, 4, 6)		I/O	I/O	ADDRESS BUS (A_0~A_7) DATA BUS (D_0~D_7)	ADDRESS BUS* (A_8~A_{15})	AS(O)	R/$\overline{W}$(O)
EXPANDED	(Mode 5)	I/O	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS* (A_0~A_7)	$\overline{IOS}$(O)	R/$\overline{W}$(O)
NON-MUX	(Mode 1)	ADDRESS BUS (A_0~A_7)	I/O	DATA BUS (D_0~D_7)	ADDRESS BUS (A_8~A_{15})	Not Used	R/$\overline{W}$(O)

*These lines can be substituted for I/O (Input Only) (except Mode 0, 2, 4).

I = Input
O = Output
R/$\overline{W}$ = Read/Write
$\overline{IS3}$ = Input Strobe
$\overline{OS3}$ = Output Strobe
$\overline{IOS}$ = I/O Select
SC = Strobe Control
AS = Address Strobe

Table 4 Mode Selection Summary

Mode	P_{23} (PC2)	P_{21} (PC1)	P_{10} (PC0)	ROM	RAM	Interrupt Vectors	Bus Mode	Operating Mode
7	H	H	H	I	I	I	I	Single Chip
6	H	H	L	I	I	I	MUX(3)	Multiplexed/Partial Decode
5	H	L	H	I	I	I	NMUX(3)	Non-Multiplexed/Partial Decode
4	H	L	L	E(1)	I	E	MUX	Multiplexed/RAM
3	L	H	H	—	—	—	—	Not Used
2	L	H	L	E(1)	I	E	MUX	Multiplexed/RAM
1	L	L	H	E(1)	I	E	NMUX	Non-Multiplexed
0	L	L	L	I	I	I(2)	MUX	Multiplexed Test

LEGEND :
I — Internal
E — External
MUX — Multiplexed
NMUX — Non-Multiplexed
L — Logic "0"
H — Logic "1"

(NOTES)
1) Internal ROM is disabled.
2) Reset vector is external for 4 cycles after $\overline{RES}$ goes "high".
3) Idle lines of Port 4 address outputs can be assigned to Input Port.

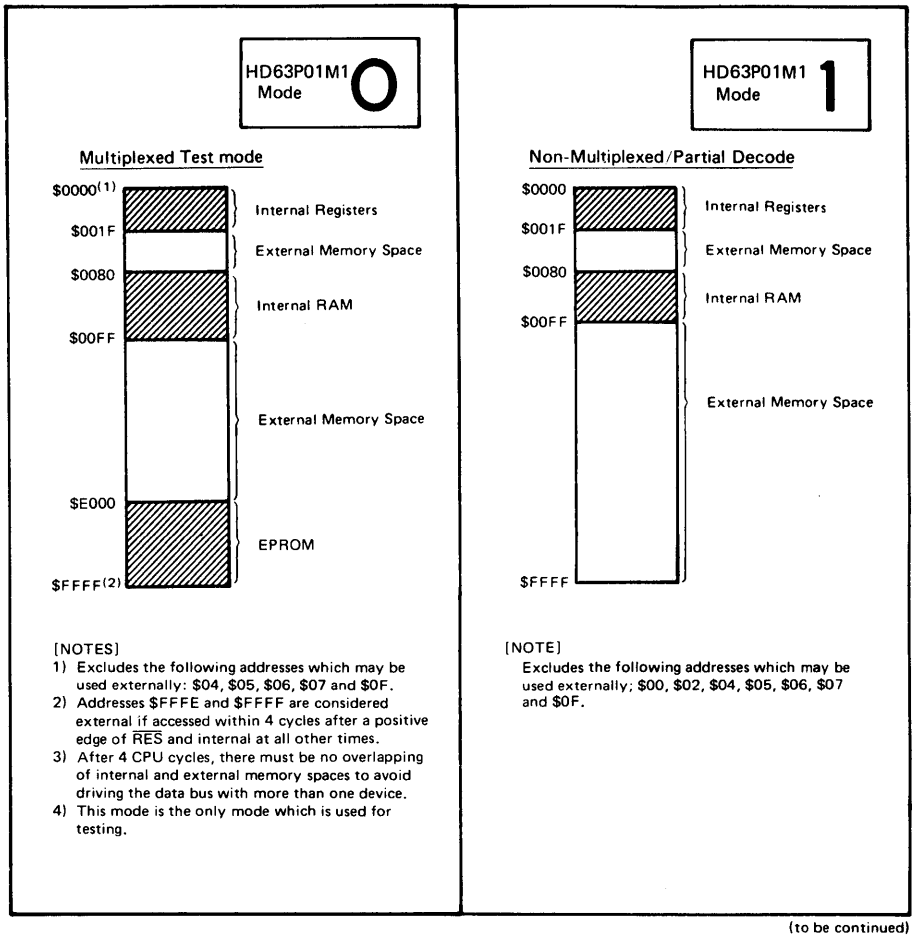
■ Memory Map

The MCU can provide up to 65k byte address space depending on the operating mode. Fig. 19 shows a memory map for each operating mode. The first 32 locations of each map are for the MCU's internal register only, as shown in Table 5.

Table 5 Internal Register Area

Register	Address
Port 1 Data Direction Register****	00*
Port 2 Data Direction Register****	01
Port 1 Data Register	02*
Port 2 Data Register	03
Port 3 Data Direction Register****	04**
Port 4 Data Direction Register****	05***
Port 3 Data Register	06**
Port 4 Data Register	07***
Timer Control and Status Register	08
Counter (High Byte)	09
Counter (Low Byte)	0A
Output Compare Register (High Byte)	0B
Output Compare Register (Low Byte)	0C
Input Capture Register (High Byte)	0D
Input Capture Register (Low Byte)	0E
Port 3 Control and Status Register	0F**
Rate and Mode Control Register	10
Transmit/Receive Control and Status Register	11
Receive Data Register	12
Transmit Data Register	13
RAM Control Register	14
Reserved	15-1F

* External address in Mode 1
** External address in Modes 0, 1, 2, 4, 6; cannot be accessed in Mode 5
*** External address in Modes 0, 1, 2, 4
**** 1 = Output, 0 = Input

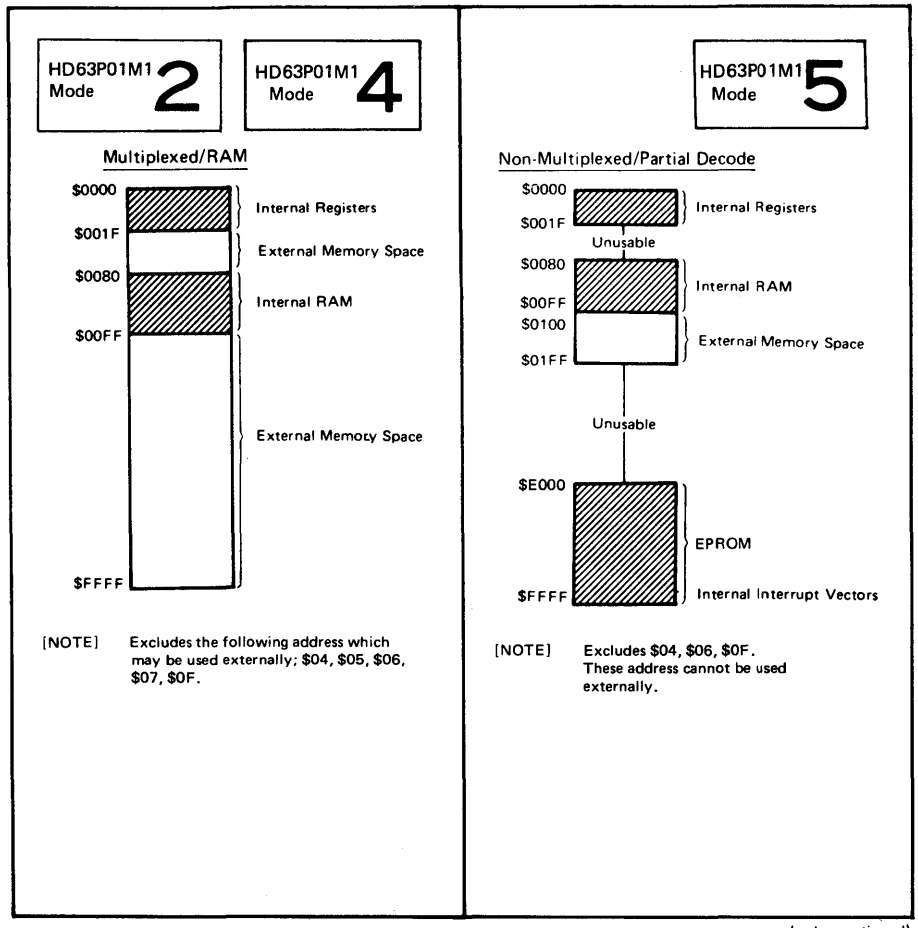


[NOTES]
1) Excludes the following addresses which may be used externally: $04, $05, $06, $07 and $0F.
2) Addresses $FFFE and $FFFF are considered external if accessed within 4 cycles after a positive edge of $\overline{RES}$ and internal at all other times.
3) After 4 CPU cycles, there must be no overlapping of internal and external memory spaces to avoid driving the data bus with more than one device.
4) This mode is the only mode which is used for testing.

[NOTE]
Excludes the following addresses which may be used externally; $00, $02, $04, $05, $06, $07 and $0F.

(to be continued)

Figure 19 HD63P01M1 Memory Maps



(to be continued)

Figure 19 HD63P01M1 Memory Maps

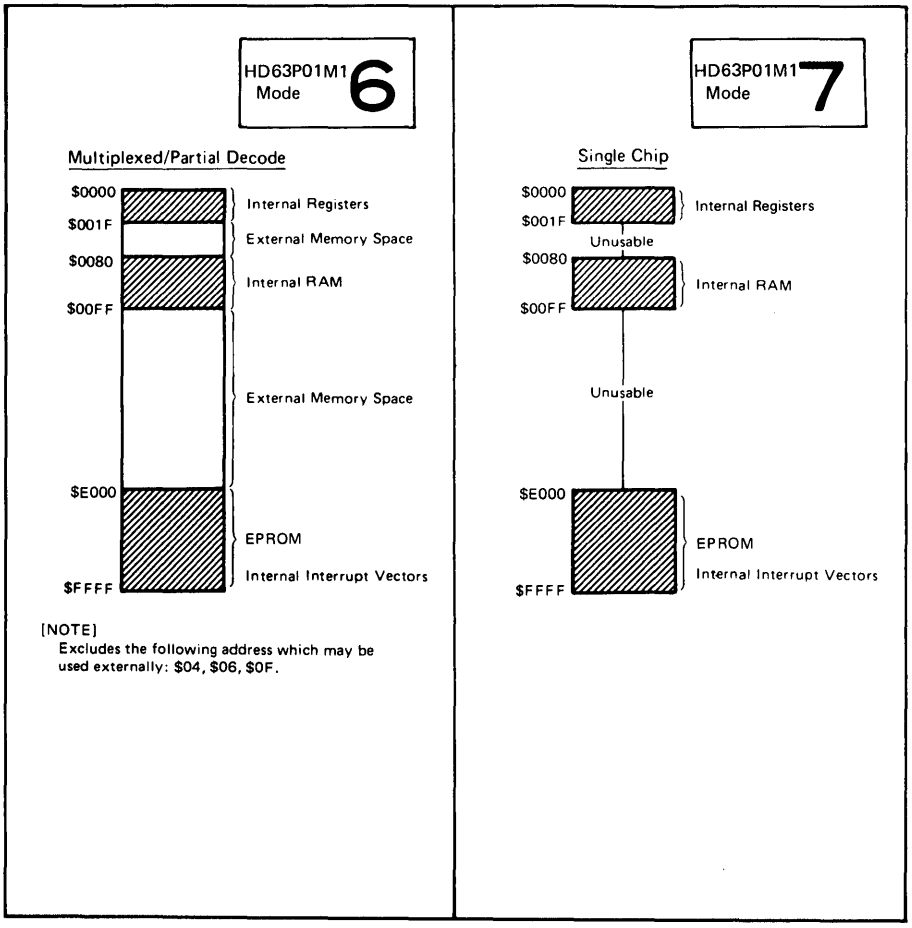


Figure 19 HD63P01M1 Memory Maps

■ PROGRAMMABLE TIMER

The HD63P01M1 contains 16-bit programmable timer which may be used to make measurement of input waveform. In addition to that it can generate an output waveform by itself. For both input and output waveform, the pulse width may vary from a few microseconds to several seconds.

The timer hardware consists of

- an 8-bit control and status register
- a 16-bit free running counter
- a 16-bit output compare register,
- a 16-bit input capture register

A block diagram of the timer is shown in Figure 20.

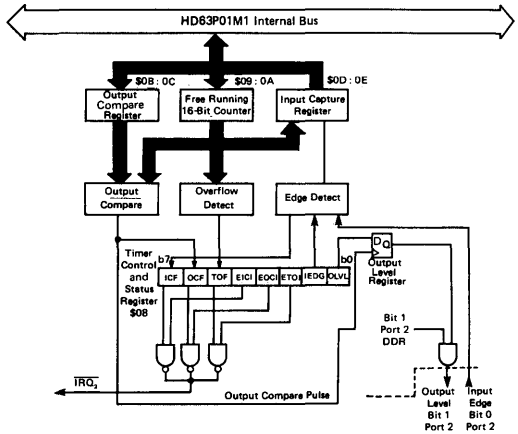


Figure 20 Programmable Timer Block Diagram

• Free Running Counter ($0009: $000A)

The key element in the programmable timer is a 16-bit free running counter, that is driven by an E (Enable) clock to increment its values. The counter value will be read out by the CPU software at any time with no effects on the counter. Reset will clear the counter.

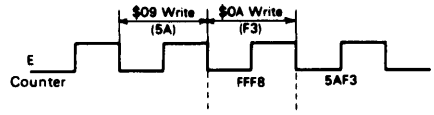
When the MSB of this counter is read, the LSB is stored in temporary latch. The data is fetched from this latch by the subsequent read of LSB. Thus consistent double byte data can be read from the counter.

When the CPU writes arbitrary data to the MSB ($09), the value of $FFF8 is being pre-set to the counter ($09, $0A) regardless of the write data value. Then the CPU writes arbitrary data to the LSB ($0A), the data is set to the "Low" byte of the counter, at the same time, the data preceedingly written in the MSB ($09) is set to "High" byte of the counter.

When the data is written to this counter, a double byte store instruction (ex. STD) must be used. If only the MSB of counter is written, the counter is set to $FFF8.

The counter value written to the counter using the double byte store instruction is shown in Figure 21.

To write to the counter may disturb serial operations, so it should be inhibited during using the SCI in internal clock mode.



(5AF3 written to the counter)

Figure 21 Counter Write Timing

• Output Compare Register ($000B:$000C)

This is a 16-bit read/write register which is used to control an output waveform. The contents of this register are constantly being compared with current value of the free running counter.

When the contents match with the value of the free running counter, a flag (OCF) in the timer control/status register (TCSR) is set and the current value of an output level Bit (OLVL) in the TCSR is transferred to Port 2 bit 1. When bit 1 of the Port 2 data direction register is "1" (output), the OLVL value will appear on the bit 1 of Port 2. Then, the value of Output Compare Register and Output level bit may be changed for the next compare.

The output compare register is set to $FFFF during reset.

The compare function is inhibited at the cycle of writing to the high byte of the output compare register and at the cycle just after that to ensure valid compare. It is also inhibited in same manner at writing to the free running counter.

In order to write a data to Output Compare Register, a double byte store instruction (ex. STD) must be used.

• Input Capture Register ($000D:$000E)

The input capture register is a 16-bit read-only register used to hold the current value of free running counter when the proper transition of an external input signal occurs.

The input transition change required to trigger the counter transfer is controlled by the input edge bit (IEDG).

To allow the external input signal to go in the edge detect unit, the bit of the Data Direction Register corresponding to bit 0 of Port 2 must have been cleared (to zero).

To insure input capture in all cases, the width of an input pulse requires at least 2 Enable cycles.

• Timer Control/Status Register (TCSR) ($0008)

This is an 8-bit register. All 8 bits are readable and the lower 5-bit may be written. The upper 3 bits are read-only, indicating the timer status information as is shown below.

(1) A proper transition has been detected on the input pin (ICF).
(2) A match has been found between the value in the free running counter and the output compare register (OCF).
(3) When counting up to $0000 (TOF).

Each flag has an individual enable bit in TCSR which determines whether or not an interrupt request may occur ($\overline{IRQ_2}$). If the I-bit in Condition Code Register has been cleared, a priority vectored address occurs corresponding to each flag. A description of each bit is as follows.

Timer Control / Status Register

7	6	5	4	3	2	1	0	
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$0008

Bit 0 OLVL (Output Level); When a match is found in the value between the counter and the output com-

pare register, this bit is transferred to the Port 2 bit 1. If the DDR corresponding to Port 2 bit 1 is set "1", the value will appear on the output pin of Port 2 bit 1.

Bit 1 IEDG (Input Edge): This bit control which transition of an input of Port 2 bit 0 will trigger the data transfer from the counter to the input capture register. The DDR corresponding to Port 2 bit 0 must be cleared in advance of using this function. When IEDG = 0, trigger takes place on a negative edge ("High" to "Low" transition). When IEDG = 1, trigger takes place on a positive edge ("Low" to "High" transition).

Bit 2 ETOI (Enable Timer Overflow Interrupt); When set, this bit enables TOF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 3 EOCI (Enable Output Compare Interrupt); When set, this bit enables OCF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 4 EICI (Enable Input Capture Interrupt); When set, this bit enables ICF interrupt to generate the interrupt request ($\overline{IRQ_2}$). When cleared, the interrupt is inhibited.

Bit 5 TOF (Timer Over Flow Flag); This read-only bit is set at the transition of $FFFF to $0000 of the counter. It is cleared by CPU read of TCSR (with TOF set) followed by an CPU read of the counter ($0009).

Bit 6 OCF (Output Compare Flag); This read-only bit is set when a match is found in the value between the output compare register and the counter. It is cleared by a read of TCSR (with OCF set) followed by an CPU write to the output compare register ($000B or $000C).

Bit 7 ICF (Input Capture Flag); The read-only bit is set by a proper transition on the input, and is cleared by a read of TCSR (with ICF set) followed by an CPU read of Input Capture Register ($000D).

Reset will clear each bit of Timer Control and Status Register.

■ SERIAL COMMUNICATION INTERFACE

The HD63P01M1 contains a full-duplex asynchronous Serial Communication Interface (SCI). SCI may select the several kinds of the data rate. It consists of a transmitter and a receiver which operate independently but with the same data format and the same data rate. Both the transmitter and receiver communicate with the CPU via the data bus and with the outside world through Port 2 bit 2, 3 and 4. Description of hardware, software and register is as follows.

• Wake-Up Feature

In typical multiprocessor applications the software protocol will usually have the designated address at the initial byte of the message. The purpose of Wake-Up feature is to have the non-selected MCU neglect the remainder of the message. Thus the non-selected MCU can inhibit the all further interrupt process until the next message begins.

Wake-Up feature is re-enabled by a ten consecutive "1"s which indicates an idle transmit line. Therefore software protocol must put an idle period between the messages and must prevent it within the message.

With this hardware feature, the non-selected MCU is re-enabled or ("waked-up") by the next message.

• Programmable Options

The HD63P01M1 has the following programmable features.

- data format; standard mark/space (NRZ)
- clock source; external or internal
- baud rate; one of 4 rates per given E clock frequency or 1/8 of external clock
- wake-up feature; enabled or disabled
- interrupt requests; enabled or masked individually for transmitter and receiver
- clock output; internal clock enabled or disabled to Port 2 bit 2
- Port 2 (bits 3, 4); dedicated or not dedicated to serial I/O individually

• Serial Communication Hardware

The serial communications hardware is controlled by 4 registers as shown in Figure 22. The registers include:

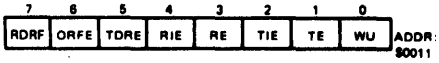
- an 8-bit control/status register
- a 4-bit rate/mode control register (write-only)
- an 8-bit read-only receive data register
- an 8-bit write-only transmit data register

Besides these 4 registers, Serial I/O utilizes Port 2 bit 3 (input) and bit 4 (output). Port 2 bit 2 can be used when an option is selected for the internal-clock-out or the external-clock-in.

• Transmit/Receive Control Status Register (TRCSR)

TRCS Register consists of 8 bits which all may be read while only bits 0 to 4 may be written. The register is initialized to $20 on $\overline{RES}$. The bits of the TRCS register are explained below.

Transmit / Receive Control Status Register



Bit 0 WU (Wake Up); Set by software and cleared by hardware on receipt of ten consecutive "1"s. While this bit is "1", RDRF and ORFE flags are not set even if data are received or errors are detected. Therefore received data are ignored. It should be noted that RE flag must have already been set in advance of WU flag's set.

Bit 1 TE (Transmit Enable); This bit enables transmitter. When this bit is set, bit 4 of Port 2 DDR is also forced to be set. It remains set even if TE is cleared. Preamble of ten consecutive "1"s is transmitted just after this bit is set, and then transmitter becomes ready to send data.
If this bit is cleared, the transmitter is disabled and serial I/O affects nothing on Port 2 bit 4.

Bit 2 TIE (Transmit Interrupt Enable); When this bit is set, TDRE (bit 5) causes an $\overline{IRQ_2}$ interrupt. When cleared TDRE interrupt is masked.

Bit 3 RE (Receive Enable); When set, Port 2 bit 3 can be used as an input of receive regardless of DDR value for this bit. When cleared, the receiver is disabled.

Bit 4 RIE (Receive Interrupt Enable); When this bit is set, RDRF (bit /) or ORFE (bit 6) cause an $\overline{IRQ_2}$ interrupt. When cleared, this interrupt is masked.

Bit 5 TDRE (Transmit Data Register Empty); When the data is transferred from the Transmit Data Register to Output Shift Register, this bit is set by hardware. The bit is cleared by reading the status register followed by writing the next new data into the Transmit Data Register. TDRE is initialized to 1 by $\overline{RES}$.

Bit 6 ORFE (Over Run Framing Error); When overrun or framing error occurs (receive only), this bit is set by hardware. Over Run Error occurs if the attempt is made to transfer the new byte to the receive data register while the RDRF is "1". Framing Error occurs when the bit counter is not synchronized with the boundary of the byte in the receiving bit stream. When Framing Error is detected, RDRF is not set. Therefore Framing Error can be distinguished from Overrun Error. That is, if ORFE is "1" and RDRF is "1", Overrun Error is detected. Otherwise Framing Error occurs. The bit is cleared by reading the status register followed by reading the receive data register, or by $\overline{RES}$.

Bit 7 RDRF (Receive Data Register Full); This bit is set by hardware when the data is transferred from the receive shift register to the receive data register. It is cleared by reading the status register followed by reading the receive data register, or by $\overline{RES}$.

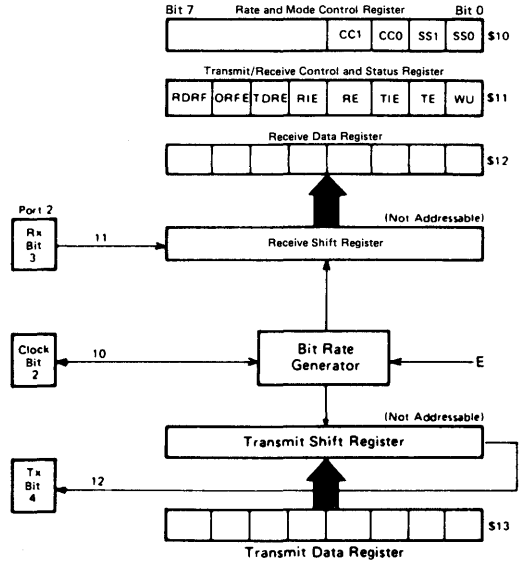


Figure 22 Serial I/O Register

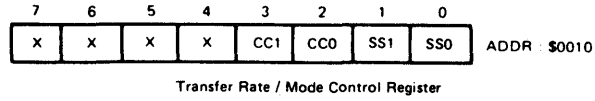


Table 6 SCI Bit Times and Transfer Rates

SS1 : SS0	XTAL	2.4576 MHz	4.0 MHz	4.9152MHz
	E	614.4 kHz	1.0 MHz	1.2288MHz
0 0	E ÷ 16	26 μs/38,400 Baud	16 μs/62,500 Baud	13 μs/76,800Baud
0 1	E ÷ 128	208μs/4,800 Baud	128 μs/7812.5 Baud	104.2μs/ 9,600Baud
1 0	E ÷ 1024	1.67ms/600 Baud	1.024ms/976.6 Baud	833.3μs/ 1,200Baud
1 1	E ÷ 4096	6.67ms/150 Baud	4.096ms/244.1 Baud	3.333ms/ 300Baud

Table 7 SCI Format and Clock Source Control

CC1: CC0	Format	Clock Source	Port 2 Bit 2	Port 2 Bit 3	Port 2 Bit 4
0 0	–	–	–	–	–
0 1	NRZ	Internal	Not Used***	••	••
1 0	NRZ	Internal	Output*	••	••
1 1	NRZ	External	Input	••	••

* Clock output is available regardless of values for bits RE and TE.
** Bit 3 is used for serial input if RE = "1" in TRCS.
Bit 4 is used for serial output if TE = "1" in TRCS.
*** This pin can be used as I/O port.

- **Transfer rate/Mode Control Register (RMCR)**

The register controls the following serial I/O functions:
- Bauds rate
- data format
- clock source
- Port 2 bit 2 feature

It is 4-bit write-only register, cleared by $\overline{RES}$. The 4 bits are considered as a pair of 2-bit fields. The lower 2 bits control the bit rate of internal clock while the upper 2 bits control the format and the clock select logic.

Bit 0 SS0, Bit 1 SS1 — Speed Select

These bits select the Baud rate for the internal clock. The rates selectable are function of E clock frequency of the CPU. Table 6 lists the available Baud Rates.

Bit2 CC0, Bit 3 CC1 — Clock Control/Format Select

They control the data format and the clock select logic. Table 7 defines the bit field.

- **Internally Generated Clock**

If the user wish to use externally an internal clock of the serial I/O, the following requirements should be noted.
- CC1, CC0 must be set to "10".
- The maximum clock rate must be E/16.
- The clock rate is equal to the bit rate.
- The values of RE and TE have no effect.

- **Externally Generated Clock**

If the user wish to supply an external clock to the Serial I/O, the following requirements should be noted.
- The CC1, CC0 must be set to "11" (See Table 7).
- The external clock must be set to 8 times of the desired baud rate.
- The maximum external clock frequency is E/2 clock.

- **Serial Operations**

The serial I/O hardware must be initialized by the software before operation. The sequence will be normally as follows.
- Writing the desired operation control bits of the Rate and Mode Control Register.
- Writing the desired operation control bits of the TRCS register.

If Port 2 bit 3, 4 are used for serial I/O, TE, RE bits may be kept set. When TE, RE bit are cleared during SCI operation, and subsequently set again, it should be noted that TE, RE must be kept "0" for at least one bit time of the current baud rate. If TE, RE are set again within one bit time, there may be the case where the initializing of internal function for transmitter and receiver does not take place correctly.

- **Transmit Operation**

Data transmission is enabled by the TE bit in the TRCS register. When set, the output of the transmit shift register is connected with Port 2 bit 4 which is unconditionally configured as an output.

After $\overline{RES}$, the user should initialize both the RMC register and the TRCS register for desired operation. Setting the TE bit causes a transmission of ten-bit preamble of "1"s. Following the preamble, internal synchronization is established and the transmitter is ready to operate. Then either of the following states exists.

(1) If the transmit data register is empty (TDRE = 1), the consecutive "1"s are transmitted indicating an idle states.
(2) If the data has been loaded into the Transmit Data Register (TDRE = 0), it is transferred to the output shift register and data transmission begins.

During the data transfer, the start bit ("0") is first transferred. Next the 8-bit data (beginning at bit 0) and finally the stop bit ("1"). When the contents of the Transmit Data Register is transferred to the output shift register, the hardware sets the TDRE flag bit: If the CPU fails to respond to the flag within the proper time, TDRE is kept set and then a continuous string of 1's is sent until the data is supplied to the data register.

- **Receive Operation**

The receive operation is enabled by the RE bit. The serial input is connected with Port 2 bit 3. The receiver operation is determined by the contents of the TRCS and RMC register. The received bit stream is synchronized by the first "0" (start bit). During 10-bit time, the data is strobed approximately at the center of each bit. If the tenth bit is not "1" (stop bit), the system assumes a framing error and the ORFE is set.

If the tenth bit is "1", the data is transferred to the receive data register, and the RDRF flag is set. If the tenth bit of the next data is received and still RDRF is preserved set, then ORFE is set indicating that an overrun error has occurred.

After the CPU read of the status register as a response to RDRF flag or ORFE flag, followed by the CPU read of the receive data register, RDRF or ORFE will be cleared.

■ RAM CONTROL REGISTER

The register assigned to the address $0014 gives a status information about standby RAM.

RAM Control Register

	7	6	5	4	3	2	1	0
$0014	STBY PWR	RAME	x	x	x	x	x	x

Bit 0 Not used.
Bit 1 Not used.
Bit 2 Not used.

Bit 3 Not used.
Bit 4 Not used.
Bit 5 Not used.
Bit 6 RAM Enable.

Using this control bit, the user can disable the RAM. RAM Enable bit is set on the positive edge of $\overline{RES}$ and RAM is enabled. The program can write "1" or "0". If RAME is cleared, the RAM address becomes external address and the CPU may read the data from the outside memory.

Bit 7 Standby Bit

This bit can be read or written by the user program. It is cleared when the V_{CC} voltage is removed. Normally this bit is set by the program before going into stand-by mode. When the CPU recovers from stand-by mode, this bit should be checked. If it is "1", the data of the RAM is retained during stand-by and it is valid.

■ GENERAL DESCRIPTION OF INSTRUCTION SET

The HD63P01M1 has an upward object code compatible with the HD6801 to utilize all instruction sets of the HMCS6800. The execution time of the key instruction is reduced to increase the system through-put. In addition, the bit operation instruction, the exchange instruction between the index and the accumulator, the sleep instruction are added. This section describes:

- CPU programming model (See Fig. 23)
- Addressing modes
- Accumulator and memory manipulation instructions (See Table 8)
- New instructions
- Index register and stack manipulation instructions (See Table 9)
- Jump and branch instructions (See Table 10)
- Condition code register manipulation instructions (See Table 11)
- Op-code map (See Table 12)
- Cycle-by-Cycle Operation (See Table 13)

● CPU Programming Model

The programming model for the HD63P01M1 is shown in Figure 23. The double accumulator is physically the same as the accumulator A concatenated with the accumulator B, so that the contents of A and B is changed with executing operation of an accumulator D.

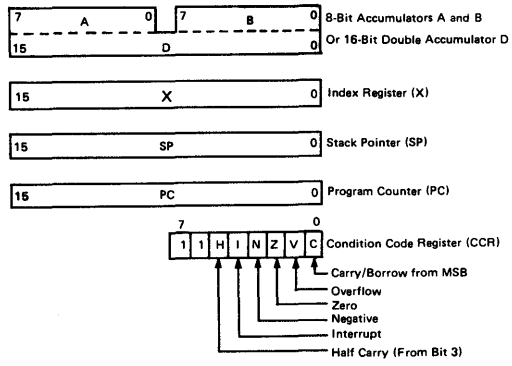


Figure 23 CPU Programming Model

● CPU Addressing Modes

The HD63P01M1 has seven address modes which depend on both of the instruction type and the code. The address mode for every instruction is shown along with execution time given in terms of machine cycles (Table 8 to 12). When the clock frequency is 4 MHz, the machine cycles will be microseconds.

Accumulator (ACCX) Addressing

Only the accumulator (A or B) is addressed. Either accumulator A or B is specified by one-byte instructions.

Immediate Addressing

In this mode, the operand is stored in the second byte of the instruction except that the operand in LDS and LDX, etc are stored in the second and the third byte. These are two or three-byte instructions.

Direct Addressing

In this mode, the second byte of instruction indicates the address where the operand is stored. Direct addressing allows the user to directly address the lowest 256 Bytes in the machine locations zero through 255. Improved execution times are achieved by storing data in these locations. For system configuration, it is recommended that these locations should be RAM and be utilized preferably for user's data realm. These are two-byte instructions except the AIM, OIM, EIM and TIM which have three- byte.

Extended Addressing

In this mode, the second byte indicates the upper 8 bits addresses where the operand is stored, while the third byte indicates the lower 8 bits. This is an absolute address in memory. These are three-byte instructions.

Indexed Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the Index Register. For each of AIM, OIM, EIM and TIM instructions, the contents of the third byte are added to the lower 8 bits in the Index Register. In addition, the resulting "carry" is added to the upper 8 bits in the Index Register. The result is used for addressing memory. Because the modified address is held in the Temporary Address Register, there is no change to the Index Register. These are two-byte instructions but AIM, OIM, EIM, TIM have three-byte.

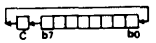
Implied Addressing

In this mode, the instruction itself gives the address; stack pointer, index register, etc. These are 1-byte instructions.

Relative Addressing

In this mode, the contents of the second byte is added to the lower 8 bits in the program counter. The resulting carry or borrow is added to the upper 8 bits. This helps the user to address the data within a range of −126 to +129 bytes of the current execution instruction. These are two-byte instructions.

Table 8 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M} \rightarrow M$	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A} \rightarrow A$	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B} \rightarrow B$	•	•	↕	↕	R	S
Complement, 2's (Negate)	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3				M / A / B 	•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

Note) Condition Code Register will be explained in Note of Table 11.

(to be continued)

Table 8 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑤	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑤	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑤	↕
Double Shift Left, Arithmetic	ASLD													05	1	1	C A7 A0 B7 B0	•	•	↕	↕	⑤	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	1	1	A	•	•	R	↕	⑥	↕
	LSRB													54	1	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1	0 A7 A0 B7 B0 C	•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B - M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A - B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B - 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

Note) Condition Code Register will be explained in Note of Table 11.

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

- **New Instructions**

In addition to the HD6801 Instruction Set, the HD63P01M1 has the following new instructions:

AIM----(M) · (IMM) → (M)
Evaluates the AND of the immediate data and the memory, places the result in the memory.

OIM----(M) + (IMM) → (M)
Evaluates the OR of the immediate data and the memory, places the result in the memory.

EIM----(M) ⊕ (IMM) → (M)
Evaluates the EOR of the immediate data and the contents of memory, places the result in memory.

TIM----(M) · (IMM)
Evaluates the AND of the immediate data and the memory, changes the flag of associated condition code register

Each instruction has three bytes; the first is op-code, the second is immediate data, the third is address modifier.

XGDX--(ACCD) ↔ (IX)
Exchanges the contents of accumulator and the index register.

SLP----The MPU is brought to the sleep mode. For sleep mode, see the "sleep mode" section.

Table 9 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X−M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M+1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{sp}, SP − 1 → SP X_H → M_{sp}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{sp} → X_H SP + 1 → SP, M_{sp} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD↔IX	•	•	•	•	•	•

Note) Condition Code Register will be explained in Note of Table 11.

Table 10 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——		⑧		——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

Note) *WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 11.

Table 11 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	⑩					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

[NOTE 1] Condition Code Register Notes: (Bit set if test is true and cleared otherwise)
- ① (Bit V) Test: Result = 10000000?
- ② (Bit C) Test: Result ≠ 00000000?
- ③ (Bit C) Test: BCD Character of high-order byte greater than 9? (Not cleared if previously set)
- ④ (Bit V) Test: Operand = 10000000 prior to execution?
- ⑤ (Bit V) Test: Operand = 01111111 prior to execution?
- ⑥ (Bit V) Test: Set equal to N⊕C=1 after the execution of instructions
- ⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
- ⑧ (All Bit) Load Condition Code Register from Stack.
- ⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
- ⑩ (All Bit) Set according to the contents of Accumulator A.
- ⑪ (Bit C) Result of Multiplication Bit 7=1 of ACCB?

[NOTE 2] CLI instructions and interrupt.
If interrupt mask-bit is set (1="1") and interrupt is requested ($\overline{IRQ_1}$ = "0" or $\overline{IRQ_2}$ = "0"),, and then CLI instruction is executed, the CPU responds as follows.
1 the next instruction of CLI is one-machine cycle instruction.
Subsequent two instructions are executed before the interrupt is responded.
That is, the next and the next of the next instruction are executed.
2 the next instruction of CLI is two-machine cycle (or more) instruction.
Only the next instruction is executed and then the CPU jump to the interrupt routine.
Even if TAP instruction is used, instead of CLI, the same thing occurs.

Table 12 OP-Code Map

OP CODE						ACC A	ACC B	IND	EXT / DIR*	ACCA or SP IMM	DIR	IND	EXT	ACCB or X IMM	DIR	IND	EXT	
HI		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC					CPX			LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE
* Only for instructions of AIM, OIM, EIM, TIM

- **Instruction Execution Cycles**

In the HMCS6800 series, the execution cycle of each instruction is the number of cycles between the start of the current instruction fetch and just before the start of the subsequent instruction fetch.

The HD63P01M1 uses a mechanism of the pipeline control for the instruction fetch and the subsequent instruction fetch is performed during the current instruction being executed.

Therefore, the method to count instruction cycles used in the HMCS6800 series cannot be applied to the instruction cycles such as MULT, PULL, DAA and XGDX in the HD63P01M1.

Table 13 provides the information about the relationship among each data on the Address Bus, Data Bus, and R/$\overline{W}$ status in cycle-by-cycle basis during the execution of each instruction.

Table 13 Cycle-by-Cycle Operation

Address Mode & Instructions		Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMMEDIATE						
ADC	ADD	2	1	Op Code Address+1	1	Operand Data
AND	BIT		2	Op Code Address+2	1	Next Op Code
CMP	EOR					
LDA	ORA					
SBC	SUB					
ADDD	CPX	3	1	Op Code Address+1	1	Operand Data (MSB)
LDD	LDS		2	Op Code Address+2	1	Operand Data (LSB)
LDX	SUBD		3	Op Code Address+3	1	Next Op Code
DIRECT						
ADC	ADD	3	1	Op Code Address+1	1	Address of Operand (LSB)
AND	BIT		2	Address of Operand	1	Operand Data
CMP	EOR		3	Op Code Address+2	1	Next Op Code
LDA	ORA					
SBC	SUB					
STA		3	1	Op Code Address+1	1	Destination Address
			2	Destination Address	0	Accumulator Data
			3	Op Code Address+2	1	Next Op Code
ADDD	CPX	4	1	Op Code Address+1	1	Address of Operand (LSB)
LDD	LDS		2	Address of Operand	1	Operand Data (MSB)
LDX	SUBD		3	Address of Operand+1	1	Operand Data (LSB)
			4	Op Code Address+2	1	Next Op Code
STD	STS	4	1	Op Code Address+1	1	Destination Address (LSB)
STX			2	Destination Address	0	Register Data (MSB)
			3	Destination Address+1	0	Register Data (LSB)
			4	Op Code Address+2	1	Next Op Code
JSR		5	1	Op Code Address+1	1	Jump Address (LSB)
			2	FFFF	1	Restart Address (LSB)
			3	Stack Pointer	0	Return Address (LSB)
			4	Stack Pointer−1	0	Return Address (MSB)
			5	Jump Address	1	First Subroutine Op Code
TIM		4	1	Op Code Address+1	1	Immediate Data
			2	Op Code Address+2	1	Address of Operand (LSB)
			3	Address of Operand	1	Operand Data
			4	Op Code Address+3	1	Next Op Code
AIM	EIM	6	1	Op Code Address+1	1	Immediate Data
OIM			2	Op Code Address+2	1	Address of Operand (LSB)
			3	Address of Operand	1	Operand Data
			4	FFFF	1	Restart Address (LSB)
			5	Address of Operand	0	New Operand Data
			6	Op Code Address+3	1	Next Op Code

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
INDEXED					
JMP	3	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Jump Address	1	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	Op Code Address+2	1	Next Op Code
STA	4	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Accumulator Data
		4	Op Code Address+2	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data (MSB)
		4	IX+Offset+1	1	Operand Data (LSB)
		5	Op Code Address+2	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	0	Register Data (MSB)
		4	IX+Offset+1	0	Register Data (LSB)
		5	Op Code Address+2	1	Next Op Code
JSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	IX+Offset	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	IX+Offset	0	New Operand Data
		6	Op Code Address+2	1	Next Op Code
TIM	5	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	IX+Offset	1	Operand Data
		4	IX+Offset	0	00
		5	Op Code Address+2	1	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	Immediate Data
		2	Op Code Address+2	1	Offset
		3	FFFF	1	Restart Address (LSB)
		4	IX+Offset	1	Operand Data
		5	FFFF	1	Restart Address (LSB)
		6	IX+Offset	0	New Operand Data
		7	Op Code Address+3	1	Next Op Code

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
EXTEND					
JMP	3	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	Jump Address	1	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Op Code Address+3	1	Next Op Code
STA	4	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Accumulator Data
		4	Op Code Address+3	1	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data (MSB)
		4	Address of Operand+1	1	Operand Data (LSB)
		5	Op Code Address+3	1	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	Destination Address (MSB)
		2	Op Code Address+2	1	Destination Address (LSB)
		3	Destination Address	0	Register Data (MSB)
		4	Destination Address+1	0	Register Data (LSB)
		5	Op Code Address+3	1	Next Op Code
JSR	6	1	Op Code Address+1	1	Jump Address (MSB)
		2	Op Code Address+2	1	Jump Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	Stack Pointer	0	Return Address (LSB)
		5	Stack Pointer−1	0	Return Address (MSB)
		6	Jump Address	1	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	FFFF	1	Restart Address (LSB)
		5	Address of Operand	0	New Operand Data
		6	Op Code Address+3	1	Next Op Code
CLR	5	1	Op Code Address+1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	Address of Operand (LSB)
		3	Address of Operand	1	Operand Data
		4	Address of Operand	0	00
		5	Op Code Address+3	1	Next Op Code

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Accumulator Data
		4	Op Code Address+1	1	Next Op Code
PULX	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Index Register (LSB)
		4	Stack Pointer−1	0	Index Register (MSB)
		5	Op Code Address+1	1	Next Op Code
RTS	5	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Return Address (MSB)
		4	Stack Pointer+2	1	Return Address (LSB)
		5	Return Address	1	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	FFFF	1	Restart Address (LSB)
		4	FFFF	1	Restart Address (LSB)
		5	FFFF	1	Restart Address (LSB)
		6	FFFF	1	Restart Address (LSB)
		7	FFFF	1	Restart Address (LSB)

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
IMPLIED					
WAI	9	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Stack Pointer−2	0	Index Register (LSB)
		6	Stack Pointer−3	0	Index Register (MSB)
		7	Stack Pointer−4	0	Accumulator A
		8	Stack Pointer−5	0	Accumulator B
		9	Stack Pointer−6	0	Conditional Code Register
RTI	10	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer+1	1	Conditional Code Register
		4	Stack Pointer+2	1	Accumulator B
		5	Stack Pointer+3	1	Accumulator A
		6	Stack Pointer+4	1	Index Register (MSB)
		7	Stack Pointer+5	1	Index Register (LSB)
		8	Stack Pointer+6	1	Return Address (MSB)
		9	Stack Pointer+7	1	Return Address (LSB)
		10	Return Address	1	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer − 1	0	Return Address (MSB)
		5	Stack Pointer − 2	0	Index Register (LSB)
		6	Stack Pointer − 3	0	Index Register (MSB)
		7	Stack Pointer − 4	0	Accumulator A
		8	Stack Pointer − 5	0	Accumulator B
		9	Stack Pointer − 6	0	Conditional Code Register
		10	Vector Address FFFA	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	Next Op Code
		2	FFFF	1	Restart Address (LSB)
		↑	FFFF		High Impedance-Non MPX Mode
		Sleep			Address Bus -MPX Mode
		↓			↓
		3	FFFF		Restart Address (LSB)
		4	Op Code Address+1		Next Op Code

— Continued —

Table 13 Cycle-by-Cycle Operation (Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	Data Bus
RELATIVE					
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	Branch Offset
		2	FFFF	1	Restart Address (LSB)
		3	{Branch Address……Test="1" Op Code Address+1…Test="0"	1	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	Offset
		2	FFFF	1	Restart Address (LSB)
		3	Stack Pointer	0	Return Address (LSB)
		4	Stack Pointer−1	0	Return Address (MSB)
		5	Branch Address	1	First Op Code of Subroutine

■ LOW POWER CONSUMPTION MODE

The HD63P01M1 has two low power consumption modes; sleep and standby mode.

• Sleep Mode

On execution of SLP instruction, the MCU is brought to the sleep mode. In the sleep mode, the CPU sleeps (the CPU clock becomes inactive), but the contents of the registers in the CPU are retained. In this mode, the peripherals of CPU will remain active. So the operations such as transmit and receive of the SCI data and counter may keep in operation. In this mode, the power consumption is reduced to about 1/6 the value of a normal operation.

The escape from this mode can be done by interrupt, $\overline{RES}$, $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode (This will be mentioned later). When interrupt is requested to the CPU and accepted, the sleep mode is released, then the CPU is brought in the operation mode and jumps to the interrupt routine. When the CPU has masked the interrupt after recovering from the sleep mode, the next instruction of SLP starts to execute. However, in such a case that the timer interrupt is inhibited on the timer side, the sleep mode cannot be released due to the absence of the interrupt request to the CPU.

This sleep mode is available to reduce an average power consumption in the applications of the HD63P01M1 which may not be always running.

• Standby Mode

Bringing $\overline{STBY}$ "Low", the CPU becomes reset and all clocks of the HD63P01M1 become inactive. It goes into the standby mode. This mode remarkably reduces the power consumptions of the HD63P01M1.

In the standby mode, if the HD63P01M1 is continuously supplied with power, the contents of RAM is retained. The standby mode should escape by the reset start. The following is the typical application of this mode.

First, $\overline{NMI}$ routine stacks the MCU's internal information and the contents of SP in RAM, disables RAME bit of RAM control register, sets the Standby bit, and then goes into the standby mode. If the Standby bit keeps set on reset start, it means that the power has been kept during standby mode and the contents of RAM is normally guaranteed. The system recovery may be possible by returning SP and bringing into the condition before the standby mode has started. The timing relation for each line in this application is shown in Figure 24.

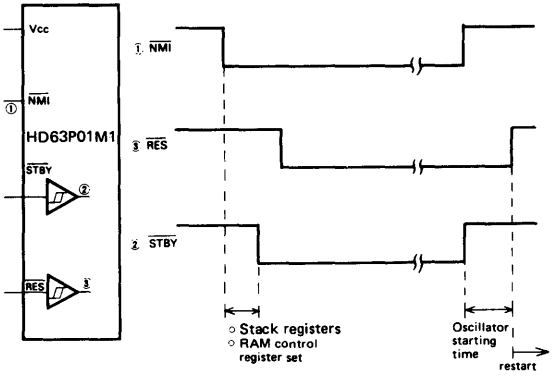


Figure 24 Standby Mode Timing

■ ERROR PROCESSING

When the HD63P01M1 fetches an undefined instruction or fetches an instruction from unusable memory area, it generates the highest priority internal interrupt, that may protect from system upset due to noise or a program error.

• Op-Code Error

Fetching an undefined op-code, the HD63P01M1 will stack the CPU register as in the case of a normal interrupt and vector to the TRAP ($FFEE, $FFEF), that has a second highest priority ($\overline{RES}$ is the highest).

• Address Error

When an instruction is fetched from other than a resident ROM, RAM, or an external memory area, the CPU starts the same interrupt as op-code error. In the case which the instruction is fetched from external memory area and that area is not usable, the address error cannot be detected.

The addresses which cause address error in particular mode are shown in Table 14.

This feature is applicable only to the instruction fetch, not to normal read/write of data accessing.

Table 14 Address Error

Mode	0	1	2, 4	5	6	7
Address	$0000	$0000	$0000	$0000	$0000	$0000
	↓	↓	↓	↓	↓	↓
	$001F	$001F	$001F	$007F	$001F	$007F
				$0200		$0100
				↓		↓
				$DFFF		$DFFF

System Flow chart of HD63P01M1 is shown in Fig. 25.

Transitions among the active mode, sleep mode, standby mode and reset are shown in Fig. 26.

Figures 27, 28, 29 and 30 shows a system configuration.

