TS 800 HARDWARE THEORY OF OPERATION

This Theory of Operation describes the hardware layout, functions and operations.

1.0 INTRODUCTION

The TS 800 is a satellite user station which is designed to be used along with a TeleVideo Systems service processor (TS806 or TS816). Logically, TS800 is divided into two portions, a smart terminal portion and a Z-80 computer portion. The terminal portion is identical to the 950 terminal except that some of the circuits are modified to communicate with the Z-80 computer portion; (Block Diagram is shown in Figure 1). TS-80 has two ports; one of the ports is the RS-232 printer port which has the same functions as in the 950 terminal. The second port is a high speed RS-422 port which is used to interface to the TeleVideo Systems service processor. Since the TS800 does not have local file storage, the operating system and user programs are retreived from the TeleVideo Systems service processor through the RS-422 communication link.

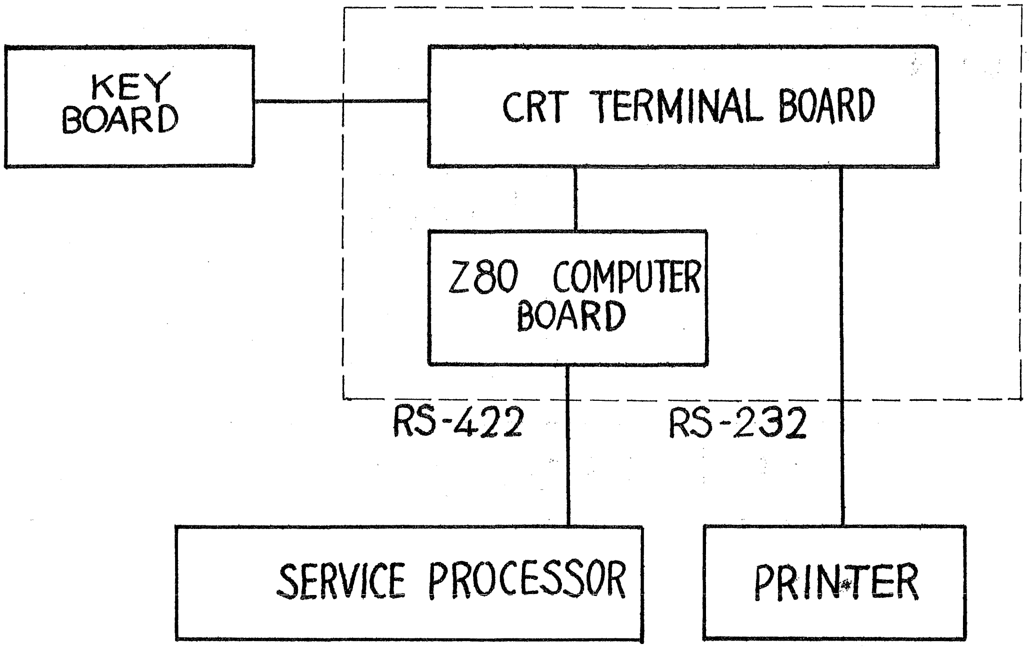


FIGURE 1 SYSTEM BLOCK DIAGRAM

2.0 Z-80 COMPUTER BOARD DESCRIPTION

This section describes only the Z-80 computer portion hardware. For the TS800 terminal portion, please refer to Section six. The Z-80 computer portion contains four Z-80A family chips (the CPU, CTC, SIO and DMA); 64K bytes of dynamic RAM, 4K bytes of ROM, 1K bytes of static RAM and all the required control logic. The block diagram of the Z-80 computer board is shown in Figure 2.

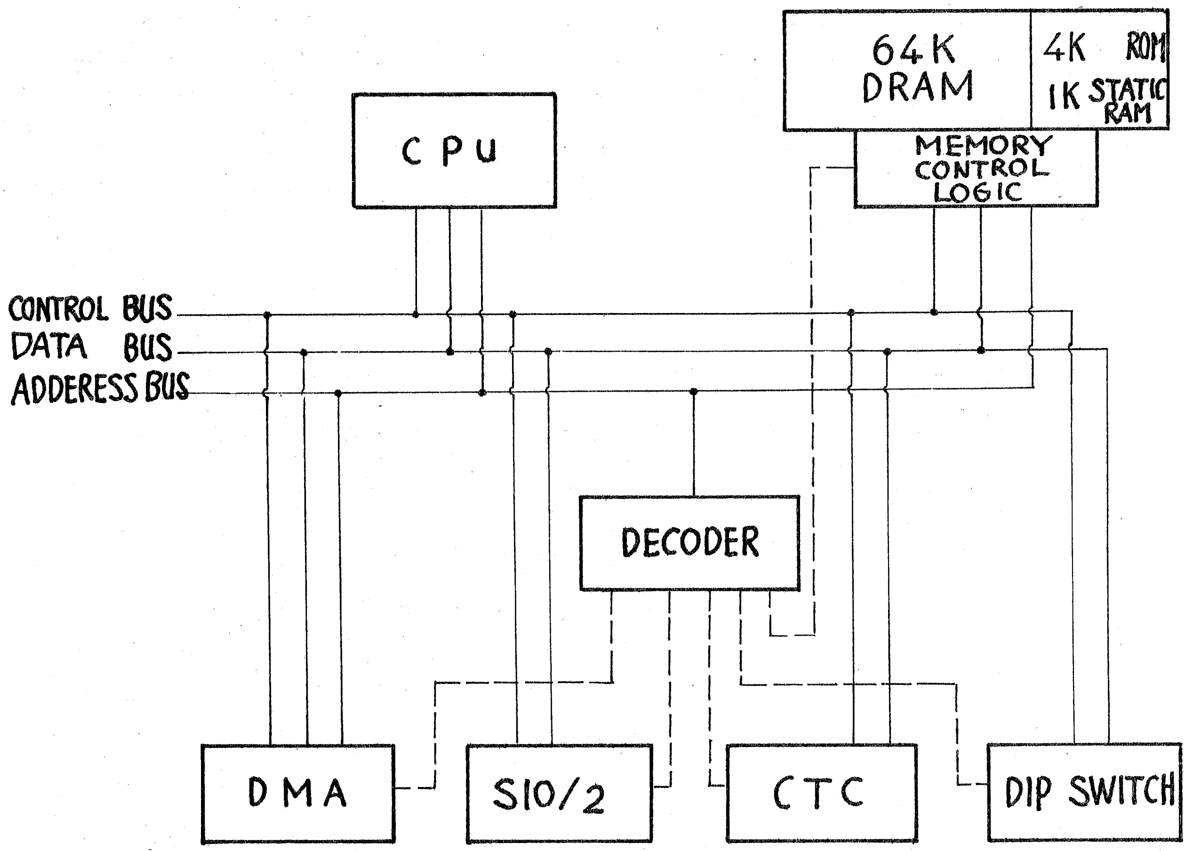


FIGURE 2 BLOCK DIAGRAM OF TS-80 COMPUTER BOARD

All the Z-80A chips are driven by a 4 MHz system clock, which is also used to generate the necessary timings for the memory control logic. Z80A-CPU has a 16 bit memory address bus. The lower eight address lines are also used to address up to 256 input/output devices. During instruction fetch cycles, the