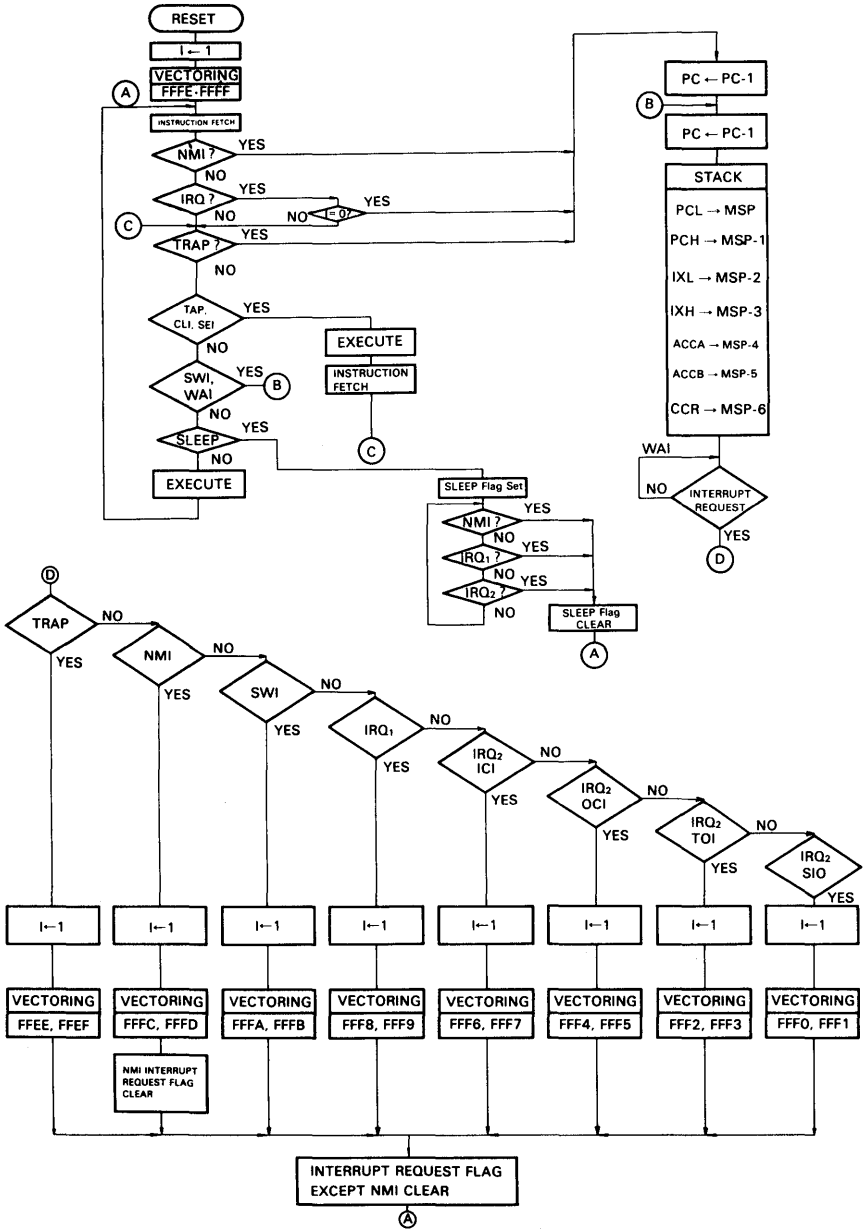


Figure 25 HD63P01M1 System Flow Chart

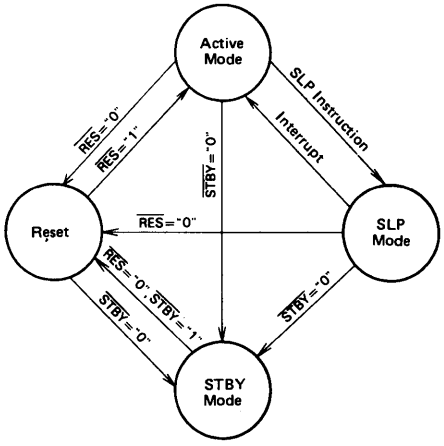


Figure 26 Transitions among Active Mode, Standby Mode, Sleep Mode, and Reset

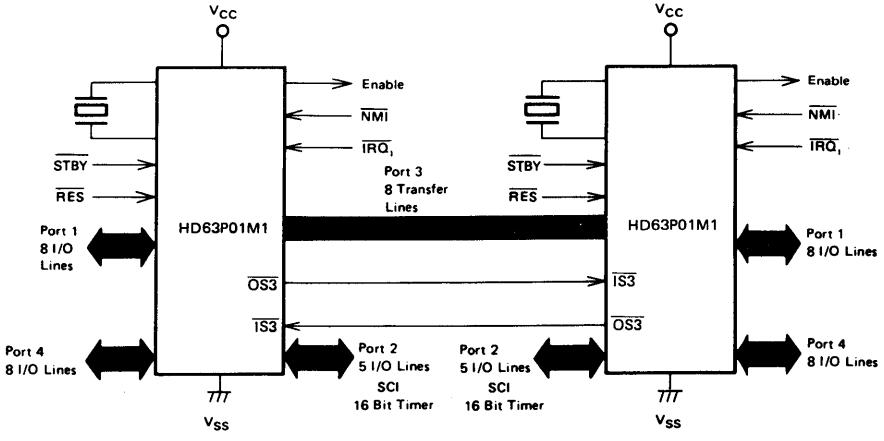


Figure 27 HD63P01M1 MCU Single-Chip Dual Processor Configuration

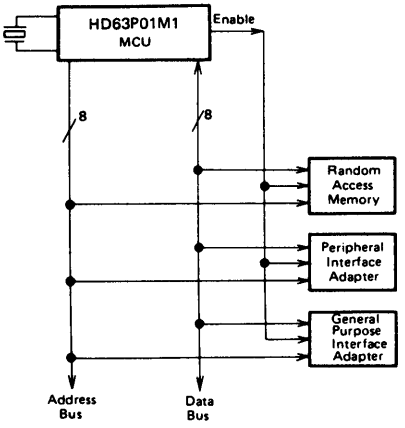


Figure 28 HD63P01M1 MCU Expanded Non-Multiplexed Mode (Mode 5)

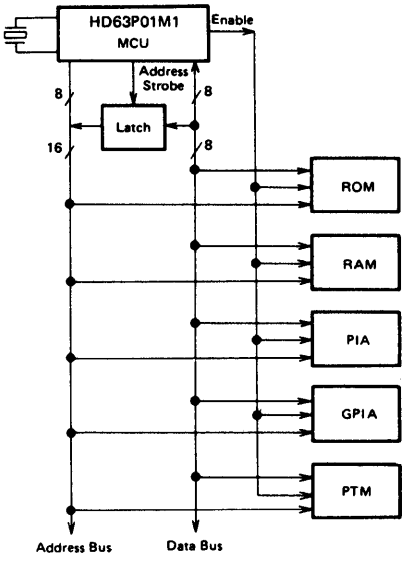


Figure 29 HD63P01M1 MCU Expanded Multiplexed Mode (Modes 2, 4 and 6)

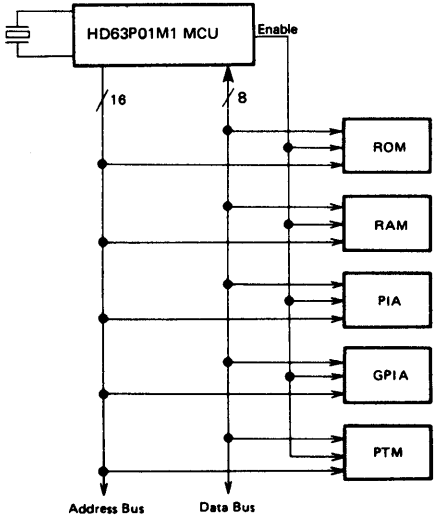
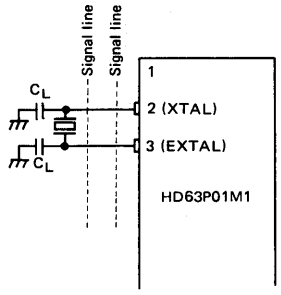


Figure 30 HD63P01M1 MCU Expanded Non-Multiplexed Mode (Mode 1)

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 31, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this . Crystal and C_L must be put as near the HD63P01M1 as possible.



Do not use this kind of print board design.

Figure 31 Precaution to the boad design of oscillation circuit

■ PIN CONDITIONS AT SLEEP AND STANDBY STATE

• Sleep State

The conditions of power supply pins (pins 1 and 21), clock pins (pins 2 and 3), input pins (pins 4, 5, 6 and 7) and E clock pin (pin 40) are the same as those of operation. Refer to Table 15 for the other pin conditions. Both address ($A_0 \sim A_{12}$) and chip enable ($\overline{CE}$) for the EPROM are in "1" state.

• Standby State

Only power supply pins (pins 1 and 21) and $\overline{STBY}$ pin (pin 7) are active. As for the clock pin EXTAL(pin3),its input is fixed internally so the MCU is not influenced by the pin conditions. XTAL (pin 2) is in "1" output. All the other pins are in high impedance. Both address ($A_0 \sim A_{12}$) and chip enable ($\overline{CE}$) for the EPROM are in "1" output.

Table 15 Pin Condition in Sleep Mode

Pin \ Mode		0	1	2, 4	5	6	7
Port 1 $P_{10} \sim P_{17}$	Function	I/O Port	Lower Address Bus	I/O Port	←	←	←
	Condition	Keep the condition just before sleep	Output "1"	Keep the condition just before sleep	←	←	←
Port 2 $P_{20} \sim P_{24}$	Function	I/O Port	←	←	←	←	←
	Condition	Keep the condition just before sleep	←	←	←	←	←
Port 3 $P_{30} \sim P_{37}$	Function	$\overline{E}$: Lower Address Bus E: Data Bus	Data Bus	$\overline{E}$: Lower Address Bus E: Data Bus	Data Bus	$\overline{E}$: Lower Address Bus E: Data Bus	I/O Port
	Condition	$\overline{E}$: Output "1" E: High Impedance	High Impedance	$\overline{E}$: Output "1" E: High Impedance	High Impedance	$\overline{E}$: Output "1" E: High Impedance	Keep the condition just before sleep
Port 4 $P_{40} \sim P_{47}$	Function	Upper Address	←	←	Lower Address Bus or Input Port	Upper Address Bus or Input Port	I/O Port
	Condition	Output "1"	←	←	Address Bus: Output "1" Port: Keep the condition just before sleep	←	Keep the condition just before sleep
SC_2		Output "1" (Read Condition)	←	←	←	←	Output "1"
SC_1		Output Address Strobe	←	←	Output "1"	Output Address Strobe	Input Pin

Table 16 Pin Condition during Reset

Pin \ Mode	0	1	2, 4	5	6	7
Port 1 $P_{10} \sim P_{17}$	high impedance (input)	←	←	←	←	←
Port 2 $P_{20} \sim P_{24}$	high impedance (input)	←	←	←	←	←
Port 3 $P_{30} \sim P_{37}$	$\overline{E}$: "1" output E: "1" output (Note) (high impedance)	high impedance	$\overline{E}$: "1" output E: "1" output (Note) (high impedance)	high impedance	$\overline{E}$: "1" output E: "1" output (Note) (high impedance)	high impedance (input)
Port 4 $P_{40} \sim P_{47}$	high impedance (input)	←	←	←	←	←
SC_2	"1" output (READ)	←	←	←	←	"1" output
SC_1	$\overline{E}$: "1" output E: high impedance	←	←	"1" output	$\overline{E}$: "1" output E: high impedance	high impedance (input)
$A_0 \sim A_{12}$, $\overline{CE}$	"1" output	←	←	←	←	←

[Note] In mode 0, 2, 4, 6, port 3 is set to "1" output state during E = "1" and it causes the conflict with the output of external memory. Following 1 and 2 should be done to avoid the conflict;
(1) Construct the system that disables the external memory during reset.
(2) Add 4.7kΩ pull-down resistance to the SC_1 pin (AS) to make SC_1 pin "0" level during E = "1".
This operation makes port 3 high impedance state.

■ PRECAUTION TO EMULATE THE HD6301V1 BY HD63P01M1

The internal EPROM of the HD63P01M1 provides 8k bytes address space located from $E000 through $FFFF. The followings should be noted to emulate the HD6301V1 (4k bytes internal ROM) with the HD63P01M1.

1. Mode 5 (Expanded Non-multiplexed Mode) and Mode 7 (Single Chip Mode)
   Use 4k bytes of EPROM address space located from $F000 through $FFFF.
2. Mode 6 (Expanded Multiplexed Mode)
   Use 4k bytes of EPROM address space located from $F000 through $FFFF. But do not use 4k bytes from $E000 through $EFFF because these addresses are internal for the HD63P01M1, while these are external for the HD6301V1.
3. Mode 1, 2, 4
   No need to be careful, since ROM address is external in these cases.

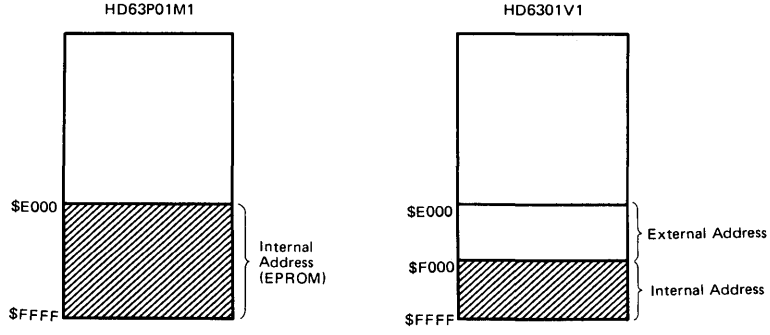


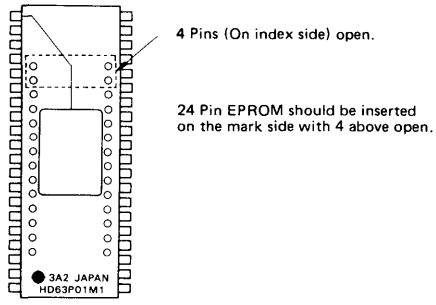
Figure 32 Address Map of Mode 6

■ PRECAUTION TO USE THE EPROM ON-PACKAGE 8 BIT SINGLE CHIP MICROCOMPUTER

Please pay attention to the followings, since this MCU has special structure with pin socket on the package.

(1) Don't apply high static voltage or surge voltage over MAXIMUM RATINGS to the socket pins as well as the LSI pins. If not, that may cause permanent damage to the device.

(2) When using 32k EPROM (24 pin), insert it on the mark side and let the four above pins open.



(3) When using this in production like mask ROM type single chip microcomputer, pay attention to the followings to keep the good contact between the EPROM pins and socket pins.

(a) When soldering the LSI on a print circuit board, the recommended condition is
Temperature : lower than 250°C
Time : within 10 sec.

(b) Note that the detergent or coating will not get in the socket during flux washing or board coating after soldering, because that may cause bad effect on socket contact.

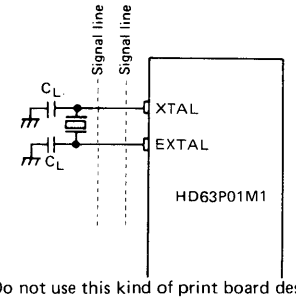
(c) Avoid permanent application of this under the condition of vibratory place and system.

(d) The socket, inserted and pulled repeatedly loses its contactability. It is recommended to use new one when applied in production.

Ask our sales agent about anything unclear.

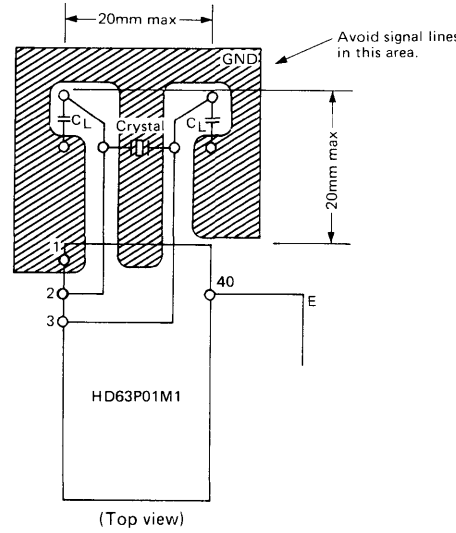
■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig. 33, there is a case that the cross talk disturbs the normal oscillation if signal lines are put near the oscillation circuit. When designing a board, pay attention to this. Crystal and C_L must be put as near the HD63P01M1 as possible.



Do not use this kind of print board design.

Figure 33 Precaution to the boad design of oscillation circuit



(Top view)

Figure 34 Example of Oscillation Circuits in Board Design

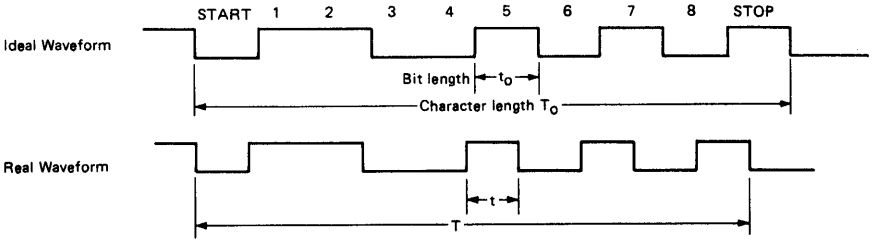
■ RECEIVE MARGIN OF THE SCI

Receive margin of the SCI contained in the HD63P01M1 is shown in Table 17.

Note: SCI = Serial Communication Interface

Table 17

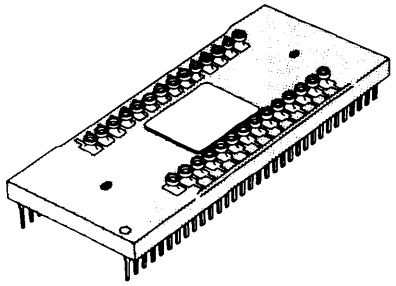
Bit distortion tolerance (t−to) /to	Character distortion tolerance (T−To) /To
±25%	±3.75%



HD63P05Y0, HD63PA05Y0, HD63PB05Y0
CMOS MCU (Microcomputer Unit)

The HD63P05Y0 is an CMOS 8-bit single-chip microcomputer unit which has a 4k-byte or 8k-byte EPROM on the package. It is compatible with the HD6305Y0 except for ROM which is not included in the HD63P05Y0. It can be used not only for debugging and evaluating the internal program of HD6305X0 or HD6305Y0, but also for small-sized production preceding mask ROM.

HD63P05Y0, HD63PA05Y0,
HD63PB05Y0



(DC-64SP)

■ FEATURES

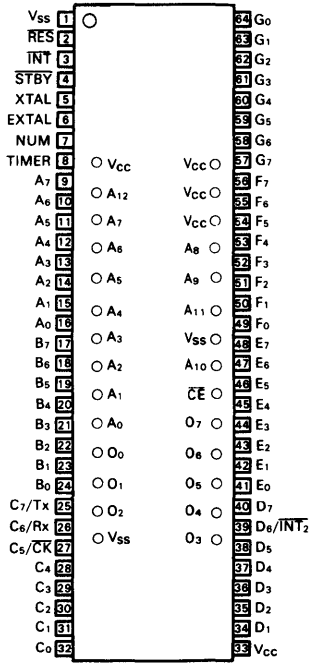
- Pin compatible with HD6305X0 and HD6305Y0
- 256-byte of RAM
- A total of 55 terminals, including 32 I/O's, 7 inputs and 16 outputs.
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes – Wait, Stop and Standby Mode
- Minimum instruction cycle time
  HD63P05Y0 . . . . . 1 μs (f = 1 MHz)
  HD63PA05Y0 . . . . 0.67 μs (f = 1.5 MHz)
  HD63PB05Y0 . . . . . 0.5 μs (f = 2 MHz)
- Similar to HD6800 instruction set
- Bit manipulation
- Bit test and branch
- Versatile interrupt handling
- Full set of conditional branches
- New instructions – STOP, WAIT, DAA
- Applicable to 4k or 8k bytes of EPROM
  4k bytes; HN482732A
  8k bytes; HN482764, HN27C64

■ TYPE OF PRODUCTS

Type No.	Bus Timing	Applied EPROM
HD63P05Y0	1 MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PA05Y0	1.5 MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PB05Y0	2 MHz	HN482732A-25, HN482764, HN27C64-25

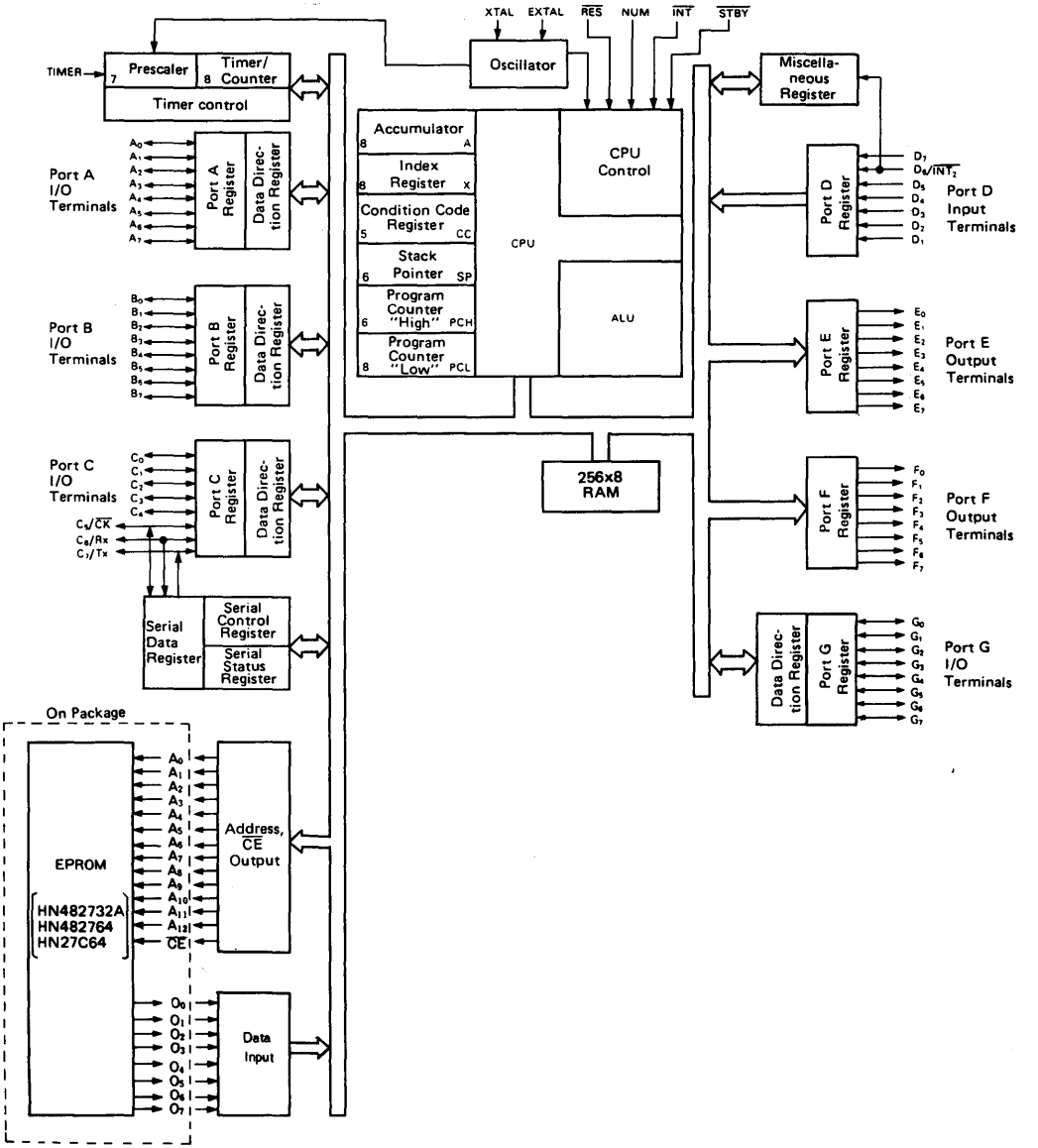
(Note) EPROM is not attached to the MCU.

■ PIN ARRANGEMENT



(Top View)

■ BLOCK DIAGRAM



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 ~ +7.0	V
Input voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating temperature	T_{opr}	0 ~ +70	°C
Storage temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$.

■ ELECTRICAL CHARACTERISTICS

• **DC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)**

Item		Symbol	Test condition	min	typ	max	Unit
Input voltage "High"	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC} − 0.5	−	V_{CC} + 0.3	V
	EXTAL			V_{CC} x 0.7	−	V_{CC} + 0.3	V
	Others			2.0	−	V_{CC} + 0.3	V
Input voltage "Low"	All Input	V_{IL}		−0.3	−	0.8	V
Current *** dissipation	Operating	I_{CC}	f = 1MHz*	−	5	10	mA
	Wait			−	2	5	mA
	Stop			−	2	10	μA
	Standby			−	2	10	μA
Input leakage current	$\overline{TIMER}$, $\overline{INT}$, $D_1 \sim D_7$, $\overline{STBY}$	$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} − 0.5V	−	−	1	μA
Three-state current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $G_0 \sim G_7$, $E_0 \sim E_7$**, $F_0 \sim F_7$**	$\|I_{TSI}\|$		−	−	1	μA
Input capacity	All terminals	C_{in}	f = 1MHz, V_{in} = 0V	−	−	15	pF

* The value at f = xMHz can be calculated by the following equation: I_{CC} (f= xMHz) = I_{CC} (f= 1MHz) multiplied by x
** At standby mode
*** All output and $\overline{RES}$ terminals are open ($V_{IH\ min}$ = V_{CC}−1.0V, $V_{IL\ max}$ = 0.8V), and I_{CC} of EPROM is not included.

• AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item	Symbol	Test condition	HD63P05Y0			HD63PA05Y0			HD63PB05Y0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock frequency	f_{cl}		0.4	–	4	0.4	–	6	0.4	–	8	MHz
Cycle time	t_{cyc}		1.0	–	10	0.666	–	10	0.5	–	10	μs
$\overline{INT}$ pulse width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ pulse width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ pulse width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
TIMER pulse width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation start time (crystal)	t_{OSC}	C_L = 22pF ± 20% R_s = 60Ω max	–	–	20	–	–	20	–	–	20	ms
Reset delay time	t_{RHL}	External cap. 2.2μF	80	–	–	80	–	–	80	–	–	ms

• Port Electrical Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item		Symbol	Test condition	min	typ	max	Unit
Output voltage "High"	Ports A, B, C, G, E, F	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC} - 0.7	–	–	V
Output voltage "Low"		V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input voltage "High"	Ports A, B, C, D, G	V_{IH}		2.0	–	V_{CC} + 0.3	V
Input voltage "Low"		V_{IL}		- 0.3	–	0.8	V
Input leakage current		$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} - 0.5V	–	–	1	μA

• SCI Timing (V_{CC} = 5.0V±10%, V_{SS} = GND and T_a = 0 ~ +70°C unless otherwise specified)

Item	Symbol	Test condition	HD63P05Y0			HD63PA05Y0			HD63PB05Y0			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock cycle	t_{Scyc}	Fig. 1, Fig. 2	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data output delay time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data set-up time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data hold time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

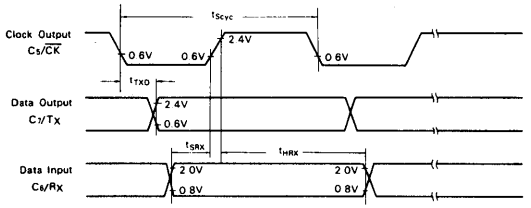


Figure 1 SCI Timing (Internal Clock)

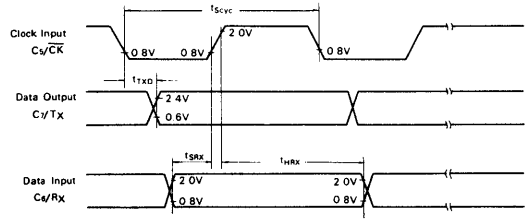
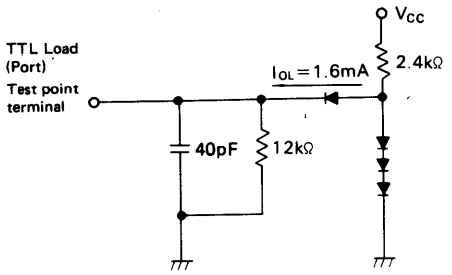


Figure2 SCI Timing(External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ.

Figure 3 Test Load

■DESCRIPTION ON PIN FUNCTIONS

Here is the description of HD63P05Y0 MCU input and output signals.

●V_{CC}, V_{SS}

Power is supplied to the MCU using these two pins. When the operating voltage of the EPROM is 5.0V ± 5%, change V_{CC} according to that of EPROM.

●$\overline{INT}$, $\overline{INT_2}$

Used for requesting an external interrupt to the MCU. For details, see "INTERRUPT". The $\overline{INT_2}$ is used as the port D_6 pin.

●XTAL, EXTAL

Are input pins to the internal clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic oscillator is connected to these pins. For instance, in order to obtain the system clock 1 MHz, a 4 MHz resonant fundamental crystal is useful because the divide-by-4 circuitry is included. EXTAL accepts an external clock input of duty 50% (±10%) to drive, then the internal clock is a quarter the frequency of the external clock. External drive frequency will be 4 or less times the maximum internal clock. For external driving, no XTAL should be connected. Refer to "INTERNAL OSCILLATOR" for using these input pins.

●TIMER

Is an external input pin to control the internal Timer. For details, see "TIMER".

●$\overline{RES}$

Is used for resetting MCU. For details, see "RESET".

●NUM

Is not for user application. It must be grounded to V_{SS}.

●INPUT/OUTPUT PINS (A_0~A_7, B_0~B_7, C_0~C_7, G_0~G_7)

32 pins consist of four 8-bit I/O ports (A, B, C, G). Each of them is used as input or output pin, through program control of the data direction register. For details, see "I/O PORTS".

●INPUT PINS (D_1 ~ D_7)

Are 7 input-only pins compatible with the TTL and CMOS. D_6 is used as $\overline{INT_2}$. When the D_6 is used as the port, set the $\overline{INT_2}$ interrupt mask bit of the miscellaneous register to "1" to prevent an $\overline{INT_2}$ from accidental interruption.

●OUTPUT PINS (E_0 ~ E_7, F_0 ~ F_7)

Are 16 output-only pins compatible with the TTL and CMOS.

●$\overline{STBY}$

Used for bringing the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and internal situation is reset. Fore details, see "STANDBY MODE". The following are I/O pins for serial communication interface (SCI), and used as ports C_5, C_6, and C_7. For details, see "SERIAL COMMUNICATION INTERFACE".

●$\overline{CK}$ (C_5)

Used to input or output clocks when receiving or transmitting serial data.

●Rx (C_6)

Used to receive serial data.

●Tx (C_7)

Used to transmit serial data.

■MEMORY MAP

The memory map of the HD63P05Y0 MCU is shown in Fig. 4. During interrupt, the contents of the registers are saved in the stack as shown in Fig. 5. The saving begins with the lower byte (PCL) of the program counter. Then the stack pointer value is decremented, and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CC) are stacked in this order. In subroutine calls, only the contents of the program counter (PCH and PCL) are stacked.

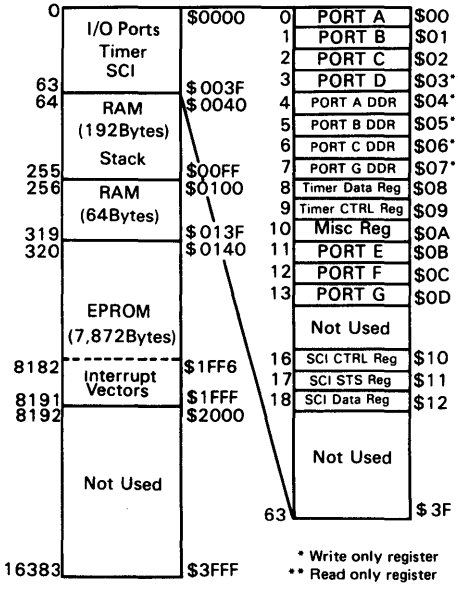
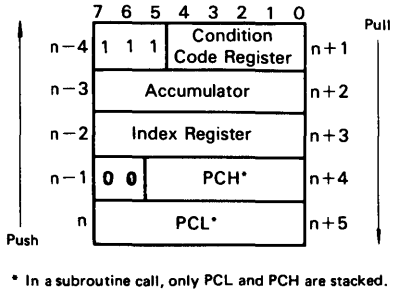


Figure 4 Memory Map of HD63P05Y0 MCU



* In a subroutine call, only PCL and PCH are stacked.

Figure 5 Sequence of Interrupt Stacking

■ REGISTERS

There are five registers which the programmers can handle.

● Accumulator (A)

The accumulator is a general purpose 8-bit register which holds operands, the results of arithmetic operations or data processing.

● Index Register (X)

The index register is an 8-bit register used for the index addressing mode. It contains an 8-bit value to be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instruction. The index register may also be used as a temporary storage area.

● Program Counter (PC)

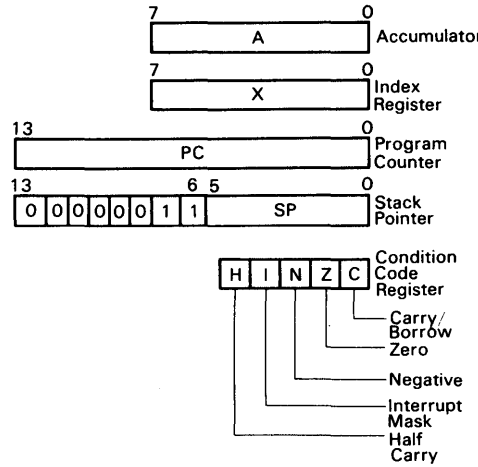


Figure 6 Programming Model

The program counter is a 14-bit register which contains the address of the next instruction to be executed.

● Stack Pointer (SP)

The stack pointer is a 14-bit register which indicates the address of the next free location in the stack. Initially, the stack pointer is set to $00FF. It is decremented as data is pushed in, and incremented as it is pulled out. The upper 8 bits of the stack pointer are fixed to 00000011.

During an MCU reset or when the reset stack pointer (RSP) instruction is executed, the pointer is set to the location $00FF. A subroutine or interrupt may be nested down to location $00C1 which allows programmers to use up to 31 levels of subroutine call or 12 levels of interrupt response.

● Condition Code Register (CC)

The condition code register is a 5-bit register. Each bit indicates the result of the executed instruction. These bits can be individually tested by conditional branch instructions. The CC bits are as follows.

Half Carry (H): Used to indicate a carry occurring between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts to be masked except for software ones. If an interrupt occurs while the bit I is set, the interrupt is latched, and processed as soon as the interrupt mask bit (I) is reset. (Exactly, the interrupt enters the processing routine after the instruction next to the CLI is executed.)

Negative (N): Used to indicate that the result of the latest arithmetic operation, logical operation or data processing is negative (Bit 7 is logical "1").

Zero (Z): Used to indicate that the result of the latest arithmetic operation, logical operation or data processing is zero.

Carry/Borrow (C): Shows a carry or borrow occurring in the latest arithmetic operation. This bit is also affected by the Bit Test and Branch, Shift and

Rotate instructions.

■INTERRUPT

There are six different types of interrupt: external interrupt ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, TIMER 2), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER, and SCI and TIMER 2 respectively generate the same vector address. When an interrupt occurs, the program in execution stops and CPU state at the interrupt is saved onto the stack. In addition, the interrupt causes the interrupt mask bit (I) in the condition code register to be set and obtains the start address of the interrupt routine from an assigned interrupt vector address before the interrupt routine starts from the state address. The system exits from the interrupt routine by RTI instruction. When the RTI instruction is executed, the CPU state before the interrupt (saved in the stack) is pulled and the CPU starts the program again from the next step to the interrupted one. Table 1. lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$1FFE, $1FFF
SWI	2	$1FFC, $1FFD
$\overline{INT}$	3	$1FFA, $1FFB
TIMER/$\overline{INT_2}$	4	$1FF8, $1FF9
SCI/TIMER$_2$	5	$1FF6, $1FF7

A flow chart of the interrupt is shown in Fig. 7. Also a block diagram of the interrupt request source is shown in Fig. 8. In the block diagram, both the external interrupts $\overline{INT}$ and $\overline{INT_2}$ are edge trigger inputs. At the falling edge of the input, an interrupt request is generated and latched.

The $\overline{INT}$ interrupt request is automatically cleared if a program jumps to the $\overline{INT}$ routine. In the case of $\overline{INT_2}$, the

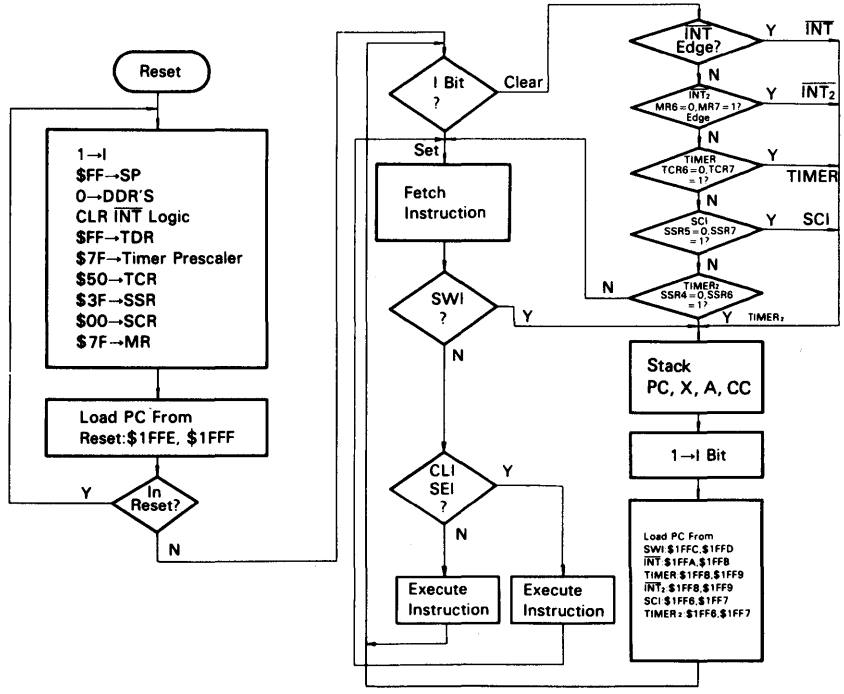


Figure 7 Interrupt Flowchart

interrupt request is cleared when "0" is written in bit 7 of the miscellaneous register. For external interrupts($\overline{INT}$,$\overline{INT_2}$), internal timer interrupts (TIMER, TIMER2) and serial interrupt (SCI), these interrupt requests are held, but not operated, while bit I of the condition code register is set. Immediately after the bit I is cleared, the corresponding interrupt is activated.

The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by bit 6 of the timer control register, the SCI interrupt by bit 5 of the serial status register and the TIMER2 interrupt by bit 4 of the serial status register.

The state of the $\overline{INT}$ pin is tested by BIL or BIH instructions. The $\overline{INT}$ falling edge detector circuit and its latch circuit are independent of tests by these instructions. The state of $\overline{INT_2}$ pin is also independent.

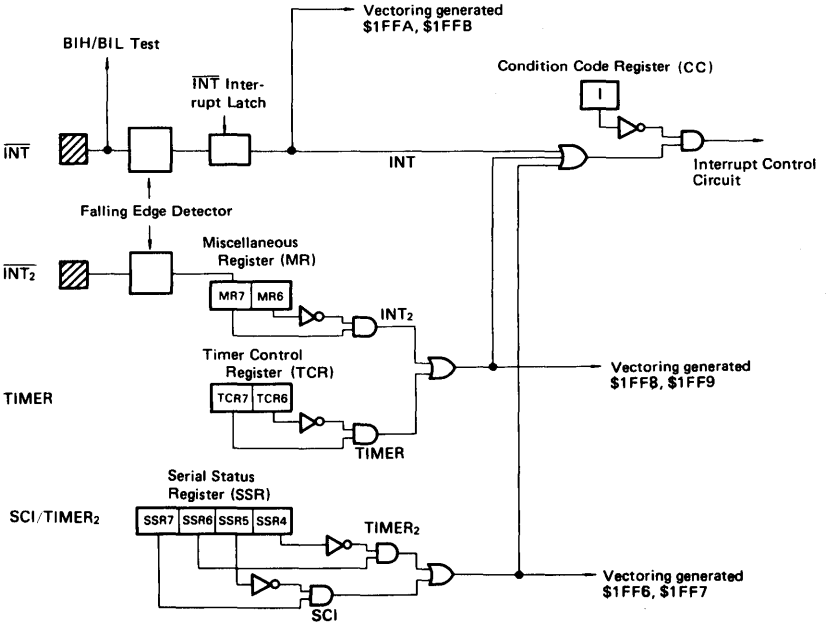
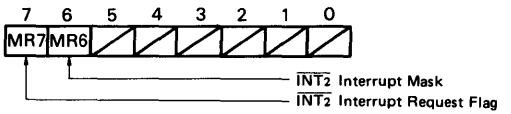


Figure 8 Interrupt Request Generation Circuitry

- **Miscellaneous Register (MR: $000A)**

The interrupt vector address for external interrupt $\overline{INT_2}$ is the same as that for the TIMER interrupt, as shown in Table 1. For this reason, a special register called a miscellaneous register (MR: $000A) is available for $\overline{INT_2}$ interrupt control. Bit 7 of the miscellaneous register is of $\overline{INT_2}$ interrupt request flag. When the falling edge is detected at the $\overline{INT_2}$ pin, "1" is set in bit 7. The software in the interrupt routine (vector address: $1FF8, $1FF9) checks to see if it is $\overline{INT_2}$ interrupt. Bit 7 is reset by software. Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If the bit is set to "1", the $\overline{INT_2}$ interrupt is disabled.

Miscellaneous Register (MR;$000A)



Both "READ" and "WRITE" are possible with bit 7, but "1" can not be written to in this bit by software. Therefore, interrupt requests by software are not possible. By resetting, bit 7 is cleared and bit 6 is entered "1".

- **TIMER**

The MCU timer block diagram is shown in Fig. 9. The 8-bit counter is loaded under program control and is decremented by the clock input. When the timer data register (TDR) reaches 0, the timer interrupt request bit (bit 7) in the timer control register is set. The MCU responds to this interrupt by saving the present CPU state in the stack, fetching the timer interrupt routine address from address $1FF8 and $1FF9. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also disable the timer interrupt. The source clock for the timer can be either an external signal from the timer input pin or the internal E signal (oscillator clock divided by 4). If the E signal is selected as the source, the clock input can be gated by the input to the timer input pin.

When the timer counter reaches "0", it starts counting down from $FF. The count can be monitored at any time by reading the timer data register. This function allows knowledge of the length of time after a timer interrupt with a program, without destroying the contents of the counter.

When the MCU is reset, both the prescaler and counter return to the initial state of logical "1". At the same time, the timer interrupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set. Write "0" in the timer interrupt request bit (bit 7) to clear it.

TCR7	Timer interrupt request
0	Absent
1	Present

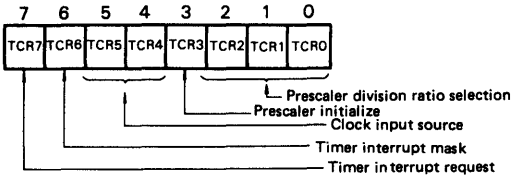
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

• Timer Control Register (TCR; $0009)

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After resetting, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$FF" immediately after the reset.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under timer terminal control
1	0	No clock input (counting stopped)
1	1	Event input from timer terminal

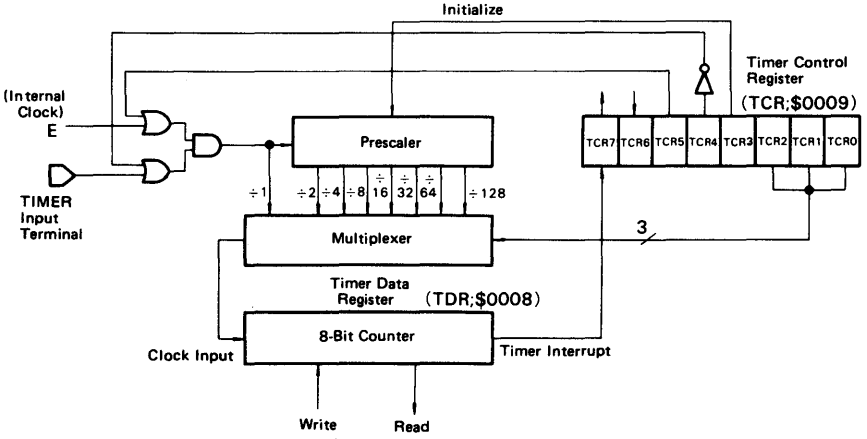


Figure 9 Timer Block Diagram

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

The prescaler is initialized by writing "1" in bit 3. The bit is always "0", when "READ". A prescaler division ratio is selected by a combination of the three bits (bits 0, 1 and 2) of the timer control register (See Table 3). There are eight division ratios; ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128.

After resetting, the TCR returns to the ÷1 mode. The timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. The bit is cleared by writing "0" into it.

■SERIAL COMMUNICATION INTERFACE (SCI)

Used for 8-bit data communication. Transfer rate ranges from 1μs to about 32 ms (when oscillated at 4 MHz), and there are sixteen selections.

The SCI consists of three registers, one octal counter and one prescaler. (See Fig. 10) The SCI communicates with the CPU through the data bus, and with peripherals through bits 5, 6 and 7 of port C. Operations of the registers and data transfer are described below.

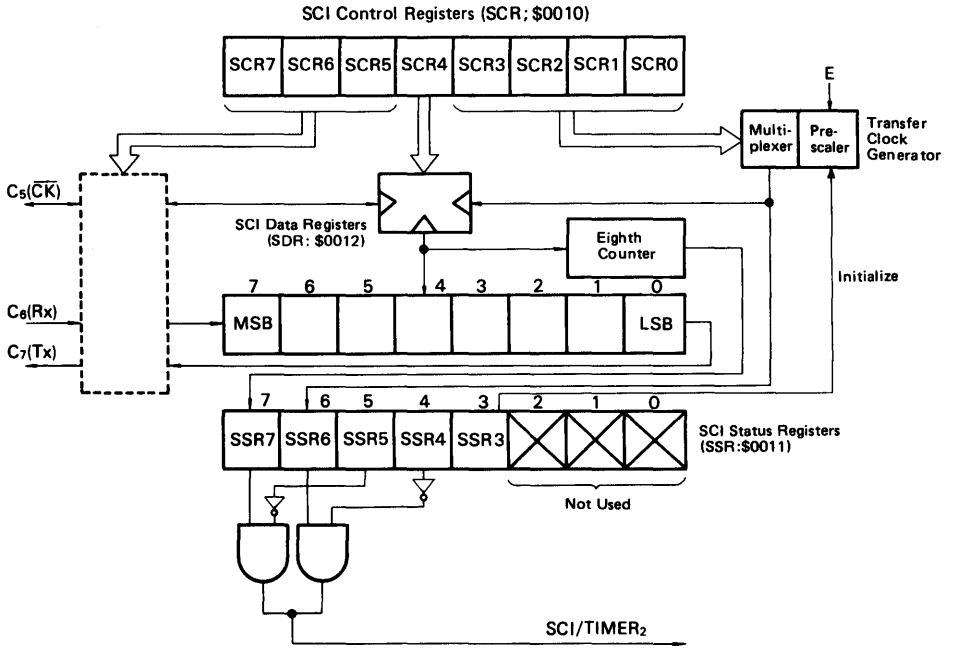


Figure 10 SCI Block Diagram

•SCI Control Register (SCR; $0010)

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

SCR7	C_7 terminal
0	Used as I/O terminal (by DDR).
1	Serial data output (DDR output)

SCR6	C_6 terminal
0	Used as I/O terminal (by DDR).
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	C_5 terminal
0	0	–	Used as I/O terminal (by DDR).
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)

When this bit is set, the DDR corresponding to the C_7 becomes "1" and this terminal serves for output of SCI data. After resetting the bit is cleared to "0".

Bit 6 (SCR6)

When this bit is set, the DDR corresponding to the C_6 becomes "0" and this terminal serves for input of SCI data. After resetting the bit is cleared to "0".

Bits 5 and 4 (SCR5, SCR4)

These bits are used to select a clock source. After resetting the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)

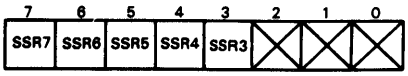
These bits are used to select a transfer clock rate. After resetting the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

•SCI Data Register (SDR; $0012)

A serial-parallel conversion register that is used for transfer of data.

•SCI Status Register (SSR; $0011)



Bit 7 (SSR7)

Bit 7 is the SCI interrupt request bit which is set on completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5="1". The bit can be cleared by writing "0" into it.

Bit 6 (SSR6)

Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is commonly used with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When resetting , the bit is cleared. It can also be cleared by writing "0" into it. (For details, see $TIMER_2$).

Bit 5 (SSR5)

Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When resetting, it is set to "1".

Bit 4 (SSR4)

Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When resetting, it is set to "1".

Bit 3 (SSR3)

When "1" is written into this bit, the prescaler of the transfer clock generator is initialized. When "READ", the bit is always "0".

Bits 2 ~ 0

Not used.

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

• Data Transmission

By writing the desired control bits into the SCI control registers, a transfer rate and a transfer clock source are determined and bits 7 and 5 of port C are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the C_7/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock (See Fig. 11). When 8 bits of data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register with the rising edge of the last serial clock. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the C_7/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 to 3 of the SCI control register is ignored, and the $C_5/\overline{CK}$ terminal is set as input. If the internal clock has been selected, the $C_5/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 to 3 of the SCI control register.

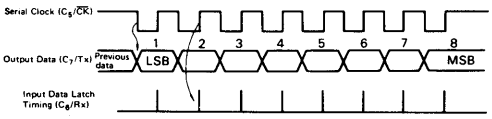


Figure 11 SCI Timing Chart

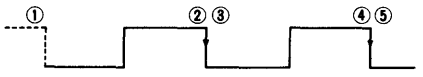
• Data Reception

By writing the desired control bits into the SCI control register, a transfer rate and a transfer clock source are determined and bit 6 and 5 of port C are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed after reading subsequent received data. It must be taken after reset and after not reading subsequent received data.)

The data from the C_6/Rx terminal is input to the SCI data register synchronously with the rising edge of the serial clock (see Fig. 11). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the data is received synchronously with the clock from the $C_5/\overline{CK}$ terminal. If the internal clock has been selected, the $C_5/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

• $TIMER_2$

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 to 0 of the SCI control register (4 μs to approx. 32 ms (when oscillated at 4 MHz)) is input to bit 6 of the SCI status register and the $TIMER_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, $TIMER_2$ can be used as a reload counter or clock.



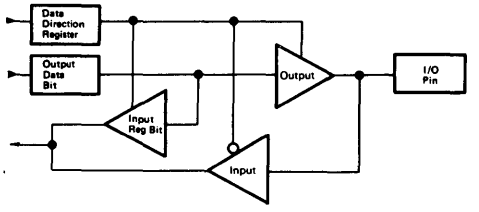
① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : $TIMER_2$ interrupt request
③, ⑤ : $TIMER_2$ interrupt request bit cleared

$TIMER_2$ is commonly used with the SCI transfer clock generator. If wanting to use $TIMER_2$ independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

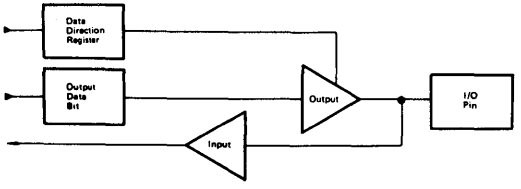
■I/O PORTS

There are 32 input/output terminals (ports A, B, C, G). Each I/O terminal can be selected for either input or output by the data direction register. Specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B or C reads latched data if it has been programmed as output, even with the output load, the output level fluctuating. (See Fig. 12-a.) For port G, in this case, the level of the pin is always read when it is read. (See Fig. 12-b.) This implies that, even when "1" stays output, port G may read "0" if the load condition causes the output voltage less than 2.0V.



Bit of data direction register	Bit of output data	Status of output	Input to MCU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

a. Ports A, B and C



b. Port G

Figure 12 Input/Output Port Diagram

When resetting the data direction register and data register go to "0" and all input/output terminals are used as input.

There are 16 output-only terminals (ports E and F). Each of them can also read. In this case, latched data is read even with the output terminal level being fluctuated by the output load (as with ports A, B and C).

When resetting, "Low" level is output from each output terminal.

Seven input-only terminals are available (port D). Writing to these ones is invalid.

All input/output terminals, output terminals and input terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports or input ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite their not being used.

■RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 13.) On power up, the reset input must be held "Low" for at least t_{OSC} to assure that the internal oscillator is stabilized. A sufficient delay time can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 14.

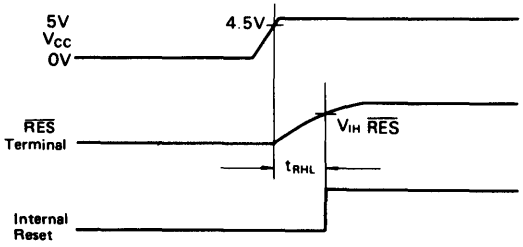


Figure 13 Power On and Reset Timing

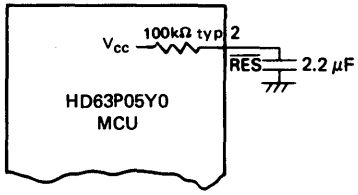


Figure 14 Input Reset Delay Circuit

■INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0MHz) or ceramic oscillator between pins 5 and 6 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 15. Figs. 16 and 17 illustrate the specifications and typical arrangement of the crystal.

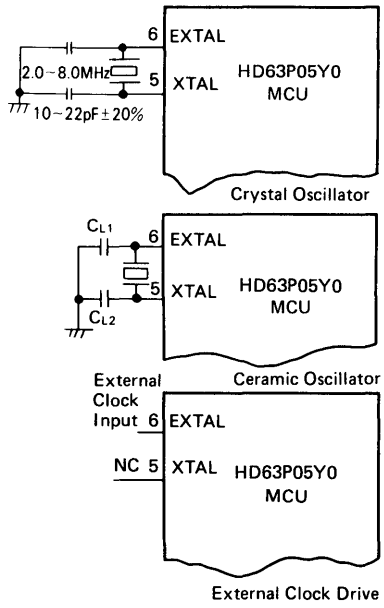


Figure 15 Internal Oscillator Circuit

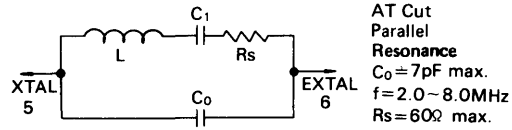
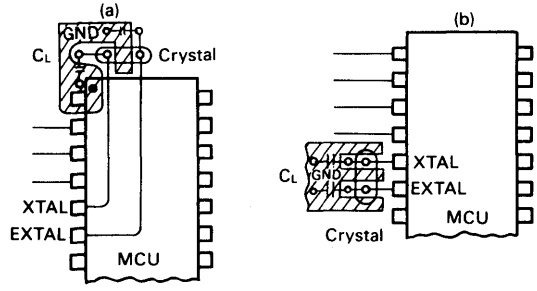


Figure 16 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 17 Typical Crystal Arrangement

■LOW POWER DISSIPATION MODE

The HD63P05Y0 has three low power dissipation modes: wait, stop and standby.

• Wait Mode

When a WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers, RAM and I/O terminals hold the condition just before entering the wait mode. Both address ($A_0 \sim A_{12}$) and chip enable ($\overline{CE}$) for the EPROM are in "1" state.

Release from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER$_2$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode is released and the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after release from the wait mode the MCU executes the instruction following WAIT. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER2) is masked by the timer control register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 18 shows a flowchart of the wait function.

• Stop Mode

When STOP instruction is being executed, the MCU enters the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive but the RAM, register and I/O terminals hold the condition they had just before entering the stop mode. Both address ($A_0 \sim A_{12}$) and chip enable ($\overline{CE}$) for the EPROM are in "1" state.

Release from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode.

When an interrupt is requested and accepted by the CPU, the stop mode is released and the CPU is brought in the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after release from the stop mode, the MCU executes the instruction following STOP. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 19 shows the flowchart of the stop function. Fig. 20 shows a timing chart of the return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$="0" must exceed t_{osc} to assure stabilized oscillation.

• Standby Mode

The MCU enters the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are held. The I/O terminals turn to high-impedance state. Both address ($A_0 \sim A_{12}$) and chip enable ($\overline{CE}$) for the EPROM are in "1" state. The standby mode should be released by bringing $\overline{STBY}$ "High". The CPU must be restarted by resetting. The

timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 21.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 22.

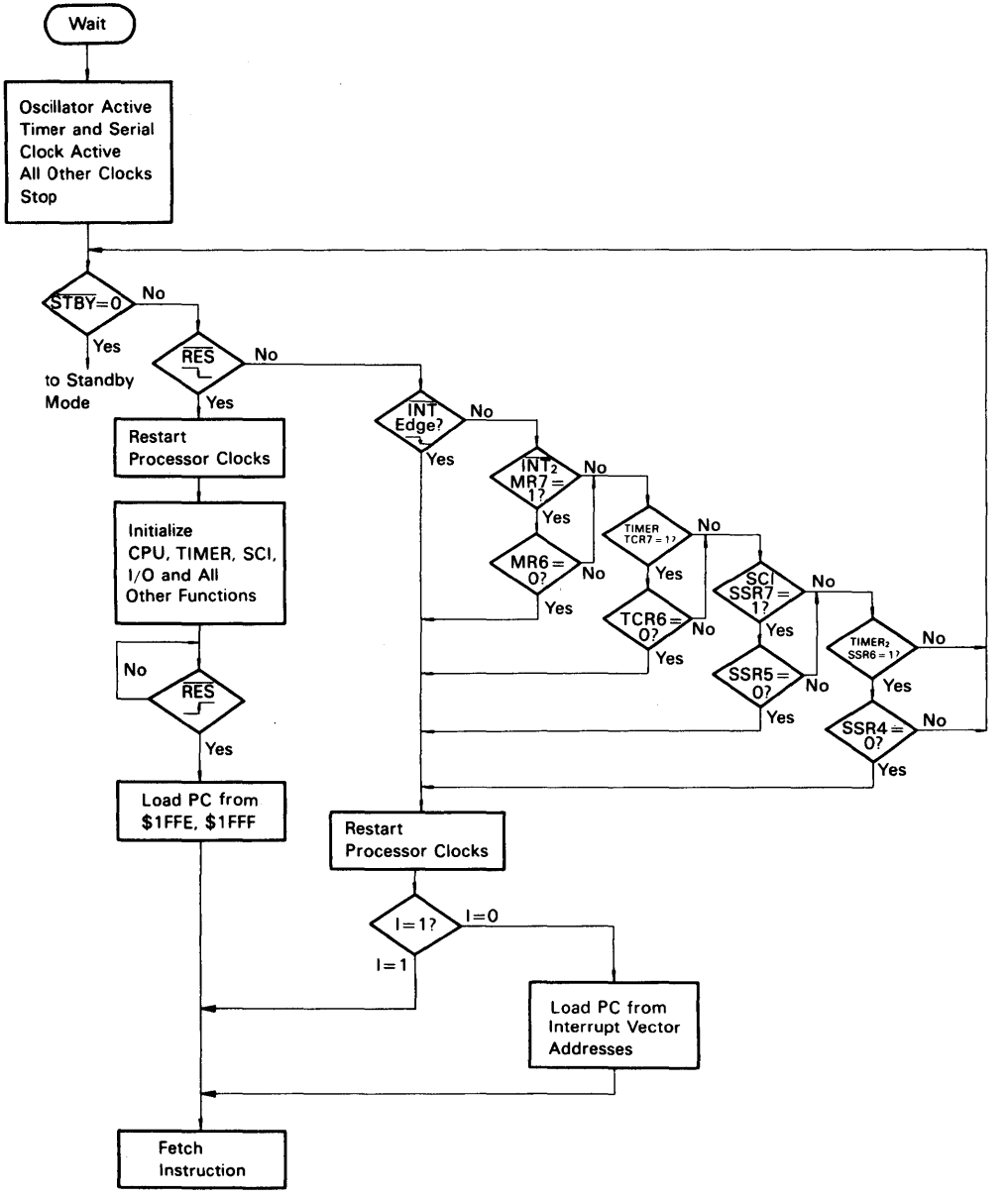


Figure 18 Wait Mode Flow Chart

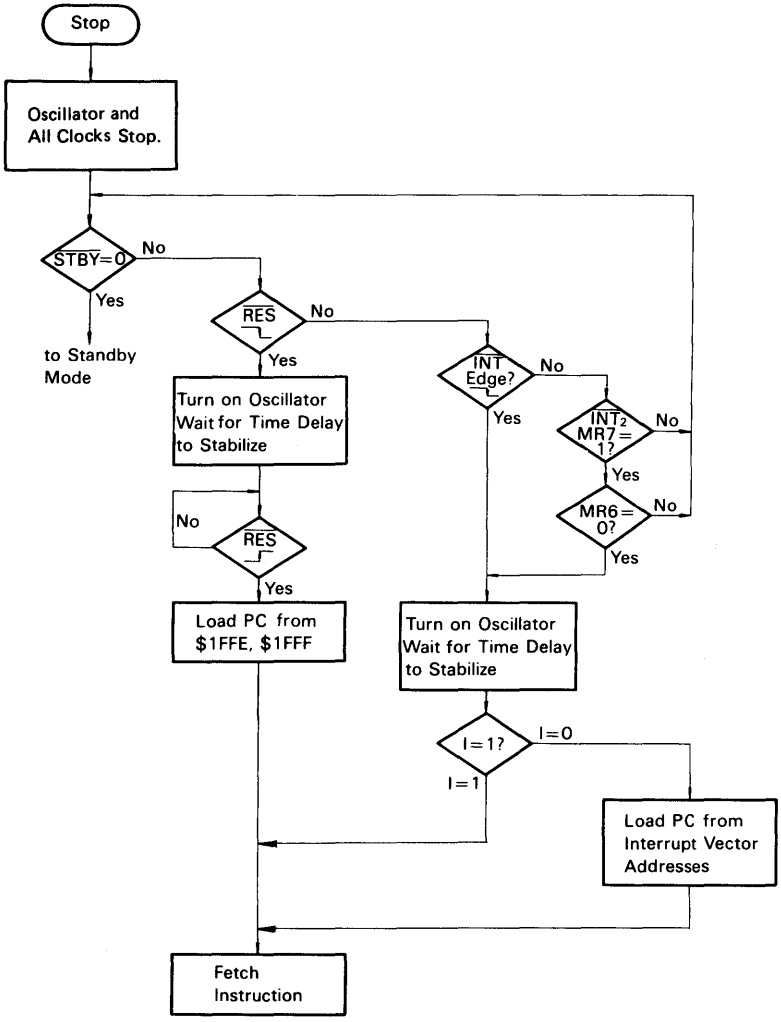


Figure 19 Stop Mode Flow Chart

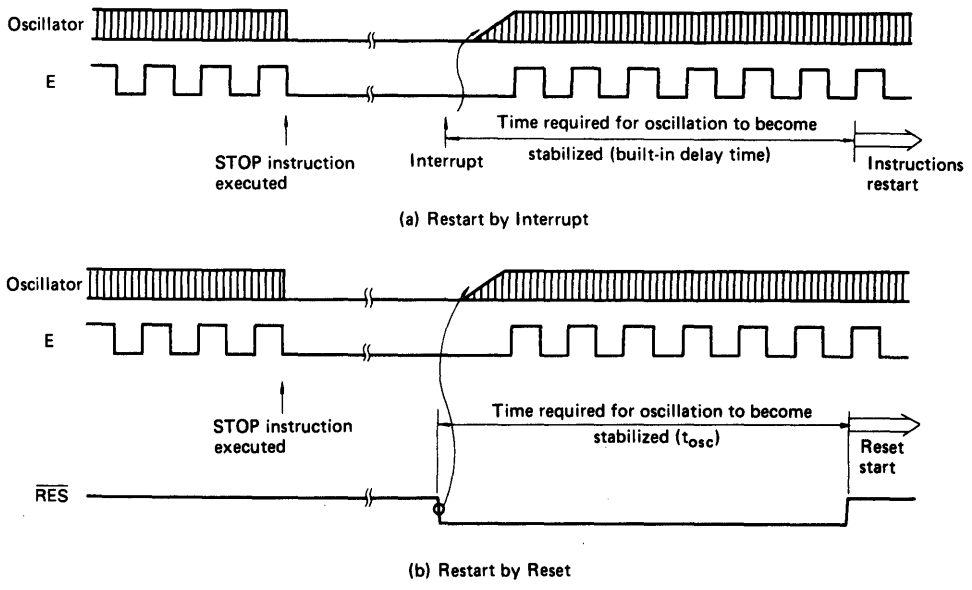


Figure 20 Timing Chart of Releasing from Stop Mode

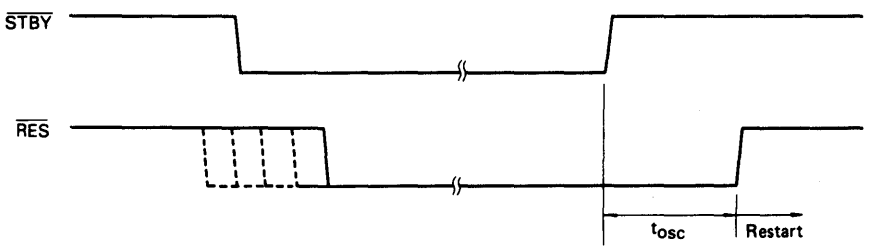


Figure 21 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Hold	Hold	Hold	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Hold	Hold	Hold	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Hold	High impedance	$\overline{STBY}$="High"

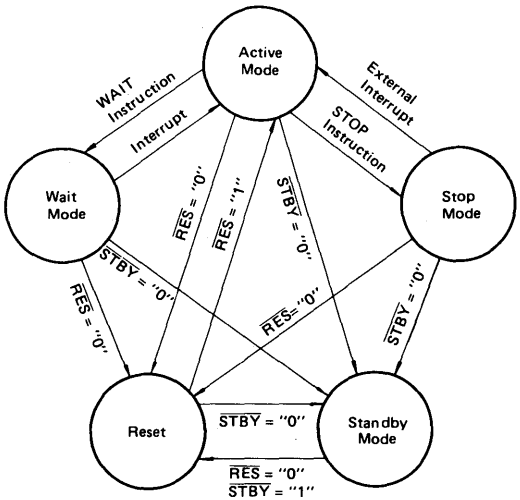


Figure 22 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig.23, the cross talk may disturb normal oscillation if signal lines are set near the oscillation circuit. When designing a board, be careful of this. Crystal and C_L must be put near XTAL and EXTAL pins as possible.

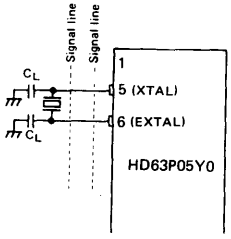


Figure 23 Precaution to the board design of oscillation circuit

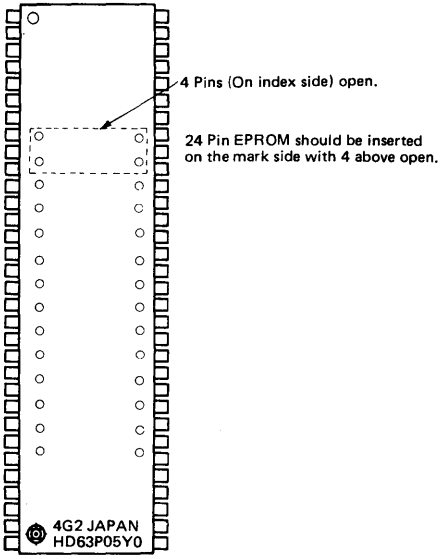
■ PRECAUTION TO USE THE EPROM ON-PACKAGE 8-BIT SINGLE-CHIP MICROCOMPUTER

Please be careful of the following, since this MCU has a special structure with pin socket on the package.

(1) Don't apply high static voltage or surge voltage over MAXIMUM RATINGS to the socket pins as well as the LSI pins. If so, that may cause permanent damage to the device.

(2) When using 32k EPROM (24-pin), insert it leaving the four pins above open.

(3) When inserting this into system products like mask ROM type single chip microcomputer, be careful of the following to give effective contact between the EPROM pins and socket pins.



(a) When soldering the LSI onto a printed circuit board, the recommended condition is
Temperature: lower than 250°C
Time: within 10 sec.

(b) Be careful that detergent or coating does not get into the socket during flux washing or board coating after soldering, because that may cause bad effect on socket contact.

(c) Avoid permanent application of this under conditions

of continuous vibration.

(d) The socket, repeatedly inserted and removed, loses its contactability. It is recommended to use new one when used in production.