CPU sends out refresh addresses and the refresh enable signal to the memory control which refreshes 64K bytes of dynamic RAM. The TS800 has five input/output devices which are the DMA, SIO, CTC, DIP switch, LED memory latch. The data bus is an eight bit bidirectional bus. The third bus is the control bus which includes the following signals: M1-, MREQ-, IORQ-, RD-, WR-, RFSH-, WAIT-, INT-, RESET-, BUSRQ-, and BUSAK-. The Z80A-CTC is a four channel programmable, timing generating chip. It can be programmed as a timer or as a counter. The main function of this CTC in TS800 is to provide transmit and receive clocks for the RS-422 communication interface and communication between the Z80 computer board and the terminal portion of the board.

The Z80A-SIO is a dual channel serial input/output controller. Channel A is programmed to use SDLC mode for the RS-422 interface (fixed 800K bits/sec) and channel B is programmed to use asynchronous mode for communication between Z-80 portion and terminal portion (fixed 19.2K baud).

The Z80A-DMA is a direct memory access controller. When it is enabled, it controls the data flow between the memory and channel A of the SIO in this system. In order to guarantee CPU refreshing dynamic memory within every 2 msec, the DMA is programmed to be used in the burst transfer mode in which the DMA releases all buses when Channel A of the SIO is not requesting any data transfer. The TS800 computer board uses four switch positions for system options (positions 1, 2, 3 and 4 of Sw 1). The functions of these four positions is software dependent. The main memory in TS800 computer board contains eight of 64K X 1 dynamic memory chips. Besides these 64K bytes of dynamic memory, 4K bytes of ROM and 1K bytes of static RAM are used for initialization, "boot" up and system diagnostics every time power is turned on. A detailed description of all memory will be covered in the next section.

3.0 FUNCTION OF THE SYSTEM

The TS800 clock generator generates a 8 MHz and 4 MHz clock frequencies which are supplied to the CPU, SIO, CTC, DMA and memory control logic. After power is first turned on, the system is reset to the idle state. All system chips must be programmed individually to the correct operating mode. CTC channel 0 and channel 1 supply different clock frequencies to SIO channel A and channel B, respectively. SIO channel A is used for RS-422 interface and SIO channel B is used for serial interface between Z80 computer portion and terminal portion of the board.

The DMA is used during data communication between the TS800 and TeleVideo Systems service processors. During DMA operations, the DMA uses all buses directly for high speed data transfer.

There are three different kinds of memory devices in the TS800 computer board (dynamic RAM, static RAM and ROM .). After power is first turned on, a latch will automatically select 4K bytes of ROM as the memory space accessible by CPU. This memory space is shown in Figure 3.

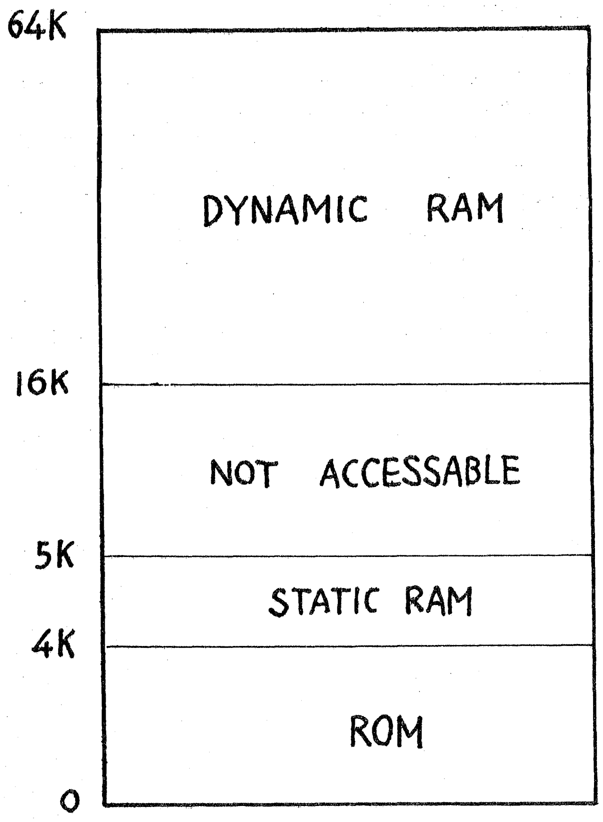


FIGURE 3 MEMORY CONFIGURATION AFTER RESET

The ROM locations from 0 to 4K contain all system boot up, initialization and diagnostic programs. The static RAM locations from 4K to 5K are used as a scratch pad area for the "ROM" programs. The operating system will be loaded into the higher dynamic RAM area between 16K to 64K area. Upon completion of loading the operating system, a single instruction (out [04H], A) will be executed and the memory space is switched to all 64K bytes of dynamic RAM as shown in Figure 4.