■BIT MANIPULATION

The HD63P05Y0 MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM within page 0 or an I/O port (except the write-only registers such as the data direction register). Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM on page 0, or I/O can be manipulated, the user may use a bit within the RAM on page 0 as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 24 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the memory. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
        :
SELF 1. BRCLR 0, PORT A, SELF 1
        BSET 1, PORT A
        BCLR 1, PORT A
        :
```

Figure 24 Example of Bit Manipulation

■ADDRESSING MODES

Ten different addressing modes are available to the HD63P05Y0 MCU.

• Immediate

See Fig. 25. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 26. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. 192 byte RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 27. The extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 28. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 29. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 30. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 31. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 32. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 33. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The value of the test bit is written in the carry bit of the condition code register.

• Implied

See Fig. 34. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

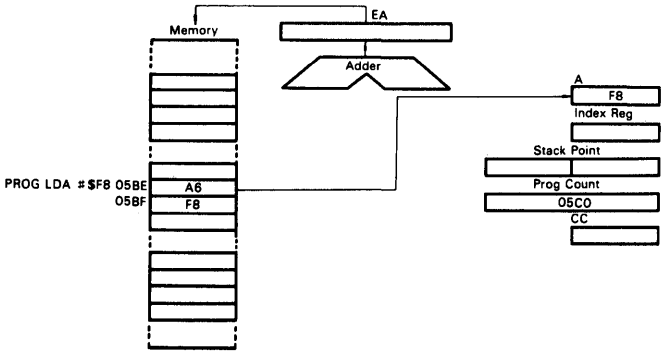


Figure 25 Example of Immediate Addressing

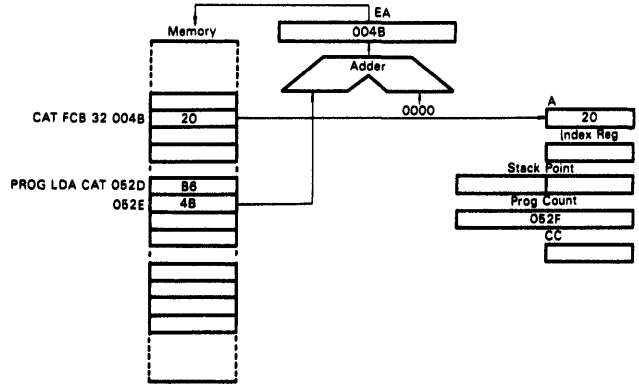


Figure 26 Example of Direct Addressing

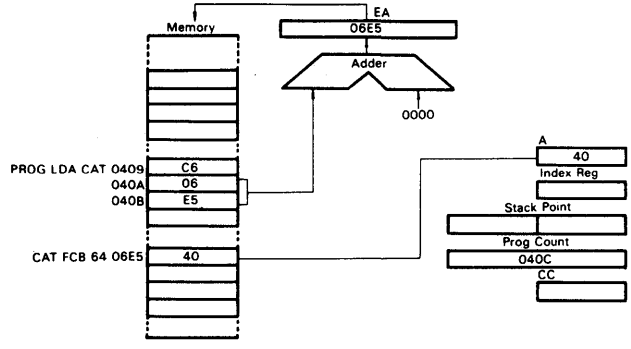


Figure 27 Example of Extended Addressing

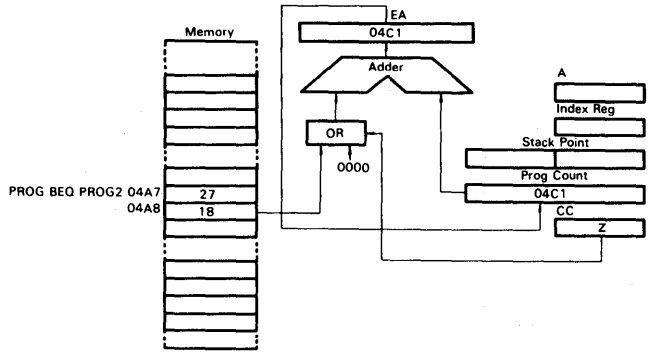


Figure 28 Example of Relative Addressing

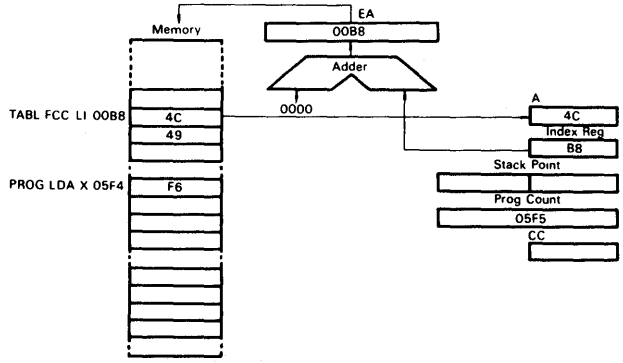


Figure 29 Example of Indexed (No Offset) Addressing

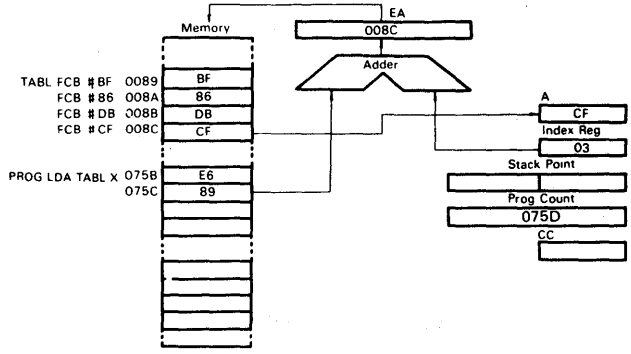


Figure 30 Example of Index (8-bit Offset) Addressing

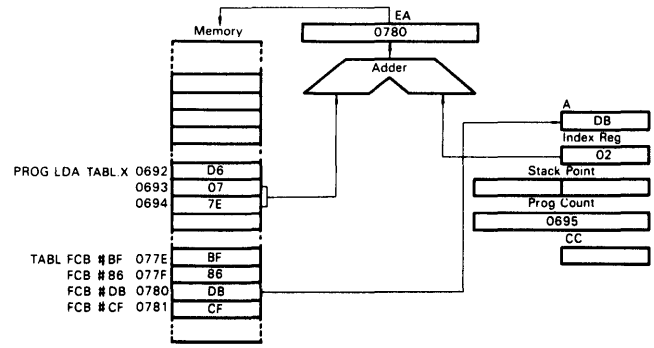


Figure 31 Example of Index (16-bit Offset) Addressing

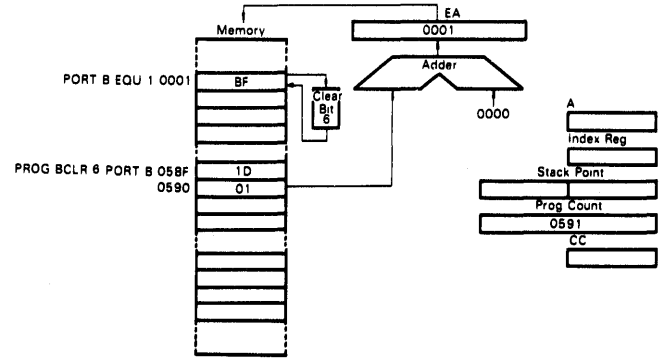


Figure 32 Example of Bit Set/Clear Addressing

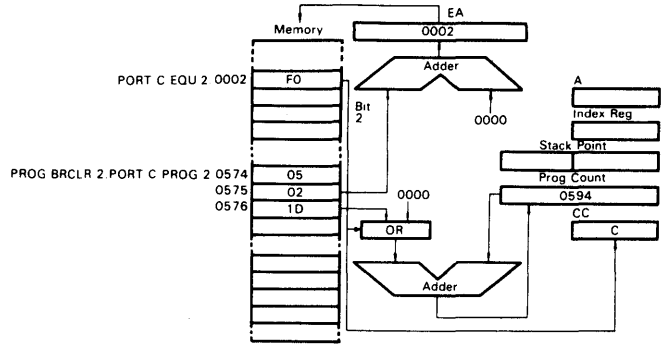


Figure 33 Example of Bit Test and Branch Addressing

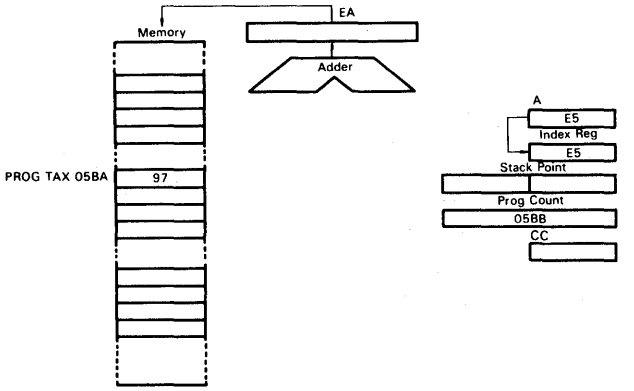


Figure 34 Example of Implied Addressing

■INSTRUCTION SET

There are 62 basic instructions available to the HD63P05Y0 MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the MCU in the alphabetical order.

• Operation Code Map

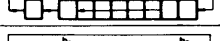
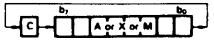
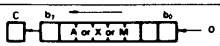
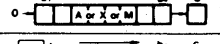
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate			Direct			Extended			Indexed (No Offset)			Indexed (8-Bit Offset)			Indexed (16-Bit Offset)			Boolean/ Arithmetic Operation	Condition Code: H	I	N	Z	C
		OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~						
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	^	^	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	^	^	•
Store A in Memory	STA				B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	^	^	•
Store X in Memory	STX				BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	^	^	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	^	•	^	^	^
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	^	•	^	^	^
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	^	^	^
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	^	^	^
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	^	^	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	^	^	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	^	^	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	^	^	^
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	^	^	^
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	^	^	•
Jump Unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A)			Implied(X)			Direct			Indexed (No Offset)			Indexed (8-Bit Offset)			Boolean/Arithmetic Operation	Condition Code: H	I	N	Z	C
		OP	#	~	OP	#	~	OP	#	~	OP	#	~	OP	#	~							
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	^	^	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	^	^	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	^	^	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	^	^	^
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	^	^	^
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	^	^	^
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	^	^	^
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	^	^
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	^	^	^
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	^	^	^
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	^	^	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	•	•	•	•	•
Branch Never	BRN	21	2	3	None	•	•	•	•	•
Branch IF Higher	BHI	22	2	3	C+Z=0	•	•	•	•	•
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	•	•	•	•	•
Branch IF Carry Clear	BCC	24	2	3	C=0	•	•	•	•	•
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	•	•	•	•	•
Branch IF Carry Set	BCS	25	2	3	C=1	•	•	•	•	•
(Branch IF Lower)	(BLO)	25	2	3	C=1	•	•	•	•	•
Branch IF Not Equal	BNE	26	2	3	Z=0	•	•	•	•	•
Branch IF Equal	BEQ	27	2	3	Z=1	•	•	•	•	•
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	•	•	•	•	•
Branch IF Half Carry Set	BHCS	29	2	3	H=1	•	•	•	•	•
Branch IF Plus	BPL	2A	2	3	N=0	•	•	•	•	•
Branch IF Minus	BMI	2B	2	3	N=1	•	•	•	•	•
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	•	•	•	•	•
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	•	•	•	•	•
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	•	•	•	•	•
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	•	•	•	•	•
Branch to Subroutine	BSR	AD	2	5	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	•	•	•	•	^
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	•	•	•	•	^
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	•	•	•	•	•
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes: Implied OP	#	~	Boolean Operation	Condition Code: H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	•	•	•	•	•
Transfer X to A	TXA	9F	1	2	X→A	•	•	•	•	•
Set Carry Bit	SEC	99	1	1	1→C	•	•	•	•	1
Clear Carry Bit	CLC	98	1	1	0→C	•	•	•	•	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	•	1	•	•	•
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	•	0	•	•	•
Software Interrupt	SWI	83	1	10		•	1	•	•	•
Return from Subroutine	RTS	81	1	5		•	•	•	•	•
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	•	•	•	•	•
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	•	•	•	•	•
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	•	•	∧	∧	∧*
Stop	STOP	8E	1	4		•	•	•	•	•
Wait	WAIT	8F	1	4		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes: Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	Condition Code: H	I	N	Z	C
ADC		×	×	×		×	×	×			∧	•	∧	∧	∧
ADD		×	×	×		×	×	×			∧	•	∧	∧	∧
AND		×	×	×		×	×	×			•	•	∧	∧	•
ASL	×		×			×	×				•	•	∧	∧	∧
ASR	×		×			×	×				•	•	∧	∧	∧
BCC					×						•	•	•	•	•
BCLR									×		•	•	•	•	•
BCS					×						•	•	•	•	•
BEQ					×						•	•	•	•	•
BHCC					×						•	•	•	•	•
BHCS					×						•	•	•	•	•
BHI					×						•	•	•	•	•
(BHS)					×						•	•	•	•	•
BIH					×						•	•	•	•	•
BIL					×						•	•	•	•	•
BIT		×	×	×		×	×	×			•	•	∧	∧	•
(BLO)					×						•	•	•	•	•
BLS					×						•	•	•	•	•
BMC					×						•	•	•	•	•
BMI					×						•	•	•	•	•
BMS					×						•	•	•	•	•
BNE					×						•	•	•	•	•
BPL					×						•	•	•	•	•
BRA					×						•	•	•	•	•

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

(to be continued)

Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						●	●	●	●	●
BRCLR										×	●	●	●	●	^
BRSET										×	●	●	●	●	^
BSET									×		●	●	●	●	●
BSR					×						●	●	●	●	●
CLC	×										●	●	●	●	0
CLI	×										●	0	●	●	●
CLR	×		×			×	×				●	●	0	1	●
CMP		×	×	×		×	×	×			●	●	^	^	^
COM	×		×			×	×				●	●	^	^	1
CPX		×	×	×		×	×	×			●	●	^	^	^
DAA	×										●	●	^	^	^
DEC	×		×			×	×				●	●	^	^	●
EOR		×	×	×		×	×	×			●	●	^	^	●
INC	×		×			×	×				●	●	^	^	●
JMP			×	×		×	×	×			●	●	●	●	●
JSR			×	×		×	×	×			●	●	●	●	●
LDA		×	×	×		×	×	×			●	●	^	^	●
LDX		×	×	×		×	×	×			●	●	^	^	●
LSL	×		×			×	×				●	●	^	^	^
LSR	×		×			×	×				●	●	0	^	^
NEG	×		×			×	×				●	●	^	^	^
NOP	×										●	●	●	●	●
ORA		×	×	×		×	×	×			●	●	^	^	●
ROL	×		×			×	×				●	●	^	^	^
ROR	×		×			×	×				●	●	^	^	^
RSP	×										●	●	●	●	●
RTI	×										?	?	?	?	?
RTS	×										●	●	●	●	●
SBC		×	×	×		×	×	×			●	●	^	^	^
SEC	×										●	●	●	●	1
SEI	×										●	1	●	●	●
STA			×	×		×	×	×			●	●	^	^	●
STOP	×										●	●	●	●	●
STX			×	×		×	×	×			●	●	^	^	●
SUB		×	×	×		×	×	×			●	●	^	^	^
SWI	×										●	1	●	●	●
TAX	×										●	●	●	●	●
TST	×		×			×	×				●	●	^	^	●
TXA	×										●	●	●	●	●
WAIT	×										●	●	●	●	●

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry Borrow
I	Interrupt Mask	/	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	●	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	← HIGH
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRSET0	BSET0	BRA	NEG					RTI*		SUB						0
1	BRCLR0	BCLR0	BRN						RTS*		CMP						1
2	BRSET1	BSET1	BHI								SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*		CPX						3
4	BRSET2	BSET2	BCC	LSR							AND						4
5	BRCLR2	BCLR2	BCS								BIT						5
6	BRSET3	BSET3	BNE	ROR							LDA						6
7	BRCLR3	BCLR3	BEQ	ASR						TAX*		STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL						CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL						SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC						CLI*	ORA						A
B	BRCLR5	BCLR5	BMI							SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC						RSP*		JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)		DAA*	NOP	BSR*	JSR(+2)		JSR(+1)	JSR(+2)	D
E	BRSET7	BSET7	BIL							STOP*	—	LDX					E
F	BRCLR7	BCLR7	BIH	CLR						WAIT*	TXA*		STX			STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5		1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

LOW

(NOTES) 1. "—" is an undefined operation code.

2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD63P05Y0:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

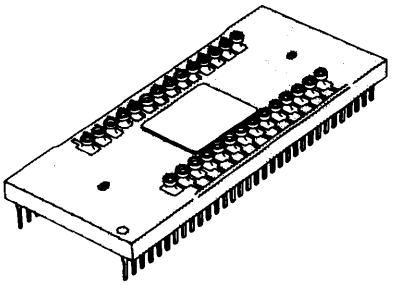
STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

HD63P05Y1, HD63PA05Y1, HD63PB05Y1
CMOS MCU (Microcomputer Unit)

—PRELIMINARY—

The HD63P05Y1 is a CMOS 8-bit single-chip microcomputer unit which has a 4k-byte or 8k-byte EPROM on the package. It is compatible with the HD6305Y1 except for ROM which is not included in the HD63P05Y1. It can be used not only for debugging and evaluating the internal program of HD6305X1 or HD6305Y1, but also for small-sized production preceding mask ROM.

HD63P05Y1, HD63PA05Y1,
HD63PB05Y1



(DC-64SP)

■ FEATURES

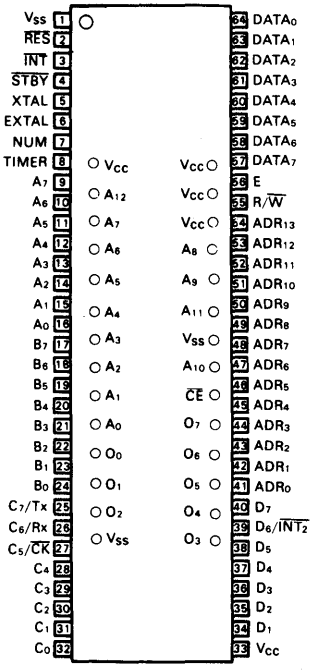
- Pin compatible with HD6305X1 and HD6305Y1
- 256-byte of RAM
- A total of 31 terminals, including 24 I/O's, 7 inputs
- Two timers
  - 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes – Wait, Stop and Standby Mode
- Minimum instruction cycle time
  HD63P05Y1 . . . . . 1 μs (f = 1 MHz)
  HD63PA05Y1 . . . . 0.67 μs (f = 1.5 MHz)
  HD63PB05Y1 . . . . . 0.5 μs (f = 2 MHz)
- Similar to HD6800 instruction set
- Bit manipulation
- Bit test and branch
- Versatile interrupt handling
- Full set of conditional branches
- New instructions – STOP, WAIT, DAA
- Applicable to 4k or 8k bytes of EPROM
  4k bytes; HN482732A
  8k bytes; HN482764, HN27C64

■ TYPE OF PRODUCTS

Type No.	Bus Timing	Applied EPROM
HD63P05Y1	1 MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PA05Y1	1.5 MHz	HN482732A-30, HN482764-3, HN27C64-30
HD63PB05Y1	2 MHz	HN482732A-25, HN482764, HN27C64-25

(Note) EPROM is not attached to the MCU.

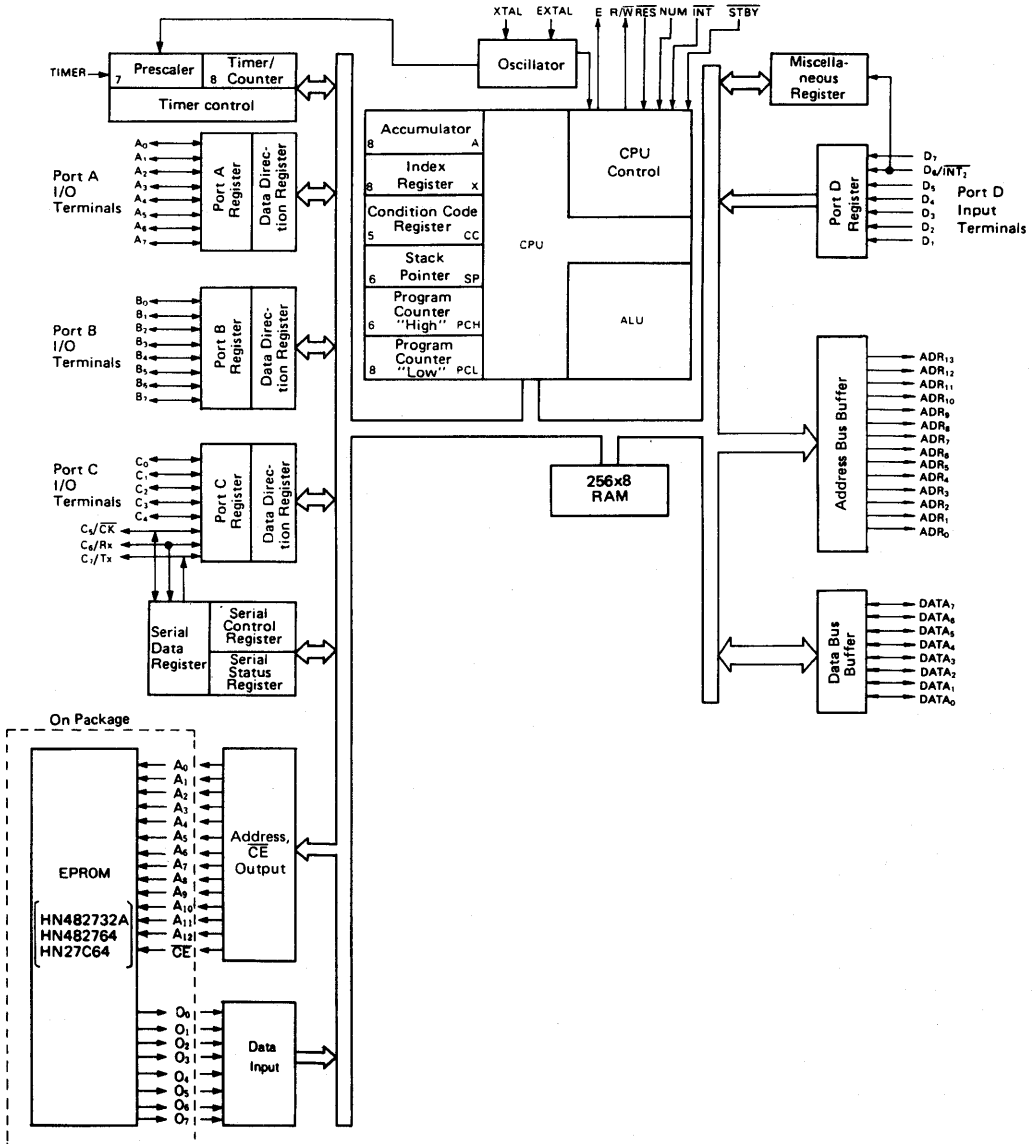
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply voltage	V_{CC}	−0.3 ~ +7.0	V
Input voltage	V_{in}	−0.3 ~ V_{CC} + 0.3	V
Operating temperature	T_{opr}	0 ~ +70	°C
Storage temperature	T_{stg}	−55 ~ +150	°C

[NOTE] These products have a protection circuit in their input terminals against high electrostatic voltage or high electric fields. Notwithstanding, be careful not to apply any voltage higher than the absolute maximum rating to these high input impedance circuits. To assure normal operation, we recommended V_{in}, V_{out}; $V_{SS} \leqq (V_{in}$ or $V_{out}) \leqq V_{CC}$

■ ELECTRICAL CHARACTERISTICS

• **DC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND, T_a = 0 ~ +70°C, unless otherwise specified.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Input voltage "High"	$\overline{RES}$, $\overline{STBY}$	V_{IH}		V_{CC} − 0.5	−	V_{CC} + 0.3	V
	EXTAL			V_{CC} x 0.7	−	V_{CC} + 0.3	V
	Others			2.0	−	V_{CC} + 0.3	V
Input voltage "Low"	All Input	V_{IL}		−0.3	−	0.8	V
Current *** dissipation	Operating	I_{CC}	f = 1MHz*	−	5	10	mA
	Wait			−	2	5	mA
	Stop			−	2	10	μA
	Standby			−	2	10	μA
Input leakage current	$\overline{TIMER}$, $\overline{INT}$, $D_1 \sim D_7$, $\overline{STBY}$	$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} − 0.5V	−	−	1	μA
Three-state current	$A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$, $ADR_0 \sim ADR_{13}$,** $DATA_0 \sim DATA_7$, E**, R/$\overline{W}$**	$\|I_{TSI}\|$		−	−	1	μA
Input capacity	All terminals	C_{in}	f = 1MHz, V_{in} = 0V	−	−	15	pF

* The value at f = xMHz can be calculated by the following equation: I_{CC} (f= xMHz) = I_{CC} (f= 1MHz) multiplied by x
** At standby mode
*** All output and $\overline{RES}$ terminals are open ($V_{IH\ min}$ = V_{CC}-1.0V, $V_{IL\ max}$ = 0.8V), and I_{CC} of EPROM is not included.

● **AC Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND, T_a = 0 ~ +70°C, unless otherwise specified.)**

Item	Symbol	Test Condition	HD63P05Y1			HD63PA05Y1			HD63PB05Y1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Cycle Time	t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time	t_{Er}		–	–	20	–	–	20	–	–	20	ns
Enable Fall Time	t_{Ef}		–	–	20	–	–	20	–	–	20	ns
Enable Pulse Width ("High" Level)	PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width ("Low" Level)	PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address Delay Time	t_{AD}		–	–	250	–	–	190	–	–	180	ns
Address Hold Time	t_{AH}		40	–	–	30	–	–	20	–	–	ns
Data Delay Time	t_{DW}		–	–	250	–	–	160	–	–	120	ns
Data Hold Time (Write)	t_{HW}		40	–	–	30	–	–	20	–	–	ns
Data Set-up Time (Read)	t_{DSR}		80	–	–	60	–	–	50	–	–	ns
Data Hold Time (Read)	t_{HR}		0	–	–	0	–	–	0	–	–	ns

● **Port Electrical Characteristics (V_{CC} = 5.0V ± 10%, V_{SS} = GND, T_a = 0 ~ +70°C, unless otherwise specified.)**

Item		Symbol	Test Condition	min	typ	max	Unit
Output voltage "High"	Ports A, B, C	V_{OH}	I_{OH} = -200μA	2.4	–	–	V
			I_{OH} = -10μA	V_{CC} - 0.7	–	–	V
Output voltage "Low"		V_{OL}	I_{OL} = 1.6mA	–	–	0.55	V
Input voltage "High"	Ports A, B, C, D	V_{IH}		2.0	–	V_{CC} + 0.3	V
Input voltage "Low"		V_{IL}		- 0.3	–	0.8	V
Input leakage current		$\|I_{IL}\|$	V_{in} = 0.5 ~ V_{CC} - 0.5V	–	–	1	μA

● **Port Timing (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD63P05Y1			HD63PA05Y1			HD63PB05Y1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Port Data Set-up Time (Port A, B, C, D)	t_{PDS}	Fig. 2	200	–	–	200	–	–	200	–	–	ns
Port Data Hold Time (Port A, B, C, D)	t_{PDH}		200	–	–	200	–	–	200	–	–	ns
Port Data Delay Time (Port A, B, C)	t_{PDW}	Fig. 3	–	–	300	–	–	300	–	–	300	ns

- **Control Signal Timing (V_{CC} = 5.0V±10%, V_{SS} = GND, Ta = 0 ~ +70°C, unless otherwise noted.)**

Item	Symbol	Test Condition	HD63P05Y1			HD63PA05Y1			HD63PB05Y1			Unit
			min	typ	max	min	typ	max	min	typ	max	
$\overline{INT}$ Pulse Width	t_{IWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{INT_2}$ Pulse Width	t_{IWL2}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
$\overline{RES}$ Pulse Width	t_{RWL}		5	–	–	5	–	–	5	–	–	t_{cyc}
Control Set-up Time	t_{CS}	Fig. 5	250	–	–	250	–	–	250	–	–	ns
Timer Pulse Width	t_{TWL}		t_{cyc} +250	–	–	t_{cyc} +200	–	–	t_{cyc} +200	–	–	ns
Oscillation Start Time (Crystal)	t_{OSC}	Fig.5, Fig.20*	–	–	20	–	–	20	–	–	20	ms
Reset Delay Time	t_{RHL}	Fig. 19	80	–	–	80	–	–	80	–	–	ms

* C_L = 22pF ±20%, R_s = 60Ω max.

- **SCI Timing (V_{CC} = 5.0V±10%, V_{SS} = GND, T_a = 0 ~ +70°C, unless otherwise specified.)**

Item	Symbol	Test Condition	HD63P05Y1			HD63PA05Y1			HD63PB05Y1			Unit
			min	typ	max	min	typ	max	min	typ	max	
Clock cycle	t_{Scyc}	Fig. 6, Fig. 7	1	–	32768	0.67	–	21845	0.5	–	16384	μs
Data output delay time	t_{TXD}		–	–	250	–	–	250	–	–	250	ns
Data set-up time	t_{SRX}		200	–	–	200	–	–	200	–	–	ns
Data hold time	t_{HRX}		100	–	–	100	–	–	100	–	–	ns

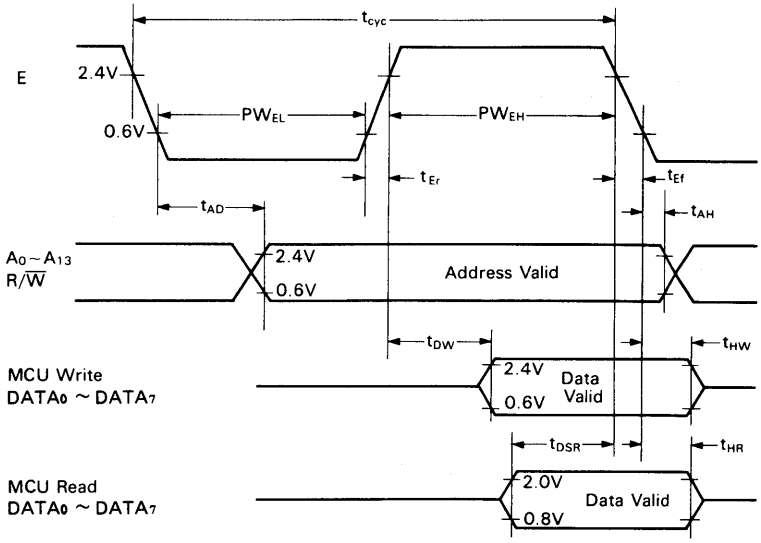


Figure 1 Bus Timing

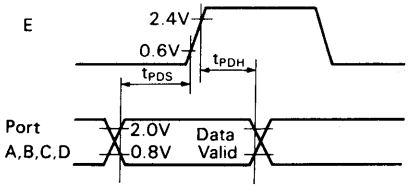


Figure 2 Port Data Set-up and Hold Times (MCU Read)

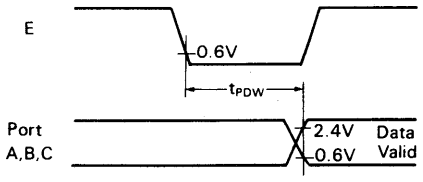


Figure 3 Port Data Delay Time (MCU Write)

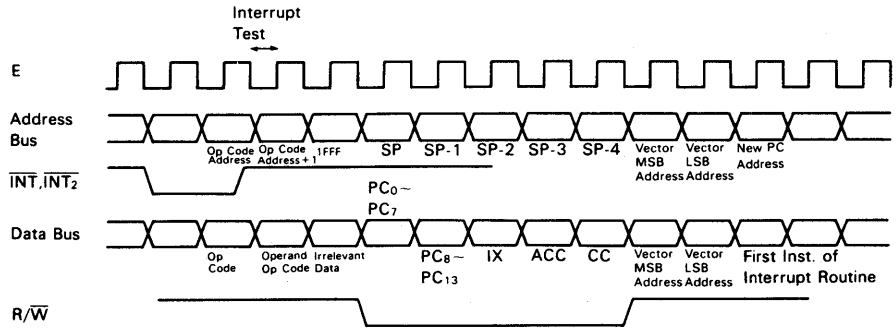


Figure 4 Interrupt Sequence

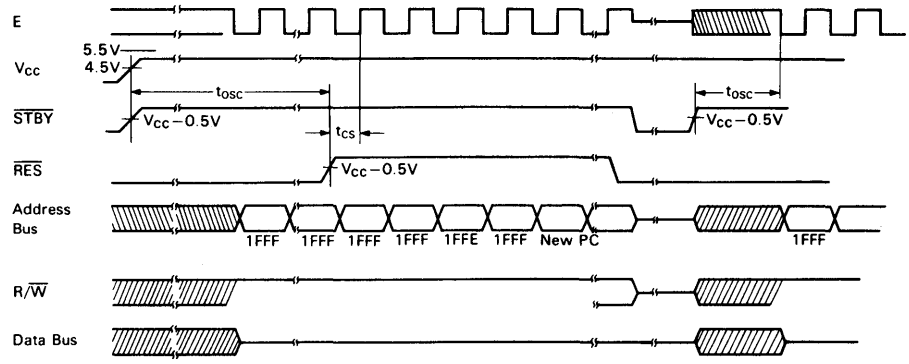


Figure5 Reset Timing

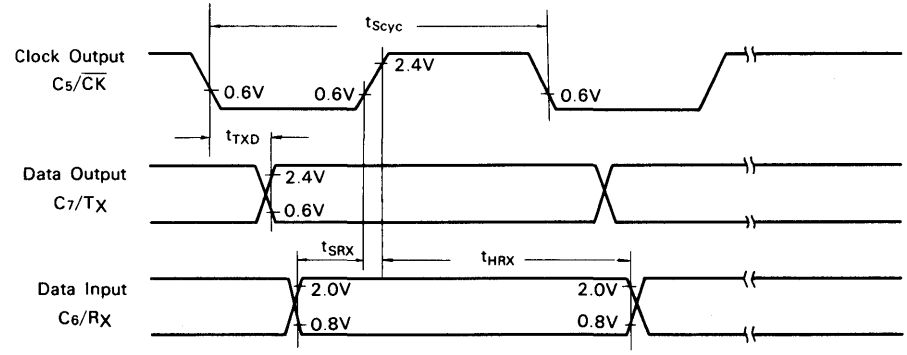


Figure6 SCI Timing (Internal Clock)

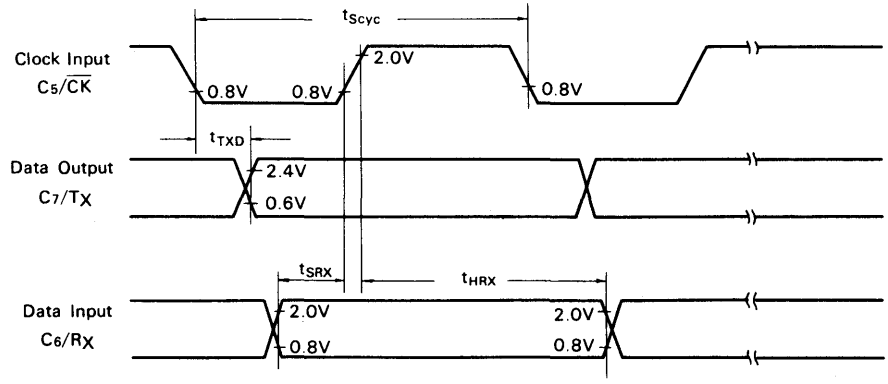
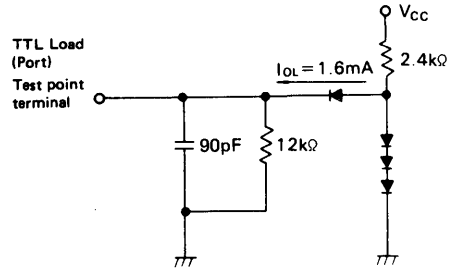


Figure7 SCI Timing(External Clock)



[NOTES] 1. The load capacitance includes stray capacitance caused by the probe, etc.
2. All diodes are 1S2074 Ⓗ

Figure 8 Test Load

■DESCRIPTION ON PIN FUNCTIONS

Here is the description of HD63P05Y1 MCU input and output signals.

●V_{CC}, V_{SS}

Power is supplied to the MCU using these two pins. When the operating voltage of the EPROM is 5.0V ± 5%, change V_{CC} according to that of EPROM.

●$\overline{INT}$, $\overline{INT_2}$

Used for requesting an external interrupt to the MCU. For details, see "INTERRUPT". The $\overline{INT_2}$ is used as the port D_6 pin.

●XTAL, EXTAL

Are input pins to the internal clock circuit. A crystal oscillator (AT cut, 2.0 to 8.0 MHz) or ceramic oscillator is connected to these pins. For instance, in order to obtain the system clock 1 MHz, a 4 MHz resonant fundamental crystal is useful because the divide-by-4 circuitry is included. EXTAL accepts an external clock input of duty 50% (±10%) to drive, then the internal clock is a quarter the frequency of the external clock. External drive frequency will be 4 or less times the maximum internal clock. For external driving, no XTAL should be connected. Refer to "INTERNAL OSCILLATOR" for using these input pins.

●TIMER

Is an external input pin to control the internal Timer. For details, see "TIMER".

●$\overline{RES}$

Is used for resetting MCU. For details, see "RESET".

●NUM

Is not for user application. It must be connected to V_{CC} through 1kΩ resistance.

●Input/Output Pins ($A_0 \sim A_7$, $B_0 \sim B_7$, $C_0 \sim C_7$)

24 pins consist of three 8-bit I/O ports (A, B, C). Each of them is used as input or output pin, through program control of the data direction register. For details, see "I/O PORTS".

●Input Pins ($D_1 \sim D_7$)

Are 7 input-only pins compatible with the TTL and CMOS. D_6 is used as $\overline{INT_2}$. When the D_6 is used as the port, set the $\overline{INT_2}$ interrupt mask bit of the miscellaneous register to "1" to prevent an $\overline{INT_2}$ from accidental interruption.

●Enable (E)

Supplies E clock. Output is a single-phase, TTL compatible and 1/4 crystal oscillation frequency or 1/4 external clock frequency. It can drive one TTL load and a 90pF condenser.

●Read/Write (R/$\overline{W}$)

Is an output pin compatible with the TTL. This indicates to peripheral and memory devices whether MCU is in Read ("High"), or in Write ("Low"). The normal standby state is Read ("High"). Its output can drive one TTL load and a 90pF condenser.

●Data Bus ($DATA_0 \sim DATA_7$)

Are three-state buffers compatible with the TTL. Each of them can drive one TTL load and 90pF.

●Address Bus ($ADR_0 \sim ADR_{13}$)

Are compatible with the TTL and can drive one TTL load and 90pF.

●$\overline{STBY}$

Used for bringing the MCU into the standby mode. With $\overline{STBY}$ at "Low" level, the oscillation stops and internal situation is reset. For details, see "STANDBY MODE". The following are I/O pins for serial communication interface (SCI), and used as ports C_5, C_6, and C_7. For details, see "SERIAL COMMUNICATION INTERFACE".

●$\overline{CK}$ (C_5)

Used to input or output clocks when receiving or transmitting serial data.

●Rx (C_6)

Used to receive serial data.

●Tx (C_7)

Used to transmit serial data.

■MEMORY MAP

The memory map of the HD63P05Y1 MCU is shown in Fig. 9. During interrupt, the contents of the registers are saved in the stack as shown in Fig. 10. The saving begins with the lower byte (PCL) of the program counter. Then the stack pointer value is decremented, and the higher byte (PCH) of the program counter, index register (X), accumulator (A) and condition code register (CC) are stacked in this order. In subroutine calls, only the contents of the program counter (PCH and PCL) are stacked.

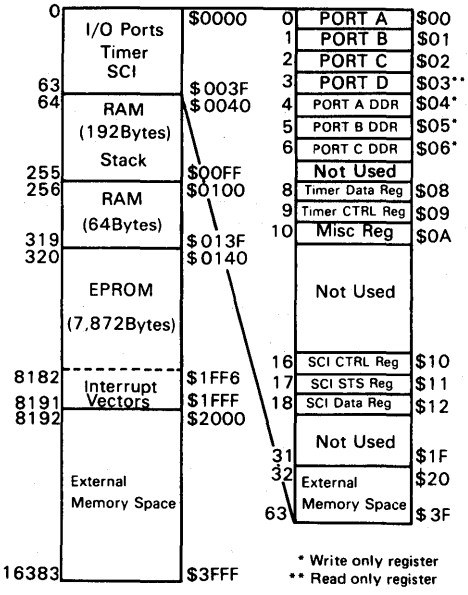


Figure 9 Memory Map of HD63P05Y1 MCU

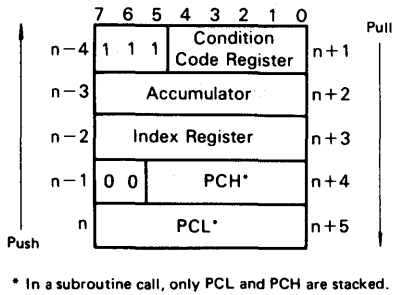


Figure 10 Sequence of Interrupt Stacking

■ REGISTERS

There are five registers which the programmers can handle.

● Accumulator (A)

The accumulator is a general purpose 8-bit register which holds operands, the results of arithmetic operations or data processing.

● Index Register (X)

The index register is an 8-bit register used for the index addressing mode. It contains an 8-bit value to be added to an instruction value to create an effective address. The index register can also be used for data manipulations using the read-modify-write instruction. The index register may also be used as a temporary storage area.

● Program Counter (PC)

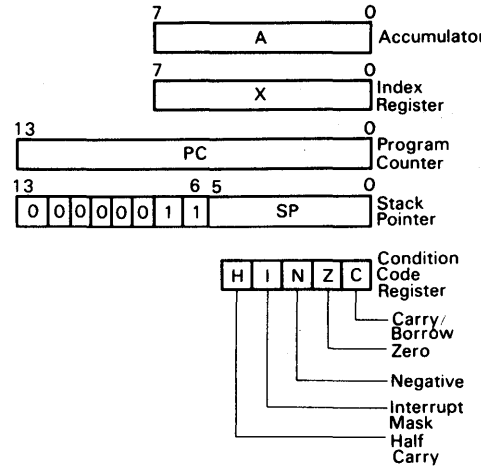


Figure 11 Programming Model

The program counter is a 14-bit register which contains the address of the next instruction to be executed.

● Stack Pointer (SP)

The stack pointer is a 14-bit register which indicates the address of the next free location in the stack. Initially, the stack pointer is set to $00FF. It is decremented as data is pushed in, and incremented as it is pulled out. The upper 8 bits of the stack pointer are fixed to 00000011.

During an MCU reset or when the reset stack pointer (RSP) instruction is executed, the pointer is set to the location $00FF. A subroutine or interrupt may be nested down to location $00C1 which allows programmers to use up to 31 levels of subroutine call or 12 levels of interrupt response.

● Condition Code Register (CC)

The condition code register is a 5-bit register. Each bit indicates the result of the executed instruction. These bits can be individually tested by conditional branch instructions. The CC bits are as follows.

Half Carry (H): Used to indicate a carry occurring between bits 3 and 4 during an arithmetic operation (ADD, ADC).

Interrupt (I): Setting this bit causes all interrupts to be masked except for software ones. If an interrupt occurs while the bit I is set, the interrupt is latched, and processed as soon as the interrupt mask bit (I) is reset. (Exactly, the interrupt enters the processing routine after the instruction next to the CLI is executed.)

Negative (N): Used to indicate that the result of the latest arithmetic operation, logical operation or data processing is negative (Bit 7 is logical "1").

Zero (Z): Used to indicate that the result of the latest arithmetic operation, logical operation or data processing is zero.

Carry/Borrow (C): Shows a carry or borrow occurring in the latest arithmetic operation. This bit is also affected by the Bit Test and Branch, Shift and

Rotate instructions.

■INTERRUPT

There are six different types of interrupt: external interrupt ($\overline{INT}$, $\overline{INT_2}$), internal timer interrupts (TIMER, TIMER 2), serial interrupt (SCI) and interrupt by an instruction (SWI).

Of these six interrupts, the $\overline{INT_2}$ and TIMER, and SCI and TIMER 2 respectively generate the same vector address. When an interrupt occurs, the program in execution stops and CPU state at the interrupt is saved onto the stack. In addition, the interrupt causes the interrupt mask bit (I) in the condition code register to be set and obtains the start address of the interrupt routine from an assigned interrupt vector address before the interrupt routine starts from the state address. The system exits from the interrupt routine by RTI instruction. When the RTI instruction is executed, the CPU state before the interrupt (saved in the stack) is pulled and the CPU starts the program again from the next step to the interrupted one. Table 1. lists the priority of interrupts and their vector addresses.

Table 1 Priority of Interrupts

Interrupt	Priority	Vector Address
$\overline{RES}$	1	$1FFE, $1FFF
SWI	2	$1FFC, $1FFD
$\overline{INT}$	3	$1FFA, $1FFB
TIMER/$\overline{INT_2}$	4	$1FF8, $1FF9
SCI/TIMER$_2$	5	$1FF6, $1FF7

A flow chart of the interrupt is shown in Fig. 12. Also a block diagram of the interrupt request source is shown in Fig. 13. In the block diagram, both the external interrupts $\overline{INT}$ and $\overline{INT_2}$ are edge trigger inputs. At the falling edge of the input, an interrupt request is generated and latched.

The $\overline{INT}$ interrupt request is automatically cleared if a program jumps to the $\overline{INT}$ routine. In the case of $\overline{INT_2}$, the

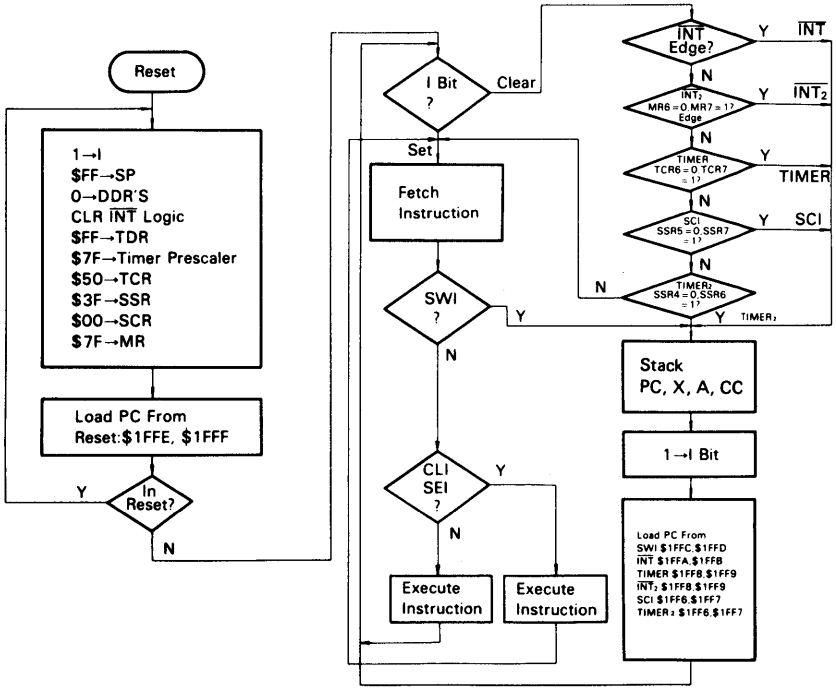


Figure 12 Interrupt Flowchart

interrupt request is cleared when "0" is written in bit 7 of the miscellaneous register. For external interrupts($\overline{INT}$,$\overline{INT_2}$), internal timer interrupts (TIMER, TIMER2) and serial interrupt (SCI), these interrupt requests are held, but not operated, while bit I of the condition code register is set. Immediately after the bit I is cleared, the corresponding interrupt is activated.

The $\overline{INT_2}$ interrupt can be masked by setting bit 6 of the miscellaneous register; the TIMER interrupt by bit 6 of the timer control register, the SCI interrupt by bit 5 of the serial status register and the TIMER2 interrupt by bit 4 of the serial status register.

The state of the $\overline{INT}$ pin is tested by BIL or BIH instructions. The $\overline{INT}$ falling edge detector circuit and its latch circuit are independent of tests by these instructions. The state of $\overline{INT_2}$ pin is also independent.

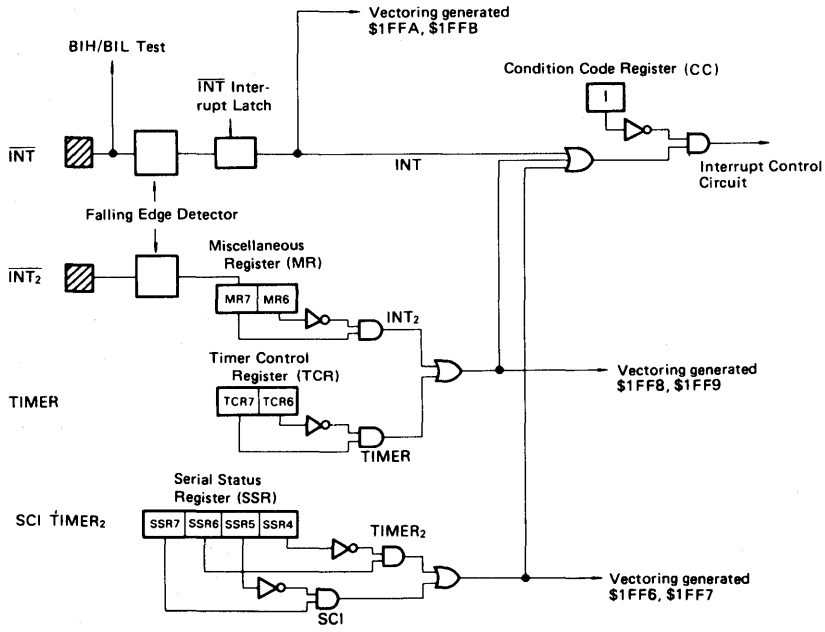
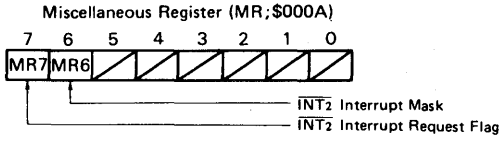


Figure 13 Interrupt Request Generation Circuitry

- **Miscellaneous Register (MR: $000A)**

The interrupt vector address for external interrupt $\overline{INT_2}$ is the same as that for the TIMER interrupt, as shown in Table 1. For this reason, a special register called a miscellaneous register (MR: $000A) is available for $\overline{INT_2}$ interrupt control. Bit 7 of the miscellaneous register is of $\overline{INT_2}$ interrupt request flag. When the falling edge is detected at the $\overline{INT_2}$ pin, "1" is set in bit 7. The software in the interrupt routine (vector address: $1FF8, $1FF9) checks to see if it is $\overline{INT_2}$ interrupt. Bit 7 is reset by software. Bit 6 is the $\overline{INT_2}$ interrupt mask bit. If the bit is set to "1", the $\overline{INT_2}$ interrupt is disabled.



Both "READ" and "WRITE" are possible with bit 7, but "1" can not be written to in this bit by software. Therefore, interrupt requests by software are not possible. By resetting, bit 7 is cleared and bit 6 is entered "1".

- **TIMER**

The MCU timer block diagram is shown in Fig. 14. The 8-bit counter is loaded under program control and is decremented by the clock input. When the timer data register (TDR) reaches 0, the timer interrupt request bit (bit 7) in the timer control register is set. The MCU responds to this interrupt by saving the present CPU state in the stack, fetching the timer interrupt routine address from address $1FF8 and $1FF9. The timer interrupt can be masked by setting the timer interrupt mask bit (bit 6) in the timer control register. The mask bit (I) in the condition code register can also disable the timer interrupt. The source clock for the timer can be either an external signal from the timer input pin or the internal E signal (oscillator clock divided by 4). If the E signal is selected as the source, the clock input can be gated by the input to the timer input pin.

When the timer counter reaches "0", it starts counting down from $FF. The count can be monitored at any time by reading the timer data register. This function allows knowledge of the length of time after a timer interrupt with a program, without destroying the contents of the counter.

When the MCU is reset, both the prescaler and counter return to the initial state of logical "1". At the same time, the timer interrupt request bit (bit 7) is cleared and the timer interrupt mask bit (bit 6) is set. Write "0" in the timer interrupt request bit (bit 7) to clear it.

TCR7	Timer interrupt request
0	Absent
1	Present

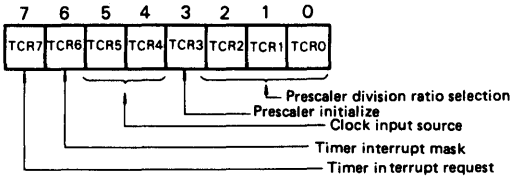
TCR6	Timer interrupt mask
0	Enabled
1	Disabled

• Timer Control Register (TCR; $0009)

Selection of a clock source, selection of a prescaler frequency division ratio, and a timer interrupt can be controlled by the timer control register (TCR; $0009).

For the selection of a clock source, any one of the four modes (see Table 2) can be selected by bits 5 and 4 of the timer control register (TCR).

Timer Control Register (TCR; $0009)



After resetting, the TCR is initialized to "E under timer terminal control" (bit 5 = 0, bit 4 = 1). If the timer terminal is "1", the counter starts counting down with "$FF" immediately after the reset.

Table 2 Clock Source Selection

TCR		Clock input source
Bit 5	Bit 4	
0	0	Internal clock E
0	1	E under timer terminal control
1	0	No clock input (counting stopped)
1	1	Event input from timer terminal

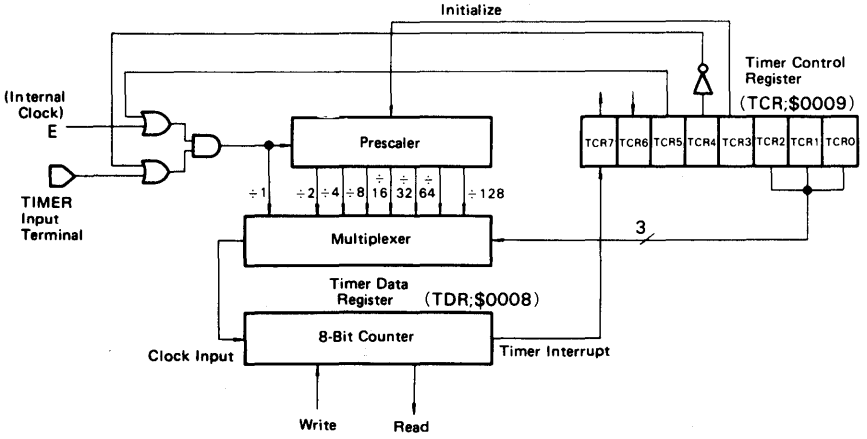


Figure 14 Timer Block Diagram

Table 3 Prescaler Division Ratio Selection

TCR			Prescaler division ratio
Bit 2	Bit 1	Bit 0	
0	0	0	÷1
0	0	1	÷2
0	1	0	÷4
0	1	1	÷8
1	0	0	÷16
1	0	1	÷32
1	1	0	÷64
1	1	1	÷128

The prescaler is initialized by writing "1" in bit 3. The bit is always "0", when "READ". A prescaler division ratio is selected by a combination of the three bits (bits 0, 1 and 2) of the timer control register (See Table 3). There are eight division ratios; ÷1, ÷2, ÷4, ÷8, ÷16, ÷32, ÷64 and ÷128.

After resetting, the TCR returns to the ÷1 mode. The timer interrupt is enabled when the timer interrupt mask bit is "0", and disabled when the bit is "1". When a timer interrupt occurs, "1" is set in the timer interrupt request bit. The bit is cleared by writing "0" into it.

■SERIAL COMMUNICATION INTERFACE (SCI)

Used for 8-bit data communication. Transfer rate ranges from 1μs to about 32 ms (when oscillated at 4 MHz), and there are sixteen selections.

The SCI consists of three registers, one octal counter and one prescaler. (See Fig. 15) The SCI communicates with the CPU through the data bus, and with peripherals through bits 5, 6 and 7 of port C. Operations of the registers and data transfer are described below.

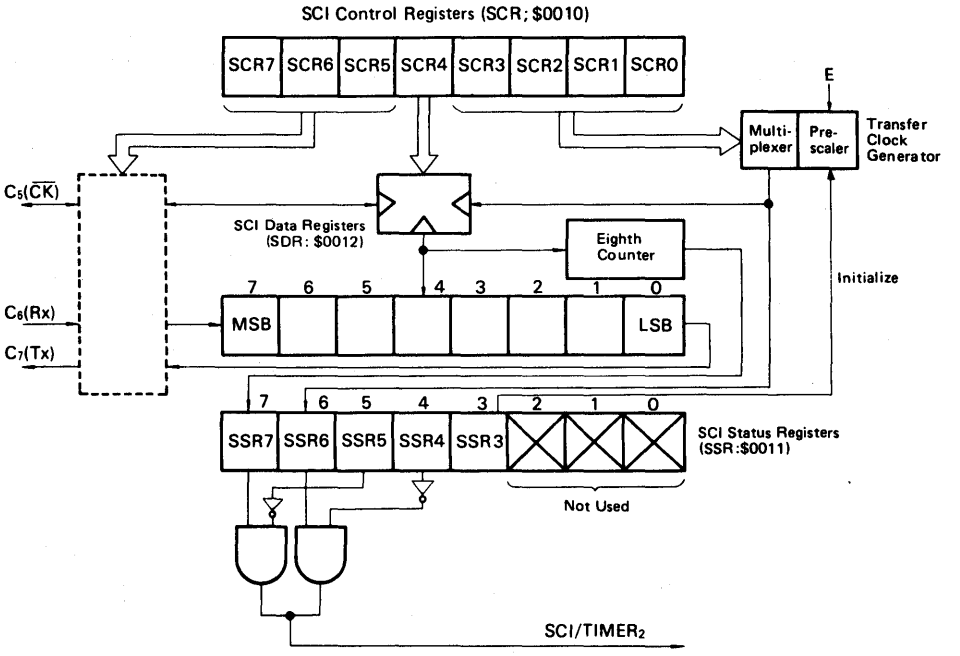


Figure 15 SCI Block Diagram

•SCI Control Register (SCR; $0010)

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0

SCR7	C_7 terminal
0	Used as I/O terminal (by DDR).
1	Serial data output (DDR output)

SCR6	C_6 terminal
0	Used as I/O terminal (by DDR).
1	Serial data input (DDR input)

SCR5	SCR4	Clock source	C_5 terminal
0	0	–	Used as I/O terminal (by DDR).
0	1	–	
1	0	Internal	Clock output (DDR output)
1	1	External	Clock input (DDR input)

Bit 7 (SCR7)

When this bit is set, the DDR corresponding to the C_7 becomes "1" and this terminal serves for output of SCI data. After resetting the bit is cleared to "0".

Bit 6 (SCR6)

When this bit is set, the DDR corresponding to the C_6 becomes "0" and this terminal serves for input of SCI data. After resetting the bit is cleared to "0".

Bits 5 and 4 (SCR5, SCR4)

These bits are used to select a clock source. After resetting the bits are cleared to "0".

Bits 3 ~ 0 (SCR3 ~ SCR0)

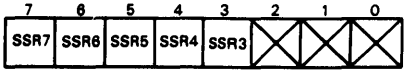
These bits are used to select a transfer clock rate. After resetting the bits are cleared to "0".

SCR3	SCR2	SCR1	SCR0	Transfer clock rate	
				4.00 MHz	4.194 MHz
0	0	0	0	1 μs	0.95 μs
0	0	0	1	2 μs	1.91 μs
0	0	1	0	4 μs	3.82 μs
0	0	1	1	8 μs	7.64 μs
≀	≀	≀	≀	≀	≀
1	1	1	1	32768 μs	1/32 s

•SCI Data Register (SDR; $0012)

A serial-parallel conversion register that is used for transfer of data.

•SCI Status Register (SSR; $0011)



Bit 7 (SSR7)

Bit 7 is the SCI interrupt request bit which is set on completion of transmitting or receiving 8-bit data. It is cleared when reset or data is written to or read from the SCI data register with the SCR5="1". The bit can be cleared by writing "0" into it.

Bit 6 (SSR6)

Bit 6 is the $TIMER_2$ interrupt request bit. $TIMER_2$ is commonly used with the serial clock generator, and SSR6 is set each time the internal transfer clock falls. When resetting , the bit is cleared. It can also be cleared by writing "0" into it. (For details, see $TIMER_2$).

Bit 5 (SSR5)

Bit 5 is the SCI interrupt mask bit which can be set or cleared by software. When it is "1", the SCI interrupt (SSR7) is masked. When resetting, it is set to "1".

Bit 4 (SSR4)

Bit 4 is the $TIMER_2$ interrupt mask bit which can be set or cleared by software. When the bit is "1", the $TIMER_2$ interrupt (SSR6) is masked. When resetting, it is set to "1".

Bit 3 (SSR3)

When "1" is written into this bit, the prescaler of the transfer clock generator is initialized. When "READ", the bit is always "0".

Bits 2 ~ 0

Not used.

SSR7	SCI interrupt request
0	Absent
1	Present

SSR6	$TIMER_2$ interrupt request
0	Absent
1	Present

SSR5	SCI interrupt mask
0	Enabled
1	Disabled

SSR4	$TIMER_2$ interrupt mask
0	Enabled
1	Disabled

• Data Transmission

By writing the desired control bits into the SCI control registers, a transfer rate and a transfer clock source are determined and bits 7 and 5 of port C are set at the serial data output terminal and the serial clock terminal, respectively. The transmit data should be stored from the accumulator or index register into the SCI data register. The data written in the SCI data register is output from the C_7/Tx terminal, starting with the LSB, synchronously with the falling edge of the serial clock (See Fig. 16). When 8 bits of data have been transmitted, the interrupt request bit is set in bit 7 of the SCI status register with the rising edge of the last serial clock. This request can be masked by setting bit 5 of the SCI status register. Once the data has been sent, the 8th bit data (MSB) stays at the C_7/Tx terminal. If an external clock source has been selected, the transfer rate determined by bits 0 to 3 of the SCI control register is ignored, and the $C_5/\overline{CK}$ terminal is set as input. If the internal clock has been selected, the $C_5/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 to 3 of the SCI control register.

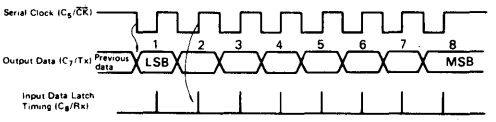


Figure 16 SCI Timing Chart

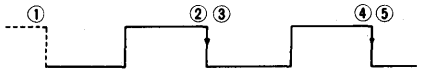
• Data Reception

By writing the desired control bits into the SCI control register, a transfer rate and a transfer clock source are determined and bit 6 and 5 of port C are set at the serial data input terminal and the serial clock terminal, respectively. Then dummy-writing or -reading the SCI data register, the system is ready for receiving data. (This procedure is not needed after reading subsequent received data. It must be taken after reset and after not reading subsequent received data.)

The data from the C_6/Rx terminal is input to the SCI data register synchronously with the rising edge of the serial clock (see Fig. 16). When 8 bits of data have been received, the interrupt request bit is set in bit 7 of the SCI status register. This request can be masked by setting bit 5 of the SCI status register. If an external clock source have been selected, the transfer rate determined by bits 0 ~ 3 of the SCI control register is ignored, and the data is received synchronously with the clock from the $C_5/\overline{CK}$ terminal. If the internal clock has been selected, the $C_5/\overline{CK}$ terminal is set as output and clocks are output at the transfer rate selected by bits 0 ~ 3 of the SCI control register.

• $TIMER_2$

The SCI transfer clock generator can be used as a timer. The clock selected by bits 3 to 0 of the SCI control register (4 μs to approx. 32 ms (when oscillated at 4 MHz)) is input to bit 6 of the SCI status register and the $TIMER_2$ interrupt request bit is set at each falling edge of the clock. Since interrupt requests occur periodically, $TIMER_2$ can be used as a reload counter or clock.



① : Transfer clock generator is reset and mask bit (bit 4 of SCI status register) is cleared.
②, ④ : TIMER2 interrupt request
③, ⑤ : TIMER2 interrupt request bit cleared

TIMER2 is commonly used with the SCI transfer clock generator. If wanting to use TIMER2 independently of the SCI, specify "External" (SCR5 = 1, SCR4 = 1) as the SCI clock source.

If "Internal" is selected as the clock source, reading or writing the SDR causes the prescaler of the transfer clock generator to be initialized.

■I/O PORTS

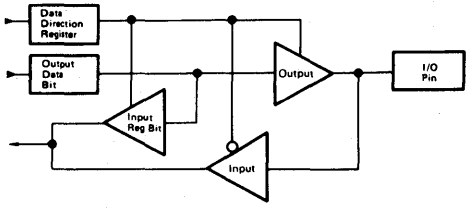
There are 24 input/output terminals (ports A, B, C). Each I/O terminal can be selected for either input or output by the data direction register. Specifically, an I/O port will be input if "0" is written in the data direction register, and output if "1" is written in the data direction register. Port A, B or C reads latched data if it has been programmed as output, even with the output load, the output level fluctuating. (See Fig. 17).

When resetting the data direction register and data register go to "0" and all input/output terminals are used as input.

Seven input-only terminals are available (port D). Writing to these ones is invalid.

All input/output terminals and input terminals are TTL compatible and CMOS compatible in respect of both input and output.

If I/O ports or input ports are not used, they should be connected to V_{SS} via resistors. With none connected to these terminals, there is the possibility of power being consumed despite their not being used.



Bit of data direction register	Bit of output data	Status of output	Input to MCU
1	0	0	0
1	1	1	1
0	X	3-state	Pin

Figure 17 Input/Output Port Diagram

■RESET

The MCU can be reset either by external reset input ($\overline{RES}$) or power-on reset. (See Fig. 18). On power up, the reset input must be held "Low" for at least t_{osc} to assure that the internal oscillator is stabilized. A sufficient delay time can be obtained by connecting a capacitance to the $\overline{RES}$ input as shown in Fig. 19.

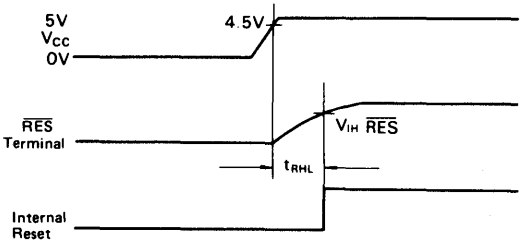


Figure 18 Power On and Reset Timing

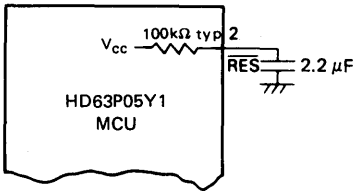


Figure 19 Input Reset Delay Circuit

■INTERNAL OSCILLATOR

The internal oscillator circuit is designed to meet the

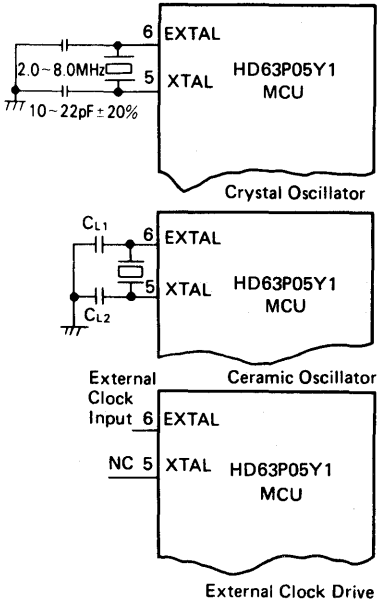


Figure 20 Internal Oscillator Circuit

requirement for minimum external configurations. It can be driven by connecting a crystal (AT cut 2.0 ~ 8.0MHz) or ceramic oscillator between pins 5 and 6 depending on the required oscillation frequency stability.

Three different terminal connections are shown in Fig. 20. Figs. 21 and 22 illustrate the specifications and typical arrangement of the crystal.

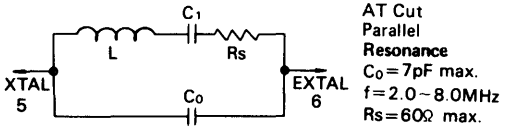
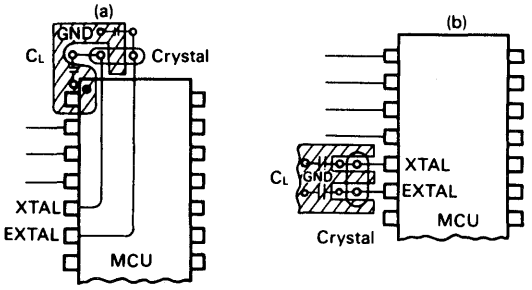


Figure 21 Parameters of Crystal



[NOTE] Use as short wirings as possible for connection of the crystal with the EXTAL and XTAL terminals. Do not allow these wirings to cross others.

Figure 22 Typical Crystal Arrangement

■LOW POWER DISSIPATION MODE

The HD63P05Y1 has three low power dissipation modes: wait, stop and standby.

• Wait Mode

When a WAIT instruction being executed, the MCU enters into the wait mode. In this mode, the oscillator stays active but the internal clock stops. The CPU stops but the peripheral functions – the timer and the serial communication interface – stay active. (NOTE: Once the system has entered the wait mode, the serial communication interface can no longer be retriggered.) In the wait mode, the registers, RAM and I/O terminals hold the condition just before entering the wait mode. Both address (A_0 ~ A_{12}) and chip enable ($\overline{CE}$) for the EPROM are in "1" state.

Release from this mode can be done by interrupt ($\overline{INT}$, TIMER/$\overline{INT_2}$ or SCI/TIMER2), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode. (This will be mentioned later.)

When interrupt is requested to the CPU and accepted, the wait mode is released and the CPU is brought to the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after release from the wait mode the MCU executes the instruction following WAIT. If an interrupt other than the $\overline{INT}$ (i.e., TIMER/$\overline{INT_2}$ or SCI/TIMER2) is masked by the timer control register, miscellaneous register or serial status register, there is no interrupt request to the CPU, so the wait mode cannot be released.

Fig. 23 shows a flowchart of the wait function.

• Stop Mode

When STOP instruction is being executed, the MCU enters the stop mode. In this mode, the oscillator stops and the CPU and peripheral functions become inactive but the RAM, register and I/O terminals hold the condition they had just before entering the stop mode. Both address (A_0 ~ A_{12}) and chip enable ($\overline{CE}$) for the EPROM are in "1" state.

Release from this mode can be done by an external interrupt ($\overline{INT}$ or $\overline{INT_2}$), $\overline{RES}$ or $\overline{STBY}$. The $\overline{RES}$ resets the MCU and the $\overline{STBY}$ brings it into the standby mode.

When an interrupt is requested and accepted by the CPU, the stop mode is released and the CPU is brought in the operation mode and vectors to the interrupt routine. If the interrupt is masked by the I bit of the condition code register, after release from the stop mode, the MCU executes the instruction following STOP. If the $\overline{INT_2}$ interrupt is masked by the miscellaneous register, there is no interrupt request to the MCU, so the stop mode cannot be released.

Fig. 24 shows the flowchart of the stop function. Fig. 25 shows a timing chart of the return to the operation mode from the stop mode.

For releasing from the stop mode by an interrupt, oscillation starts upon input of the interrupt and, after the internal delay time for stabilized oscillation, the CPU becomes active. For restarting by $\overline{RES}$, oscillation starts when the $\overline{RES}$ goes "0" and the CPU restarts when the $\overline{RES}$ goes "1". The duration of $\overline{RES}$="0" must exceed t_{osc} to assure stabilized oscillation.

• Standby Mode

The MCU enters the standby mode when the $\overline{STBY}$ terminal goes "Low". In this mode, all operations stop and the internal condition is reset but the contents of the RAM are held. The I/O terminals turn to high-impedance state. Both address (A_0 ~ A_{12}) and chip enable ($\overline{CE}$) for the EPROM are in "1" state. The standby mode should be released by bringing $\overline{STBY}$ "High". The CPU must be restarted by resetting. The timing of input signals at the $\overline{RES}$ and $\overline{STBY}$ terminals is shown in Fig. 26.

Table 4 lists the status of each parts of the MCU in each low power dissipation modes. Transitions between each mode are shown in Fig. 27.

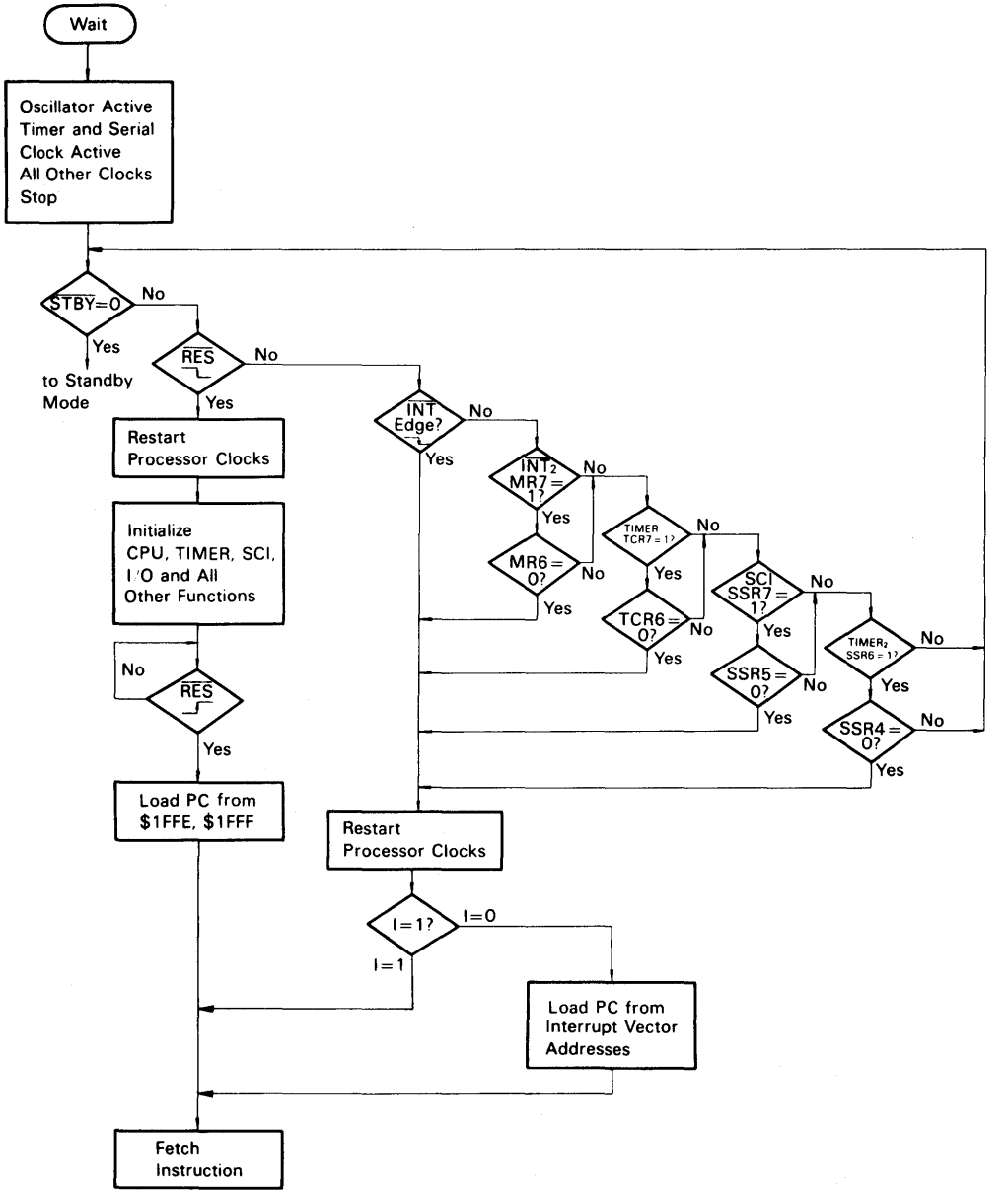


Figure 23 Wait Mode Flow Chart

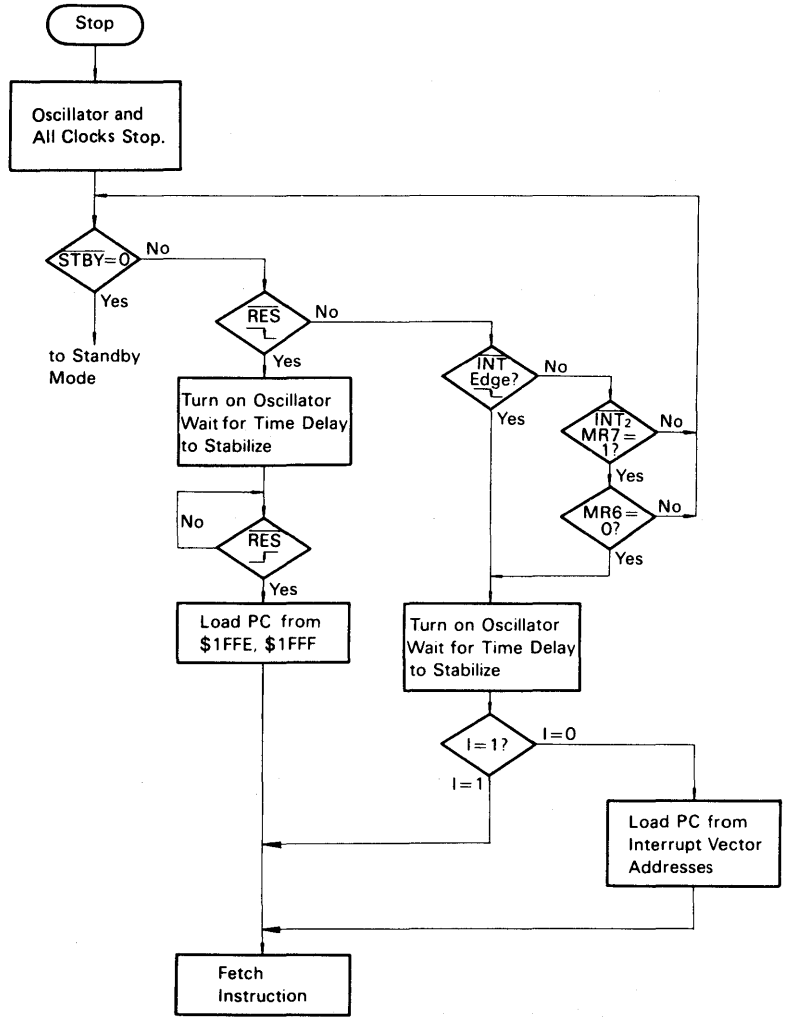


Figure 24 Stop Mode Flow Chart

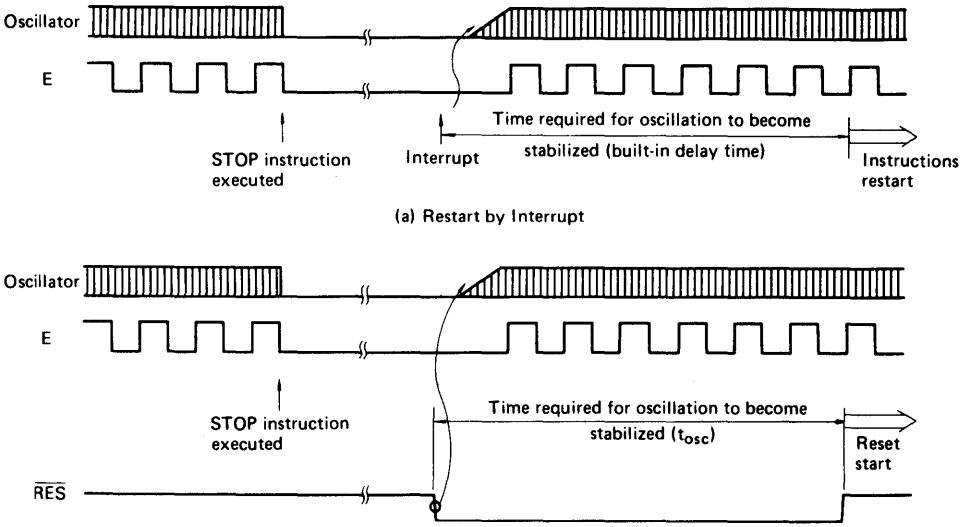


Figure 25 Timing Chart of Releasing from Stop Mode

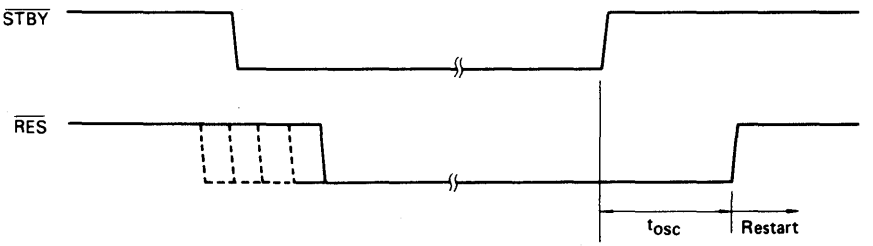


Figure 26 Timing Chart of Releasing from Standby Mode

Table 4 Status of Each Part of MCU in Low Power Dissipation Modes

Mode	Start		Condition						Escape
			Oscillator	CPU	Timer, Serial	Register	RAM	I/O terminal	
WAIT	Software	WAIT instruction	Active	Stop	Active	Hold	Hold	Hold	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$, each interrupt request of TIMER, $TIMER_2$, SCI
STOP		STOP instruction	Stop	Stop	Stop	Hold	Hold	Hold	$\overline{STBY}$, $\overline{RES}$, $\overline{INT}$, $\overline{INT_2}$
Standby	Hardware	$\overline{STBY}$="Low"	Stop	Stop	Stop	Reset	Hold	High impedance	$\overline{STBY}$="High"

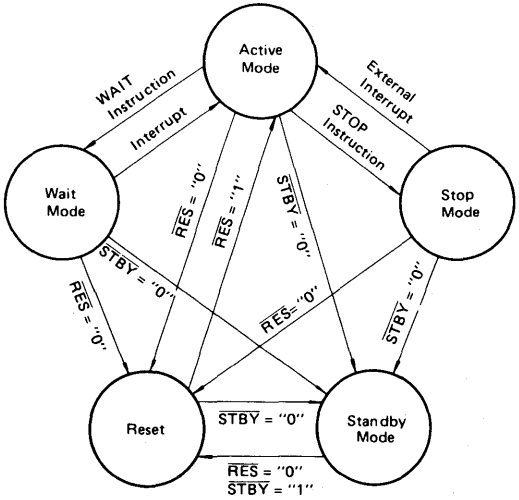


Figure 27 Transitions among Active Mode, Wait Mode, Stop Mode, Standby Mode and Reset

■ PRECAUTION TO THE BOARD DESIGN OF OSCILLATION CIRCUIT

As shown in Fig.28, the cross talk may disturb normal oscillation if signal lines are set near the oscillation circuit. When designing a board, be careful of this. Crystal and C_L must be put near XTAL and EXTAL pins as possible.

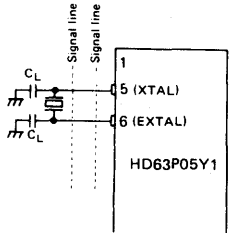


Figure 28 Precaution to the board design of oscillation circuit

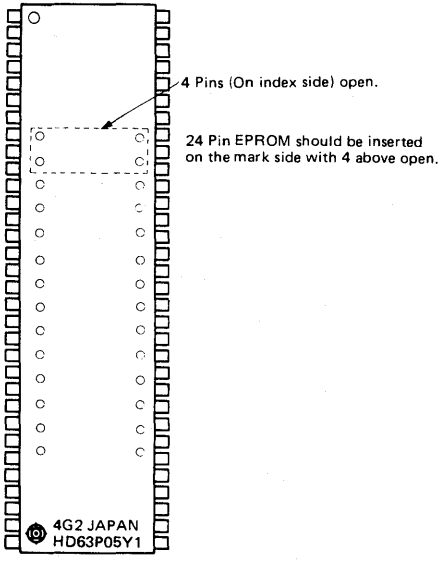
■ PRECAUTION TO USE THE EPROM ON-PACKAGE 8-BIT SINGLE-CHIP MICROCOMPUTER

Please be careful of the following, since this MCU has a special structure with pin socket on the package.

(1) Don't apply high static voltage or surge voltage over MAXIMUM RATINGS to the socket pins as well as the LSI pins. If so, that may cause permanent damage to the device.

(2) When using 32k EPROM (24-pin), insert it leaving the four pins above open.

(3) When inserting this into system products like mask ROM type single chip microcomputer, be careful of the following to give effective contact between the EPROM pins and socket pins.



(a) When soldering the LSI onto a printed circuit board, the recommended condition is
Temperature: lower than 250°C
Time: within 10 sec.

(b) Be careful that detergent or coating does not get into the socket during flux washing or board coating after soldering, because that may cause bad effect on socket contact.

(c) Avoid permanent application of this under conditions

of continuous vibration.

(d) The socket, repeatedly inserted and removed, loses its contactability. It is recommended to use new one when used in production.

■BIT MANIPULATION

The HD63P05Y1 MCU can use a single instruction (BSET or BCLR) to set or clear one bit of the RAM within page 0 or an I/O port (except the write-only registers such as the data direction register). Every bit of memory or I/O within page 0 ($00 ~ $FF) can be tested by the BRSET or BRCLR instruction; depending on the result of the test, the program can branch to required destinations. Since bits in the RAM on page 0, or I/O can be manipulated, the user may use a bit within the RAM on page 0 as a flag or handle a single I/O bit as an independent I/O terminal. Fig. 29 shows an example of bit manipulation and the validity of test instructions. In the example, the program is configured assuming that bit 0 of port A is connected to a zero cross detector circuit and bit 1 of the same port to the trigger of a triac.

The program shown can activate the triac within a time of 10μs from zero-crossing through the use of only 7 bytes on the memory. The on-chip timer provides a required time of delay and pulse width modulation of power is also possible.