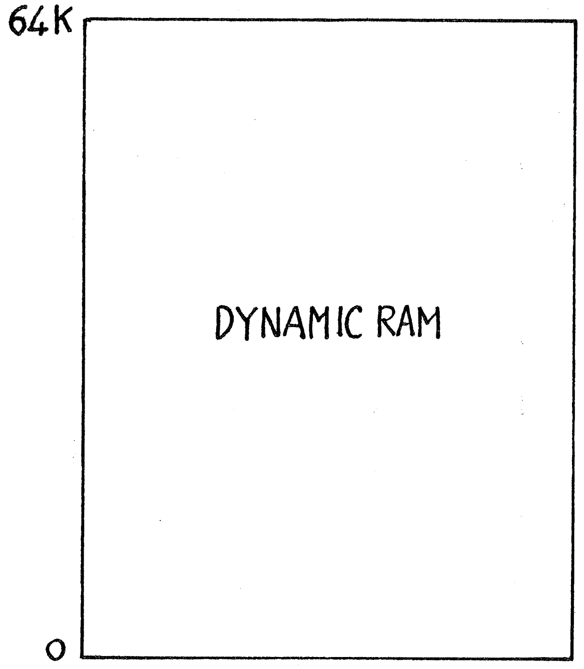


FIGURE 4 MEMORY CONFIGURATION AFTER INSTRUCTION 'OUT 04H, A'

The functions of the memory control logic is to supply necessary read/write signals, address latching signals and refresh signal to the dynamic RAM. Because of the difference in speed of the three types of memories, one cycle of 'wait state' is inserted to the read/write cycle when the CPU accesses the ROM or the static RAM location.

4.0 OPERATION OF THE SYSTEM

The clock generating circuit generates a 4 MHz system clock which supplies all the Z-80A family chips and memory control circuit. A counter 74LS161A (A62) is used to divide down the 16 MHz clock from the oscillator to 4 MHz clock. Clock output voltage is a critical signal for all the Z-80A family chips. The high output voltage of the clock must be between 4.4 V and 5.3 V and the low output voltage of the clock must be between -0.3V to 0.45V as shown in Figure 5. In order to satisfy this requirement, a transistor (2N2907) is used to pull the clock output high voltage to about 5V.

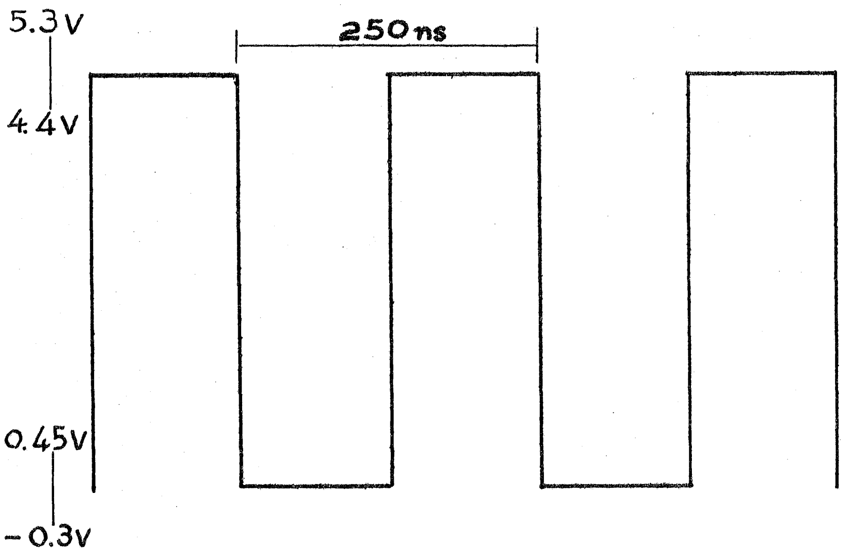
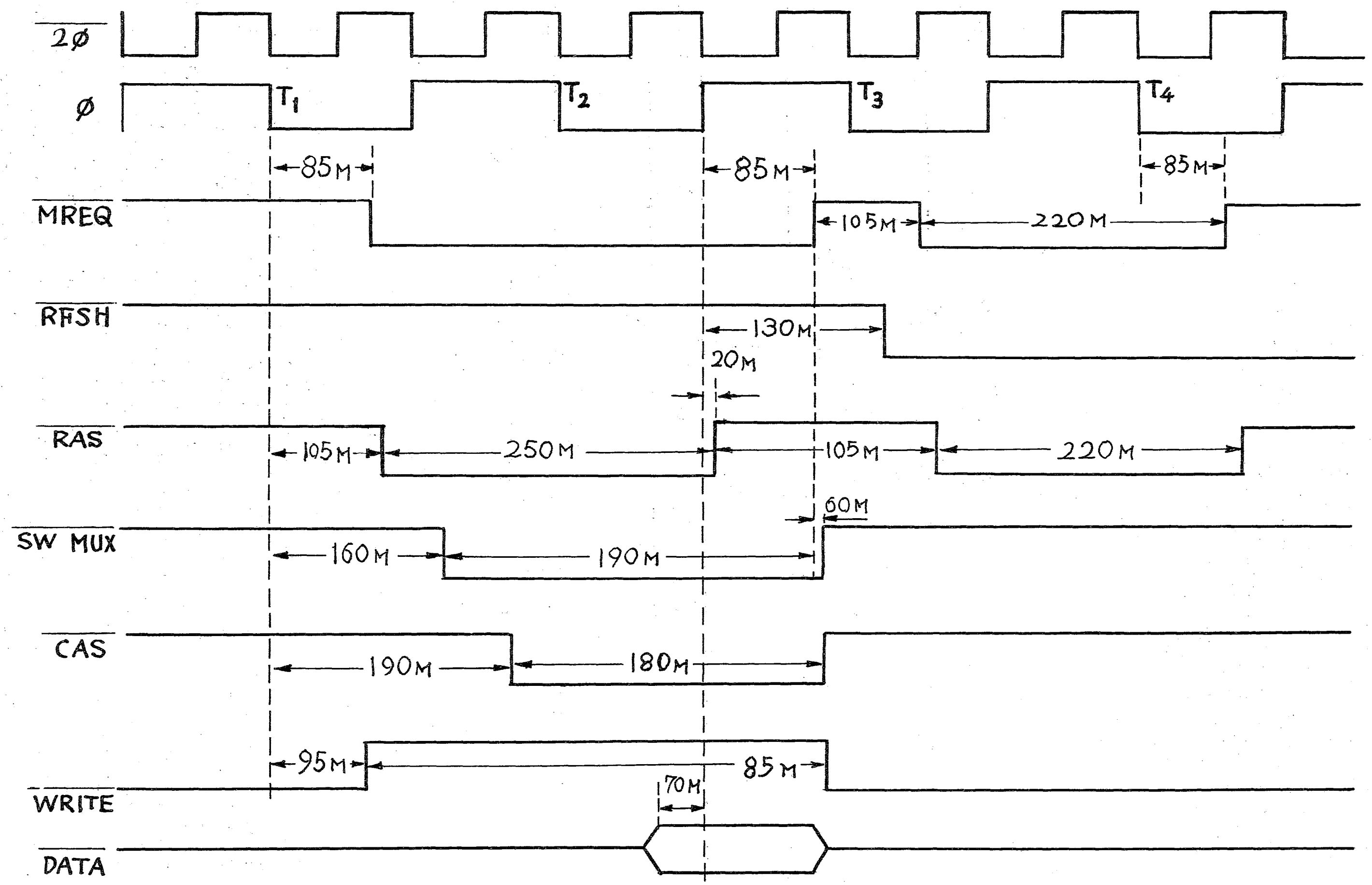