```
        :
SELF 1. BRCLR 0, PORT A, SELF 1
        BSET 1, PORT A
        BCLR 1, PORT A
        :
```

Figure 29 Example of Bit Manipulation

■ADDRESSING MODES

Ten different addressing modes are available to the HD63P05Y1 MCU.

• Immediate

See Fig. 30. The immediate addressing mode provides access to a constant which does not vary during execution of the program.

This access requires an instruction length of 2 bytes. The effective address (EA) is PC and the operand is fetched from the byte that follows the operation code.

• Direct

See Fig. 31. In the direct addressing mode, the address of the operand is contained in the 2nd byte of the instruction. The user can gain direct access to memory up to the lower 255th address. 192 byte RAM and I/O registers are on page 0 of address space so that the direct addressing mode may be utilized.

• Extended

See Fig. 32. The extended addressing is used for referencing to all addresses of memory. The EA is the contents of the 2 bytes that follow the operation code. An extended addressing instruction requires a length of 3 bytes.

• Relative

See Fig. 33. The relative addressing mode is used with branch instructions only. When a branch occurs, the program counter is loaded with the contents of the byte following the operation code. EA = (PC) + 2 + Rel., where Rel. indicates a signed 8-bit data following the operation code. If no branch occurs, Rel. = 0. When a branch occurs, the program jumps to any byte in the range +129 to −127. A branch instruction requires a length of 2 bytes.

• Indexed (No Offset)

See Fig. 34. The indexed addressing mode allows access up to the lower 255th address of memory. In this mode, an instruction requires a length of one byte. The EA is the contents of the index register.

• Indexed (8-bit Offset)

See Fig. 35. The EA is the contents of the byte following the operation code, plus the contents of the index register. This mode allows access up to the lower 511th address of memory. Each instruction when used in the index addressing mode (8-bit offset) requires a length of 2 bytes.

• Indexed (16-bit Offset)

See Fig. 36. The contents of the 2 bytes following the operation code are added to content of the index register to compute the value of EA. In this mode, the complete memory can be accessed. When used in the indexed addressing mode (16-bit offset), an instruction must be 3 bytes long.

• Bit Set/Clear

See Fig. 37. This addressing mode is applied to the BSET and BCLR instructions that can set or clear any bit on page 0. The lower 3 bits of the operation code specify the bit to be set or cleared. The byte that follows the operation code indicates an address within page 0.

• Bit Test and Branch

See Fig. 38. This addressing mode is applied to the BRSET and BRCLR instructions that can test any bit within page 0 and can be branched in the relative addressing mode. The byte to be tested is addressed depending on the contents of the byte following the operation code. Individual bits within the byte to be tested are specified by the lower 3 bits of the operation code. The 3rd byte represents a relative value which will be added to the program counter when a branch condition is established. Each of these instructions should be 3 bytes long. The value of the test bit is written in the carry bit of the condition code register.

• Implied

See Fig. 39. This mode involves no EA. All information needed for execution of an instruction is contained in the operation code. Direct manipulation on the accumulator and index register is included in the implied addressing mode. Other instructions such as SWI and RTI are also used in this mode. All instructions used in the implied addressing mode should have a length of one byte.

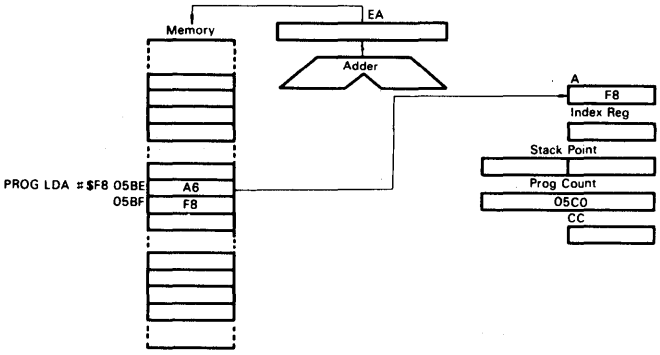


Figure 30 Example of Immediate Addressing

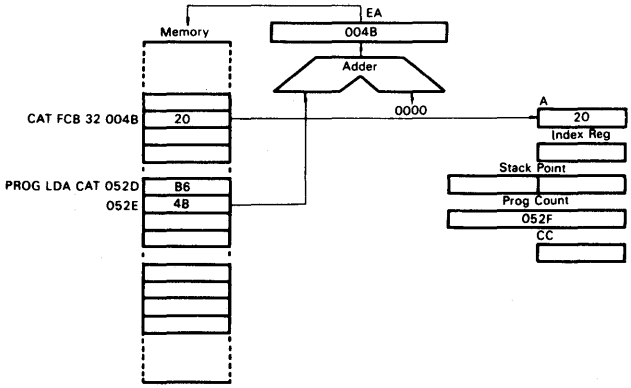


Figure 31 Example of Direct Addressing

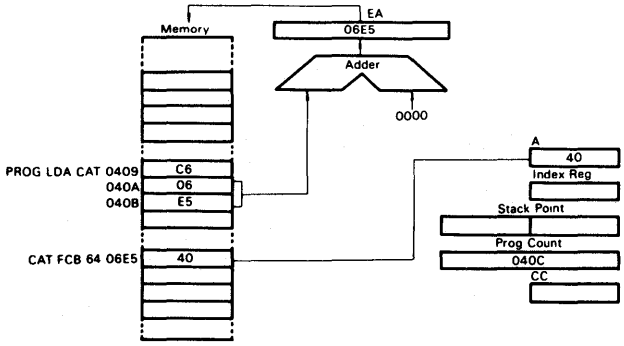


Figure 32 Example of Extended Addressing

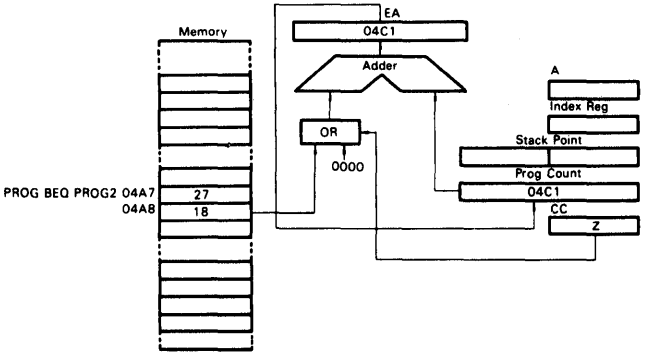


Figure 33 Example of Relative Addressing

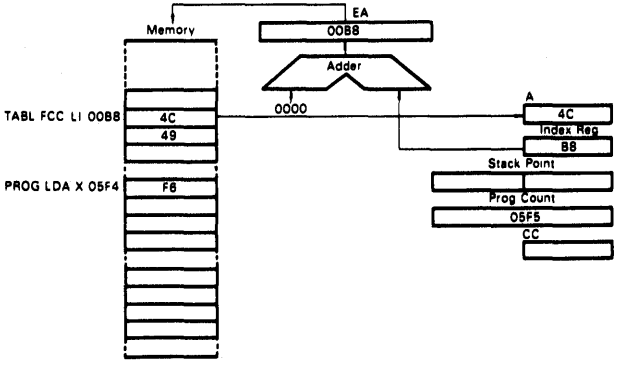


Figure 34 Example of Indexed (No Offset) Addressing

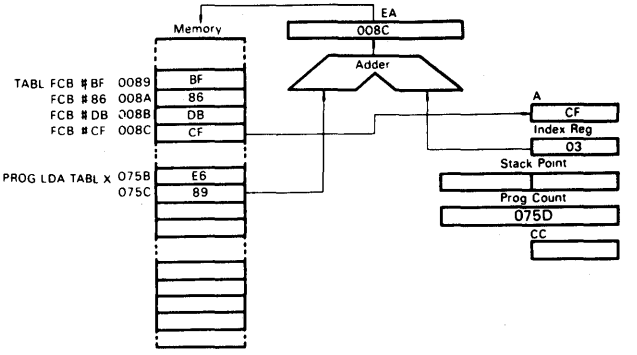


Figure 35 Example of Indexed (8-bit Offset) Addressing

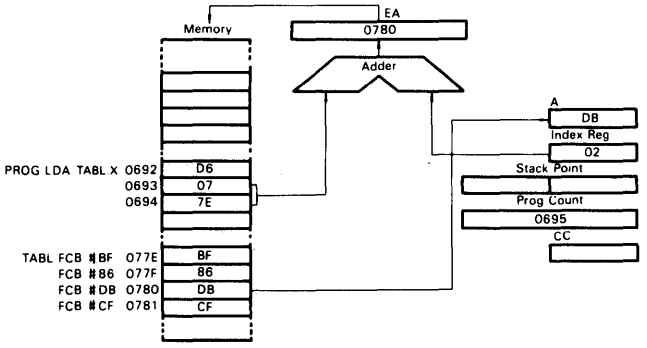


Figure 36 Example of Indexed (16-bit Offset) Addressing

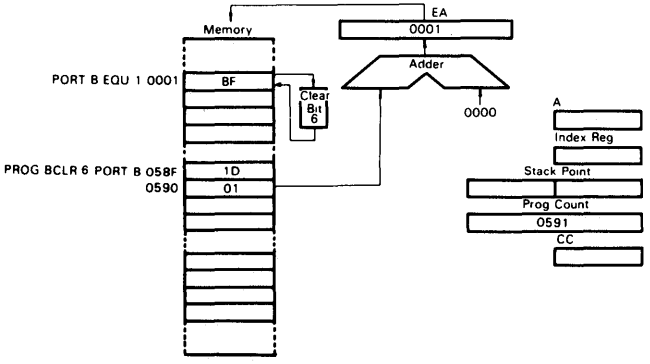


Figure 37 Example of Bit Set/Clear Addressing

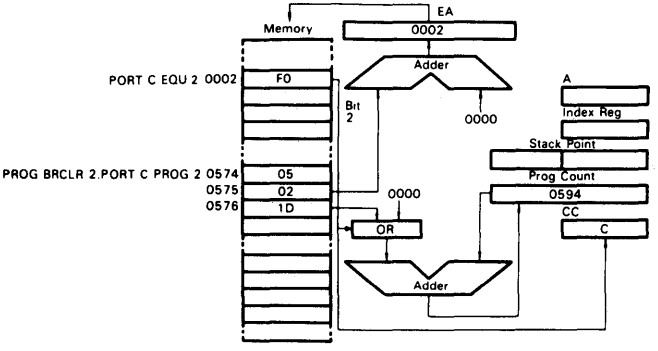


Figure 38 Example of Bit Test and Branch Addressing

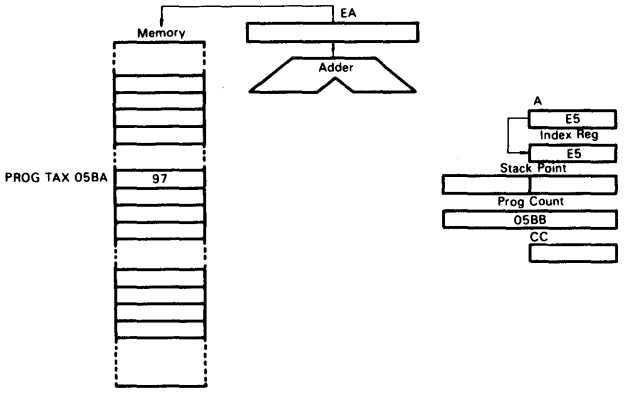


Figure 39 Example of Implied Addressing

■ INSTRUCTION SET

There are 62 basic instructions available to the HD63P05Y1 MCU. They can be classified into five categories: register/memory, read/modify/write, branch, bit manipulation, and control. The details of each instruction are described in Tables 5 through 11.

• Register/Memory Instructions

Most of these instructions use two operands. One operand is either an accumulator or index register. The other is derived from memory using one of the addressing modes used on the MCU. There is no register operand in the unconditional jump instruction (JMP) and the subroutine jump instruction (JSR). See Table 5.

• Read/Modify/Write Instructions

These instructions read a memory or register, then modify or test its contents, and write the modified value into the memory or register. Zero test instruction (TST) does not write data, and is handled as an exception in the read/modify/write group. See Table 6.

• Branch Instructions

A branch instruction branches from the program sequence in progress if a particular condition is established. See Table 7.

• Bit Manipulation Instructions

These instructions can be used with any bit located up to the lower 255th address of memory. Two groups are available; one for setting or clearing and the other for bit testing and branching. See Table 8.

• Control Instructions

The control instructions control the operation of the MCU which is executing a program. See Table 9.

• List of Instructions in Alphabetical Order

Table 10 lists all the instructions used on the MCU in the alphabetical order.

• Operation Code Map

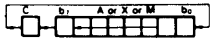
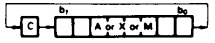
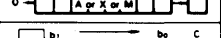
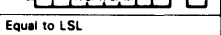
Table 11 shows the operation code map for the instructions used on the MCU.

Table 5 Register/Memory Instructions

Operations	Mnemonic	Addressing Modes: Immediate OP	#	~	Direct OP	#	~	Extended OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Indexed (16-Bit Offset) OP	#	~	Boolean/ Arithmetic Operation	Condition Code H	I	N	Z	C
Load A from Memory	LDA	A6	2	2	B6	2	3	C6	3	4	F6	1	3	E6	2	4	D6	3	5	M→A	•	•	∧	∧	•
Load X from Memory	LDX	AE	2	2	BE	2	3	CE	3	4	FE	1	3	EE	2	4	DE	3	5	M→X	•	•	∧	∧	•
Store A in Memory	STA				B7	2	3	C7	3	4	F7	1	4	E7	2	4	D7	3	5	A→M	•	•	∧	∧	•
Store X in Memory	STX				BF	2	3	CF	3	4	FF	1	4	EF	2	4	DF	3	5	X→M	•	•	∧	∧	•
Add Memory to A	ADD	AB	2	2	BB	2	3	CB	3	4	FB	1	3	EB	2	4	DB	3	5	A+M→A	∧	•	∧	∧	∧
Add Memory and Carry to A	ADC	A9	2	2	B9	2	3	C9	3	4	F9	1	3	E9	2	4	D9	3	5	A+M+C→A	∧	•	∧	∧	∧
Subtract Memory	SUB	A0	2	2	B0	2	3	C0	3	4	F0	1	3	E0	2	4	D0	3	5	A−M→A	•	•	∧	∧	∧
Subtract Memory from A with Borrow	SBC	A2	2	2	B2	2	3	C2	3	4	F2	1	3	E2	2	4	D2	3	5	A−M−C→A	•	•	∧	∧	∧
AND Memory to A	AND	A4	2	2	B4	2	3	C4	3	4	F4	1	3	E4	2	4	D4	3	5	A · M→A	•	•	∧	∧	•
OR Memory with A	ORA	AA	2	2	BA	2	3	CA	3	4	FA	1	3	EA	2	4	DA	3	5	A+M→A	•	•	∧	∧	•
Exclusive OR Memory with A	EOR	A8	2	2	B8	2	3	C8	3	4	F8	1	3	E8	2	4	D8	3	5	A⊕M→A	•	•	∧	∧	•
Arithmetic Compare A with Memory	CMP	A1	2	2	B1	2	3	C1	3	4	F1	1	3	E1	2	4	D1	3	5	A−M	•	•	∧	∧	∧
Arithmetic Compare X with Memory	CPX	A3	2	2	B3	2	3	C3	3	4	F3	1	3	E3	2	4	D3	3	5	X−M	•	•	∧	∧	∧
Bit Test Memory with A (Logical Compare)	BIT	A5	2	2	B5	2	3	C5	3	4	F5	1	3	E5	2	4	D5	3	5	A · M	•	•	∧	∧	•
Jump Unconditional	JMP				BC	2	2	CC	3	3	FC	1	2	EC	2	3	DC	3	4		•	•	•	•	•
Jump to Subroutine	JSR				BD	2	5	CD	3	6	FD	1	5	ED	2	5	DD	3	6		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 6 Read/Modify/Write Instructions

Operations	Mnemonic	Addressing Modes: Implied(A) OP	#	~	Implied(X) OP	#	~	Direct OP	#	~	Indexed (No Offset) OP	#	~	Indexed (8-Bit Offset) OP	#	~	Boolean/Arithmetic Operation	Condition Code H	I	N	Z	C
Increment	INC	4C	1	2	5C	1	2	3C	2	5	7C	1	5	6C	2	6	A+1→A or X+1→X or M+1→M	•	•	∧	∧	•
Decrement	DEC	4A	1	2	5A	1	2	3A	2	5	7A	1	5	6A	2	6	A−1→A or X−1→X or M−1→M	•	•	∧	∧	•
Clear	CLR	4F	1	2	5F	1	2	3F	2	5	7F	1	5	6F	2	6	00→A or 00→X or 00→M	•	•	0	1	•
Complement	COM	43	1	2	53	1	2	33	2	5	73	1	5	63	2	6	$\overline{A}$→A or $\overline{X}$→X or $\overline{M}$→M	•	•	∧	∧	1
Negate (2's Complement)	NEG	40	1	2	50	1	2	30	2	5	70	1	5	60	2	6	00−A→A or 00−X→X or 00−M→M	•	•	∧	∧	∧
Rotate Left Thru Carry	ROL	49	1	2	59	1	2	39	2	5	79	1	5	69	2	6		•	•	∧	∧	∧
Rotate Right Thru Carry	ROR	46	1	2	56	1	2	36	2	5	76	1	5	66	2	6		•	•	∧	∧	∧
Logical Shift Left	LSL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6		•	•	∧	∧	∧
Logical Shift Right	LSR	44	1	2	54	1	2	34	2	5	74	1	5	64	2	6		•	•	0	∧	∧
Arithmetic Shift Right	ASR	47	1	2	57	1	2	37	2	5	77	1	5	67	2	6		•	•	∧	∧	∧
Arithmetic Shift Left	ASL	48	1	2	58	1	2	38	2	5	78	1	5	68	2	6	Equal to LSL	•	•	∧	∧	∧
Test for Negative or Zero	TST	4D	1	2	5D	1	2	3D	2	4	7D	1	4	6D	2	5	A−00 or X−00 or M−00	•	•	∧	∧	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 7 Branch Instructions

Operations	Mnemonic	Addressing Modes			Branch Test	Condition Code				
		Relative								
		OP	#	~		H	I	N	Z	C
Branch Always	BRA	20	2	3	None	•	•	•	•	•
Branch Never	BRN	21	2	3	None	•	•	•	•	•
Branch IF Higher	BHI	22	2	3	C+Z=0	•	•	•	•	•
Branch IF Lower or Same	BLS	23	2	3	C+Z=1	•	•	•	•	•
Branch IF Carry Clear	BCC	24	2	3	C=0	•	•	•	•	•
(Branch IF Higher or Same)	(BHS)	24	2	3	C=0	•	•	•	•	•
Branch IF Carry Set	BCS	25	2	3	C=1	•	•	•	•	•
(Branch IF Lower)	(BLO)	25	2	3	C=1	•	•	•	•	•
Branch IF Not Equal	BNE	26	2	3	Z=0	•	•	•	•	•
Branch IF Equal	BEQ	27	2	3	Z=1	•	•	•	•	•
Branch IF Half Carry Clear	BHCC	28	2	3	H=0	•	•	•	•	•
Branch IF Half Carry Set	BHCS	29	2	3	H=1	•	•	•	•	•
Branch IF Plus	BPL	2A	2	3	N=0	•	•	•	•	•
Branch IF Minus	BMI	2B	2	3	N=1	•	•	•	•	•
Branch IF Interrupt Mask Bit is Clear	BMC	2C	2	3	I=0	•	•	•	•	•
Branch IF Interrupt Mask Bit is Set	BMS	2D	2	3	I=1	•	•	•	•	•
Branch IF Interrupt Line is Low	BIL	2E	2	3	INT=0	•	•	•	•	•
Branch IF Interrupt Line is High	BIH	2F	2	3	INT=1	•	•	•	•	•
Branch to Subroutine	BSR	AD	2	5	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 8 Bit Manipulation Instructions

Operations	Mnemonic	Addressing Modes						Boolean/ Arithmetic Operation	Branch Test	Condition Code				
		Bit Set/Clear			Bit Test and Branch									
		OP	#	~	OP	#	~			H	I	N	Z	C
Branch IF Bit n is set	BRSET n(n=0···7)	—	—	—	2·n	3	5	——	Mn=1	•	•	•	•	∧
Branch IF Bit n is clear	BRCLR n(n=0···7)	—	—	—	01+2·n	3	5	——	Mn=0	•	•	•	•	∧
Set Bit n	BSET n(n=0···7)	10+2·n	2	5	—	—	—	1→Mn	——	•	•	•	•	•
Clear Bit n	BCLR n(n=0···7)	11+2·n	2	5	—	—	—	0→Mn	——	•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

Table 9 Control Instructions

Operations	Mnemonic	Addressing Modes (Implied) OP	#	~	Boolean Operation	Condition Code H	I	N	Z	C
Transfer A to X	TAX	97	1	2	A→X	•	•	•	•	•
Transfer X to A	TXA	9F	1	2	X→A	•	•	•	•	•
Set Carry Bit	SEC	99	1	1	1→C	•	•	•	•	1
Clear Carry Bit	CLC	98	1	1	0→C	•	•	•	•	0
Set Interrupt Mask Bit	SEI	9B	1	2	1→I	•	•	•	•	•
Clear Interrupt Mask Bit	CLI	9A	1	2	0→I	•	0	•	•	•
Software Interrupt	SWI	83	1	10		•	1	•	•	•
Return from Subroutine	RTS	81	1	5		•	•	•	•	•
Return from Interrupt	RTI	80	1	8		?	?	?	?	?
Reset Stack Pointer	RSP	9C	1	2	$FF→SP	•	•	•	•	•
No-Operation	NOP	9D	1	1	Advance Prog. Cntr. Only	•	•	•	•	•
Decimal Adjust A	DAA	8D	1	2	Converts binary add of BCD charcters into BCD format	•	•	∧	∧	∧*
Stop	STOP	8E	1	4		•	•	•	•	•
Wait	WAIT	8F	1	4		•	•	•	•	•

Symbols: Op = Operation
= Number of bytes
~ = Number of cycles

* Are BCD characters of upper byte 10 or more? (They are not cleared if set in advance.)

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes: Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	Condition Code H	I	N	Z	C
ADC		×	×	×		×	×	×			∧	•	∧	∧	∧
ADD		×	×	×		×	×	×			∧	•	∧	∧	∧
AND		×	×	×		×	×	×			•	•	∧	∧	•
ASL	×		×			×	×				•	•	∧	∧	∧
ASR	×		×			×	×				•	•	∧	∧	∧
BCC					×						•	•	•	•	•
BCLR									×		•	•	•	•	•
BCS					×						•	•	•	•	•
BEQ					×						•	•	•	•	•
BHCC					×						•	•	•	•	•
BHCS					×						•	•	•	•	•
BHI					×						•	•	•	•	•
(BHS)					×						•	•	•	•	•
BIH					×						•	•	•	•	•
BIL					×						•	•	•	•	•
BIT		×	×	×		×	×	×			•	•	∧	∧	•
(BLO)					×						•	•	•	•	•
BLS					×						•	•	•	•	•
BMC					×						•	•	•	•	•
BMI					×						•	•	•	•	•
BMS					×						•	•	•	•	•
BNE					×						•	•	•	•	•
BPL					×						•	•	•	•	•
BRA					×						•	•	•	•	•

(to be continued)

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	∧	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 10 Instruction Set (in Alphabetical Order)

Mnemonic	Addressing Modes										Condition Code				
	Implied	Immediate	Direct	Extended	Relative	Indexed (No Offset)	Indexed (8-Bit)	Indexed (16-Bit)	Bit Set/ Clear	Bit Test & Branch	H	I	N	Z	C
BRN					×						•	•	•	•	•
BRCLR										×	•	•	•	•	^
BRSET										×	•	•	•	•	^
BSET									×		•	•	•	•	•
BSR					×						•	•	•	•	•
CLC	×										•	•	•	•	0
CLI	×										•	0	•	•	•
CLR	×		×			×	×				•	•	0	1	•
CMP		×	×	×		×	×	×			•	•	^	^	^
COM	×		×			×	×				•	•	^	^	1
CPX		×	×	×		×	×	×			•	•	^	^	^
DAA	×										•	•	^	^	^
DEC	×		×			×	×				•	•	^	^	•
EOR		×	×	×		×	×	×			•	•	^	^	•
INC	×		×			×	×				•	•	^	^	•
JMP			×	×		×	×	×			•	•	•	•	•
JSR			×	×		×	×	×			•	•	•	•	•
LDA		×	×	×		×	×	×			•	•	^	^	•
LDX		×	×	×		×	×	×			•	•	^	^	•
LSL	×		×			×	×				•	•	^	^	^
LSR	×		×			×	×				•	•	0	^	^
NEG	×		×			×	×				•	•	^	^	^
NOP	×										•	•	•	•	•
ORA		×	×	×		×	×	×			•	•	^	^	•
ROL	×		×			×	×				•	•	^	^	^
ROR	×		×			×	×				•	•	^	^	^
RSP	×										•	•	•	•	•
RTI	×										?	?	?	?	?
RTS	×										•	•	•	•	•
SBC		×	×	×		×	×	×			•	•	^	^	^
SEC	×										•	•	•	•	1
SEI	×										•	1	•	•	•
STA			×	×		×	×	×			•	•	^	^	•
STOP	×										•	•	•	•	•
STX			×	×		×	×	×			•	•	^	^	•
SUB		×	×	×		×	×	×			•	•	^	^	^
SWI	×										•	1	•	•	•
TAX	×										•	•	•	•	•
TST	×		×			×	×				•	•	^	^	•
TXA	×										•	•	•	•	•
WAIT	×										•	•	•	•	•

Condition Code Symbols:

H	Half Carry (From Bit 3)	C	Carry/Borrow
I	Interrupt Mask	^	Test and Set if True, Cleared Otherwise
N	Negative (Sign Bit)	•	Not Affected
Z	Zero	?	Load CC Register From Stack

Table 11 Operation Code Map

	Bit Manipulation		Branch	Read/Modify/Write					Control		Register/Memory						
	Test & Branch	Set/ Clear	Rel	DIR	A	X	,X1	,X0	IMP	IMP	IMM	DIR	EXT	,X2	,X1	,X0	
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	← HIGH
0	BRSET0	BSET0	BRA	NEG					RTI*		SUB						0
1	BRCLR0	BCLR0	BRN						RTS*		CMP						1
2	BRSET1	BSET1	BHI								SBC						2
3	BRCLR1	BCLR1	BLS	COM					SWI*		CPX						3
4	BRSET2	BSET2	BCC	LSR							AND						4
5	BRCLR2	BCLR2	BCS								BIT						5
6	BRSET3	BSET3	BNE	ROR							LDA						6
7	BRCLR3	BCLR3	BEQ	ASR						TAX*		STA				STA(+1)	7
8	BRSET4	BSET4	BHCC	LSL/ASL						CLC	EOR						8
9	BRCLR4	BCLR4	BHCS	ROL						SEC	ADC						9
A	BRSET5	BSET5	BPL	DEC						CLI*	ORA						A
B	BRCLR5	BCLR5	BMI							SEI*	ADD						B
C	BRSET6	BSET6	BMC	INC						RSP*		JMP(−1)					C
D	BRCLR6	BCLR6	BMS	TST(−1)	TST			TST(−1)	DAA*	NOP	BSR*	JSR(+2)		JSR(+1)		JSR(+2)	D
E	BRSET7	BSET7	BIL						STOP*	—	LDX						E
F	BRCLR7	BCLR7	BIH	CLR					WAIT*	TXA*		STX				STX(+1)	F
	3/5	2/5	2/3	2/5	1/2	1/2	2/6	1/5	1/*	1/1	2/2	2/3	3/4	3/5	2/4	1/3	

LOW

(NOTES) 1. "—" is an undefined operation code.

2. The lowermost numbers in each column represent a byte count and the number of cycles required (byte count/number of cycles). The number of cycles for the mnemonics asterisked (*) is as follows:

RTI	8	TAX	2
RTS	5	RSP	2
SWI	10	TXA	2
DAA	2	BSR	5
STOP	4	CLI	2
WAIT	4	SEI	2

3. The parenthesized numbers must be added to the cycle count of the particular instruction.

- **Additional Instructions**

The following new instructions are used on the HD63P05Y1:

DAA Converts the contents of the accumulator into BCD code.

WAIT Causes the MCU to enter the wait mode. For this mode, see the topic, Wait Mode.

STOP Causes the MCU to enter the stop mode. For this mode, see the topic, Stop Mode.

HD63701V0, HD637A01V0, HD637B01V0

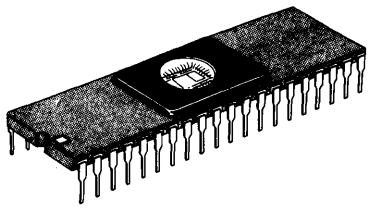
CMOS MCU (Microcomputer Unit)

—ADVANCE INFORMATION—

The HD63701V0 is an 8-bit CMOS single-chip microcomputer unit, pin compatible with the HD6301V. 4kB EPROM, 192 bytes RAM, Serial Communication Interface (SCI), parallel I/O ports and multi function timer are incorporated in the HD63701V0. It is bus compatible with HMCS6800. Execution time of key instructions are improved and several new instructions are added to increase system throughput. The HD63701V0 can be expanded up to 65k words. Like the HMCS6800 family, I/O level is TTL compatible with +5.0V single power supply. As HD63701V0 is fabricated by the advanced CMOS process technology, power dissipation is extremely reduced. In addition to that, HD63701V0 has Sleep Mode and Standby Mode at lower power dissipation mode. Therefore flexible low power consumption application is possible.

On chip EPROM can be programmed by the same procedure as that of 27C256 or 27256.

HD63701V0C, HD637A01V0C, HD637B01V0C



(DC-40)

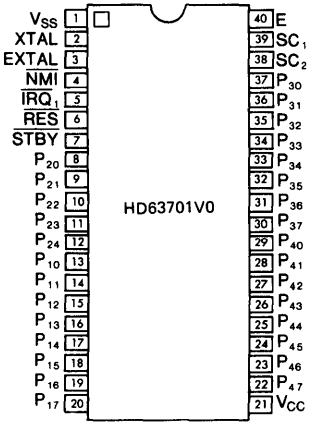
■ FEATURES

- Instruction Set Compatible with HD6301 Family
- Abundant On-Chip Functions
  4kB EPROM, 192 Bytes RAM, 29 Parallel I/O Lines, 2 Lines of Data Strobe, 16-bit Timer, Serial Communication Interface
- Low Power Consumption Mode: Sleep Mode, Standby Mode
- Minimum Instruction Execution Time
  1 μs (f=1MHz), 0.67 μs (f=1.5MHz), 0.5 μs (f=2MHz)
- Bit Manipulation, Bit Test Instruction
- Protection from System Upset: Address Trap, Op-Code Trap
- Up to 65k Words Address Space
- Wide Operation Range
  V_{CC}=5V±10% (f=0.1 to 2.0MHz)

■ TYPE OF PRODUCTS

Type No.	Bus Timing
HD63701V0	1.0 MHz
HD637A01V0	1.5 MHz
HD637B01V0	2.0 MHz

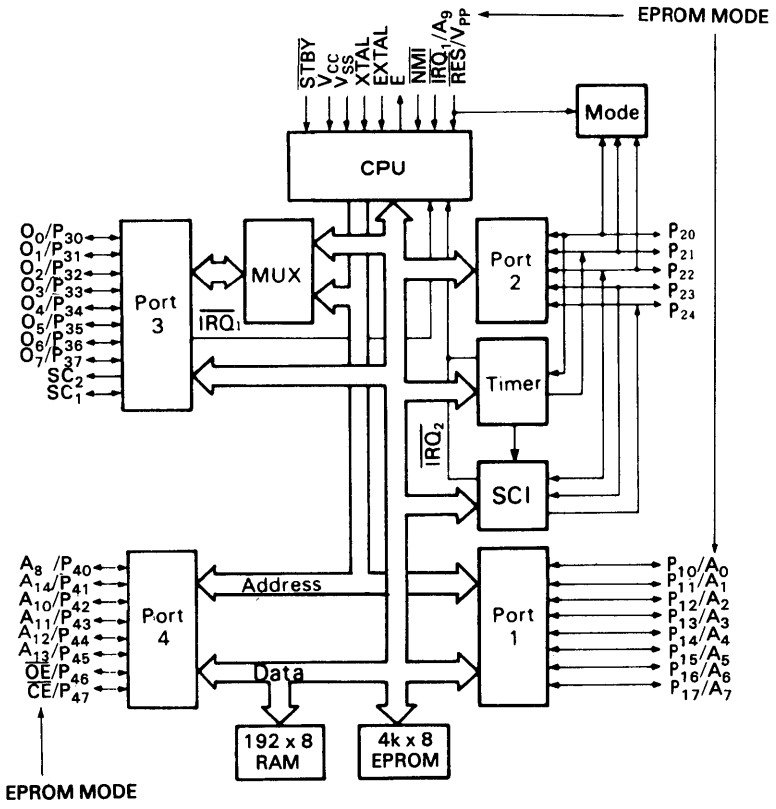
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BLOCK DIAGRAM**



HD63701X0, HD637A01X0, HD637B01X0
CMOS MCU (Microcomputer Unit)

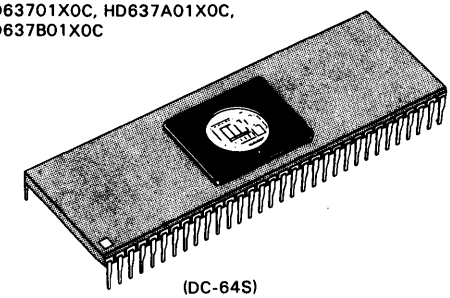
—PRELIMINARY—

The HD63701X0 is a high performance 8-bit CMOS single chip microcomputer unit (MCU) which, including 4k bytes of EPROM, is pin compatible with the HD6301X0.

The HD63701X0 contains 4k bytes of EPROM, 192 bytes of RAM, serial communication interface and 53 parallel I/O pins in addition to CPU. It includes functions of halt, memory ready, low speed access and releasing external bus at system expansion.

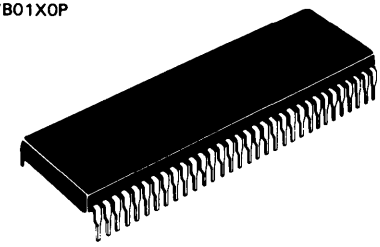
The HD63701X0 is available in a hermetically sealed 64-pin shrunk ceramic package which includes a window that allows for EPROM erasure and in a 64-pin shrunk plastic package which is one-time-programmable type. It can be programmed in the same method as 2732A type EPROM.

HD63701X0C, HD637A01X0C, HD637B01X0C



(DC-64S)

HD63701X0P, HD637A01X0P, HD637B01X0P



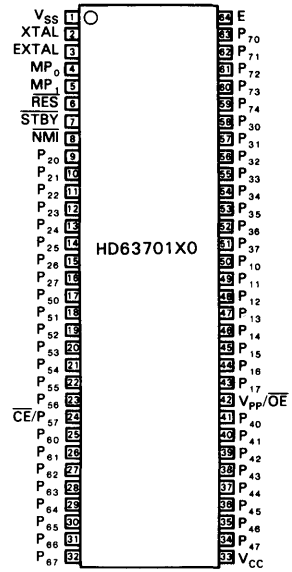
(DP-64S)

■ FEATURES

- Instruction Set Compatible with the HD6301X0
- 4k Bytes of EPROM (compatible with 2732A type)
- 192 Bytes of RAM
- 53 Parallel I/O Pins
  - 24 I/O Common Pins (Port 2, 3, 6)
  - 21 Output Pins (Port1, 4, 7)
  - 8 Input Pins (Port 5)
- Driving Darlington Transistor (Port 2, 6)
- 16-bit Programmable Timer
  - Input Capture Register x 1
  - Free Running Counter x 1
  - Output Compare Register x 2
- 8-bit Reloadable Timer
  - External Event Count
  - Square Wave Occurrence
- Serial Communication Interface (SCI)
  - Asynchronous Mode/Clock Synchronous Mode
  - 3 Transfer Formats (Asynchronous Mode)
  - 6 Clock Sources
- Memory Ready for Low Speed Memory Access
- Halt
- Error-Detection (Address Error, Op-code Error)
- Interrupts — 3 External, 7 Internal
- Operation Mode
  - MCU Mode
    - Mode 1 — Expanded (Internal ROM Inhibited)
    - Mode 2 — Expanded (Internal ROM Valid)
    - Mode 3 — Single-chip Mode
  - EPROM Mode
- Up to 65k Bytes of Address Space
- Low Power Dissipation Mode
  - Sleep
  - Standby
- Minimum Instruction Execution Time — 0.5μs (f=2.0MHz)
- Wide Operation Range

V_{CC}=5V±10%
- f=0.1 to 1.0MHz; HD63701X0
- f=0.1 to 1.5MHz; HD637A01X0
- f=0.1 to 2.0MHz; HD637B01X0

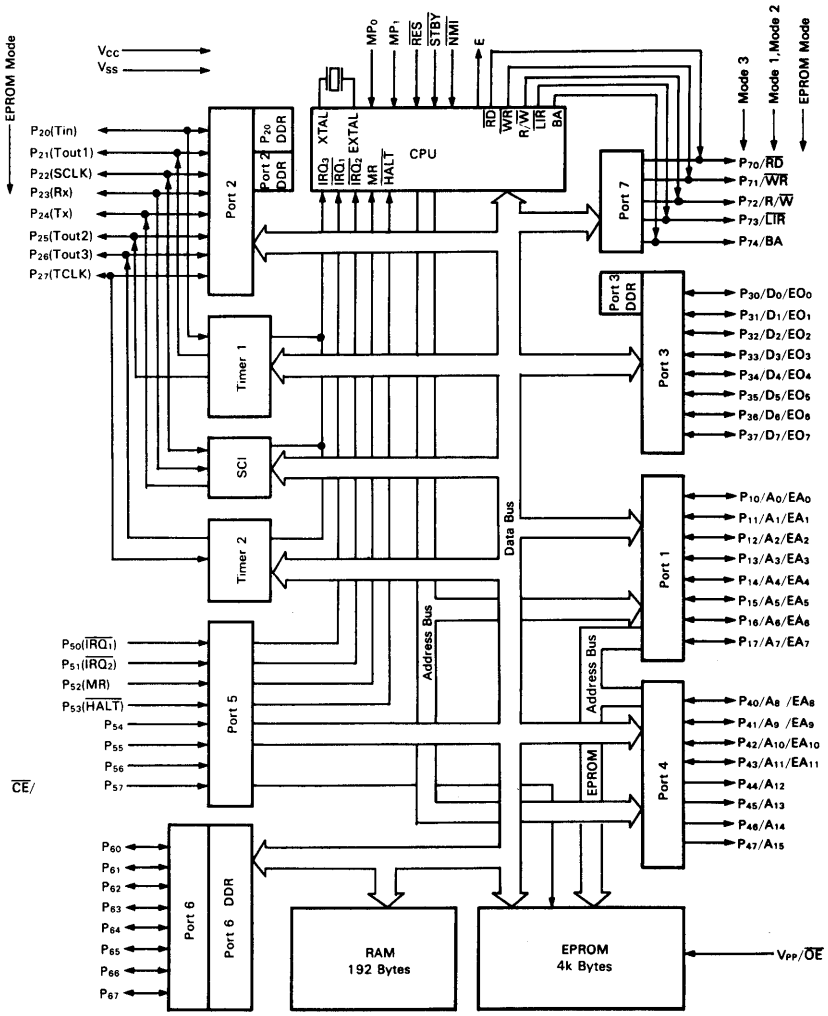
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



■ ABSOLUTE MAXIMUM RATINGS

Item	Symbol	Value	Unit
Supply Voltage	V_{CC}	-0.3 ~ +7.0	V
Program Voltage	V_{PP}	-0.3 ~ 22	V
Input Voltage	V_{in}	-0.3 ~ V_{CC}+0.3	V
Operating Temperature	T_{opr}	0 ~ +70	°C
Storage Temperature	T_{stg}	-55 ~ +125	°C

(Note) This product has protection circuits in input terminal from high static electricity voltage and high electric field. But be careful not to apply overvoltage more than maximum ratings to these high input impedance protection circuits. To assure the normal operation, we recommend V_{in}, V_{out}: $V_{SS} \leqq (V_{in} \text{ or } V_{out}) \leqq V_{CC}$.

■ MCU ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC} =5V±10%, V_{SS} = V_{PP} =0V, Ta= 0 ~ +70°C, unless otherwise noted.)

Item		Symbol	Test Condition	min	typ	max	Unit
Input "High" Voltage	$\overline{RES}$, $\overline{STBY}$, MP_0, MP_1	V_{IH}		V_{CC}-0.5	–	V_{CC}+0.3	V
	EXTAL			V_{CC}x0.7	–		
	P_{22} (SCLK)***			2.4	–		
	Other Inputs			2.2	–		
Input "Low" Voltage	All Inputs	V_{IL}		-0.3	–	0.8	V
Input Leakage Current	$\overline{NMI}$, $\overline{RES}$, $\overline{STBY}$, MP_0, MP_1, Port 5	$\|I_{in}\|$	V_{in}=0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Three State (off-state) Leakage Current	Ports 1, 2, 3, 4, 6, 7	$\|I_{TSI}\|$	V_{in}=0.5 ~ V_{CC}-0.5V	–	–	1.0	μA
Output "High" Voltage	All Outputs	V_{OH}	I_{OH}=-200μA	2.4	–	–	V
			I_{OH}=-10μA	V_{CC}-0.7	–	–	V
Output "Low" Voltage	Ports 2, 6	V_{OL}	I_{OL}=1.6mA	–	–	0.5	V
	Other Outputs			–	–	0.4	V
Darlington Drive Current	Ports 2, 6	$-I_{OH}$	V_{out}=1.5V	1.0	–	10.0	mA
Input Capacitance	All Inputs (Except V_{PP}/$\overline{OE}$)	C_{in}	V_{in}=0V, f=1MHz, Ta=25°C	–	–	12.5	pF
	V_{PP}/$\overline{OE}$			–	–	25	pF
Standby Current	Non Operation	I_{STB}		–	3.0	15.0	μA
Current Dissipation*		I_{SLP}	Sleeping (f=1MHz**)	–	1.5	3.0	mA
			Sleeping (f=1.5MHz**)	–	2.3	4.5	mA
			Sleeping (f=2MHz**)	–	3.0	6.0	mA
		I_{CC}	Operating (f=1MHz**)	–	7.0	10.0	mA
			Operating (f=1.5MHz**)	–	10.5	15.0	mA
			Operating (f=2MHz**)	–	14.0	20.0	mA
RAM Standby Voltage		V_{RAM}		2.0	–	–	V

*V_{IH} min = V_{CC}-1.0V, V_{IL} max = 0.8V (All output terminals are at no load.)

**Current Dissipation of the operating or sleeping condition is proportional to the operating frequency. So the typ. or max. values about Current Dissipations at x MHz operation are decided according to the following formula;

typ. value (f = x MHz) = typ. value (f = 1MHz) x x
max. value (f = x MHz) = max. value (f = 1MHz) x x

***Synchronous clock input use only.

- **AC CHARACTERISTICS** (V_{CC}=5V±10%, V_{SS}=V_{PP}=0V, Ta=0 ~ +70°C, unless otherwise noted.)

BUS TIMING

Item		Symbol	Test Condition	HD63701X0			HD637A01X0			HD637B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Cycle Time		t_{cyc}	Fig. 1	1	–	10	0.666	–	10	0.5	–	10	μs
Enable Rise Time		t_{Er}		–	–	25	–	–	25	–	–	25	ns
Enable Fall Time		t_{Ef}		–	–	25	–	–	25	–	–	25	ns
Enable Pulse Width "High" Level*		PW_{EH}		450	–	–	300	–	–	220	–	–	ns
Enable Pulse Width "Low" Level*		PW_{EL}		450	–	–	300	–	–	220	–	–	ns
Address, R/$\overline{W}$ Delay Time*		t_{AD}		–	–	250	–	–	190	–	–	160	ns
Data Delay Time	Write	t_{DDW}		–	–	200	–	–	160	–	–	120	ns
Data Set-up Time	Read	t_{DSR}		80	–	–	70	–	–	70	–	–	ns
Address, R/$\overline{W}$ Hold Time*		t_{AH}		70	–	–	45	–	–	30	–	–	ns
Data Hold Time	Write*	t_{HW}		70	–	–	50	–	–	35	–	–	ns
	Read	t_{HR}		0	–	–	0	–	–	0	–	–	ns
$\overline{RD}$, $\overline{WR}$ Pulse Width*		PW_{RW}		450	–	–	300	–	–	220	–	–	ns
$\overline{RD}$, $\overline{WR}$ Delay Time		t_{RWD}		–	–	40	–	–	40	–	–	40	ns
$\overline{RD}$, $\overline{WR}$ Hold Time		t_{HRW}		–	–	30	–	–	30	–	–	25	ns
$\overline{LIR}$ Delay Time		t_{DLR}		–	–	200	–	–	160	–	–	120	ns
$\overline{LIR}$ Hold Time		t_{HLR}		10	–	–	10	–	–	10	–	–	ns
MR Set-up Time*		t_{SMR}	Fig. 2	400	–	–	280	–	–	230	–	–	ns
MR Hold Time*		t_{HMR}		–	–	90	–	–	40	–	–	0	ns
E Clock Pulse Width at MR		PW_{EMR}		–	–	9	–	–	9	–	–	9	μs
Processor Control Set-up Time		t_{PCS}	Fig. 3, 10, 11	200	–	–	200	–	–	200	–	–	ns
Processor Control Rise Time		t_{PCr}	Fig. 2, 3	–	–	100	–	–	100	–	–	100	ns
Processor Control Fall Time		t_{PCf}		–	–	100	–	–	100	–	–	100	ns
BA Delay Time		t_{BA}	Fig. 3	–	–	250	–	–	190	–	–	160	ns
Oscillator Stabilization Time		t_{RC}	Fig. 11	20	–	–	20	–	–	20	–	–	ms
Reset Pulse Width		PW_{RST}		3	–	–	3	–	–	3	–	–	t_{cyc}

* These timings change in approximate proportion to t_{cyc}. The figures in this characteristics represent those when t_{cyc} is minimum (= in the highest speed operation).

PERIPHERAL PORT TIMING

Item		Symbol	Test Condition	HD63701X0			HD637A01X0			HD637B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Peripheral Data Set-up Time	Ports 2, 3, 5, 6	t_{PDSU}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Peripheral Data Hold Time	Ports 2, 3, 5, 6	t_{PDH}	Fig. 5	200	–	–	200	–	–	200	–	–	ns
Delay Time (Enable positive Transition to Peripheral Data Valid)	Ports 1, 2, 3, 4, 6, 7	t_{PWED}	Fig. 6	–	–	800	–	–	630	–	–	550	ns

TIMER, SCI TIMING

Item		Symbol	Test Condition	HD63701X0			HD637A01X0			HD637B01X0			Unit
				min	typ	max	min	typ	max	min	typ	max	
Timer 1 Input Pulse Width		t_{PWT}	Fig. 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Delay Time (Enable Positive Transition to Timer Output)		t_{TOD}	Fig. 7	–	–	400	–	–	400	–	–	400	ns
SCI Input Clock Cycle	Async. Mode	t_{Scyc}	Fig. 8	1.0	–	–	1.0	–	–	1.0	–	–	t_{cyc}
	Clock Sync.		Fig. 4, 8	2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
SCI Transmit Data Delay Time (Clock Sync. Mode)		t_{TXD}	Fig. 4	–	–	200	–	–	200	–	–	200	ns
SCI Receive Data Set-up Time (Clock Sync. Mode)		t_{SRX}		290	–	–	290	–	–	290	–	–	ns
SCI Receive Data Hold Time (Clock Sync. Mode)		t_{HRX}		100	–	–	100	–	–	100	–	–	ns
SCI Input Clock Pulse Width		t_{PWSCK}	Fig. 8	0.4	–	0.6	0.4	–	0.6	0.4	–	0.6	t_{Scyc}
Timer 2 Input Clock Cycle		t_{tcyc}		2.0	–	–	2.0	–	–	2.0	–	–	t_{cyc}
Timer 2 Input Clock Pulse Width		t_{PWTCK}		200	–	–	200	–	–	200	–	–	ns
Timer 1·2, SCI Input Clock Rise Time		t_{CKr}		–	–	100	–	–	100	–	–	100	ns
Timer 1·2, SCI Input Clock Fall Time		t_{CKf}		–	–	100	–	–	100	–	–	100	ns

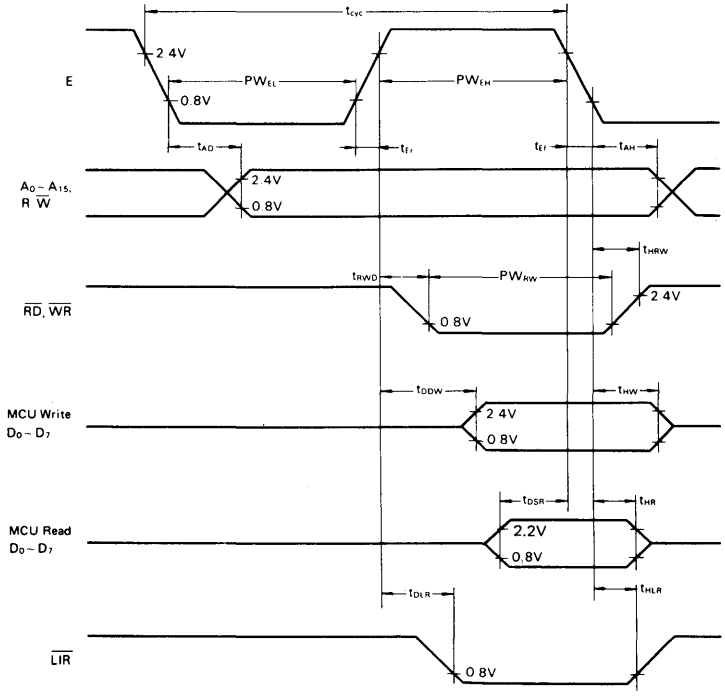


Figure 1 Mode 1, Mode 2 Bus Timing

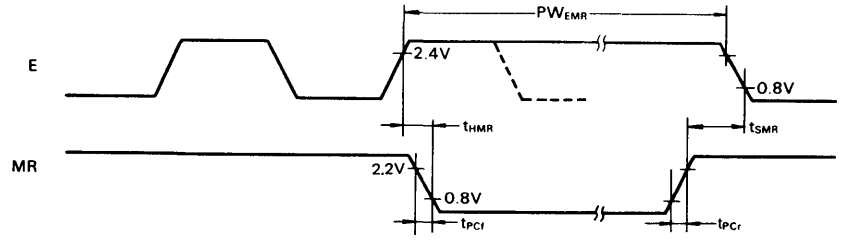


Figure 2 Memory Ready and E Clock Timing

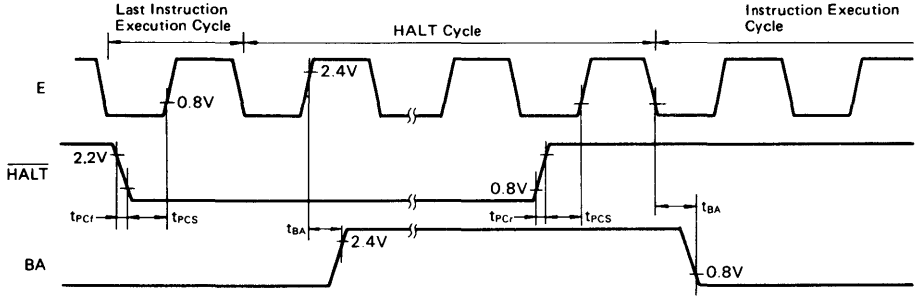


Figure 3 $\overline{\text{HALT}}$ and BA Timing

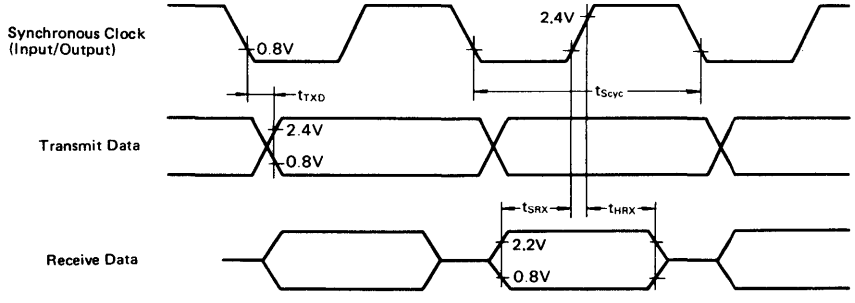


Figure 4 SCI Clocked Synchronous Timing

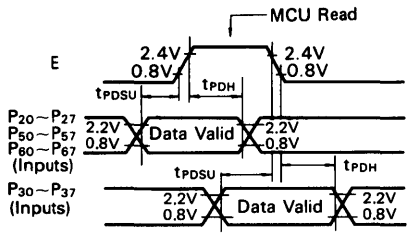


Figure 5 Port Data Set-up and Hold Times (MCU Read)

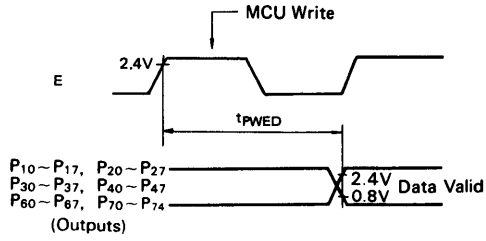
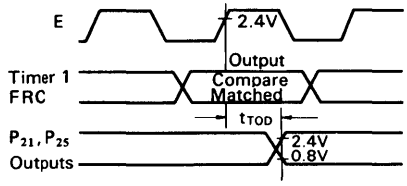
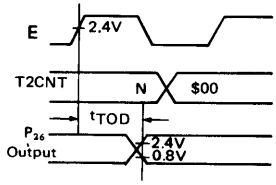


Figure 6 Port Data Delay Times (MCU Write)

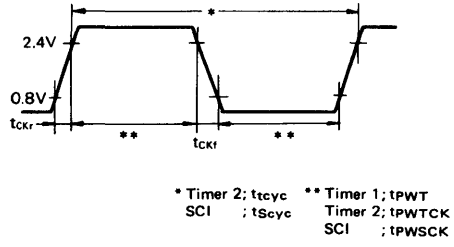


(a) Timer 1 Output Timing



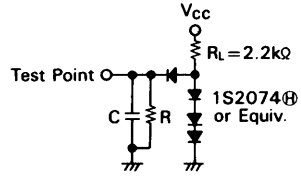
(b) Timer 2 Output Timing

Figure 7 Timer Output Timing



* Timer 2; t_{TCYC} ** Timer 1; t_{PWT}
SCI ; t_{SCYC} Timer 2; t_{PWTCK}
SCI ; t_{PWSCK}

Figure 8 Timer 1·2, SCI Input Clock Timing



C=90pF for Port 1, Port 3, Port 4, E
=30pF for Port 2, Port 6, Port 7
R=12kΩ for Port 1~Port 4, Port 6, Port 7, E

Figure 9 Bus Timing Test Loads (TTL Load)

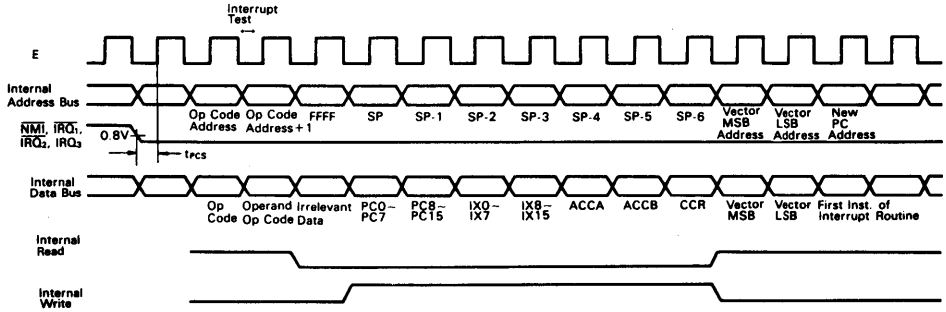


Figure 10 Interrupt Sequence

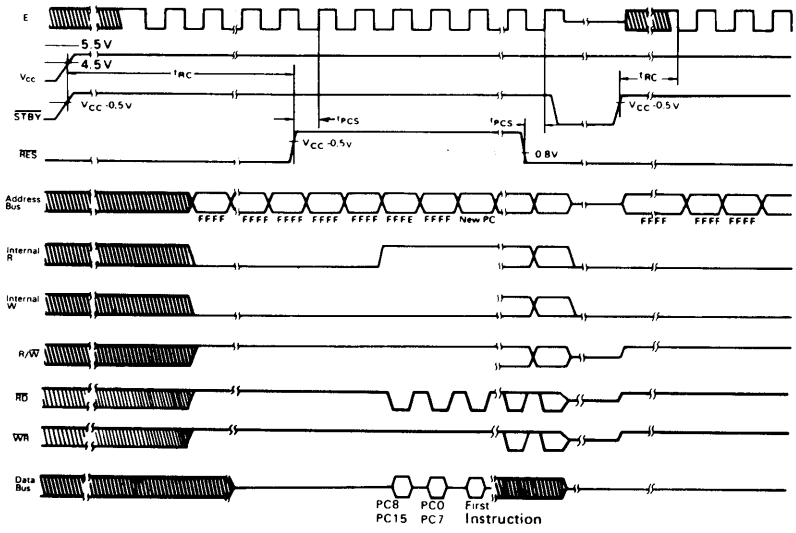


Figure 11 Reset Timing

■ EPROM PROGRAMMING ELECTRICAL CHARACTERISTICS

• DC CHARACTERISTICS (V_{CC}=5V±10%, V_{PP}=21V±0.5V, V_{SS}=0V, Ta=25°C±5°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Program Voltage	V_{PP}		20.5	21	21.5	V
Program Current	I_{PP}	V_{PP}=21V	–	–	30	mA
Input Leakage Current	I_{LI}	V_{in}=5.25V/0.4V	–	–	10	μA
Input "Low" Voltage	V_{IL}		-0.1	–	0.6	V
Input "High" Voltage	V_{IH}		2.2	–	V_{CC}+1.0	V
Output "Low" Voltage	V_{OL}	I_{OL}=1.6mA	–	–	0.4	V
Output "High" Voltage	V_{OH}	I_{OH}=-200 μA	2.4	–	–	V

• AC CHARACTERISTICS (V_{CC}=5V±10%, V_{PP}=21V±0.5V, V_{SS}=0V, Ta=25°C±5°C, unless otherwise noted.)

Item	Symbol	Test Condition	min	typ	max	Unit
Address Set-up Time	t_{AS}		2	–	–	μs
Address Hold Time	t_{AH}		0	–	–	μs
$\overline{OE}$ Set-up Time	t_{OES}		2	–	–	μs
$\overline{OE}$ Hold Time	t_{OEH}		2	–	–	μs
Data Set-up Time	t_{DS}		2	–	–	μs
Data Hold Time	t_{DH}		2	–	–	μs
Output Disable Delay Time	t_{DF}		0	–	130	ns
Data Valid from $\overline{CE}$	t_{DV}	$\overline{CE}$=V_{IL}, $\overline{OE}$=V_{IL}	–	–	1	μs
$\overline{CE}$ Pulse Width	t_{PW}		45	50	55	ms
$\overline{OE}$ Pulse Rise Time	t_{PRT}		50	–	–	ns
V_{pp} Recovery Time	t_{VR}		2	–	–	μs

(Note) t_{DF} is defined when output becomes open because output level can not be refered.

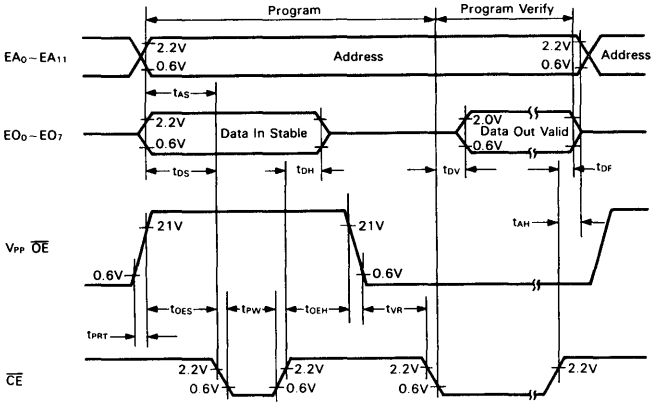


Figure 12 EPROM Programming Timing

■ FUNCTIONAL PIN DESCRIPTION

● V_{CC}, V_{SS}

V_{CC} and V_{SS} provide power to the MCU with 5V±10% supply. V_{SS} pin should be tied to ground.

● XTAL, EXTAL

These two pins interface a crystal (an AT-cut type). Divide-by-four circuit is on chip. When 4MHz crystal is used, the system clock is 1MHz for example.

EXTAL pin may be driven with an external clock of 45 to 55% duty, and one fourth frequency of the external clock is produced in the LSI. The external clock frequency should be less

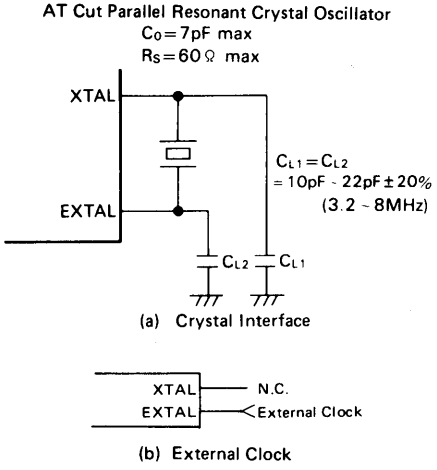


Figure 13 Connection Circuit

than four times of the maximum frequency. When using the external clock, XTAL pin should be open. Fig. 13 shows an example of connection circuit. The crystal and C_{L1}, C_{L2} should be mounted as close as possible to XTAL and EXTAL pins. Any line must not cross the line between the crystal and XTAL, EXTAL.

● $\overline{STBY}$

This pin is used for standby mode or EPROM mode.

In standby mode, the oscillation may be stopped. To retain the contents of RAM at standby, "0" should be written into RAM enable bit (RAMW). RAME is the bit 6 of the RAM/port 5 control register at $0014. RAM is disabled by this operation and its contents is sustained. Refer to "LOW POWER DISSIPATION MODE" for standby mode.

When this pin and Mode Program pins, MP_0 and MP_1, are "Low" level, the MCU is in EPROM mode. Refer to "PROGRAMMING THE EPROM" for details.

● Reset ($\overline{RES}$)

This pin is used to reset the MCU's internal state and provide a startup procedure. During power up, $\overline{RES}$ pin must be held below "Low" level for more than 20 ms.

The CPU registers (accumulator, index register, stack pointer, condition code register except for interrupt mask bit), RAM and data registers of ports are not initialized during reset, so their contents are unknown in a startup procedure.

To reset the MCU during operation, $\overline{RES}$ should be held "Low" for at least 3 system-clock cycles. At the 3rd cycle, all the address buses become "High". When $\overline{RES}$ remains "Low", the address buses keep "High". If $\overline{RES}$ turns "High", the MCU restart sequence is:

(1) Latch the value of the mode program pins: MP_0 and MP_1.
(2) Initialize each internal register (refer to Table 5).
(3) Set the interrupt mask bit. For the CPU to recognize the maskable interrupts $\overline{IRQ_1}$, $\overline{IRQ_2}$ and IRQ_3, this bit should be cleared in advance.
(4) Put the contents (=start address) of the last two addresses ($FFFE, $FFFF) into the program counter and start the program from this address. (Refer to Table 1).

* The MCU is unable to accept a reset input until the clock becomes normal oscillation after power on (max. 20ms). During this transient time, the MCU and I/O pins are undefined. Please be aware of this for system designing.

● Enable (E)

This pin provides a TTL-compatible clock used for bus synchronization. Its frequency is one fourth that of the internal oscillator or external clock. This pin can drive one TTL load and 90pF capacitance.

● Non-Maskable Interrupt ($\overline{NMI}$)

When the negative edge of the input signal is detected at this pin, the CPU will begin a non-maskable interrupt sequence. But the current instruction will be completed before it responds to the request. The interrupt mask bit of the condition code register doesn't affect non-maskable interrupt at all.

When the interrupt occurs, the contents of the program counter, the index register, the accumulators and the condition code register will be pushed onto the stack. Upon completion of this sequence, a vector is fetched from $FFFC and $FFFD, transferred their contents to the program counter and the non-maskable interrupt service routine starts. After reset, the stack pointer should be initialized on an appropriate memory area before $\overline{NMI}$ input.

● Interrupt Request ($\overline{IRQ_1}$, $\overline{IRQ_2}$)

These are level-sensitive pins which request an internal interrupt sequence. At interrupt request, the CPU will complete the current instruction before it responds to the request. If the interrupt mask in the condition code register is clear, the CPU will begin an interrupt sequence; if set, the interrupt request will be ignored. When the sequence starts, the contents of the program counter, the index register, the accumulators and the condition code register will be pushed onto the stack, then the interrupt mask bit will be set and inhibits all maskable interrupt. Finally, a vector is fetched from an address depicted in Table 1 and transferred to the program counter, and instruction execution is resumed.

The external interrupt pins, $\overline{IRQ_1}$ and $\overline{IRQ_2}$ are also used for port pins P_{50} and P_{51}, so it is controlled by Bit 0 and 1 of the RAM/port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for details.

One of the internal interrupts, ICI, OCI, TOI, CMI or SIO can generate an internal interrupt (IRQ_3). IRQ_3 function is just the same as $\overline{IRQ_1}$ or $\overline{IRQ_2}$ except the vector address. Fig. 14 shows the block diagram of the interrupt circuit.

● Mode Program (MP_0, MP_1)

These two pins decide the operation mode. Refer to "MODE SELECTION" for more details.

Table 1 Interrupt Vector Memory Map

Priority	Vector		Interrupt
	MSB	LSB	
Highest	FFFE	FFFF	$\overline{RES}$
	FFEE	FFEF	TRAP
	FFFC	FFFD	$\overline{NMI}$
	FFFA	FFFB	SWI (Software Interrupt)
	FFF8	FFF9	$\overline{IRQ_1}$
	FFF6	FFF7	ICI (Timer 1 Input Capture)
	FFF4	FFF5	OCI (Timer 1 Output Compare 1, 2)
	FFF2	FFF3	TOI (Timer 1 Overflow)
	FFEC	FFED	CMI (Timer 2 Counter Match)
	FFEA	FFEB	$\overline{IRQ_2}$
Lowest	FFF0	FFF1	SIO (RDRF+ORFE+TDRE)

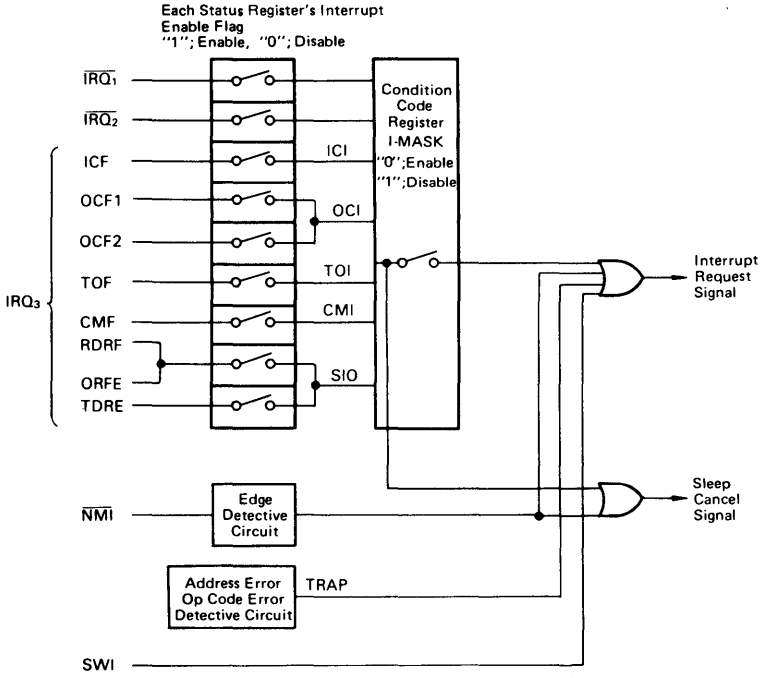


Figure 14 Interrupt Circuit Block Diagram

The following signal descriptions are applied only for expanded mode.

- **Read/Write (R/$\overline{W}$; P_{72})**

This signal, usually in read state ("High"), shows whether the MCU is in read ("High") or write ("Low") state. This can drive one TTL load and 30pF capacitance.

- **$\overline{RD}$, $\overline{WR}$ (P_{70}, P_{71})**

These outputs will turn "Low" when the CPU read/write operation is completed. This enables the CPU easy to access the peripheral LSI with $\overline{RD}$ and $\overline{WR}$ input pins. These pins can drive one TTL load and 30pF capacitance.

- **Load Instruction Register ($\overline{LIR}$; P_{73})**

This is output for the instruction opecode on data bus (active low). This pin can drive one TTL load and 30pF capacitance.

- **Memory Ready (MR; P_{52})**

This input is used to stretch the system clock's "High" period in order to access low-speed memories. During this signal being in "High", the system clock operates in normal sequence. But in "Low", the "High" period of the system clock will be stretched in integral multiples of the cycle time. This allows the CPU to interface with low-speed memories (See Fig. 2). Up to 9μs can be stretched.

During internal address access or nonvalid memory access, MR is prohibited internally to prevent decrease of operation speed. Even in the halt state, MR can also stretch "High" period of system clock to allow peripheral devices to access low-speed memories. As this pin is used also for P_{52}, an enable bit is provided at bit 2 of the RAM/port 5 control register at $0014. Refer to "RAM/PORT 5 CONTROL REGISTER" for more details.

- **Halt ($\overline{HALT}$; P_{53})**

This input is used to stop instruction execution or to release buses free. When this signal turns "Low", the CPU will be in the halt state after completing the current instruction. During the halt state, BA (P_{74}) is in "High", and an address bus, data bus, $\overline{RD}$, $\overline{WR}$ and R/$\overline{W}$ are high impedance. When an interrupt is requested in the halt state, the CPU responds to the interrupt request after the halt is cancelled.

(Note) When the CPU is interrupt wait state in WAI instruction execution, $\overline{HALT}$ should be held "High". If $\overline{HALT}$ turns "Low", the CPU may malfunction after releasing the halt state. Refer to "APPLICATION NOTES" for details.

- **Bus Available (BA; P_{74})**

This output is normally "Low" but "High" when the CPU accepts $\overline{HALT}$ and releases the buses. The HD6800 and HD6802 make BA "High" and release the buses at WAI execution, while the HD63701X0 doesn't make BA "High" under the same condition.

The following pin functions are applied only in EPROM mode. Refer to "THE EPROM PROGRAMMING" for details of EPROM mode.

- **Chip Enable ($\overline{CE}$; P_{57})**

This pin is input for programming and verifying the EPROM. When this pin is "Low" level, EPROM will be enable.

The EPROM can not be programmed or verified in "High" level.

- **Program Voltage/Output Enable ($V_{PP}/\overline{OE}$)**

This pin is used for program voltage and data output control in verification.

Data from Port 3 (EO_0 to EO_7) can be programmed into the EPROM when applying 21V±0.5V to V_{PP} and holding $\overline{CE}$ in "Low" level. The EPROM address is provided to Port 1 and Port 4 (EA_0 to EA_{11}). In verification, the EPROM data is output from Port 3 (EO_0 to EO_7) when this pin is "Low" level. In "High" level, Port 3 will be high-impedance. In MCU mode, this pin should be connected to V_{SS}.

■ PORT

The HD63701X0 has six 8-bit ports and a 5-bit port. Table 2 gives the address of ports and the data direction register and Fig. 15 the block diagrams of each port.

Table 2 Port and Data Direction Register Address

Port	Port Address	Data Direction Register
Port 1	$0002	–
Port 2	$0003	$0001
Port 3	$0006	$0004
Port 4	$0007	–
Port 5	$0015	–
Port 6	$0017	$0016
Port 7	$0018	–

- **Port 1**

In MCU mode, port 1 is used for an 8-bit output port. In mode 3, port 1 is high impedance during reset, and keeps the state even after reset is released. When the CPU writes on the port 1 data register, the written data will appear at Port 1. Once port 1 gets in the output state, it operates as an output till reset. The CPU can read the Port 1 data register for the bit manipulation instruction.

In mode 1 and 2, port 1 is used for lower address buses. This port can drive one TTL load and 90pF capacitance.

In EPROM mode, port 1 is lower address bus (EA_0 to EA_7) for the EPROM.

- **Port 2**

An 8-bit input/output port. Its I/O state depends on the data direction register (DDR) of port 2 which provides two bits; bit 0 decides the I/O direction of P_{20} and bit 1 the I/O direction of P_{21} to P_{27} ("0" for input, "1" for output).

Port 2 is also used for the timers and the SCI. When used for the timers and the SCI, P_{21} to P_{27} are decided I/O regardless of the DDR (except for P_{20}).

Port 2 Data Direction Register

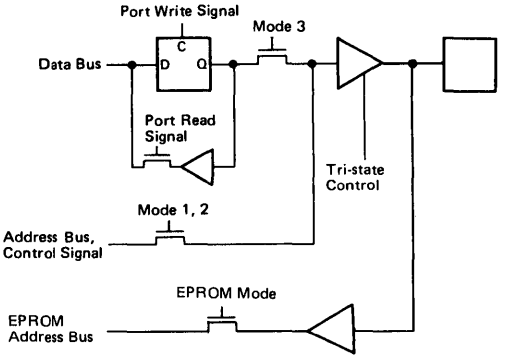
7	6	5	4	3	2	1	0	
–	–	–	–	–	–	DDR 1~7	DDR 0	$0001

The DDR of port 2 is cleared at reset and port 2 is configured as an input. This port can drive one TTL and 30pF. In addition, it is capable of sinking 1mA current at Vout=1.5V to drive directly the base of Darlington transistors.

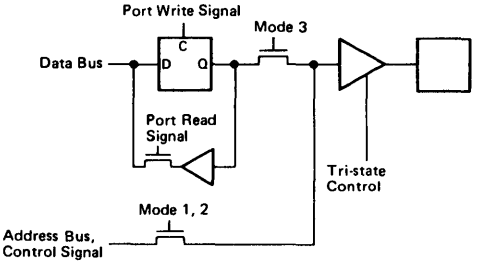
- **Port 3**

An 8-bit I/O port. I/O state depends on the DDR of Port 3 which has only one bit ("0" for input and "1" for output). It is cleared at reset. In mode 1 and 2, port 3 is used for data bus. This port can drive one TTL load and 90pF capacitance.

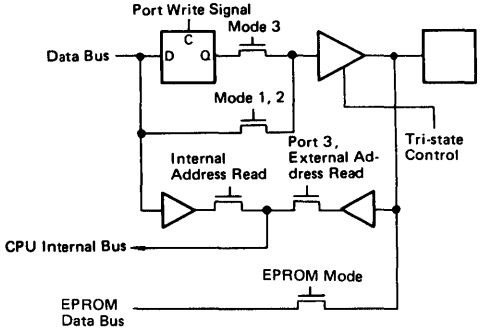
Port 3 is used for data bus (EO_0 to EO_7) of EPROM in EPROM mode. In this case, I/O state of Port 3 is selected by $\overline{OE}$ but not the DDR.



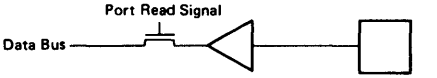
Port 1, Port 4 (Bit 0 to 3)



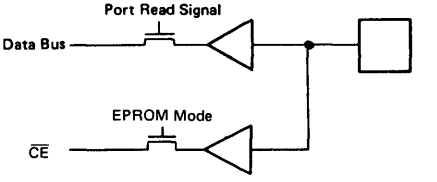
Port 4 (Bit 4 to 7), Port 7



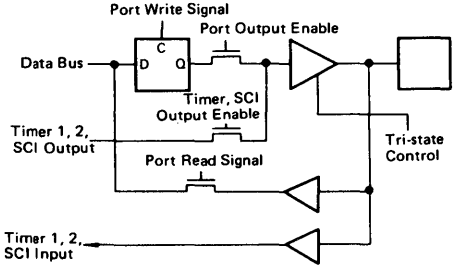
Port 3



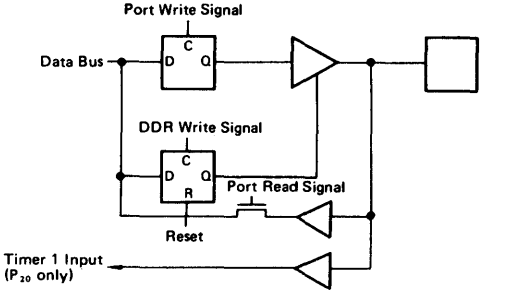
Port 5



Port 5 (Bit 7)



Port 2



Port 6, Port 2 (Bit 0)

Figure 15 Port Block Diagram

Port 3 Data Direction Register

7	6	5	4	3	2	1	0	
–	–	–	–	–	–	–	Port 3 DDR	$0004

- **Port 4**

In MCU mode, port 4 is used for an 8-bit output port like Port 1. In mode 1 and 2, it is used for upper address bus.

In EPROM mode, P_{40} to P_{43} are used for upper address bus (EA_8 to EA_{11}) of EPROM.

- **Port 5**

An 8-bit input port. The lower 4 bits are used for interrupt, MR, $\overline{HALT}$, and P_{57} is $\overline{CE}$ for the EPROM control.

- **Port 6**

An 8-bit I/O port. This port is programmable as either input or output under software control of the corresponding the DDR ("0" for input, "1" for output). This port can drive one TTL load and 30pF. The DDR of port 6 is cleared at reset. In addition, it is capable to sinking 1mA current at Vout=1.5V to drive directly the base of Darlington transistors.

- **Port 7**

A 5-bit output port. In mode 3, port 7 is high impedance during reset and keeps the state even after reset is released. When the CPU writes on the port 7 data register, the written data will appear at Port 7. Once port 7 gets in the output state, it operates as an output till reset. The CPU can read the data register for the bit manipulation instruction. In this case b_7 to b_5 are "1".

In mode 1 and 2, port 7 is used for control signals ($\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$ and BA). This port can drive one TTL load and 30pF.

■ RAM/PORT 5 CONTROL REGISTER

The control register located at $0014 controls on-chip RAM and port 5.

RAM/Port 5 Control Register

7	6	5	4	3	2	1	0	
STBY PWR	RAME	–	–	HLTE	MRE	IRQ_2 E	IRQ_1 E	$0014

Bit 0, Bit 1 $\overline{IRQ_1}$, $\overline{IRQ_2}$ Enable Bit (IRQ_1E, IRQ_2E)

When using P_{50} and P_{51} for interrupt pins, write "1" in these bits. When "0", the CPU doesn't accept an external interrupt or a sleep cancellation by the external interrupt. These bits are cleared at reset.

Bit 2 Memory Ready Enable Bit (MRE)

When using P_{52} for an input for Memory Ready signal, write "1" in this bit. When "0", the memory ready function is prohibited and P_{52} is for port. In mode 3, the memory ready function is prohibited regardless of the value of this bit. This bit is set at reset.

Bit 3 Halt Enable Bit (HLTE)

When using P_{53} for an input for Halt signal, write "1" in this bit. When "0", the halt function is prohibited. In mode 3, the halt function is prohibited regardless of the value of this bit. This bit is set at reset.

(Note) When using P_{52} and P_{53} for port in mode 1 and 2, MRE and MLTE must be cleared after reset. If P_{52} or P_{53} turns "Low" before MRE and HLTE are cleared, the memory ready function or the halt function will not be prohibited.

Bit 4, Bit 5 Not Used.

Bit 6 RAM Enable (RAME)

The RAM is controlled by this bit. It is set at reset and the RAM is enabled. This bit is programmable by software. When the RAM is disabled (=logic "0"), the CPU can access an external memory. This bit should be cleared at the beginning of standby mode to protect the RAM data.

Bit 7 Standby Power Bit (STBY PWR)

This bit is cleared whenever V_{CC} decreases below V_{RAM} (min). This is a read/write status bit by software. If this bit is set before standby mode, it indicates that V_{CC} is applied and the RAM is valid.

■ MODE SELECTION

The HD63701X0 provides two fundamental modes, MCU mode and EPROM mode. MCU mode is grouped into three; two expanded modes (mode 1, mode 2) and a single chip mode (mode 3).

These operating modes are selectable by mode program pins, MP_0 and MP_1, and standby pin, $\overline{STBY}$ as shown in Table 3.

- **Mode 1 (Expanded Mode)**

In this mode, Port 3 is data bus, Port 1 is lower address bus and Port 4 is upper address bus to interface with the HMCS6800 buses. Port 7 is used for control signal such as R/$\overline{W}$. In mode 1, the EPROM is disable and external address space are expandable up to 65k bytes (refer to Fig. 16).

- **Mode 2 (Expanded Mode)**

This mode is also expanded mode. But in mode 2, address space is expandable up to 61k bytes and the EPROM is enable (refer to Fig. 17).

- **Mode 3 (Single-chip Mode)**

In this mode, all ports are available (refer to Fig. 18).

- **EPROM Mode**

In this mode, the EPROM can be programmed. Refer to "PROGRAMMING THE EPROM" for details.

- **Mode and Ports**

Table 4 shows the MCU signals in each mode.

Table 3 Mode Selection

Mode		MP1	MP0	$\overline{STBY}$	EPROM	RAM	Interrupt Vector	Operation Mode
MCU Mode	1	"L"	"H"	*	E	I (Note 1)	E	Expanded Mode
	2	"H"	"L"	*	I	I (Note 1)	I	Expanded Mode
	3	"H"	"H"	*	I	I	I	Single-chip Mode
EPROM Mode		"L"	"L"	"L"	I	*	*	EPROM Programming Mode

"L"=Logic "0", "H"=Logic "1", I; Internal, E; External, *; Don't care
(Note 1) The RAM address area will be external by clearing RAME bit at $0014.

Table 4 MCU Signals in Each Mode

Port \ Mode	MCU Mode			EPROM Mode
	Mode 1	Mode 2	Mode 3	
Port 1	Address Bus ($A_0 \sim A_7$)	Address Bus ($A_0 \sim A_7$)	Output Port	Address Bus ($EA_0 \sim EA_7$)
Port 2	I/O Port	I/O Port	I/O Port	No use (Note 3)
Port 3	Data Bus ($D_0 \sim D_7$)	Data Bus ($D_0 \sim D_7$)	I/O Port	Data Bus ($EO_0 \sim EO_7$)
Port 4	Address Bus ($A_8 \sim A_{15}$)	Address Bus ($A_8 \sim A_{15}$)	Output Port	Address Bus ($EA_8 \sim EA_{11}$) (Note 1)
Port 5	Input Port	Input Port	Input Port	$\overline{CE}$ (P_{57}) (Note 2)
Port 6	I/O Port	I/O Port	I/O Port	No use (Note 3)
Port 7	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	$\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$, BA	Output Port	No use (Note 3)

(Note 1) Use only 4 pins P_{40} to P_{43}. P_{44} to P_{47} are not used.
(Note 2) 7 pins P_{50} to P_{56} are not used.
(Note 3) Unused ports should be connected to V_{SS}.

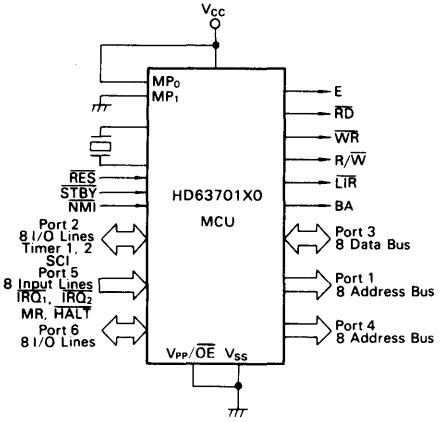


Figure 16 MCU Mode; Mode 1

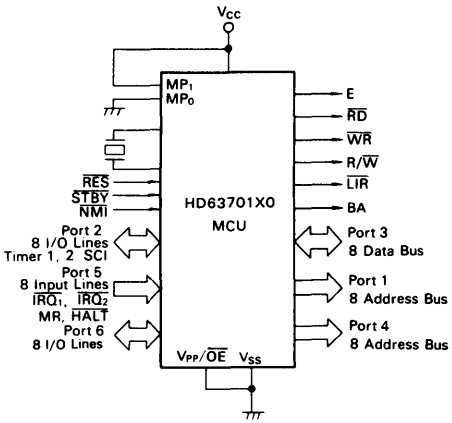


Figure 17 MCU Mode; Mode 2

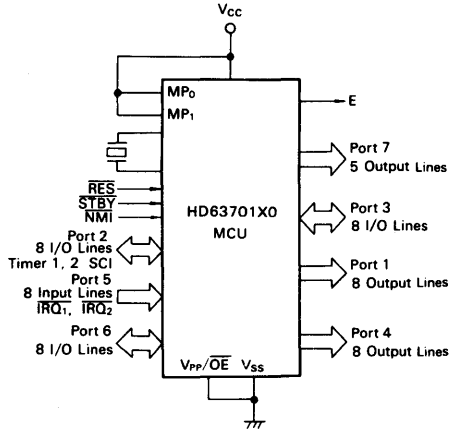


Figure 18 MCU Mode; Mode 3

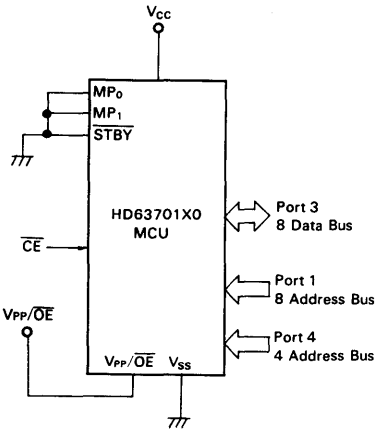


Figure 19 EPROM Mode

■ MEMORY MAP

The MCU has ability to access a 65k byte memory space depending on the operating mode. A memory map for each operating mode is shown in Fig. 20. The first 32 locations of each map are reserved for the MCU's internal register area, as shown in Table 5.

Table 5 Internal Register

Address	Registers	R/W***	Initialize at RESET
00	–	–	–
01	Port 2 Data Direction Register	W	$FC
02*	Port 1	R/W	Undefined
03	Port 2	R/W	Undefined
04*	Port 3 Data Direction Register	W	$FE
05	–	–	–
06*	Port 3	R/W	Undefined
07*	Port 4	R/W	Undefined
08	Timer Control/Status Register 1	R/W	$00
09	Free Running Counter ("High")	R/W	$00
0A	Free Running Counter ("Low")	R/W	$00
0B	Output Compare Register 1 ("High")	R/W	$FF
0C	Output Compare Register 1 ("Low")	R/W	$FF
0D	Input Capture Register ("High")	R	$00
0E	Input Capture Register ("Low")	R	$00
0F	Timer Control/Status Register 2	R/W	$10
10	Rate, Mode Control Register	R/W	$00
11	Tx/Rx Control Status Register	R/W	$20
12	Receive Data Register	R	$00
13	Transmit Data Register	W	$00
14	RAM/Port 5 Control Register	R/W	$7C or $FC
15	Port 5	R	–
16	Port 6 Data Direction Register	W	$00
17	Port 6	R/W	Undefined
18*	Port 7	R/W	Undefined
19	Output Compare Register 2 ("High")	R/W	$FF
1A	Output Compare Register 2 ("Low")	R/W	$FF
1B	Timer Control/Status Register 3	R/W	$20
1C	Time Constant Register	W	$FF
1D	Timer 2 Up Counter	R/W	$00
1E	–	–	–
1F**	Test Register	–	–

* External Address in Mode 1, 2.
** Test Register. Do not access to this register.
*** R : Read Only Register
W : Write Only Register
R/W : Read/Write Register

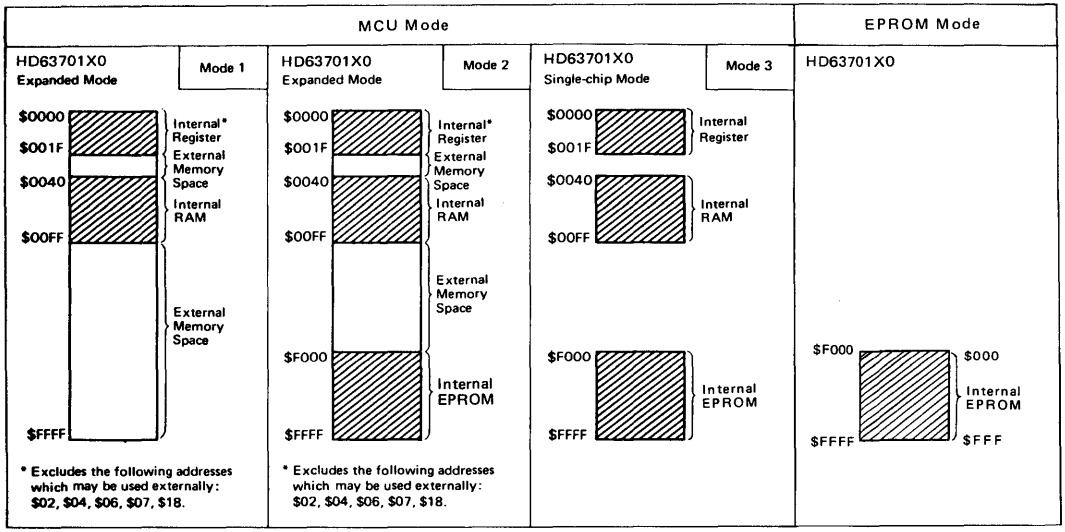


Figure 20 HD63701X0 Memory Map

■ THE EPROM PROGRAMMING

The HD63701X0 does not operate as the MCU in EPROM mode, which allows to be programmable as equivalent EPROM 2732A type. When three pins, MP_0, MP_1 and $\overline{STBY}$ should be held low, the MCU will be in EPROM mode as shown in Table 3. In this mode, P_{30} to P_{37} are used for data bus, P_{10} to P_{17} and P_{40} to P_{43} for address bus, and P_{57} for $\overline{CE}$ input shown Fig. 19. Refer to "APPLICATION NOTES" for the EPROM.

● Programming/Verification

When $\overline{CE}$ pin is held low after the program voltage (V_{PP}) is applied to $V_{PP}/\overline{OE}$ pin, the data byte can be applied to Port 3. When $V_{PP}/\overline{OE}$ pin and $\overline{CE}$ pin are held low after programming, the programmed data is output from Port 3 and user can verify the data. I/O timing of these signals are referred to Fig. 12.

When $\overline{CE}$ pin is returned to high, Port 3 will be tri-state and EPROM programming/verification will be inhibited.

Table 6 shows the condition of the each pin is EPROM mode. Unused pins should be connected to GND in EPROM mode.

● Erasure (applied only for the ceramic package with a window)

Erasure of EPROM begins to occur when the LSI is exposed to ultraviolet light (wavelength: 2537Å, an integrated does of at least: 1.5W·sec/cm). Exposing the LSI to an ultraviolet lamp of 1,200 μW/cm^2 rating for 20 to 30 minutes, at a distance of about 1 inch, should be sufficient.

(Note) If the window is stained, erasure time will be extended. Remove stains from the window with a solvent which has no influence on the package like alcohol. Don't rub the window hard but wipe out softly.

(Note for the plastic package)
It is impossible to erase the programmed EPROM of the plastic molded HD63701X0. Refer to "APPLICATION NOTES" for the plastic package.

Table 6 Pin Condition in EPROM mode

Mode \ Pin No.	V_{CC}	V_{SS}	$V_{PP}/\overline{OE}$	$\overline{CE}$	P_{30} to P_{37}	P_{10} to P_{17} P_{40} to P_{43}	MP_0, MP_1, $\overline{STBY}$	Other pins
	33	1	42	24	51 to 58	43 to 50 38 to 41	4, 5, 7	
Programming	+5	GND	V_{PP}	"L"	Data input	Address input	"L"	GND
Verification	+5	GND	"L"	"L"	Data output	Address input	"L"	GND
Inhibition of programming/verification	+5	GND	Don't care	"H"	High impedance	Don't care	"L"	GND

"H": V_{IH} level, "L": V_{IL} level

■ TIMER 1

The HD63701X0 has a 16-bit programmable timer which can simultaneously measure an input waveform and generate two independent output waveforms. The pulse width can vary from several microseconds to many seconds.

Timer 1 is configurated as follows (refer to Fig. 22).

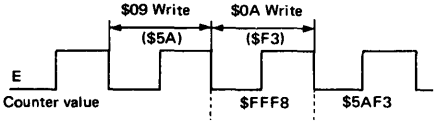
- Control/Status Register 1 (8 bit)
- Control/Status Register 2 (7 bit)
- Free Running Counter (16 bit)
- Output Compare Register 1 (16 bit)
- Output Compare Register 2 (16 bit)
- Input Capture Register (16 bit)

● Free-Running Counter (FRC) ($0009 : 000A)

The key timer element is a 16-bit Free-Running Counter which is incremented by system clock (E). The counter value is readable by software without affecting the FRC. It is cleared by reset.

A write to the high byte of the FRC ($09) will preset the high and low byte of the FRC to $FFF8 . A continuous write to the high and low byte FRC, however, will set them to the write data.

The FRC write timing will be as follows when double store instructions (STD, STX etc.) execute.



In the case of a write ($5AF3) to the FRC.

Figure 21 Counter Write Timing

● Output Compare Register (OCR) ($000B, $000C; OCR1) ($0019, $001A; OCR2)

The Output Compare Register is a 16-bit read/write register used to control an output waveform. It is always compared with the FRC on each E-cycle.

When a match is found, Output Compare Flag (OCF) in the Timer Control/Status Register (TCSR) is set. If an output enable bit (OE) in the TCSR2 is "1", an output level bit (OLVL) in the TCSR will appear at Port 21 (Tout 1) or Port 25 (Tout 2).

The OCR and OLVL can then be changed for the next compare. The OCR is set to $FFFF at reset. The compare function is inhibited for a cycle after a write to the OCR or to the high byte of the FRC. This is to set the 16-bit value valid in the register for compare. In addition, it is because $FFF8 is set at the next cycle of a write to the high byte of the FRC.

* For a write to the FRC or the OCR, 2-byte transfer instruction (such as STX etc.) should be used.

● Input Capture Register (ICR) ($000D : 000E)

The Input Capture Register is a 16-bit read only register used to store the FRC when an external input transition occures defined by input edge bit (IEDG) in the TCSR1.

In order to input the external signal to the edge detective circuit, Port 20 should be configured as an input. When an input capture occures at the next cycle of a read the high-byte of the ICR, the input capture will delay one cycle. In order to ensure the input capture, a read to the ICR needs 2-byte transfer instruction, and the input pulse width should be at least 2 system cycles. This register is cleared ($0000) at reset.

● Timer Control/Status Register 1 (TCSR1) ($0008)

The Timer Control/Status Register 1 is an 8-bit register of which all bits are readable while the lower 5 bits can be written. The upper 3 bits indicate the following timer's status.

Bit 5 The FCR has overflowed. (TOF).
Bit 6 A match has been found between the FCR and the OCR 1 (OCF1).
Bit 7 A level transition of the timer input has been detected (ICF).

The followings are each bit descriptions.

Timer Control/Status Register 1

7	6	5	4	3	2	1	0	
ICF	OCF1	TOF	EICI	EOCI1	ETOI	IEDG	OLVL1	$0008

Bit 0 OLVL1 Output Level 1
When a match is found between the FCR and the OCR1, OLVL1 will appear at Port 21 if OE1, bit 0 of the TCSR2, is set.

Bit 1 IEDG Input Edge
This bit controls which level transition will trigger the FCR transfer to the ICR. For this function, the DDR corresponding to Port 20 should be cleared.
IEDG=0, transferred on a negative edge
IEDG=1, transferred on a positive edge

Bit 2 ETOI Enable Timer Overflow Interrupt
When this bit is set, an internal interrupt (IRQ_3) is enabled for TOI. When cleared, the interrupt is inhibited.

Bit 3 EOCI1 Enable Output Compare Interrupt 1
When this bit is set, an internal interrupt (IRQ_3) is enabled for OCI1. When cleared, the interrupt is inhibited.

Bit 4 EICI Enable Input Capture Interrupt
When this bit is set, an internal interrupt (IRQ_3) is enabled for ICI. When cleared, the interrupt is inhibited.

Bit 5 TOF Timer Overflow Flag
This read only bit is set when the FCR contains all 1's. It is cleared by reading the TCSR1 followed by the FCR's high byte ($0009).

Bit 6 OCF1 Output Compare Flag 1
This read only bit is set when a match is found between the OCR1 and the FRC. It is cleared by writing to the OCR1 ($000B or $000C) followed by reading the TCSR1 or TCSR2.

Bit 7 ICF Input Capture Flag
This read only bit is set to indicate a level transition defined by IEDG. It is cleared by reading the high byte ($000D) of the ICR followed by the TCSR1 or TCSR2.

● Timer Control/Status Register 2 (TCSR2) ($000F)

The Timer Control/Status Register 2 is a 7-bit register. All bits are readable while the lower 4 bits can be written. The upper 3 bits indicate the following timer's status.

Bit 5 A match has been found between the FRC and the OCR2 (OCF2).
Bit 6 The same flag as the OCF1 of the TCSR1.
Bit 7 The same flag as the ICF of the TCSR1.

The followings are each bit descriptions.

Timer Control/Status Register 2

7	6	5	4	3	2	1	0	
ICF	OCF1	OCF2	–	EOCI2	OLVL2	OE2	OE1	$000F

Bit 0 OE1 Output Enable 1
If this bit is set, the OLVL1 will appear at Port 21 when a match is found between the FCR and the OCR1. When it is cleared, Port 21 will be I/O port. When set, it will be an output of OLVL1 automatically.

Bit 1 OE2 Output Enable 2
If this bit is set, the OLVL2 will appear at Port 25 when a match between the FCR and the OCR2. When this bit is cleared, Port 25 will be I/O port. When set, it will be an output of OLVL2 automatically.

Bit 2 OLVL2 Output Level 2
OLVL2 is transferred to Port 25 when a match is found between the FCR and the OCR2. If OE2, bit 5 of the TCSR2, is set, OLVL2 will appear at Port 25.

Bit 3 EOCI2 Enable Output Compare Interrupt 2
When this bit is set, an internal interrupt (IRQ_3) is enabled for OCI2. When cleared, the interrupt is inhibited.

Bit 4 Not Used

Bit 5 OCF2 Output Compare Flag 2
This read-only bit is set when a match is found between the FCR and the OCR2. It is cleared by writing to the OCR2 ($0019 or $001A) followed by reading the TCSR2.

Bit 6 OCF1 Output Compare Flag 1

Bit 7 ICF Input Capture Flag
OCF1 and ICF addresses are partially decoded. CPU read of the TCSR1/TCSR2 makes it possible to read OCF1 and ICF into bit 6 and bit 7.
Both the TCSR1 and TCSR2 will be cleared by reset.

(Note) If OE1 or OE2 is set before the first output compare match is found after reset, Port 21 and Port 25 will output "0" respectively.

(Note) Because the set condition of ICF precedes its reset condition, ICF is not cleared when the set condition and the reset condition occur simultaneously. The same phenomenon applies to OCF1, OCF2 or TOF respectively.

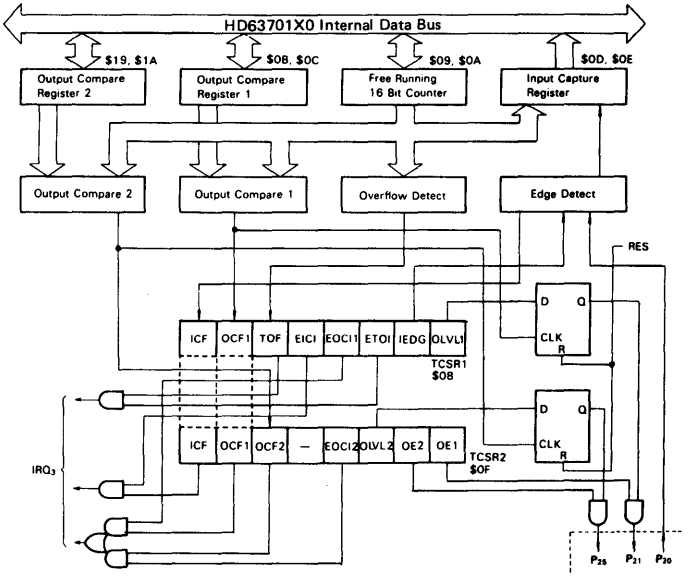


Figure 22 Timer 1 Block Diagram

■ TIMER 2

In addition to the timer 1, the HD63701X0 provides an 8-bit reloadable timer, which is capable of counting the external event. This timer 2 contains a timer output, so the MCU can generate three independent waveforms. (Refer to Fig. 23.)

The timer 2 is configured as follows:
- Control/Status Register 3 (7 bit)
- 8-bit Up Counter
- Time Constant Register (8 bit)

● Timer 2 Up Counter (T2CNT) ($001D)

This is an 8-bit up counter which is incremented by the clock controlled by CKS0 and CKS1 of the TCSR3. The T2CNT is always readable without affecting itself. In addition, any value can be written to the T2CNT by software even during counting.

The counter is cleared when a match is found between the T2CNT and the TCONR or by reset.

A write to the T2CNT at the clear cycle does not reset it but put the data to it.

● Time Constant Register (TCONR) ($001C)

The Timer Constant Register is an 8-bit write only register. It is always compared with the T2CNT.

When a match has been found, counter match flag (CMF) of the Timer Control/Status Register 3 (TCSR3) is set and the value selected by TOS0 and TOS1 of the TCSR3 will appear at Port 26. When CMF is set, the FCR will be cleared simultaneously and then a counting starts from $00. This enables regular interrupts and waveform outputs without any software support. The TCONR is set to "$FF" by reset.

● Timer Control/Status Register 3 (TCSR3) ($001B)

The Timer Control/Status Register 3 is a 7-bit register. All bits are readable while 6 bits except for CMF can be written.

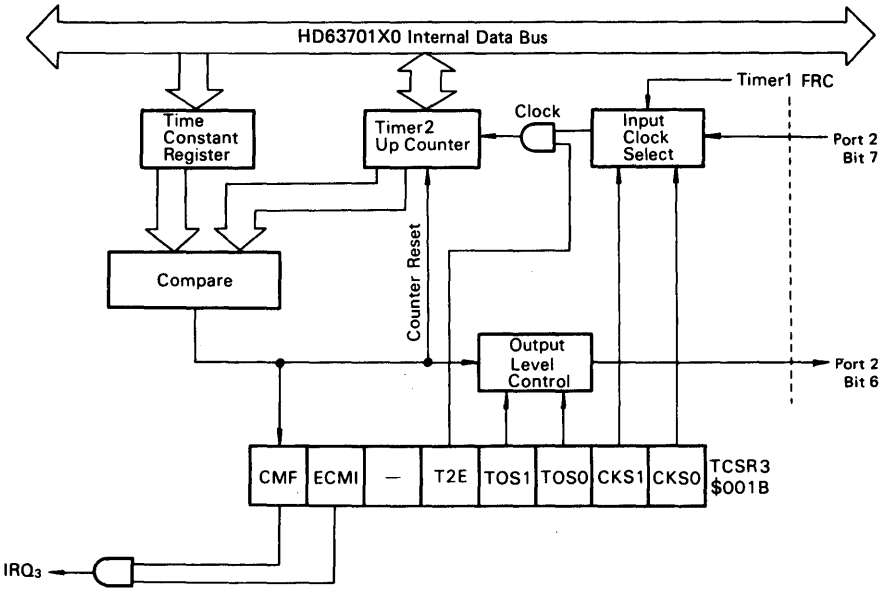


Figure 23 Timer 2 Block Diagram

The followings are each bit descriptions.

Timer Control/Status Register 3

7	6	5	4	3	2	1	0	
CMF	ECMI	—	T2E	TOS1	TOS0	CKS1	CKS0	$001B

Bit 0 CKS0 Input Clock Select 0
Bit 1 CKS1 Input Clock Select 1
An input clock to the T2CNT is selected by these bits as shown in Table 7. When an external clock is selected, Port 27 will be an input automatically. The positive edge of the external clock increments the T2CNT. The maximum external clock is half of the system clock frequency.

Table 7 Input Clock Select

CKS1	CKS0	Input Clock to the Counter
0	0	E clock
0	1	E clock/8*
1	0	E clock/128*
1	1	External clock

* These clocks come from the FRC of the timer 1. If one of these clocks is selected as an input clock to the up counter, a write to the FRC of the timer 1 should be inhibited.

Bit 2 TOS0 Timer Output Select 0
Bit 3 TOS1 Timer Output Select 1
When a match is found between the T2CNT and the TCONR, timer 2 output selected by these bits shown in Table 8 will appear at Port 26. When both TOS0 and TOS1 are cleared, Port 26 will be an I/O port.

Table 8 Timer 2 Output Select

TOS1	TOS0	Timer Output
0	0	Timer Output Inhibited
0	1	Toggle Output*
1	0	Output "0"
1	1	Output "1"

* When a match is found between the T2CNT and the TCONR, timer 2 output level is reversed. This leads to production of a square wave with 50% duty to the external without any software support.

Bit 4 T2E Timer 2 Enable Bit
When this bit is cleared, the T2CNT will stop. When set, a clock selected by CKS1 and CKS0 (Table 7) provides to the T2CNT.
(Note) P_{26} is "0" when T2E is cleared and P_{26} is configured as an output by TOS1 or TOS0. It also is "0" when T2E is set and P_{26} is configured as an output before the first counter match.
Bit 5 Not Used
Bit 6 ECMI Enable Counter Match Interrupt

When this bit is set, an internal interrupt (IRQ_3) is enabled for CMI. When cleared, the interrupt is inhibited.

Bit 7 CMF Counter Match Flag

This read only bit is set when a match is found between the T2CNT and the TCONR. It is cleared by writing "0". (It cannot be written "1" by software).

Each bit of the TCSR3 is cleared by reset.

■ SERIAL COMMUNICATION INTERFACE (SCI)

The HD63701X0 SCI provides two operation modes; one is an asynchronous mode by the NRZ format and the other is a clocked synchronous mode to transfer data synchronizing with the serial clock.

The serial interface is configured as follows:

- Transmit/Receive Control and Status Register (TRCSR)
- Rate/Mode Control Register (RMCR)
- Receive Data Register (RDR)
- Receive Data Shift Register (RDSR)
- Transmit Data Register (TDR)
- Transmit Data Shift Register (TDSR)

The SCI is initialized by software. The procedure is usually as follows:

1) Write a operation mode into each corresponding control bit of the RMCR.
2) Write a operation mode into each corresponding control bit of the TRCSR.

When setting the baud rate and operation mode, TE and RE should be "0". When TE and RE is set again, more than 1 bit cycle of the current baud rate is necessary. If set in less than 1 bit cycle, the SCI cannot be initialized occasionally.

● Asynchronous Mode

An asynchronous mode contains the following two data formats:

1 Start Bit + 8 Bit Data + 1 Stop Bit; 8 Bit Data Format
1 Start Bit + 9 Bit Data + 1 Stop Bit; 9 Bit Data Format

In 9 Bit Data Format, if the 9th bit is "1", the format of 1 Start Bit + 8 Bit Data + 2 Stop Bit

The SCI is initialized by writing desirable control bytes to the RMCR and then to the TRCSR.

The transmit operation is enabled by TE in the TRCSR. When TE is set, the output of the TDSR is connected to P_{24} which will be configured as an output regardless of the DDR, and then the serial output is initiated by transmitting to a 10-bit preamble of "1" in the 8 Bit Data Format or an 11-bit preamble of "1" in the 9 Bit Data Format. Following the preamble, the internal synchronization is established and the transmitter section is ready for operation.

At this point one of two situation exist:

1) If the TDR is empty (TDRE=1), a continuous string of ones will be sent indicating an idle line.
2) If a byte has been written to the TDR (TDRE=0), it is transferred to the TDSR, TDRE will be set and transmission will begin.

During the transfer itself, the start bit (0) is first transmitted. Then the 8 data bits or the 9 data bits (beginning with bit 0) followed by the stop bit (1) are transmitted. When the TDR has been emptied, TDRE is set.

If the MCU fails to respond to the flag within the proper time, (TDRE is still set when the next normal transfer from the TDR to the TDSR should occure) then a "1" will be sent (instead of a "0") at start bit time, followed by more 1's until more data is supplied to the TDR. No 0's will be sent while TDRE remains as "1".

The receive operation is enabled by RE which configures P_{23}. The receive operation is controlled by the contents of the TRCSR and the RMCR. The receiver bit interval is divided into 8 sub-intervals for internal synchronization. The received bit stream is synchronized by the first "0" (space) encountered. The approximate center of each bit time is strobed during the next 10 bits.

If the tenth bit is not a "1" (stop bit), a framing error is assumed and ORFE is set. If the tenth bit is a "1", the data is transferred to the RDR and interrupt flag RDRF is set. If RDRF is still set at the next tenth bit time, ORFE will be set, indicating an over-run has occurred. When the CPU responds to either flag

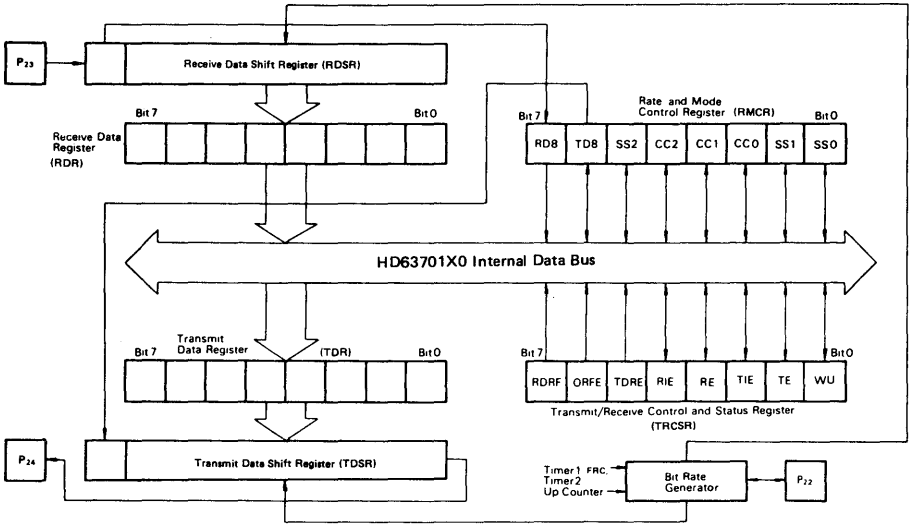


Figure 24 Serial Communication Interface Block Diagram

(RDRF or ORFE) by reading the TRCSR followed by reading the RDR, RDRF (or ORFE) will be cleared.

(Note) Clock Source in Asynchronous Mode

When using an internal clock for the SCI, the following requirements are applicable:

- Set CC1 and CC0 to "1" and "0" respectively.
- A clock is generated regardless of the value of TE, RE.
- The maximum clock rate is E÷16.
- The output clock is the same as the bit rate.

When using an external clock for the SCI, the following requirements are applicable:

- Set CC1 and CC0 in the RMCR to "1" and "1" respectively.
- The external clock should be set 16 times the desired baud rate.
- the maximum clock frequency is the same as the system clock.

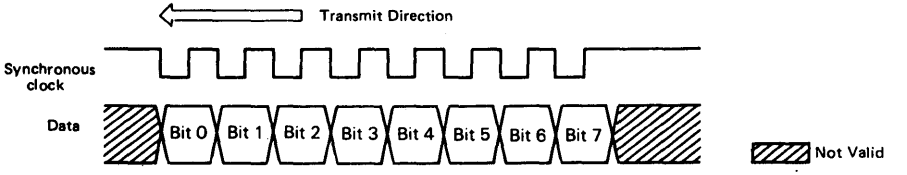
• Clocked Synchronous Mode

In the clocked synchronous mode, the transmit operation is synchronized with the clock pulse. In the clocked synchronous mode an SCI clock I/O pin is only P_{22}, so the receive and transmit operation cannot be simultaneously enabled. Therefore, TE and RE should not be set simultaneously. Fig. 25 gives a synchronous clock and a data format in the clocked synchronous mode.

The transmit operation is enabled by TE in the TRCSR. P_{24} is configured as an output regardless of the value of the corresponding DDR.

Both the RMCR and TRCSR should be set in the desirable operating conditions for data transmit.

If the user wishes to provide an external clock, the data bits (beginning with bit 0) are transmitted from P_{24}, synchronizing with 8 clock pulses supplied to P_{22}, when TDRE is "0". TDRE is set when the TDSR is "empty". More the 9th clock pulse is ignored.



- Transmit data is sent between the negative edge of a synchronous clock and the next negative edge.
- Receive data is latched at the positive edge.

Figure 25 Clocked Synchronous Mode Format

The receive operation is enabled by RE. P_{22} is configured as an input for the 8 bit external clock and P_{23} is configured as an input for the receive data. The operating mode of data receive is decided by the TRCSR and the RMCR.

If the external clock is provided, RE should be set when P_{22} is "High". The receive data is transferred to the RDSR by this clock, and RDRF is set. More the 9th clock pulse are ignored. When RDRF is cleared by reading the RDR, the MCU starts receiving the next data.

RDRF, therefore, should be cleared with P_{22} "High". When the first byte data is received, RDRF is set. After the second byte, the receive operation is enabled by clearing RDRF.

• Transmit/Receive Control and Status Register (TRCSR) ($0011)

The TRCSR is an 8 bit register which is readable. Bits 0 to 4 are also writable. This register is initialized to $20 by reset. Each bit functions as follows.

Transmit/Receive Control Status Register

7	6	5	4	3	2	1	0	
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$0011

Bit 0 WU Wake-up

In a typical multi-processor configuration, the software protocol will usually identify the address at the beginning of the message. In order to permit uninterested MCU's to ignore the remaining message, a wake-up function is available. By this, uninterested MCU can inhibit all further receive processing till the next message starts.

Then wake-up function is triggered by consecutive 1's with 1 frame length (10 bits for the 8-bit data format, or 11 bits for the 9-bit data format). The software protocol should provide the idle time between messages.

By setting this bit, the MCU stops data receive till the next message. The receive of consecutive "1" with one frame length wakes up and clears this bit and then the MCU restarts the receive operation. However, the RE flag should be set before setting this bit. In the clocked synchronous mode WU is not available, so this bit should not be set.

Bit 1 TE Transmit Enable

When this bit is set, transmit data will appear at P_{24} after one frame preamble in asynchronous mode, while in clocked synchronous mode appear immediately. This is executed regardless of the value of the corresponding DDR. When TE is cleared, the serial I/O doesn't affect P_{24}.

Bit 2 TIE Transmit Interrupt Enable

When this bit is set, an internal interrupt (IRQ_3) is enabled when TDRE (bit 5) is set. When cleared, the interrupt is inhibited.

Bit 3 RE Receive Enable

When set, P_{23} is configured as an input for the receive operation regardless of the value of the DDR. When RE is cleared, the serial I/O doesn't affect P_{23}.

Bit 4 RIE Receive Interrupt Enable

When this bit is set, an internal interrupt, IRQ_3 is enabled when RDRF (bit 7) or ORFE (bit 6) is set. When cleared, the interrupt is inhibited.

Bit 5 TDRE Transmit Data Register Empty

TDRE is set when the TDR is transferred to the TDSR in the asynchronous mode, while it is set when the TDSR is "empty" in clocked synchronous mode. This bit is cleared by reading the TRCSR and writing the new transmit data to the TDR. TDRE is set by reset.

(Note) TE should be set before clearing TDRE.

Bit 6 ORFE Overrun Framing Error

ORFE is set when an overrun or a framing error is occured (during data receive only). An overrun error occurs when a new receive data is ready to be transferred to the RDR with RDRF still set. A framing error occurs when a stop bit is "0". But in clocked synchronous mode, this bit is not affected. This bit is cleared when reading the TRCSR, then the RDR, or by reset.

Bit 7 RDRF Receive Data Register Full

RDRF is set when the RDSR is transferred to the RDR. Cleared when reading the TRCSR, then the RDR, or by reset.

(Note) When a few bits are set between bit 5 to bit 7 in the TRCSR, a read of the TRCSR is sufficient for clearing those bits. It is not necessary to read the TRCSR everytime to clear each bit.

- **Rate/Mode Control Register (RMCR)**

The RMCR controls the followings:

- Baud Rate
- Data Format
- Clock Source
- P_{22} Function

In addition, if 9-bit data format is set in the asynchronous mode, the 9th bit is put in this register. All bits are readable and writable except bit 7 (read only). This register is cleared by reset.

These bits select the baud rate when using the internal clock. Table 9 lists the available bit times and baud rates. The timer 1's FRC (SS2=0) and the timer 2's up counter (SS2=1) provide the internal clock for the SCI. When the source of the SCI internal clock is the timer 2's up counter, the desired baud rates may be selected by the TCONR shown in Table 10.

(Note) When operating the SCI with internal clock, do not write to the counter which is the source of the SCI clock.

Rate/Mode Control Register

7	6	5	4	3	2	1	0	
RD8	TD8	SS2	CC2	CC1	CC0	SS1	SS0	$0010

Bit 0 SS0
Bit 1 SS1 } Speed Select
Bit 5 SS2

Bit 2 CC0
Bit 3 CC1 } Clock Control/Format Select*
Bit 4 CC2

These bits select the data format and the clock source (refer to Table 11).

* CC0, CC1 and CC2 are cleared and the MCU will be in the clocked synchronous mode (the external clock operation) by reset. Then P_{22} is forced to be configured as an input for the clock. If using P_{22} for an output, the DDR of port 2 should be set to "1" and CC1, CC0 must be set to "01".

Table 9 SCI Bit Times and Rates

(1) Asynchronous Mode

			XTAL	2.4576MHz	4.0MHz	4.9152MHz
SS2	SS1	SS0	E	614.4kHz	1.0MHz	1 2288MHz
0	0	0	E÷16	26μs/38400Baud	16μs/62500Baud	13μs/76800Baud
0	0	1	E÷128	208μs/4800Baud	128μs/7812.5Baud	104.2μs/9600Baud
0	1	0	E÷1024	1.67ms/600Baud	1.024ms/976.6Baud	833.3μs/1200Baud
0	1	1	E÷4096	6.67ms/150Baud	4.096ms/244.1Baud	3.333ms/300Baud
1	—	—	—	*	*	*

* When SS2 is "1", Timer 2 provides SCI clocks. The baud rate is shown as follows with the TCONR as N.

$$\text{Baud Rate} = \frac{f}{32\,(N+1)} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

(2) Clocked Synchronous Mode *

			XTAL	4.0MHz	6.0MHz	8.0MHz
SS2	SS1	SS0	E	1.0MHz	1.5MHz	2.0MHz
0	0	0	E÷2	2μs/bit	1.33μs/bit	1μs/bit
0	0	1	E÷16	16μs/bit	10.7μs/bit	8μs/bit
0	1	0	E÷128	128μs/bit	85.3μs/bit	64μs/bit
0	1	1	E÷512	512μs/bit	341μs/bit	256μs/bit
1	—	—	—	**	**	**

* Bit rates in the case of internal clock operation. In the case of external clock operation, the external clock is operatable up to DC ~ 1/2 system clock.

** The bit rate is shown as follows with the TCONR as N.

$$\text{Bit Rate } (\mu s/\text{bit}) = \frac{4\,(N+1)}{f} \qquad \begin{pmatrix} f: \text{input clock frequency to the timer 2 counter} \\ N = 0 \sim 255 \end{pmatrix}$$

Table10 Baud Rate and Time Constant Register Example

Baud Rate (Baud) \ XTAL	2.4576MHz	3.6864MHz	4.0MHz	4.9152MHz	8.0MHz
110	21*	32*	35*	43*	70*
150	127	191	207	255	51*
300	63	95	103	127	207
600	31	47	51	63	103
1200	15	23	25	31	51
2400	7	11	12	15	25
4800	3	5	—	7	12
9600	1	2	—	3	—
19200	0	—	—	1	—
38400	—	—	—	0	—

*E/8 clock is provided to the timer 2's up counter.

Table 11 SCI Format and Clock Source Control

CC2	CC1	CC0	Format	Mode	Clock Source	Port 2, Bit 2	Port 2, Bit 3	Port 2, Bit 4
0	0	0	8-bit data	Clocked Synchronous	External	Input	When RE is "1", bit 3 is used for a serial input.	
0	0	1	8-bit data	Asynchronous	Internal	Not Used**		
0	1	0	8-bit data	Asynchronous	Internal	Output*		
0	1	1	8-bit data	Asynchronous	External	Input		
1	0	0	8-bit data	Clocked Synchronous	Internal	Output	When TE is "1", bit 4 is used for a serial output.	
1	0	1	9-bit data	Asynchronous	Internal	Not Used**		
1	1	0	9-bit data	Asynchronous	Internal	Output*		
1	1	1	9-bit data	Asynchronous	External	Input		

* Clock output regardless of RE or TE in the TRCSR.
** Not used for the SCI.

Bit 6 TD8 Transmit Data Bit 8
When selecting the 9-bit data format in the asynchronous mode, this bit is transmitted as the 9th data.

Bit 7 RD8 Receive Data Bit 8
When selecting the 9-bit data format in the asynchronous mode, this bit stores the 9th bit data.

■ TIMER, SCI STATUS FLAG

Table 12 shows set and clear conditions of each status flag in the timer 1, the timer 2 and the SCI.

If the flag set and clear conditions occur at the same time, the flag of the Timer 1 and the Timer 2 will be set, and the SCI cleared. Therefore the OCF1 and OCF2 of the Timer 1 may not be cleared correctly because set signal is generated periodically whenever the OCR matches the FRC. In order to clear these flags correctly, the match should be prohibited during the period between reading the TSCR and writing the OCR. For instance, these flags will be cleared correctly if the TCSR is read and the OCR is written continuously soon after matching the value of the OCR and the FCR.

Table 12 Timer 1, Timer 2 and SCI Status Flag

		Set Condition	Reset Condition
Timer 1	ICF	FRC → ICR by edge input to P_{20}.	1. Read the TCSR1 or TCSR2 then ICRH, when ICF=1 2. $\overline{RES}$=0
	OCF1	OCR1=FRC	1. Read the TCSR1 or TCSR2 then write to the OCR1H or OCR1L, when OCF1=1 2. $\overline{RES}$=0
	OCF2	OCR2=FRC	1. Read the TCSR2 then write to the OCR2H or OCR2L, when OCF2=1 2. $\overline{RES}$=0
	TOF	FRC=$FFFF+1 cycle	1. Read the TCSR1 then FRCH, when TOF=1 2. $\overline{RES}$=0
Timer 2	CMF	T2CNT=TCONR	1. Write "0" to CMF, when CMF=1 2. $\overline{RES}$=0
SCI	RDRF	Receive Shift Register → RDR	1. Read the TRCSR then RDR, when RDRF=1 2. $\overline{RES}$=0
	ORFE	1. Framing Error (Asynchronous Mode) Stop Bit = 0 2. Overrun Error (Asynchronous Mode) Receive Shift Register → RDR when RDRF=1	1. Read the TRCSR then RDR, when ORFE=1 2. $\overline{RES}$=0
	TDRE	1. Asynchronous Mode TDR → Transmit Shift Register 2. Clocked Synchronous Mode Transmit Shift Register is "empty" 3. $\overline{RES}$=0	Read the TRCSR then write to the TDR, when TDRE=1 (Note) Clear TDRE after setting TE.

(Note) 1. →; transfer
2. For example; "ICRH" means High byte of ICR.

■ LOW POWER DISSIPATION MODE

The HD63701X0 provides two low power dissipation modes; sleep and standby.

• Sleep Mode

The MCU will be in the sleep mode when SLP instruction is executed. In the sleep mode, the CPU stops and the registers' contents are retained. While the peripherals such as timers, SCI etc. continue their functions. The power dissipation of the sleep-condition is one fifth that of the operating condition.

The MCU returns from this mode by an interrupt, $\overline{RES}$ or $\overline{STBY}$; it will be reset by $\overline{RES}$ and the standby mode by $\overline{STBY}$. When the CPU responds to an interrupt request, it cancels the sleep mode, returns to the operation mode and branches to the interrupt routine. When the CPU masks this interrupt, it cancels the sleep mode and executes the next instruction. However, for example if the timer 1 or 2 prohibits a timer interrupt, the CPU doesn't cancel the sleep mode because of no interrupt request.

This sleep mode is effective to reduce the power dissipation for a system with no need of the HD63701X0's consecutive operation.

• Standby Mode

In MCU mode, the HD63701X0 stops and reset with $\overline{STBY}$ "low". In this mode, the power dissipation is reduced conspicuously. All pins except for the power supply, $\overline{STBY}$ and XTAL are detached from the MCU internally and will be the high impedance state.

While the contents of RAM is retained. The MCU returns from this mode by reset. The followings are typical usage of this mode.

Save the CPU information and SP contents on RAM by $\overline{NMI}$. Then disable the RAME bit of the RAM control register and set the STBY PWR bit to go to the standby mode. If the STBY PWR bit is still set at reset, that indicates the power is supplied to the MCU and RAM contents are retained properly. So system can restore itself by returning their pre-standby informations to the SP and the CPU. Fig. 26 depicts the timing at each pin with this example.

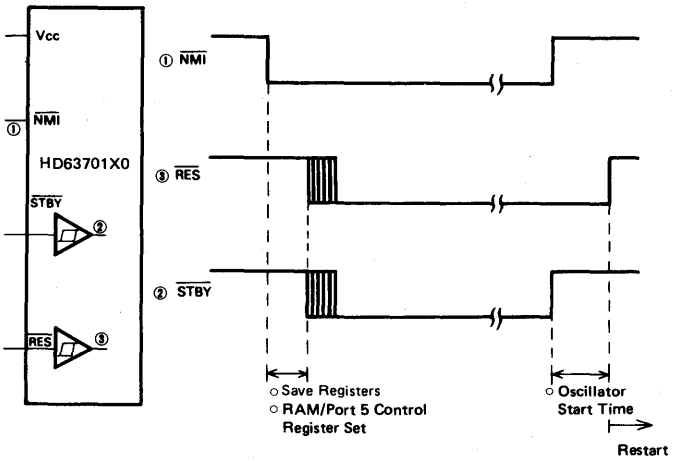


Figure 26 Standby Mode Timing

■ TRAP FUNCTION

The CPU generates an interrupt with the highest priority (TRAP) when fetching an undefined instruction or an instruction from non-memory space. The TRAP prevents the system-burst caused by noise or a program error.

• Op Code Error

When fetching an undefined op code, the CPU saves CPU registers as well as a normal interrupt and branches to the TRAP ($FFEE, $FFEF). This provides the priority next to reset.

• Address Error

When an instruction fetch is made excluding internal ROM, RAM and external memory area, the MCU generates an interrupt as well as an op code error. But on the system with no memory in its external memory area, this error processing is not applicable if an instruction fetch is made from the external non-memory area. Table 13 provides addresses where an address error occurs to each mode.

This processing is available only for an instruction fetch and is not applicable to the access of normal data read/write.

Table 13 Addresses Applicable to Address Errors

Mode	1	2	3
Address	$0000 ≀ $001F	$0000 ≀ $001F	$0000 ≀ $003F $0100 ≀ $EFFF

(Note) The TRAP interrupt provides a retry function differently from other interrupts. This is a program flow return to the address where the TRAP occurs when a sequence returns to a main routine from the TRAP interrupt routine by RTI. The retry can prevent the system burst caused by noise etc.
However, if another TRAP occurs, the program repeats the TRAP interrupt forever, so the consideration is necessary in programming.

■ INSTRUCTION SET

The HD63701X0 provides object code upward compatible with the HD6801 to utilize all instruction set of the HMCS6800. It also reduces the execution times of key instructions for throughout improvement.

Bit manipulation instruction, change instruction of the index register and accumulator and sleep instruction are also added.

The followings are explained here.

- CPU Programming Model (refer to Fig. 27)
- Addressing Mode
- Accumulator and Memory Manipulation Instruction (refer to Table 14)
- New Instruction
- Index Register and Stack Manipulation Instruction (refer to Table 15)
- Jump and Branch Instruction (refer to Table 16)
- Condition Code Register Manipulation (refer to Table 17)
- Op Code Map (refer to Table 18)

● Programming Model

Fig. 27 depicts the HD63701X0 programming model. The double accumulator D consists of accumulator A and B, so when using the accumulator D, the contents of A and B are destroyed.

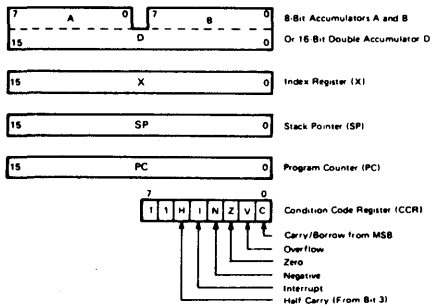


Figure 27 CPU Programming Model

● CPU Addressing Mode

The HD63701X0 provides 7 addressing modes. The addressing mode is decided by an instruction type and code. Table 14 through 18 show addressing modes of each instruction with the execution times counted by the machine cycle.

When the clock frequency is 4MHz, the machine cycle time becomes microseconds directly.

Accumulator (ACCX) Addressing

Only an accumulator is addressed and the accumulator A or B is selected. This is a one-byte instruction.

Immediate Addressing

This addressing locates a data in the second byte of an instruction. However, LDS and LDX locate a data in the second and third byte exceptionally. This addressing is a 2 or 3-byte instruction.

Direct Addressing

In this addressing mode, the second byte of an instruction shows the address where a data is stored. 256 bytes ($0 through $255) can be addressed directly. Execution times can be reduced by storing data in this area so it is recommended to make it RAM for users' data area in configurating a system. This is a 2-byte instruction, while 3 byte with regard to AIM, OIM, EIM and TIM.

Extended Addressing

In this mode, the second byte shows the upper 8 bit of the data stored address and the third byte the lower 8 bit. This indicates the absolute address of 3 byte instruction in the memory.

Indexed Addressing

The second byte of an instruction and the lower 8 bit of the index register are added in this mode. As for AIM, OIM, EIM and TIM, the third byte of an instruction and the lower 8 bits of the index register are added.

This carry is added to the upper 8 bit of the index register and the result is used for addressing the memory. The modified address is retained in the temporary address register, so the contents of the index register doesn't change. This is a 2-byte instruction except AIM, OIM, EIM and TIM (3-byte instruction).

Implied Addressing

An instruction itself specifies the address. That is, the instruction addresses a stack pointer, index register etc. This is a one-byte instruction.

Relative Addressing

The second byte of an instruction and the lower 8 bits of the program counter are added. The carry or borrow is added to the upper 8 bit. So addressing from −126 to +129 byte of the current instruction is enabled. This is a 2-byte instruction.

(Note) CLI, SEI Instructions and Interrupt Operation

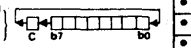
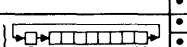
When accepting the IRQ at a preset timing with the help of CLI and SEI instructions, more than 2 cycles are necessary between the CLI and SEI instructions. For example, the following program (a) (b) don't accept the IRQ but (c) accepts it.