FIGURE 5 SYSTEM CLOCK

Each dynamic RAM used in the TS800 system is organized as 65536 X 1 bits. Eight dynamic RAM chips are used in one system (65536 X 8). Multiplexed addressing and periodic refreshing are required for this type of memory. The memory timing diagram is shown in Figure 6. The MREQ- signal and RFSH- signal from the CPU is outputed to the memory control circuit which is implemented with three D flip-flop 74S74 (A55 and A75), two NAND gates 74S00 (A74), three nor gates (74S02 (A72), 74LS02 (A73) and one 3-input NOR gate 74LS27 (A76). Two multiplexers 74S157 (A37 and A40) are used to multiplex the 16 address lines from the CPU into the 8 address lines on the dynamic memory. The row address and column address are latched internally by the falling edge of the RAS- and CAS- signals, respectively. Each memory cell in the memory chip must be refreshed at least every two ms. 'RAS only' refresh cycle and '128 refresh cycle' are used



2Ø
Ø
T1
T2
T3
T4
85M
85M
85M
MREQ
105M
220M
RFSH
130M
20M
RAS
105M
250M
105M
220M
60M
SW MUX
160M
190M
CAS
190M
180M
95M
85M
70M
WRITE
DATA

FIGURE 6

in this system. A serial resistor (68 ohm) is connected to each address line, RAS signal and CAS signal lines on the dynamic memory. The purpose of using these resistors is to reduce the signal under-shoot on the lines.

During DMA operations, the CPU stops sending refresh signals to the memory control circuit. Memory contents will be lost if DMA holds the buses for more than two ms. The other two types of memory used in this system are the static RAM and ROM. Since both static RAM and ROM are slow memory (slow access time) compared with Z-80A memory access time, one "wait cycle" is automatically inserted by the "wait control circuit" when either one of these two types of memory is accessed. Timing diagram is shown in Figure 7. This "wait control circuit" is implemented with two D flip-flop 74LS74 (A4) and one NAND gate 74S10 (A3).

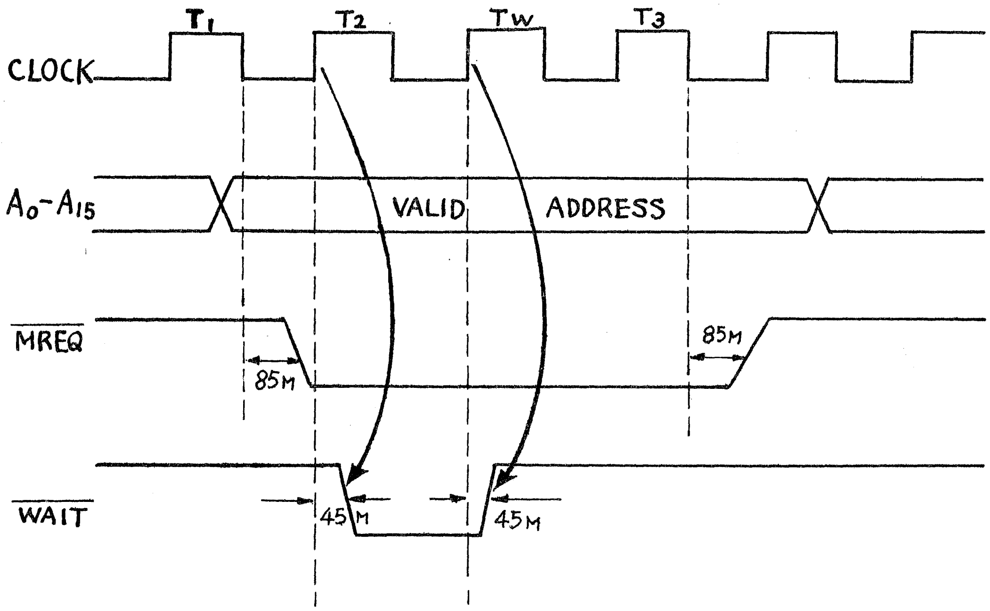


FIGURE 7 WAIT CYCLE INSERTED TIMING

The TS800 control board has a total of five physical I/O and 11 logical devices. They are dip switch, memory latch, CTC, SIO and DMA as mentioned before. During CPU I/O operations, CPU sends out I/O address on the address bus. The content of the address bus is decoded by a decoder 74LS138 (A77) to select one of the I/O devices. The following table shows all the I/O addresses of TS800 computer board.

INPUT/OUTPUT PORT ADDRESSES OF TS-80

o Dip Switch (switch 1, 2, 3, & 4 of sw 1) ----- OOH

o Memory Latch ----- 04H

o CTC Channel 0 ----- 08H SIO Channel A Clock

o CTC Channel 1 ----- 09H SIO Channel B Clock

o CTC Channel 2 ----- OAH Unused

o CTC Channel 3 ----- OBH Unused

o SIO Channel A (data) ----- OCH SDLC/DMA Interface/RS422

o SIO Channel A (command/status) ----- OEH DMA Interface/RS422

o SIO Channel B (data) ----- ODH 19.2K Baud Internal Link

o SIO Channel B (command/status) ----- OFH 19.2K Baud Internal Link

o DMA ----- 10H

I/O timing of each I/O device are shown from Figure 8 to Figure 13.

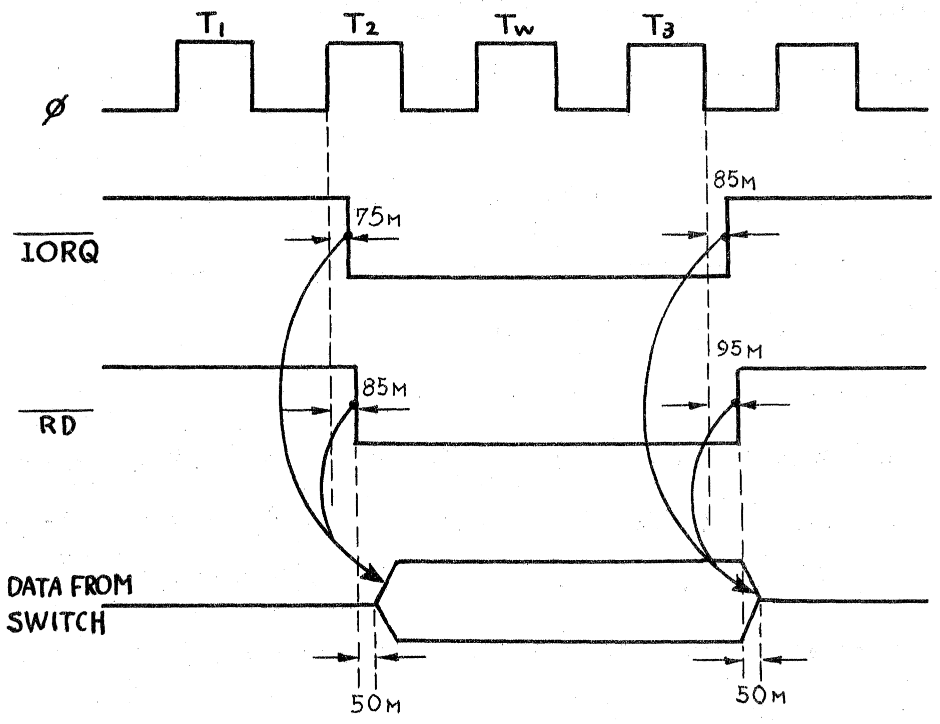


FIGURE 8 SWITCH READING TIMING

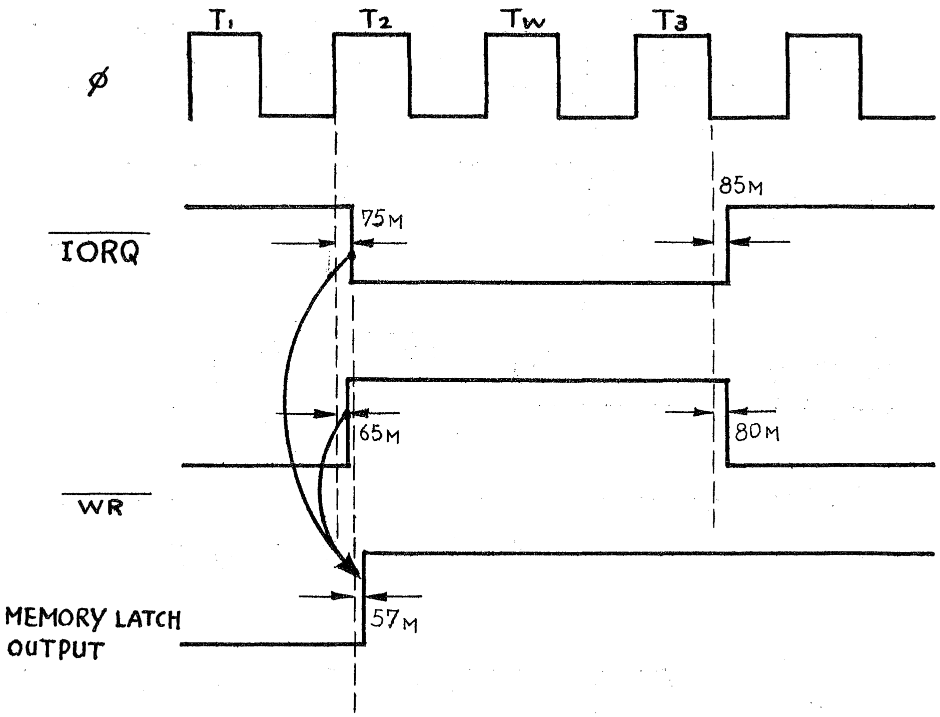
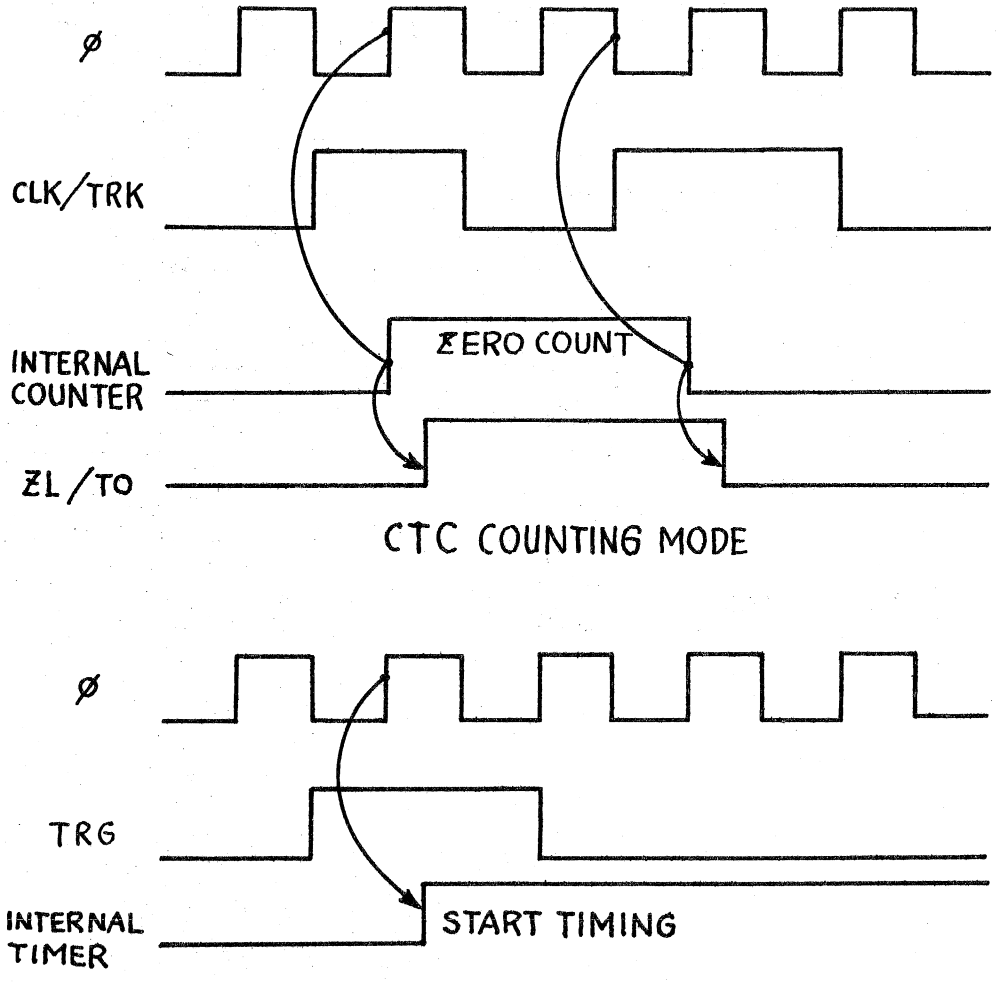