```
                                  .
                                  .
                                  .
                                CLI
CLI                             NOP
SEI                             NOP
 .                              SEI
 .                               .
 .                               .
 .                               .
 .                               .

(a)                             (c)
```

The same thing can be said to the TAP instruction instead of the CLI and SEI instructions.

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → B	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	3	3	D3	4	2	E3	5	2	F3	5	3				A : B + M : M + 1 → A : B	•	•	↕	↕	↕	↕
Add Accumulators	ABA													1B	1	1	A + B → A	↕	•	↕	↕	↕	↕
Add With Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
AND	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A·M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B·M → B	•	•	↕	↕	R	•
Bit Test	BIT A	85	2	2	95	3	2	A5	4	2	B5	4	3				A·M	•	•	↕	↕	R	•
	BIT B	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B·M	•	•	↕	↕	R	•
Clear	CLR							6F	5	2	7F	5	3				00 → M	•	•	R	S	R	R
	CLRA													4F	1	1	00 → A	•	•	R	S	R	R
	CLRB													5F	1	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
Compare Accumulators	CBA													11	1	1	A - B	•	•	↕	↕	↕	↕
Complement, 1's	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	1	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	1	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Complement, 2's	NEG							60	6	2	70	6	3				00 - M → M	•	•	↕	↕	①	②
(Negate)	NEGA													40	1	1	00 - A → A	•	•	↕	↕	①	②
	NEGB													50	1	1	00 - B → B	•	•	↕	↕	①	②
Decimal Adjust, A	DAA													19	2	1	Converts binary add of BCD characters into BCD format	•	•	↕	↕	↕	③
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	④	•
	DECA													4A	1	1	A - 1 → A	•	•	↕	↕	④	•
	DECB													5A	1	1	B - 1 → B	•	•	↕	↕	④	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A ⊕ M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B ⊕ M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	⑤	•
	INCA													4C	1	1	A + 1 → A	•	•	↕	↕	⑤	•
	INCB													5C	1	1	B + 1 → B	•	•	↕	↕	⑤	•
Load Accumulator	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double Accumulator	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M + 1 → B, M → A	•	•	↕	↕	R	•
Multiply Unsigned	MUL													3D	7	1	A x B → A : B	•	•	•	•	•	⑪
OR, Inclusive	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	4	1	A → Msp, SP - 1 → SP	•	•	•	•	•	•
	PSHB													37	4	1	B → Msp, SP - 1 → SP	•	•	•	•	•	•
Pull Data	PULA													32	3	1	SP + 1 → SP, Msp → A	•	•	•	•	•	•
	PULB													33	3	1	SP + 1 → SP, Msp → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•	↕	↕	⑥	↕
	ROLA													49	1	1		•	•	↕	↕	⑥	↕
	ROLB													59	1	1		•	•	↕	↕	⑥	↕
Rotate Right	ROR							66	6	2	76	6	3					•	•	↕	↕	⑥	↕
	RORA													46	1	1		•	•	↕	↕	⑥	↕
	RORB													56	1	1		•	•	↕	↕	⑥	↕

(Note) Condition Code Register will be explained in Note of Table 17.

(continued)

Table 14 Accumulator, Memory Manipulation Instructions

Operations	Mnemonic	Addressing Modes															Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Shift Left Arithmetic	ASL							68	6	2	78	6	3				M	•	•	↕	↕	⑥	↕
	ASLA													48	1	1	A	•	•	↕	↕	⑥	↕
	ASLB													58	1	1	B	•	•	↕	↕	⑥	↕
Double Shift Left, Arithmetic	ASLD													05	1	1		•	•	↕	↕	⑥	↕
Shift Right Arithmetic	ASR							67	6	2	77	6	3				M	•	•	↕	↕	⑥	↕
	ASRA													47	1	1	A	•	•	↕	↕	⑥	↕
	ASRB													57	1	1	B	•	•	↕	↕	⑥	↕
Shift Right Logical	LSR							64	6	2	74	6	3				M	•	•	R	↕	⑥	↕
	LSRA													44	1	1	A	•	•	R	↕	⑥	↕
	LSRB													54	1	1	B	•	•	R	↕	⑥	↕
Double Shift Right Logical	LSRD													04	1	1		•	•	R	↕	⑥	↕
Store Accumulator	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
Store Double Accumulator	STD				DD	4	2	ED	5	2	FD	5	3				A → M B → M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A − M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B − M → B	•	•	↕	↕	↕	↕
Double Subtract	SUBD	83	3	3	93	4	2	A3	5	2	B3	5	3				A : B − M : M + 1 → A : B	•	•	↕	↕	↕	↕
Subtract Accumulators	SBA													10	1	1	A − B → A	•	•	↕	↕	↕	↕
Subtract With Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A − M − C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B − M − C → B	•	•	↕	↕	↕	↕
Transfer Accumulators	TAB													16	1	1	A → B	•	•	↕	↕	R	•
	TBA													17	1	1	B → A	•	•	↕	↕	R	•
Test Zero or Minus	TST							6D	4	2	7D	4	3				M − 00	•	•	↕	↕	R	R
	TSTA													4D	1	1	A − 00	•	•	↕	↕	R	R
	TSTB													5D	1	1	B − 00	•	•	↕	↕	R	R
And Immediate	AIM				71	6	3	61	7	3							M·IMM→M	•	•	↕	↕	R	•
OR Immediate	OIM				72	6	3	62	7	3							M+IMM→M	•	•	↕	↕	R	•
EOR Immediate	EIM				75	6	3	65	7	3							M⊕IMM→M	•	•	↕	↕	R	•
Test Immediate	TIM				7B	4	3	6B	5	3							M·IMM	•	•	↕	↕	R	•

(Note) Condition Code Register will be explained in Note of Table 17.

- **Additional Instruction**

In addition to the HD6801 instruction set, the HD63701X0 prepares the following new instructions.

AIM..... (M)·(IMM) → (M)

Executes "AND" operation to immediate data and the memory contents and stores its result in the memory.

OIM..... (M) + (IMM) → (M)

Executes "OR" operation to immediate data and the memory contents and stores its result in the memory.

EIM..... (M) ⊕ (IMM) → (M)

Executes "EOR" operation to immediate data and the memory contents and stores its result in the memory.

TIM..... (M) · (IMM)

Executes "AND" operation to immediate data and changes the relative flag of the condition code register.

These area 3-byte instructions; the first byte is op code, the second immediate data and the third address modifier.

XGDX..... (ACCD) ↔ (IX)

Exchanges the contents of accumulator and the index register.

SLP

Goes to the sleep mode. Refer to "LOW POWER DISSIPATION MODE" for more details of the sleep mode.

Table 15 Index Register, Stack Manipulation Instructions

Pointer Operations	Mnemonic	Addressing Modes														Boolean/ Arithmetic Operation	Condition Code Register					
		IMMED.			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	5	2	BC	5	3				X − M:M+1	•	•	↕	↕	↕	↕
Decrement Index Reg	DEX													09	1	1	X − 1 → X	•	•	•	↕	•	•
Decrement Stack Pntr	DES													34	1	1	SP − 1 → SP	•	•	•	•	•	•
Increment Index Reg	INX													08	1	1	X + 1 → X	•	•	•	↕	•	•
Increment Stack Pntr	INS													31	1	1	SP + 1 → SP	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M → X_H, (M + 1) → X_L	•	•	⑦	↕	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M → SP_H, (M+1) → SP_L	•	•	⑦	↕	R	•
Store Index Reg	STX				DF	4	2	EF	5	2	FF	5	3				X_H → M, X_L → (M + 1)	•	•	⑦	↕	R	•
Store Stack Pntr	STS				9F	4	2	AF	5	2	BF	5	3				SP_H → M, SP_L → (M + 1)	•	•	⑦	↕	R	•
Index Reg → Stack Pntr	TXS													35	1	1	X − 1 → SP	•	•	•	•	•	•
Stack Pntr → Index Reg	TSX													30	1	1	SP + 1 → X	•	•	•	•	•	•
Add	ABX													3A	1	1	B + X → X	•	•	•	•	•	•
Push Data	PSHX													3C	5	1	X_L → M_{SP}, SP − 1 → SP X_H → M_{SP}, SP − 1 → SP	•	•	•	•	•	•
Pull Data	PULX													38	4	1	SP + 1 → SP, M_{SP} → X_H SP + 1 → SP, M_{SP} → X_L	•	•	•	•	•	•
Exchange	XGDX													18	2	1	ACCD·IX	•	•	•	•	•	•

(Note) Condition Code Register will be explained in Note of Table 17.

Table 16 Jump, Branch Instruction

Operations	Mnemonic	Addressing Modes															Branch Test	Condition Code Register					
		RELATIVE			DIRECT			INDEX			EXTEND			IMPLIED				5	4	3	2	1	0
		OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#		H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N ⊕ V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N ⊕ V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N ⊕ V) = 1	•	•	•	•	•	•
Branch If Lower Or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N ⊕ V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch To Subroutine	BSR	8D	5	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump To Subroutine	JSR				9D	5	2	AD	5	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	1	1	Advances Prog. Cntr. Only	•	•	•	•	•	•
Return From Interrupt	RTI													3B	10	1		——		⑧		——	
Return From Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt*	WAI													3E	9	1		•	⑨	•	•	•	•
Sleep	SLP													1A	4	1		•	•	•	•	•	•

(Note) * WAI puts R/$\overline{W}$ high; Address Bus goes to FFFF; Data Bus goes to the three state.
Condition Code Register will be explained in Note of Table 17.

Table 17 Condition Code Register Manipulation Instructions

Operations	Mnemonic	Addressing Modes IMPLIED OP	~	#	Boolean Operation	Condition Code Register 5 H	4 I	3 N	2 Z	1 V	0 C
Clear Carry	CLC	0C	1	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	1	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	1	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	1	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	1	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	1	1	1 → V	•	•	•	•	S	•
Accumulator A → CCR	TAP	06	1	1	A → CCR	⑩					
CCR → Accumulator A	TPA	07	1	1	CCR → A	•	•	•	•	•	•

LEGEND

OP Operation Code (Hexadecimal)
~ Number of MCU Cycles
M_{SP} Contents of memory location pointed to by Stack Pointer
Number of Program Bytes
+ Arithmetic Plus
− Arithmetic Minus
• Boolean AND
+ Boolean Inclusive OR
⊕ Boolean Exclusive OR
$\overline{M}$ Complement of M
→ Transfer into
0 Bit = Zero
00 Byte = Zero

CONDITION CODE SYMBOLS

H Half-carry from bit 3 to bit 4
I Interrupt mask
N Negative (sign bit)
Z Zero (byte)
V Overflow, 2's complement
C Carry/Borrow from/to bit 7
R Reset Always
S Set Always
‡ Set if true after test or clear
• Not Affected

(Note) Condition Code Register Notes: (Bit set if test is true and cleared otherwise)

① (Bit V) Test: Result = 10000000?
② (Bit C) Test: Result ≠ 00000000?
③ (Bit C) Test: BCD Character of high-order byte greater than 10? (Not cleared if previously set)
④ (Bit V) Test: Operand = 10000000 prior to execution?
⑤ (Bit V) Test: Operand = 01111111 prior to execution?
⑥ (Bit V) Test: Set equal to N⊕ C = 1 after the execution of instructions
⑦ (Bit N) Test: Result less than zero? (Bit 15=1)
⑧ (All Bit) Load Condition Code Register from Stack.
⑨ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exist the wait state.
⑩ (All Bit) Set according to the contents of Accumulator A.
⑪ (Bit C) Result of Multiplication Bit 7=1? (ACCB)

Table 18 OP-Code Map

OP CODE HI	LO	0000	0001	0010	0011	ACC A 0100	ACC B 0101	IND 0110	EXT/DIR 0111	ACCA or SP IMM 1000	DIR 1001	IND 1010	EXT 1011	ACCB or X IMM 1100	DIR 1101	IND 1110	EXT 1111	HI
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	LO
0000	0		SBA	BRA	TSX	NEG				SUB								0
0001	1	NOP	CBA	BRN	INS			AIM		CMP								1
0010	2			BHI	PULA			OIM		SBC								2
0011	3			BLS	PULB	COM				SUBD				ADDD				3
0100	4	LSRD		BCC	DES	LSR				AND								4
0101	5	ASLD		BCS	TXS			EIM		BIT								5
0110	6	TAP	TAB	BNE	PSHA	ROR				LDA								6
0111	7	TPA	TBA	BEQ	PSHB	ASR					STA				STA			7
1000	8	INX	XGDX	BVC	PULX	ASL				EOR								8
1001	9	DEX	DAA	BVS	RTS	ROL				ADC								9
1010	A	CLV	SLP	BPL	ABX	DEC				ORA								A
1011	B	SEV	ABA	BMI	RTI			TIM		ADD								B
1100	C	CLC		BGE	PSHX	INC				CPX				LDD				C
1101	D	SEC		BLT	MUL	TST				BSR	JSR				STD			D
1110	E	CLI		BGT	WAI			JMP		LDS				LDX				E
1111	F	SEI		BLE	SWI	CLR					STS				STX			F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

UNDEFINED OP CODE

* Only each instructions of AIM, OIM, EIM, TIM

■ CPU OPERATION

● CPU Instruction Flow

When operating, the CPU fetches an instruction from a memory and executes the required functions. This sequence starts after the reset release and repeats itself limitlessly if not affected by a special instruction or a control signal. SWI, RTI, WAI and SLP instructions are to change this operation, while $\overline{NMI}$, $\overline{IRQ_1}$, $\overline{IRQ_2}$, IRQ_3, $\overline{HALT}$ and $\overline{STBY}$ are to control it. Fig. 28 gives the CPU mode shift and Fig. 29 the CPU system flowchart. Table 19 shows the CPU operating states and port states.

● Operation at Each Instruction Cycle

Table 20 provides the operation at each instruction cycle. By the pipeline control of the HD63701X0, MULT, PUL, DAA and XGDX instructions etc. prefetch the next instruction. So attention is necessary to the counting of the instruction cycles because it is different from the existent one ----- op code fetch to the next instruction op code.

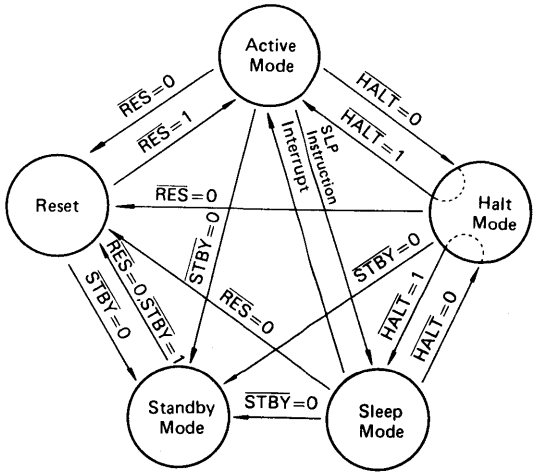


Figure 28 CPU Operation Mode Transition

Table 19 CPU Operation State and Port State

Port	Mode	Reset	STBY****	HALT***	Sleep
Port 1 ($A_0 \sim A_7$)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 2	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 3 ($D_0 \sim D_7$)	Mode 1, 2	T	T	T	T
	Mode 3				Keep
Port 4 ($A_8 \sim A_{15}$)	Mode 1, 2	H	T	T	H
	Mode 3	T			Keep
Port 5	Mode 1, 2	T	T	T	T
	Mode 3				
Port 6	Mode 1, 2	T	T	Keep	Keep
	Mode 3				
Port 7	Mode 1, 2	*	T	**	*
	Mode 3	T			Keep

H ; High, L ; Low, T ; High Impedance
* $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$, $\overline{LIR}$=H, BA=L
** $\overline{RD}$, $\overline{WR}$, R/$\overline{W}$=T, $\overline{LIR}$, BA=H
*** $\overline{HALT}$ is unacceptable in mode 3.
**** E pin goes to high impedance state.

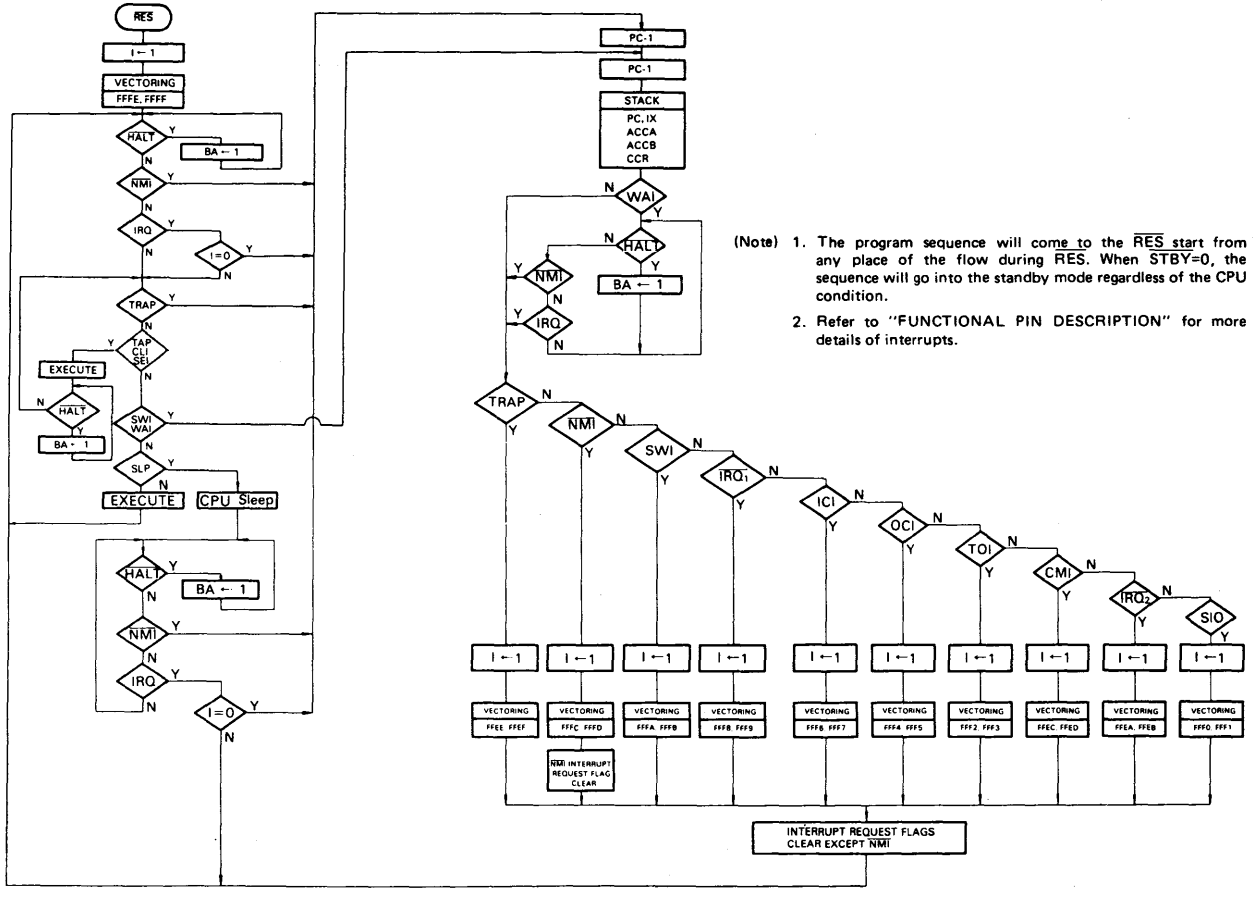


Figure 29 HD63701X0 System Flow Chart

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Table 20 Cycle-by-Cycle Operation

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMMEDIATE								
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	2	1	Op Code Address + 1	1	0	1	1	Operand Data
		2	Op Code Address + 2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	3	1	Op Code Address + 1	1	0	1	1	Operand Data (MSB)
		2	Op Code Address + 2	1	0	1	1	Operand Data (LSB)
		3	Op Code Address + 3	1	0	1	0	Next Op Code
DIRECT								
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB	3	1	Op Code Address + 1	1	0	1	1	Address of Operand (LSB)
		2	Address of Operand	1	0	1	1	Operand Data
		3	Op Code Address + 2	1	0	1	0	Next Op Code
STA	3	1	Op Code Address + 1	1	0	1	1	Destination Address
		2	Destination Address	0	1	0	1	Accumulator Data
		3	Op Code Address + 2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	4	1	Op Code Address + 1	1	0	1	1	Address of Operand (LSB)
		2	Address of Operand	1	0	1	1	Operand Data (MSB)
		3	Address of Operand + 1	1	0	1	1	Operand Data (LSB)
		4	Op Code Address + 2	1	0	1	0	Next Op Code
STD STS STX	4	1	Op Code Address + 1	1	0	1	1	Destination Address (LSB)
		2	Destination Address	0	1	0	1	Register Data (MSB)
		3	Destination Address + 1	0	1	0	1	Register Data (LSB)
		4	Op Code Address + 2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address + 1	1	0	1	1	Jump Address (LSB)
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer − 1	0	1	0	1	Return Address (MSB)
		5	Jump Address	1	0	1	0	First Subroutine Op Code
TIM	4	1	Op Code Address + 1	1	0	1	1	Immediate Data
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address + 3	1	0	1	0	Next Op Code
AIM EIM OIM	6	1	Op Code Address + 1	1	0	1	1	Immediate Data
		2	Op Code Address + 2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address + 3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
INDEXED								
JMP	3	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Jump Address	1	0	1	0	First Op Code of Jump Routine
ADC ADD AND BIT CMP EOR LDA ORA SBC SUB TST	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Accumulator Data
		4	Op Code Address+2	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data (MSB)
		4	IX+Offset+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	0	1	0	1	Register Data (MSB)
		4	IX+Offset+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+2	1	0	1	0	Next Op Code
JSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	IX+Offset	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	IX+Offset	0	1	0	1	New Operand Data
		6	Op Code Address+1	1	0	1	0	Next Op Code
TIM	5	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	IX+Offset	1	0	1	1	Operand Data
		4	IX+Offset	0	1	0	1	00
		5	Op Code Address+2	1	0	1	0	Next Op Code
AIM EIM OIM	7	1	Op Code Address+1	1	0	1	1	Immediate Data
		2	Op Code Address+2	1	0	1	1	Offset
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	IX+Offset	1	0	1	1	Operand Data
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	IX+Offset	0	1	0	1	New Operand Data
		7	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
EXTEND								
JMP	3	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	Jump Address	1	0	1	0	Next Op Code
ADC ADD TST AND BIT CMP EOR LDA ORA SBC SUB	4	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
STA	4	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Accumulator Data
		4	Op Code Address+3	1	0	1	0	Next Op Code
ADDD CPX LDD LDS LDX SUBD	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data (MSB)
		4	Address of Operand+1	1	0	1	1	Operand Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
STD STS STX	5	1	Op Code Address+1	1	0	1	1	Destination Address (MSB)
		2	Op Code Address+2	1	0	1	1	Destination Address (LSB)
		3	Destination Address	0	1	0	1	Register Data (MSB)
		4	Destination Address+1	0	1	0	1	Register Data (LSB)
		5	Op Code Address+3	1	0	1	0	Next Op Code
JSR	6	1	Op Code Address+1	1	0	1	1	Jump Address (MSB)
		2	Op Code Address+2	1	0	1	1	Jump Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Stack Pointer	0	1	0	1	Return Address (LSB)
		5	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		6	Jump Address	1	0	1	0	First Subroutine Op Code
ASL ASR COM DEC INC LSR NEG ROL ROR	6	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	Address of Operand	0	1	0	1	New Operand Data
		6	Op Code Address+3	1	0	1	0	Next Op Code
CLR	5	1	Op Code Address+1	1	0	1	1	Address of Operand (MSB)
		2	Op Code Address+2	1	0	1	1	Address of Operand (LSB)
		3	Address of Operand	1	0	1	1	Operand Data
		4	Address of Operand	0	1	0	1	00
		5	Op Code Address+3	1	0	1	0	Next Op Code

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
ABA ABX ASL ASLD ASR CBA CLC CLI CLR CLV COM DEC DES DEX INC INS INX LSR LSRD ROL ROR NOP SBA SEC SEI SEV TAB TAP TBA TPA TST TSX TXS	1	1	Op Code Address+1	1	0	1	0	Next Op Code
DAA XGDX	2	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
PULA PULB	3	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack
PSHA PSHB	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Accumulator Data
		4	Op Code Address+1	1	0	1	0	Next Op Code
PULX	4	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Data from Stack (MSB)
		4	Stack Pointer+2	1	0	1	1	Data from Stack (LSB)
PSHX	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Index Register (LSB)
		4	Stack Pointer−1	0	1	0	1	Index Register (MSB)
		5	Op Code Address+1	1	0	1	0	Next Op Code
RTS	5	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Return Address (MSB)
		4	Stack Pointer+2	1	0	1	1	Return Address (LSB)
		5	Return Address	1	0	1	0	First Op Code of Return Routine
MUL	7	1	Op Code Address+1	1	0	1	0	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	FFFF	1	1	1	1	Restart Address (LSB)
		5	FFFF	1	1	1	1	Restart Address (LSB)
		6	FFFF	1	1	1	1	Restart Address (LSB)
		7	FFFF	1	1	1	1	Restart Address (LSB)

(Continued)

Address Mode & Instructions	Cycles	Cycle #	Address Bus	R/$\overline{W}$	$\overline{RD}$	$\overline{WR}$	$\overline{LIR}$	Data Bus
IMPLIED								
WAI	9	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
RTI	10	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer+1	1	0	1	1	Conditional Code Register
		4	Stack Pointer+2	1	0	1	1	Accumulator B
		5	Stack Pointer+3	1	0	1	1	Accumulator A
		6	Stack Pointer+4	1	0	1	1	Index Register (MSB)
		7	Stack Pointer+5	1	0	1	1	Index Register (LSB)
		8	Stack Pointer+6	1	0	1	1	Return Address (MSB)
		9	Stack Pointer+7	1	0	1	1	Return Address (LSB)
		10	Return Address	1	0	1	0	First Op Code of Return Routine
SWI	12	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Stack Pointer−2	0	1	0	1	Index Register (LSB)
		6	Stack Pointer−3	0	1	0	1	Index Register (MSB)
		7	Stack Pointer−4	0	1	0	1	Accumulator A
		8	Stack Pointer−5	0	1	0	1	Accumulator B
		9	Stack Pointer−6	0	1	0	1	Conditional Code Register
		10	Vector Address FFFA	1	0	1	1	Address of SWI Routine (MSB)
		11	Vector Address FFFB	1	0	1	1	Address of SWI Routine (LSB)
		12	Address of SWI Routine	1	0	1	0	First Op Code of SWI Routine
SLP	4	1	Op Code Address+1	1	0	1	1	Next Op Code
		2	FFFF	1	1	1	1	Restart Address (LSB)
		Sleep						
		3	FFFF	1	1	1	1	Restart Address (LSB)
		4	Op Code Address+1	1	0	1	0	Next Op Code
RELATIVE								
BCC BCS BEQ BGE BGT BHI BLE BLS BLT BMT BNE BPL BRA BRN BVC BVS	3	1	Op Code Address+1	1	0	1	1	Branch Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Branch Address ····Test="1" Op Code Address+1 ··Test="0"	1	0	1	0	First Op Code of Branch Routine Next Op Code
BSR	5	1	Op Code Address+1	1	0	1	1	Offset
		2	FFFF	1	1	1	1	Restart Address (LSB)
		3	Stack Pointer	0	1	0	1	Return Address (LSB)
		4	Stack Pointer−1	0	1	0	1	Return Address (MSB)
		5	Branch Address	1	0	1	0	First Op Code of Subroutine

■ APPLICATION NOTES

● The EPROM Programming and Maintenance

(1) The EPROM Programming and Data Retention

An EPROM memory cell is programmed by hot electrons injected to the floating gate with applying high voltage at the control gate and the drain. The electrons have been trapped by the potential barrier at the polysilicon-oxide (SiO_2) by which the floating gate is completely sorrounded. The programmed cell becomes a "0".

The memory cell will be discharged by;

① Exposure to ultraviolet light; discharged by photo emitting electrons (erasure principle)

② Heat; discharged by thermal emitting electrons

③ Applied with high voltage; discharged by high electric field. Charge loss from the normal cell by case ② or ③ is negligible. But if there are some defects at the SiO_2, the cell will be rapidly discharged through the defects. Such a defective part is rejected by manufacturing screenings.

The erased, or discharged, cell is a "1".

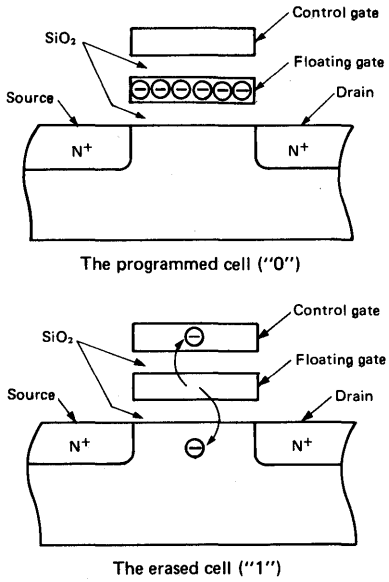


Figure 30 Cross-section of An EPROM Memory Cell

(2) Precaution of the EPROM Programming

The EPROM memory cell should be programmed with the specified voltage and timing. The higher program voltage V_{PP} or the longer program pulse width t_{PW} is applied, the more will be the quantity of electrons injected to the floating gate. However, a p-n junction will be broken permanently if V_{PP} is applied to more than maximum ratings. Especially V_{PP} overshoot of an EPROM programmer should l checked.

Negative-noise to device pins may cause a parasitic tra sistor effect and reduce the breakdown voltage.

(3) Precaution for using the MCU in the ceramic package with window

Static charge on the window surface may adversely affe the function of the MCU. The charge will be caused by rul bing the window with plastics or dry cloths, or touching charged body on it. They can be discharged by exposure ultraviolet light for a short time. It is recommended to pr gram the memory cell again after exposure, since the ele trons trapped at the floating gate will reduce. The metho to prevent static charge on the window are follows.

① Connect the body of an operator to the ground.
② Do not rub the window with plastics or dry cloths.
③ Do not use coolant sprays which contain some ions.
④ Use a conductive opaque label.

The data stored in EPROM may be losed or the MC may malfunction by photocurrent if the MCU is exposed strong light like a fluorescent lamp or the sunlight. Ther fore, it is recommended to cover the window with an opaqu label.

(4) Screening procedure of the MCU in the plastic package

In general, any standard manufacturing screening semiconductor devices will make initial failures rejected an improve reliability. The bake procedure for EPROM device accelerates any electron leakage at the floating gate. Th manufacturer tests the CPU, RAM, I/O and other logi functions in the EPROM on-chip MCU in the plastic pack age at wafer sort and final test, and rejects any devices whic do not pass the tests. It is impossible, however, to reject EI ROM defects at final test, since the EPROM memory po tion cannot be completely tested after molding in the plasti package. Therefore, it is recommended that the screenin procedure shown Fig. 31 after programming EPROM po tion.

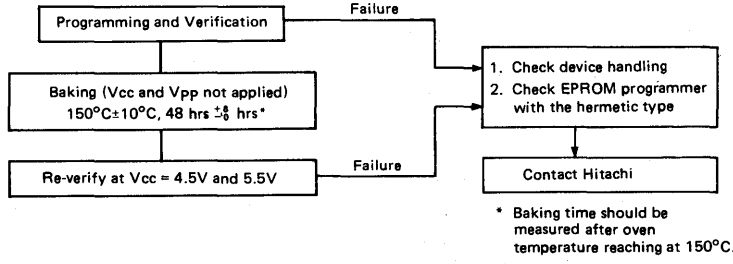


Figure 31 Recommended Screening Procedure of the MCU in The Plastic Package

(Caution) If the user experiences several consecutive programming failures, from same EPROM programmer, after following the recommended screening procedure, then call Hitachi.

(5) EPROM programmers and socket adapters

EPROM programmers and socket adapters which are rec ommended for the HD63701X0 are shown Table 21.

A socket adapter is a tool to convert from 64-pin socke to standard 24-pin socket.

Table 21 EPROM Programmers and Socket Adapters for the HD63701X0

EPROM Programmer		Socket Adapter	
Maker	Type No.	Maker	Type No.
DATA I/O (U.S.A.)	121A/121B 29A/29B	Hitachi Ltd.	H67PWA01A
		Data I/O	HD63701X0 (for 29A/29B)
AVAL CORP. (JAPAN)	PKW-1000 PKW-7000	Hitachi Ltd.	H67PWA01A
Minato Electronics Inc. (JAPAN)	M1863 M1866 7GU-2700	Hitachi Ltd.	H67PWA01B

- **Write only Register**

When a write-only register such as the DDR of the port is read by the MPU, "$FF" always appears on the data bus. Note that when an instruction which reads the memory contents and does some arithmetic operation on the contents of the write-only register, it always gets $FF as the arithmetic and logical results. AIM, OIM and EIM instructions are unable to apply especially for the bit manipulation of the DDR of the I/O port.

- **Trap Interrupt**

When execution an RTI instruction at the end of the interrupt routine, trap interrupt different from other interrupts returns to the address where the trap interrupt was generated. Attention is necessary when using several trap interrupts in the program. See Fig. 32 and 33 for details.

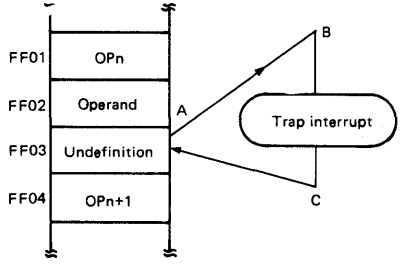


Figure 32 Fetching an Undefined Op-code

After executing OPn instruction, the HD63701X0 fetches and decodes and undefined op-code inside to generate a trap interrupt. When RTI instruction is executed in this trap interrupt servicing routine, the HD63701X0 will set $FF03 in PC, fetch the undefined code again, generate a trap interrupt and repeat ABC endless-loop.

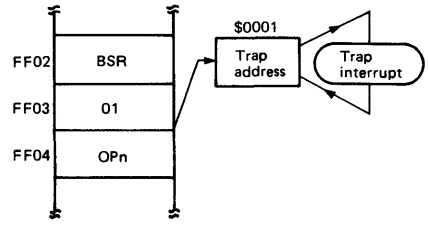


Figure 33 Fetching Erroneously

After performing BSR instruction, the branch destination address is output on an address bus to fetch the first op-code of a subroutine. If $0001 is output as an address by some mistake the HD63701X0 decodes it inside and generates a trap interrupt. When RTI instruction is performed in this trap interrupt servicing routine, the HD63701X0 will set $0001 in PC and start from this address, which causes a trap interrupt again and repeat this endless-loop.

- **Precaution for using WAI instruction**

If $\overline{\text{HALT}}$ turns "Low" in WAI execution, a CPU upset may occur since the correct vector will not be fetched after the halt state has been released. It is recommended to use BRA instruction etc. for software interrupt before $\overline{\text{HALT}}$ turns "Low" shown Fig. 35.

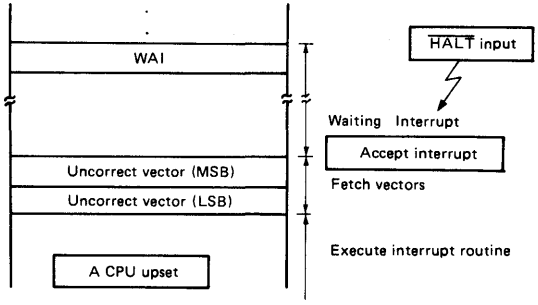


Figure 34 A CPU Upset after $\overline{\text{HALT}}$ Input in WAI Execution

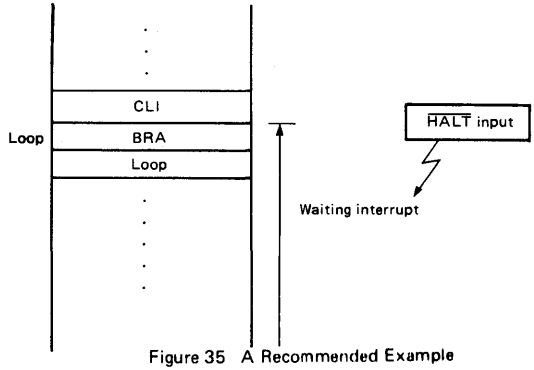


Figure 35 A Recommended Example

● **Power-on Reset**

At power-on it is necessary to hold $\overline{RES}$ "low" to reset the internal state of the device and to provide sufficient time for the oscillator to stabilize. Pay attention to the following.

* Just after power-on, the MPU doesn't enter reset state until the oscillation starts. This is because the reset signal is input internally, with the clocked synchronization as shown below.

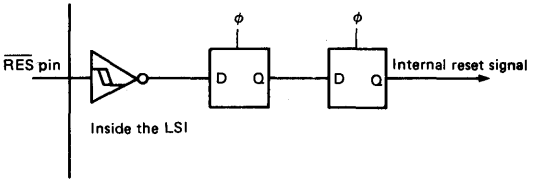
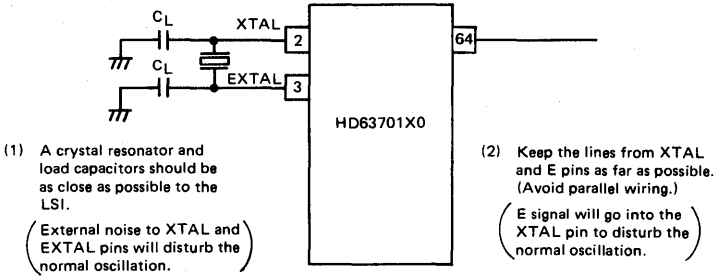


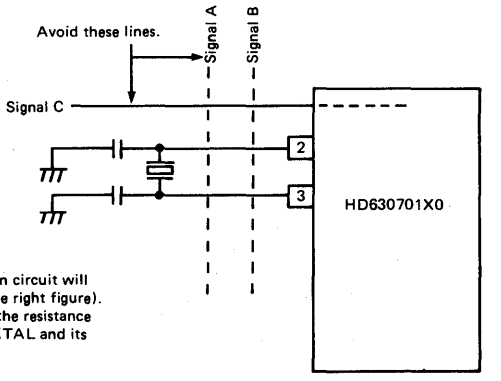
Figure 36 Reset Circuit

Thus, just after power-on the LSI state (I/O port, mode condition etc.) is unstable until the oscillation starts. If it is necessary to inform the LSI state to the external devices during this period, it needs to be done by the external circuits.

● **Board Design of Oscillation Circuits**

Keep the following in mind when connecting a crystal resonator to XTAL and EXTAL pins of the HD63701X0.





Signal lines or power supply lines near the oscillation circuit will disturb normal oscillation by their induction (see the right figure). So pay attention not to do that. In addition. keep the resistance between XTAL and its nearest pin, and between EXTAL and its nearest pin more than 10MΩ.

Figure 37 Precaution on Board Design of Oscillation Circuits

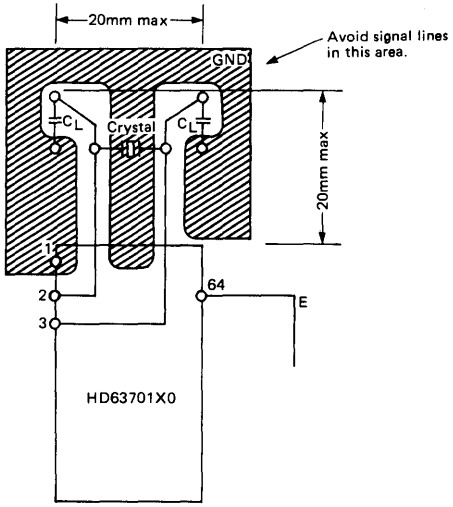


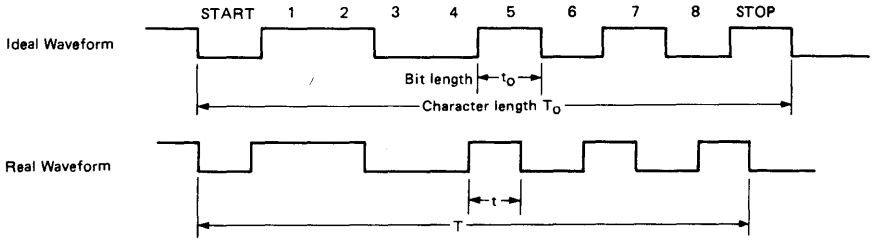
Figure 38 Example of Oscillation Circuits in Board Design

- **Receive Margin of the SCI**

Receive margin of the SCI contained in the HD63701X0 is shown in Table 22.
Note: SCI = Serial Communication Interface.

Table 22

Bit distortion tolerance $(t-t_o) / t_o$	Character distortion tolerance $(T-To) / To$
±43.7%	±4.37%



HD63705V0, HD637A05V0, HD637B05V0

CMOS MCU (Microcomputer Unit)

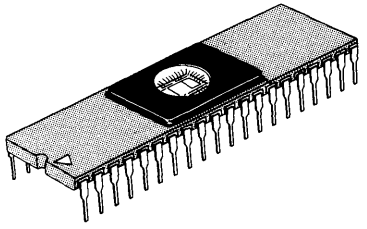
—ADVANCE INFORMATION—

The HD63705V0 is an 8-bit CMOS single chip microcomputer unit (MCU) which, including 4k bytes of EPROM, is object code compatible with the HD6305V0.

The HD63705V0 contains 4k bytes of EPROM, 192 bytes of RAM, serial communication interface and 31 parallel I/O pins in addition to CPU.

The HD63705V0 is available in a hermetically sealed 40-pin ceramic package which includes a glass window that allows for programming and EPROM erasure in the same way as 27256 type EPROM.

HD63705V0C, HD637A05V0C,
HD637B05V0C



(DC-40)

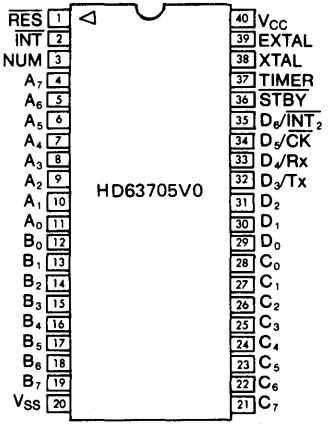
■ FEATURES

- Instruction Set compatible with the HD6305V0
- 4k Bytes of EPROM (compatible with 27256 type)
- 192 Bytes of RAM
- A total of 31 terminals
- Two timers
  - — 8-bit timer with a 7-bit prescaler (programmable prescaler; event counter)
  - — 15-bit timer (commonly used with the SCI clock divider)
- On-chip serial interface circuit (synchronized with clock)
- Six interrupts (two external, two timer, one serial and one software)
- Low power dissipation modes — Wait, Stop and Standby Mode
- Operation Mode
  - MCU Mode (Single-chip Mode)
  - EPROM Mode
- Minimum Instruction Cycle Time
  - HD63705V0 ..... 1 μs (f=1MHz)
  - HD637A05V0 ..... 0.67 μs (f=1.5MHz)
  - HD637B05V0 ..... 0.5 μs (f=2MHz)
- Wide Operating Range
  - HD63705V0 ..... f=0.1 to 1MHz (V_{CC}=5V±10%)
  - HD637A05V0 ..... f=0.1 to 1.5MHz (V_{CC}=5V±10%)
  - HD637B05V0 ..... f=0.1 to 2MHz (V_{CC}=5V±10%)

■ SOFTWARE FEATURES

- Similar to HD6800 Instruction Set
- Byte Efficient Instruction Set
- Bit Manipulation
- Bit Test and Branch
- Versatile Interrupt Handling
- Powerful Indexed Addressing for Tables
- Full Set of Conditional Branches
- 10 Powerful Addressing Modes
- New Instructions — STOP, WAIT, DAA
- Compatible with HD6805 Family

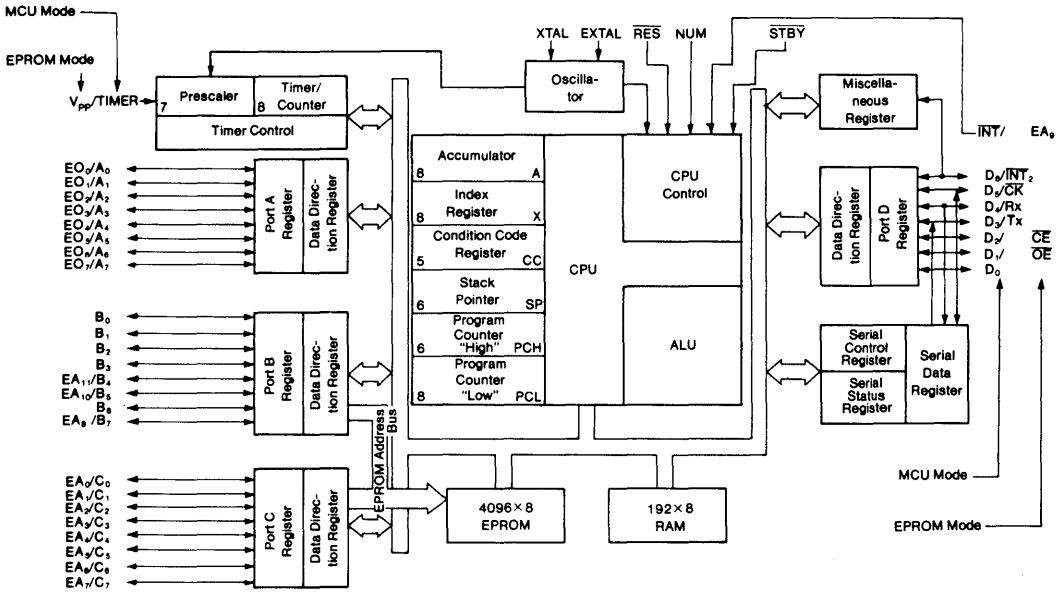
■ PIN ARRANGEMENT



(Top View)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ BLOCK DIAGRAM



INTRODUCTION OF THE RELATED DEVICES

- 8/16-bit Multi-chip Microcomputer
- 4-bit Single-chip Microcomputer
- IC Memory
- LCD Driver Series
- Gate Array
- CODEC/Filter Combo LSI
- Speech Synthesis LSI

8/16-BIT MULTI-CHIP MICROCOMPUTER

■ 8-BIT MULTI-CHIP MICROCOMPUTER

Division		Type No.	Old Type No.	LSI Characteristics: Process	Clock Frequency (MHz)	Supply Voltage (V)	Operating*** Temperature (°C)	Package†	Function	Compatibility
MPU		HD6803		NMOS	1.0	5.0	0 ~ +70	DP-40	Microprocessor +128 Bytes of RAM	MC6803
		HD6803-1			1.25					MC6803-1
		HD6303R		CMOS	1.0	5.0	0 ~ +70	DP-40 FP-54 CG-40	Microprocessor +128 Bytes of RAM	
		HD63A03R			1.5					
		HD63B03R			2.0					
		HD6303X		CMOS	1.0	5.0	0 ~ +70	DP-64S FP-80	Microprocessor +192 Bytes of RAM	
		HD63A03X			1.5					
		HD63B03X			2.0					
		HD6303Y		CMOS	1.0	5.0	0 ~ +70	DP-64S	Microprocessor +256 Bytes of RAM	
		HD63A03Y			1.5					
		HD63B03Y			2.0					
		HD6305X2		CMOS	1.0	5.0	0 ~ +70	DP-64S FP-64	Microprocessor +128 Bytes of RAM	
		HD63A05X2			1.5					
		HD63B05X2			2.0					
		HD6305Y2		CMOS	1.0	5.0	0 ~ +70	DP-64S FP-64	Microprocessor +256 Bytes of RAM	
		HD63A05Y2			1.5					
		HD63B05Y2			2.0					
		HD6800	HD46800D	NMOS	1.0	5.0	-20 ~ +75	DP-40	Microprocessor	MC6800
		HD68A00	HD468A00		1.5					MC68A00
		HD68B00	HD468B00		2.0					MC68B00
		HD6802	HD46802	NMOS	1.0	5.0	-20 ~ +75	DP-40	Microprocessor+Clock+128 Bytes of RAM	MC6802
		HD6802W		NMOS	1.0	5.0	-20 ~ +75	DP-40	Microprocessor+Clock+256 Bytes of RAM	
		HD6809		NMOS	1.0	5.0	-20 ~ +75	DP-40	High-End 8-Bit Microprocessor	MC6809
		HD68A09			1.5					MC68A09
		HD68B09			2.0					MC68B09
		HD6309**		CMOS	2.0	5.0	-20 ~ +75	DP-40	High-End 8-Bit Microprocessor	
					2.5					
					3.0					
		HD6809E		NMOS	1.0	5.0	-20 ~ +75	DP-40	High-End 8-Bit Microprocessor (External Clock Type)	MC6809E
		HD68A09E			1.5					MC68A09E
		HD68B09E			2.0					MC68B09E
		HD6309E*		CMOS	2.0	5.0	-20 ~ +75	DP-40	High-End 8-Bit Microprocessor (External Clock Type)	
					2.5					
					3.0					
Peripheral LSI	PIA	HD6821	HD46821	NMOS	1.0	5.0	-20 ~ +75	DP-40	Peripheral Interface Adapter	MC6821
		HD68A21	HD468A21		1.5					MC68A21
		HD68B21	HD468B21		2.0					MC68B21
		HD6321*		CMOS	1.0	5.0	-20 ~ +75	DP-40 FP-54	Peripheral Interface Adapter	
		HD63A21*			1.5					
		HD63B21*			2.0					
	PTM	HD6840		NMOS	1.0	5.0	-20 ~ +75	DP-28	Programmable Timer Module	MC6840
		HD68A40			1.5					MC68A40
		HD68B40			2.0					MC68B40
		HD6340*		CMOS	1.0	5.0	-20 ~ +75	DP-28	Programmable Timer Module	
		HD63A40*			1.5					
		HD63B40*			2.0					
	FDC	HD6843	HD46503S	NMOS	1.0	5.0	0 ~ +75	DP-40	Floppy Disk Controller	MC6843
		HD68A43	HD46503S-1		1.5					
	DMAC	HD6844	HD46504	NMOS	1.0	5.0	-20 ~ +75	DP-40	Direct Memory Access Controller	MC6844
		HD68A44	HD46504-1		1.5					MC68A44
		HD68B44	HD46504-2		2.0					MC68B44
	CRTC	HD6845	HD46505R	NMOS	1.0	5.0	-20 ~ +75	DP-40	CRT Controller (3.0MHz High-speed Display)	MC6845
		HD68A45	HD46505R-1		1.5					MC68A45
		HD68B45	HD46505R-2		2.0					MC68B45
		HD6845S	HD46505S	NMOS	1.0	5.0	-20 ~ +75	DP-40	CRT Controller (3.7MHz High-speed Display)	
		HD68A45S	HD46505S-1		1.5					
		HD68B45S	HD46505S-2		2.0					
	COMBO	HD6846	HD46846	NMOS	1.0	5.0	-20 ~ +75	DP-40	Combination ROM I/O Timer	MC6846
	ACIA	HD6850	HD46850	NMOS	1.0	5.0	-20 ~ +75	DP-24	Asynchronous Communications Interface Adapter	MC6850
		HD68A50	HD468A50		1.5					MC68A50
		HD6350		CMOS	1.0	5.0	-20 ~ +75	DP-24	Asynchronous Communications Interface Adapter	
		HD63A50			1.5					
		HD63850			2.0					
	SSDA	HD6852	HD46852	NMOS	1.0	5.0	-20 ~ +75	DP-24	Synchronous Serial Data Adapter	MC6852
		HD68A52	HD468A52		1.5					MC68A52
	ADU	HD46508		NMOS	1.0	5.0	-20 ~ +75	DP-40	Analog Data Acquisition Unit	
		HD46508-1			1.5					
		HD46508A			1.0					
		HD46508A-1			1.5					
	RTC	HD146818		CMOS	1.0	5.0	0 ~ +70	DP-24 FP-24	Real Time Clock Plus RAM	MC146818

* Preliminary ** Under development *** Wide Temperature Range (-40 ~ +85°C) version is available.
† DP; Plastic DIP, FP; Plastic Flat Package, CG: Glass-sealed Ceramic Leadless Chip Carrier

■ 16-BIT MULTI-CHIP MICROCOMPUTER

Division		Type No.	LSI Characteristics				† Package	Function	Compatibility
			Process	Clock Frequency (MHz)	Supply Voltage (V)	Operating Temperature (°C)			
MPU		HD68000-6	NMOS	6	5.0	0 ~ +70	DC-64	Microprocessor	MC68000L6
		HD68000-8		8					MC68000L8
		HD68000-10		10					MC68000L10
		HD68000-12		12.5					MC68000L12
		HD68000Y6		6			PGA-68		MC68000R6
		HD68000Y8		8					MC68000R8
		HD68000Y10		10					MC68000R10
		HD68000Y12		12.5					MC68000R12
		HD68000P6		6			DP-64		MC68000G6
		HD68000P8		8					MC68000G8
		HD68000PS6		6			DP-64S		—
		HD68000PS8		8					—
Peripheral LSI	DMAC	HD68450-4	NMOS	4	5.0	0 ~ +70	DC-64	Direct Memory Access Controller	MC68450L4
		HD68450-6		6					MC68450L6
		HD68450-8		8					MC68450L8
		HD68450-10		10					MC68450L10
		HD68450Y4		4			PGA-68		—
		HD68450Y6		6					—
		HD68450Y8		8					—
		HD68450Y10		10					—
	HDC	HD63463-4**	CMOS	4	5.0	0 ~ +70	DC-48	Hard Disk Controller	—
		HD63463-6**		6					—
		HD63463-8**		8					—
	ACRTC	HD63484-4*	CMOS	4	5.0	0 ~ +70	DC-64	Advanced CRT Controller	—
		HD63484-6*		6					—
		HD63484-8*		8					—

* Preliminary ** Under development † DP; Plastic DIP, DC; Ceramic DIP, PGA; Pin Grid Array

4-BIT SINGLE-CHIP MICROCOMPUTER

■ PMOS 4-BIT SINGLE-CHIP MICROCOMPUTER HMCS40 SERIES

	Family Name (Type Name)			HMCS42 (HD38702)	HMCS43 (HD38750, HD38755)	HMCS44A (HD38800, HD38805)	HMCS45A (HD38820, HD38825)	HMCS47A (HD38870)
LSI Characteristics	Supply Voltage		(V)	-10	-10	-10	-10	-10
	Power Dissipation (typ.)		(mW)	100	100	150	150	250
	Max. I/O Terminal Voltage		(V)	-50	-50	-50	-50	-50
	Operating Temperature Range *1		(°C)	-20 to +75	-20 to +75	-20 to +75	-20 to +75	-20 to +75
	Package			DP-28, DP-28S	DP-42, DP-42S	DP-42, DP-42S	FP-54, DP-64S	FP-54, DP-64S
Functions	Memory	ROM	(bits)	512 x 10 32 x 10 *2	1,024 x 10 64 x 10 *2	2,048 x 10 128 x 10 *2	2,048 x 10 128 x 10 *2	4,096 x 10
		RAM	(bits)	32 x 4	80 x 4	160 x 4	160 x 4	256 x 4
	Registers			4	6	8	6	6
	Stack Registers			2	3	4	4	4
	I/O Ports	4-Bit Data Input		22	32	32	44	44
		4-Bit Data Output		4 x 1	4 x 1	—	—	—
		Discrete Output		4 x 2	4 x 2	—	4 x 1	4 x 1
		4-Bit Data Input/Output		1 x 6	1 x 12	—	—	—
		Discrete Input/Output		—	4 x 1	4 x 4	4 x 6	4 x 6
				1 x 4	1 x 4	1 x 16	1 x 16	1 x 16
	Interrupts	External		—	2	2	2	2
		Timer/Counter		—	1	1	1	1
	Instructions	Number of Instructions		51	71	71	71	71
		Cycle Time	(μs)	10	10	10	10	10
	Built-in Clock Pulse Generator			Yes				
	Power on Reset			Yes	Yes	Yes	Yes	Yes
	Battery Back-up			—	RAM Hold	RAM Hold	RAM Hold	RAM Hold
Evaluation Chip				HD38750E HD44850E HD44857E	HD38750E HD44850E HD44857E	HD44850E HD44857E	HD44850E HD44857E	HD44857E

*1 Wide Temperature Range (-40 to +85°C) version is available.
*2 Pattern Memory

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ CMOS 4-BIT SINGLE-CHIP MICROCOMPUTER HMCS40 SERIES

	Family Name (Type Name)			HMCS42CL (HD44708) HMCS42C (HD44700)		HMCS43CL (HD44758) HMCS43C (HD44750)		HMCS44CL (HD44808) HMCS44C (HD44801)	
LSI Characteristics	Supply Voltage		(V)	3/5		3/5		3/5	
	Power Dissipation (typ.)		(mW)	0.23/1.5		0.24/1.5		0.32/2	
	Max. I/O Terminal Voltage		(V)	V_{CC}+0.3		V_{CC}+0.3		V_{CC}+0.3	
	Operating Temperature Range *1		(°C)	-20 to +75		-20 to +75		-20 to +75	
	Package			DP-28, DP-28S		DP-42, DP-42S		DP-42, DP-42S	
Functions	Memory	ROM	(bits)	512 x 10 32 x 10*3		1,024 x 10 64x10*3		2,048 x 10 128x10*3	
		RAM	(bits)	32 x 4		80 x 4		160 x 4	
	Registers			4		6		8	
	Stack Registers			2		3		4	
	I/O Ports	4-Bit Data Input		22	4 x 1	32	4 x 1	32	–
		4-Bit Data Output			4 x 2		4 x 2		–
		Discrete Output			1 x 6		1 x 12		–
		4-Bit Data Input/Output			–		4 x 1		4 x 4
		Discrete Input/Output			1 x 4		1 x 4		1 x 16
	Interrupts	External		–		2		2	
		Timer/Counter		–		1		1	
	Instructions	Number of Instructions		51		71		71	
		Cycle Time	(μs)	20/10		20/10		20/10	
	Built-in Clock Pulse Generator			Yes					
	Power on Reset			No/Yes		No/Yes		No/Yes	
	Battery Back-up			Halt		RAM Hold		Halt	
Evaluation Chip				HD44850E HD44857E		HD44850E HD44857E		HD44850E HD44857E	

*1 Wide Temperature Range (-40 to +85°C) version is available.
*2 Pattern Memory
*3 LCD DRIVE FUNCTION

LCD Drive	Common	4
	Segment	32
	Duty	Static, 1/2, 1/3, 1/4
	Bias	1/2, 1/3
Display Capability		4x32 Matrix (1/4 Duty)

Expandable using the LCD Driver HD44100H.

HMCS45CL (HD44828) HMCS45C (HD44820)	HMCS46CL (HD44848) HMCS46C (HD44840)	HMCS47CL (HD44868) HMCS47C (HD44860)	LCD-III*3 (HD44795, HD44790)	LCD-IV*3 (HD613901)
3/5	3/5	3/5	3/5	3/5
0.32/2	0.32/4	0.32/4	0.36/2.4	0.9/5.0
V_{CC}+0.3	V_{CC}+0.3	V_{CC}+0.3	V_{CC}+0.3	V_{CC}+0.3
-20 to +75	-20 to +75	-20 to +75	-20 to +75	-20 to +75
FP-54, DP-64S	DP-42, DP-42S	FP-54, DP-64S	FP-80	FP-80
2,048 x 10 128x10*2	4,096 x 10	4,096 x 10	2,048 x 10 128 x 10*2	4,096 x 10
160 x 4	256 x 4	256 x 4	160 x 4	256 x 4
6	8	6	6	6
4	4	4	4	4
44 –	32 –	44 –	32 4 x 1	32 4 x 1
4 x 1	–	4 x 1	4 x 1	4 x 1
–	–	–	–	–
4 x 6	4 x 4	4 x 6	4 x 2	4 x 2
1 x 16	1 x 16	1 x 16	1 x 16	1 x 16
2	2	2	2	2
1	1	1	1	1
71	71	71	71	71
20/10	20/5	20/5	20/10	20/5
Yes				
No/Yes	No/Yes	No/Yes	Yes	No
Halt	Halt	Halt	Halt	Halt
HD44850E HD44857E	HD44857E	HD44857E	HD44797E	HD44797E

■ CMOS 4-BIT SINGLE-CHIP MICROCOMPUTER HMCS400 SERIES

Family Name (Type Name)				HMCS404C (HD614042)	HMCS404AC* (HD614048)	HMCS404CL (HD614045)	HMCS412C** (HD614120)
LSI Characteristics	Supply Voltage		(V)	4 to 6	4.5 to 6.0	2.7 to 6.0	4 to 6
	Power Dissipation		(max.) (mW)	18	27	9	12
	Max. I/O Terminal Voltage		(V)	V_{CC}-40	V_{CC}-40	V_{CC}-40	V_{CC}-40
	Operating Temperature Range		(°C)	-20 to +75	-20 to +75	-20 to +75	-20 to +75
	Package			FP-64, DP-64S	FP-64, DP-64S	FP-64, DP-64S	DP-42
Functions	Memory	ROM	(bits)	4096 x 10	4096 x 10	4096 x 10	2048 x 10
		RAM	(bits)	256 x 4	256 x 4	256 x 4	160 x 4
	Registers			7	7	7	7
	Stack Registers			16	16	16	16
	I/O Ports	4-Bit Input		58 — 4 x 1, 2 x 1	58 — 4 x 1, 2 x 1	58 — 4 x 1, 2 x 1	36 — 1 x 1
		4-Bit Output		58 — 4 x 4	58 — 4 x 4	58 — 4 x 4	36 — 4 x 1
		4-Bit Input/Output		58 — 4 x 5	58 — 4 x 5	58 — 4 x 5	36 — 4 x 4
		1-Bit Input/Output		58 — 1 x 16	58 — 1 x 16	58 — 1 x 16	36 — 1 x 15
	Interrupts	External		2	2	2	2
		Timer/Counter		2	2	2	1
		Serial Interface		1	1	1	–
	Instructions	Number of Instructions		99	99	99	98
		Cycle Time	(μs)	2	1.33	4	2
	Built-in Clock Pulse Generator			Yes (External drive is possible)			
	Others			Low Power Dissipation Mode (Stop mode, Stand-by mode)			

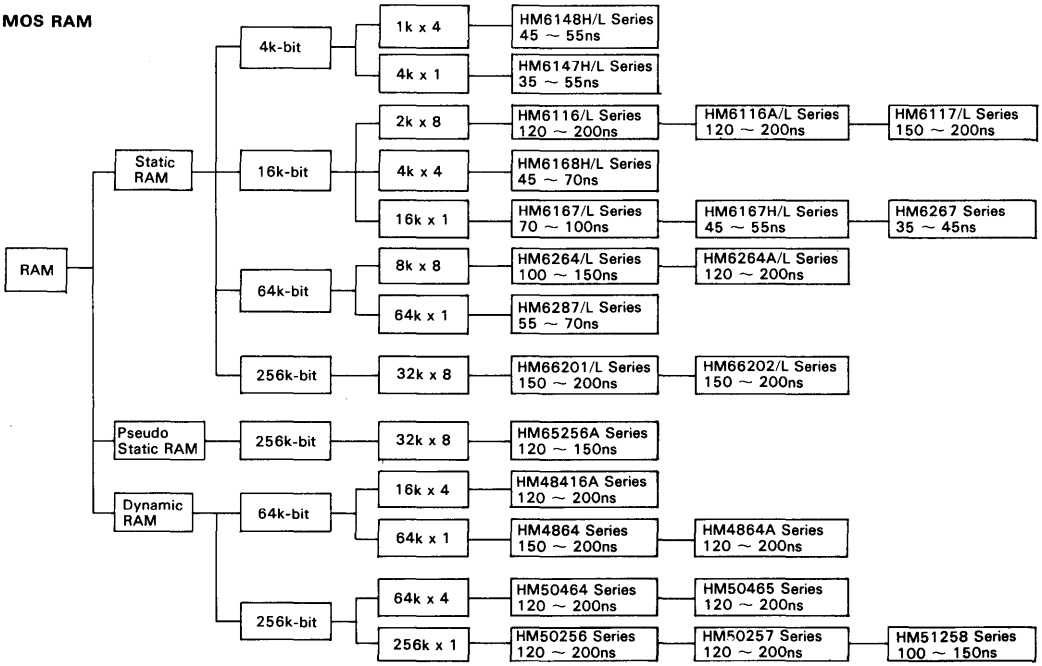
Family Name (Type Name)				HD614P080S†	HD614P180**†
LSI Characteristics	Supply Voltage		(V)	4.5 to 5.5	4.5 to 5.5
	Power Dissipation		(max.) (mW)	27	27
	Max. I/O Terminal Voltage		(V)	V_{CC}-40	V_{CC}-40
	Operating Temperature Range		(°C)	-20 to +75	-20 to +75
	Package			DC-64SP	DC-42
Functions	Memory	ROM	(bits)	∘4,096-word x 10-bit with standard EPROM 2764 ∘8,192-word x 10-bit with standard EPROM 2728	°4,096-word x 10-bit with standard EPROM 2764 °8,192-word x 10-bit with standard EPROM 27128
		RAM	(bits)	576 x 4	576 x 4
	Registers			7	7
	Stack Registers			16	16
	I/O Ports	4-Bit Input		58 — 4 x 1, 2 x 1	36 — 1 x 1
		4-Bit Output		58 — 4 x 4	36 — 4 x 1
		4-Bit Input/Output		58 — 4 x 5	36 — 4 x 4
		1-Bit Input/Output		58 — 1 x 16	36 — 1 x 15
	Interrupts	External		2	2
		Timer/Counter		2	1
		Serial Interface		1	–
	Instructions	Number of Instructions		99	98
		Cycle Time	(μs)	1.33	2
	Built-in Clock Pulse Generator			Yes (External drive is possible)	
	Others			Low Power Dissipation Mode (Stop mode, Stand-by mode)	

* Preliminary
** Under development
† EPROM on the Package Type

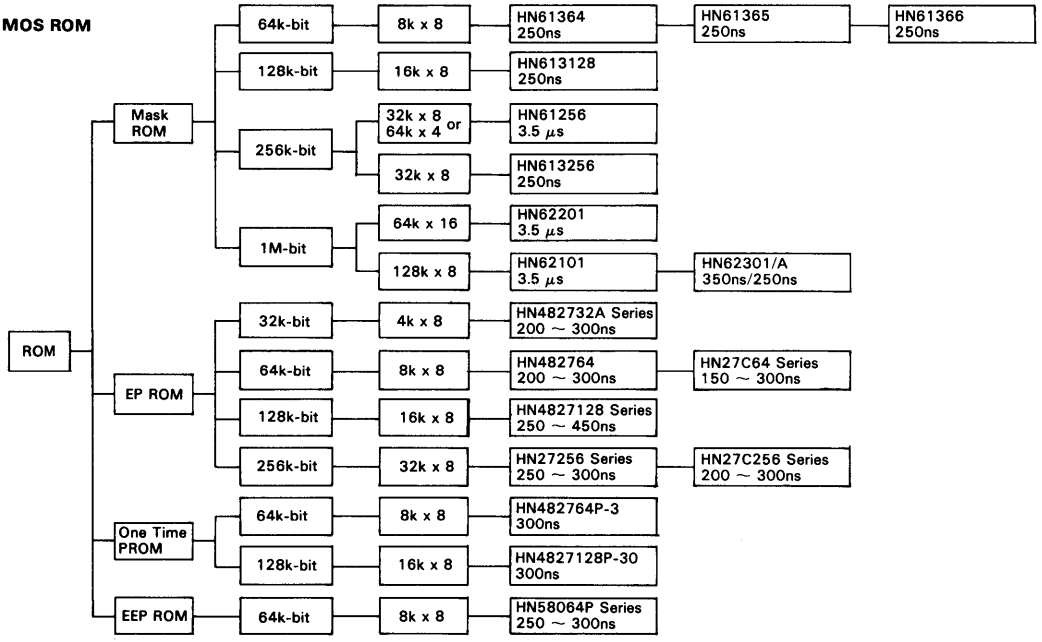
HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

IC MEMORY

■ **MOS RAM**

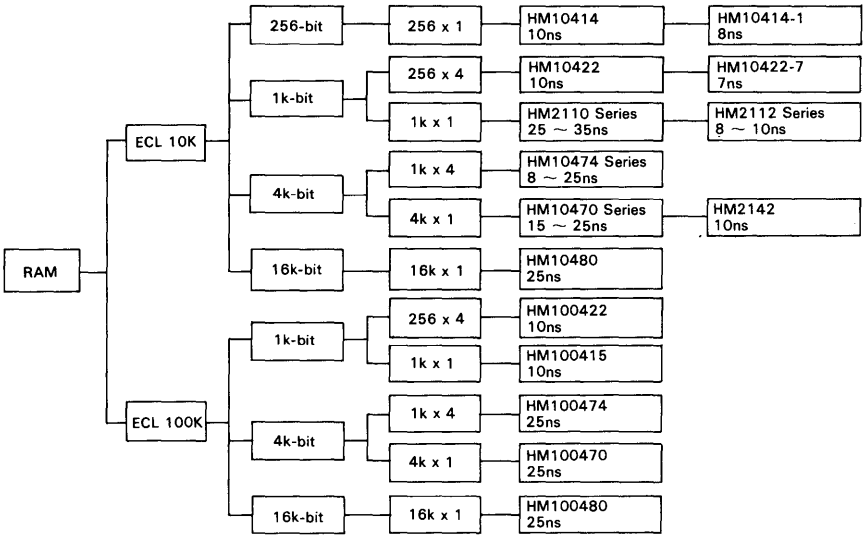


■ **MOS ROM**



HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

■ **BIPOLAR RAM**


RAM
ECL 10K
ECL 100K
256-bit
1k-bit
4k-bit
16k-bit
1k-bit
4k-bit
16k-bit
256 x 1
256 x 4
1k x 1
1k x 4
4k x 1
16k x 1
256 x 4
1k x 1
1k x 4
4k x 1
16k x 1
HM10414 10ns
HM10422 10ns
HM2110 Series 25 ~ 35ns
HM10474 Series 8 ~ 25ns
HM10470 Series 15 ~ 25ns
HM10480 25ns
HM100422 10ns
HM100415 10ns
HM100474 25ns
HM100470 25ns
HM100480 25ns
HM10414-1 8ns
HM10422-7 7ns
HM2112 Series 8 ~ 10ns
HM2142 10ns

LCD DRIVER SERIES

■ LCD DRIVER SERIES CHARACTERISTICS

Item \ Use			General			Segment Display	
LSI Characteristics	Type Number		HD44100H	HD61100A	HD61200	HD61602	HD61603
	Process		CMOS	CMOS	CMOS	CMOS	CMOS
	Supply Voltage	(V)	5*1	5*1	5*1	3 ~ 5*1	3 ~ 5*1
	Operating Temperature	(°C)	-20 ~ +75*2	-20 ~ +75	-20 ~ +75	-20 ~ +75	-20 ~ +75
	Package		FP-60	FP-100	FP-100	FP-80	FP-80
	Power Dissipation	(mW)	5.0	5.0	5.0	0.5 (5V)	0.5 (5V)
Functions	Memory — ROM	(bits)	–	–	–	–	–
	Memory — RAM	(bits)	–	–	–	51 x 4	64 x 1
	I/O — Interface (CPU)		8	8	8	14	10
	I/O — Interface (Driver IC)		2	2	2	–	–
	I/O — Interface (External ROM, RAM)		–	–	–	–	–
	Number of Instruction		–	–	–	4	4
	LCD Driver — Common		40	80	80	4	1
	LCD Driver — Segment					51	64
	LCD Driver — Duty		Free (N)	Free (N)	Free (N)	Static, 1/2, 1/3, 1/4	Static
	Display Capability		N x 40 Matrix (1/N Duty)	N x 80 Matrix (1/N Duty)	N x 80 Matrix (1/N Duty)	204 Segment (1/4 Duty)	64 Segment
Comment			SR type	SR type	SR type		

Item \ Use			Graphic Display				
LSI Characteristics	Type Number		HD44102CH	HD44103CH	HD44105H	HD61102	HD61202
	Process		CMOS	CMOS	CMOS	CMOS	CMOS
	Supply Voltage	(V)	5*1	5*1	5*1	5*1	5*1
	Operating Temperature	(°C)	-20 ~ +75	-20 ~ +75	-20 ~ +75	-20 ~ +75	-20 ~ +75
	Package		FP-80	FP-60	FP-60	FP-100	FP-100
	Power Dissipation	(mW)	2.5	4.0	4.0	3.0	2.5
Functions	Memory — ROM	(bits)	–	–	–	–	–
	Memory — RAM	(bits)	200 x 8	–	–	512 x 8	512 x 8
	I/O — Interface (CPU)		21	6	6	21	21
	I/O — Interface (Driver IC)		–	5	5	–	–
	I/O — Interface (External ROM, RAM)		–	–	–	–	–
	Number of Instruction		6	–	–	7	7
	LCD Driver — Common		–	20	32	–	–
	LCD Driver — Segment		50	–	–	64	64
	LCD Driver — Duty		1/8, 1/12, 1/16, 1/24, 1/32	1/8, 1/12, 1/16, 1/24, 1/32	1/8,1/12,1/16,1/24, 1/32, 1/48, 1/64	1/64	1/64
	Display Capability		32 x 50 Dots (1/32 Duty)	–	–	64 x 64 Dots (1/64 duty)	64 x 64 Dots (1/64 duty)
Comment			Suitable common driver is HD44105H or HD44103CH			Suitable common driver is HD61103A	Suitable common driver is HD61203

*1: Except Power Supply for LCD.
*2: -40 ~ +85°C (Special Request). Please contact Hitachi Agents.
*3: CG; Character Generator.

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

Character Display	
HD44780 (LCD-II)	HD43160AH
CMOS	CMOS
5*1	5*1
-20 ~ +75*2	-20 ~ +75
FP-80	FP-54
1.75	10.0
7200 (CG)*3	6240 (CG)*3
80 x 8/64 x 8 (CG)*3	80 x 8
11	21
4	5
–	18
11	6
16	–
40	–
1/8, 1/11, 1/16	1/8, 1/12, 1/16
16 Digits (5 x7 Dots 1/16 Duty)	–
Expandable to 80 Digits using HD44100H	Display to 80 Digits using HD44100H

Graphic Display			
HD61103A	HD61203	HD61830	HD61830B
CMOS	CMOS	CMOS	CMOS
5*1	5*1	5*1	5*1
-20 ~ +75	-20 ~ +75	-20 ~ +75	-20 ~ +75
FP-100	FP-100	FP-60	FP-60
5.0	5.0	30.0	30.0
–	–	7360 (CG)*3	7360 (CG)*3
–	–	(external 65536 x 8)	(external 65536 x
6	6	13	13
5	5	9	9
–	–	33	33
–	–	12	12
64	64	–	–
–	–	–	–
1/48, 1/64 1/96, 1/128	1/48, 1/64, 1/96, 1/128	1/1 ~ 1/128	Static 1/1 ~ 1/128
–	–	–	–
		Display to 524288 Dots using HD44100H or HD61100A.	

GATE ARRAY

CMOS Gate Array HD61J/HD61K/HD61L/HD61MM Series

■ **FEATURES**

- Fast operation
  Internal gate (2-input NAND, FO=3, AL=3mm) . . 3.5ns typ
  Input buffer (FO=3, AL = 3mm) . . . . . 9ns typ
  Output buffer (C_L=50pF) . . . . . 20ns typ
  Memory access time (HD61MM) . . . . . 60ns typ
- Low power dissipation
  At 10MHz operation (Internal gate) . . . . . 130μW/gate typ
- Abundant input and output configuration
  Allocation of all pins except power supply pins to input/output/input-output
  Output can be CMOS/open drain/3-state
- Memory on-chip (HD61MM)
  Flexibility of memory capacity and word organization
  Selection of single port/dual port memory
- Wide operation temperature range
  −20 to +75°C
- Wide package selection
  Especially plastic packages with high pin count DILP64/FPP100
- Powerful design support
  User-Defined-Macro
  Test pattern evaluation with fault simulator
  Design support at local Design Center
- Quick turn around time and reasonable development cost

■ **LINE UP**

		HD61J	HD61K	HD61L	HD61MM
Gate count		504	1080	1584	2496
I/O pin count		50	68	68	104
RAM on chip		—	—	—	available
Package	DP28	○	○	○	—
	DP40	*	*	○	*
	DP42	○	○	○	—
	DP64	—	○	○	○
	FP54	○	—	—	—
	FP80	—	○	○	—
	FP100	—	—	—	*
	DC28	○	○	○	*
	DC40	○	○	○	*
	PGA72	—	—	○	—
	PGA120	—	—	—	*
Power supply pin		4			4 8*

*Under development

Bi-CMOS Gate Array HD27K/HD27L/HD27P/HD27Q Series

■ **FEATURES**

- High speed with super low power dissipation . . . . . .
  - Internal gate: 4.0ns (Fan out=3) @0.05mW
  - Input buffer: 5.0ns (Fan out=3) @2.6mW
  - Output buffer: 8.0ns (C_L=15pF) @2.6mW
- LS TTL compatible input/output . . . . . .
  - Selectable totem-pole/3-state/open collector output
  - I_{OL}=8mA: Capable of driving 20 LS TTL's
- Output buffer can construct logic functions.
  - Saves gate stages.
- A variety of macrocell library
  - Internal gate: 44
  - Output buffer: 9
- A variety of reliable package
  - Plastic DIP 16 to 64 pins
  - Plastic FP 60 to 100 pins (under development)
- A variety of DA system support
  - Only logic diagrams and test patterns needed as an interface with the user.
- Short development time

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

	Number of gates			Number of V_{CC} and GND pins	Package	
	Internal gate (2-input NAND)	Input buffer	Output buffer		DIP (Plastic)	FP* (Plastic)
HD27K	200	18	18	2	16, 20, 28, 40 pins	–
HD27L	528	30	30	4	28, 42, 64 pins	60 pins
HD27P	966	40	40	4	28, 42, 64 pins	60, 80, 100 pins
HD27Q	1530	50	50	4	28, 42, 64 pins	60, 80, 100 pins

*Under development

CODEC/FILTER COMBO LSI

SERIES	TYPE	COMP. LAW	POWER (Typ.)	CLOCK			DECODER SHIFT	INPUT AMP	OUTPUT AMP	
				INTERNAL CLOCK	SYNC/ASYNC OPERATION	PCMBIT CLOCK RATE			TYPE	MIN LOAD
44230	HD44231B	A	60mW	DEVIDER INCLUDED	SYNC. ONLY	1536/ 1544/ 2048kHz	–	ADJUST-ABLE USING 2 RESIST	SINGLE ENDED	3 kΩ
	HD44232B	μ								
	HD44233B	A			BOTH					
	HD44234B	μ								
	HD44235	A	50mW	PLL INCLUDED	SYNC. ONLY	64- 2048kHz	–			1.2 kΩ
	HD44236	μ					○			
	HD44237	A			BOTH		–			
	HD44238	μ					○			
44230C	HD44231C	A	60mW	DEVIDER INCLUDED	SYNC. ONLY	1536/ 1544/ 2048kHz	–	FULLY UNCOMM-ITED OP-AMP		600 Ω
	HD44232C	μ								
	HD44233C	A			BOTH					
	HD44234C	μ								
	HD44235C	A		PLL INCLUDED	SYNC. ONLY	64- 2048kHz	–			
	HD44236C	μ					○			
	HD44237C	A			BOTH		–			
	HD44238C	μ					○			
44240C	HD44247C	A	80mW	PLL INCLUDED	BOTH	64- 2048kHz	–	SAME AS ABOVE	PUSH-PULL	600 Ω
	HD44248C	μ					○			

SPEECH SYNTHESIS LSI

Type	CMOS 1-chip System
Device	HD61885/7 (Speech Synthesizer) HD44881 (128k-bit Expanding ROM)
System	PARCOR
Voice channel model	10 steps digital filter
Sampling frequency	10 kHz
Bit rate (b/s)	1250 ~ 9900
Frame period (ms)	10/20
Variable speaking speed	–25%, 0, +25%
Speaking time	10 ~ 20 sec (internal ROM)

HITACHI
Hitachi America Ltd. • 2210 O'Toole Ave. • San Jose, CA 95131 • (408) 435-8300

HITACHI AMERICA, LTD.
SEMICONDUCTOR AND IC DIVISION

HEADQUARTERS

Hitachi, Ltd.
New Marunouchi Bldg., 5-1,
Marunouchi 1-chome
Chiyoda-ku, Tokyo 100, Japan
Tel: Tokyo (03) 212-1111
Telex: J22395, J22432, J24491,
J26375 (HITACHY)
Cable: HITACHY TOKYO

U.S. SALES OFFICE

Hitachi America, Ltd.
Semiconductor and IC Division
2210 O'Toole Avenue
San Jose, CA 95131
Tel: 408-435-8300
Telex: 17-1581
Twx: 910-338-2103
Fax: 408-435-2748
Fax: 408-435-2749
Fax: 408-435-2782

REGIONAL OFFICES

MID-ATLANTIC REGION

Hitachi America, Ltd.
1700 Galloping Hill Rd.
Kenilworth, NJ 07033
201/245-6400

NORTHEAST REGION

Hitachi America, Ltd.
5 Burlington Woods Drive
Burlington, MA 01803
617/229-2150

SOUTH CENTRAL REGION

Hitachi America, Ltd.
Two Lincoln Centre, Suite 865
5420 LBJ Freeway
Dallas, TX 75240
214/991-4510

NORTH CENTRAL REGION

Hitachi America, Ltd.
500 Park Blvd., Suite 415
Itasca, IL 60143
312/773-4864

NORTHWEST REGION

Hitachi America, Ltd.
2210 O'Toole Avenue
San Jose, CA 95131
408/435-2200

SOUTHWEST REGION

Hitachi America, Ltd.
18300 Von Karman Avenue, Suite 730
Irvine, CA 92715
714/553-8500

SOUTHEAST REGION

Hitachi America, Ltd.
4901 N.W. 17th Way, Suite 302
Fort Lauderdale, FL 33309
305/491-6154

DISTRICT OFFICES

- Hitachi America, Ltd.
  3800 W. 80th Street, Suite 1050
  Bloomington, MN 55431
  612/896-3444

- Hitachi America, Ltd.
  80 Washington St., Suite 302
  Poughkeepsie, NY 12601
  914/485-3400

- Hitachi America, Ltd.
  6 Parklane Blvd., #558
  Dearborn, MI 48126
  313/271-4410

- Hitachi America, Ltd.
  6161 Savoy Dr., Suite 850
  Houston, TX 77036
  713/974-0534

- Hitachi (Canadian) Ltd.
  2625 Queensview Dr.
  Ottawa, Ontario, Canada K2A 3Y4
  613/596-2777

- Hitachi America, Ltd.
  401 Harrison Oaks Blvd., Suite #317
  Cary, NC 27513
  919/481-3908

We make things possible

Hitachi America, Ltd.
Semiconductor and IC Division
2210 O'Toole Avenue, San Jose, CA 95131
1-408-435-8300

1SM

Printed in U.S.A.