FIGURE 9 MEMORY LATCH TIMING



CTC COUNTING MODE

CTC TIMING MODE

FIGURE 10 CTC TIMING

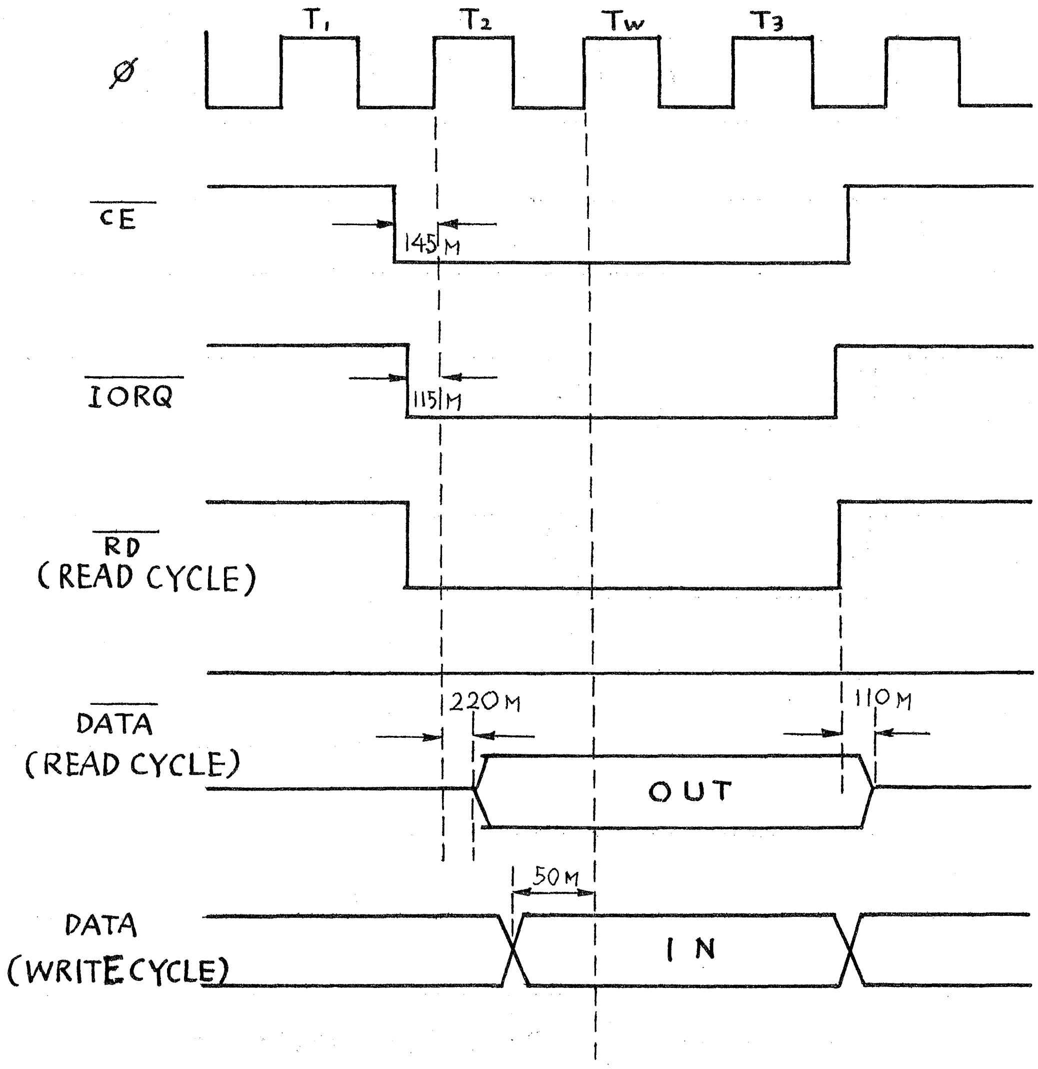


FIGURE 11 SIO READ/WRITE CYCLE

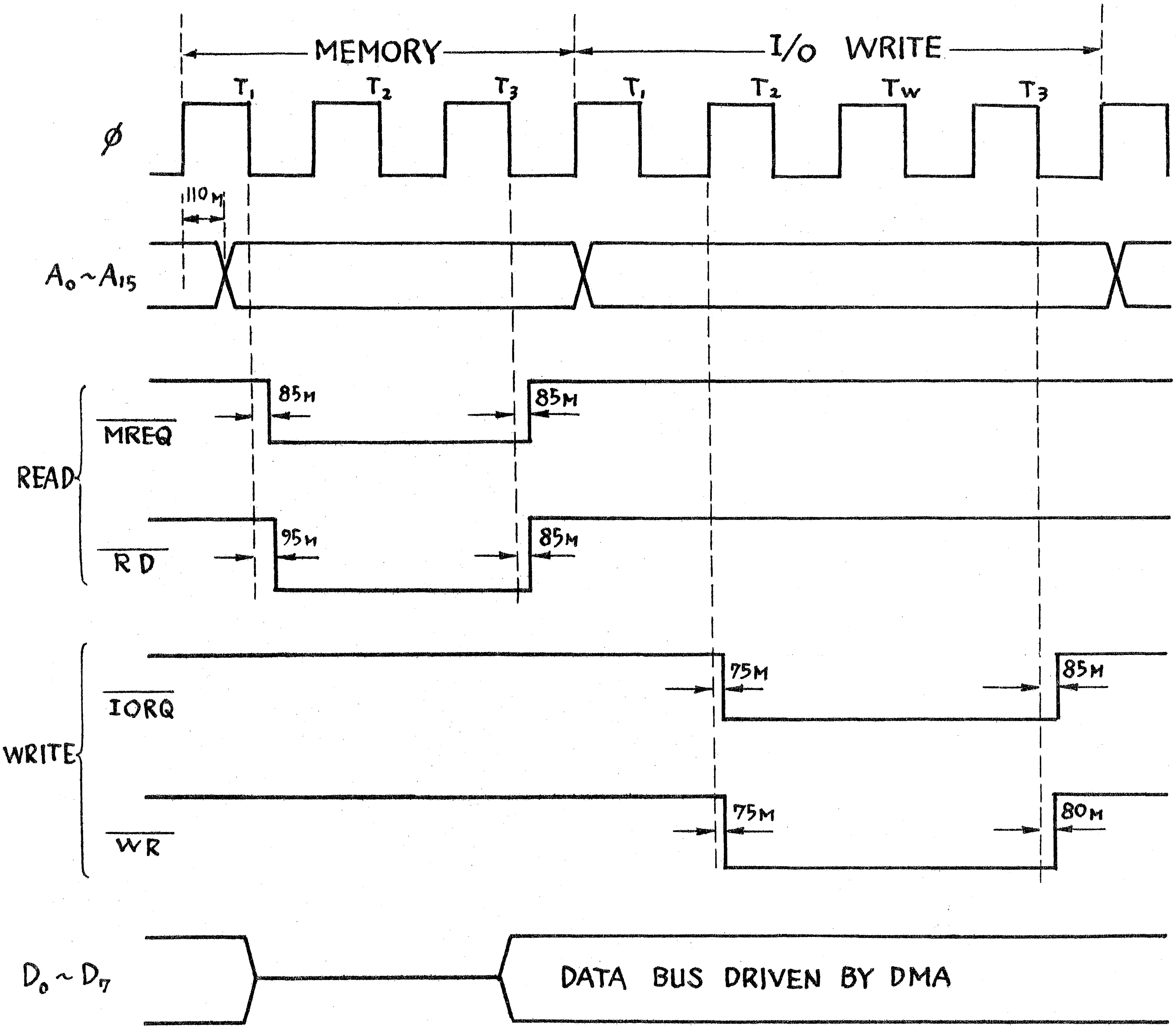


FIGURE 12 DMA MEMORY - TO - I/O TRANSFER

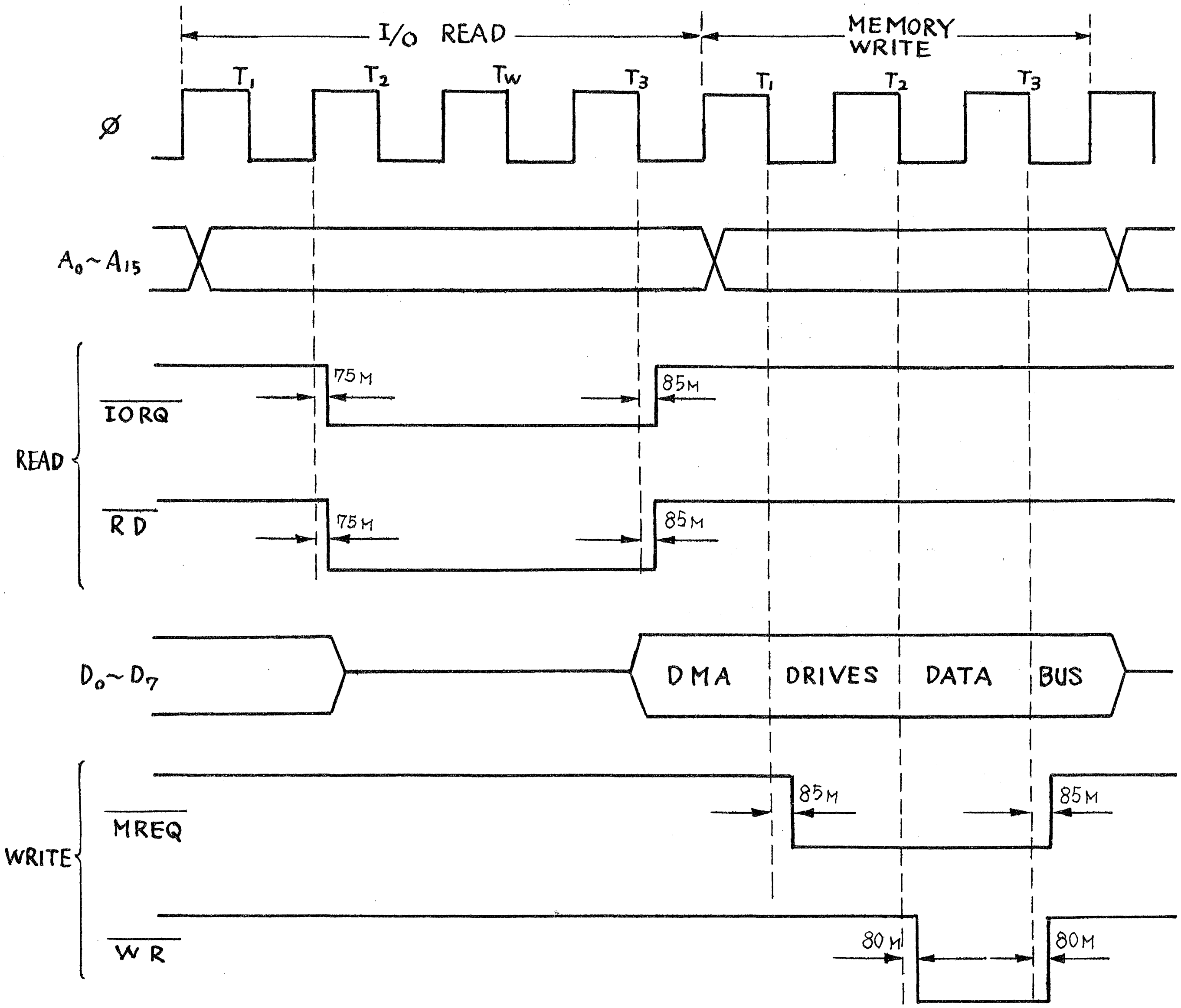


FIGURE 13 DMA I/O - TO - MEMORY TRANSFER

The following is the summary of the hardware specifications.

- o Power Requirement ----- +5V, +12, -12V
- o Power Consumption ----- Typical 5W
- o System Clock ----- 4 MHz
- o Memory
  - o 64 K bytes of dynamic RAM
  - o 4 K bytes of EPROM
  - o 1 K bytes of static RAM
- o RS-422 Communication Mode
  - o SDLC Mode
  - o 800 K bits/sec
- o Printer Port (RS-232)
  - o Asynchronous Mode
  - o 50 to 19.2 K bits/sec
- o Interrupt Priority
  - o 1st Priority ----- DMA
  - o 2nd Priority ----- SIO
  - o 3rd Priority ----- CTC

5.0 CONNECTOR AND SWITCH DESCRIPTION

The positions of all the connector on the TS800 board are shown in Figure 14.

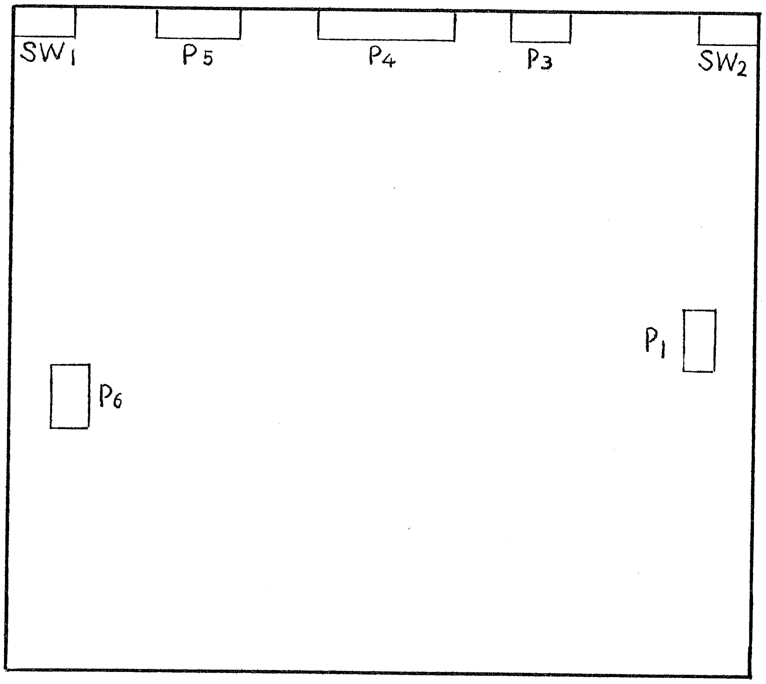


FIGURE 14 CONNECTOR POSITIONS

Table 2 shows the functions of each connector on the TS800 board.

Connector #	Description
P1	Video Signal Connector
P3	Key-board Connector
P4	RS232C Printer Interface
P5	RS422 User Connector
P6	Power Connector

TABLE 2. CONNECTOR ASSIGNMENT

5.1 The following describe the pin assignment of each connector.

P1 Video Signal Connector

Pin #	Description
1	H. Sync
2	Unused
3	Ground
4	Video
5	V. Sync

P3 Key-board Connector

Pin #	Description
1	+ 12 V
2	Ground
3	TXD -
4	RXD -

P4 RS232C Printer Interface (DCE)

Pin #	Description
1	Frame Ground
2	Transmit Data (Receive Data)
3	Receive Data (Transmit Data)
4	Request to Send
5	Clear to Send
6	Data Set Ready
7	Signal Ground
8	Data Carrier Detect
20	Data Terminal Ready (Printer Busy)

P5 RS422 User Interface

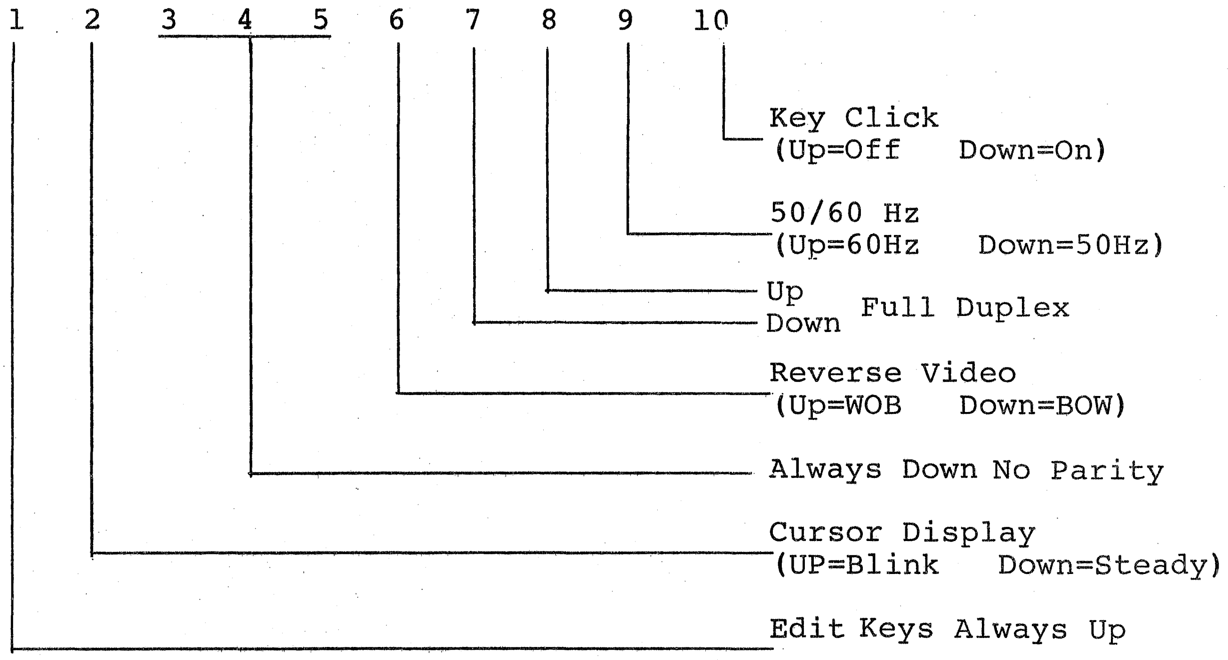
Pin #	Description
1	Ground
2	TXD +
3	RXD +
4	RTS +
5	CTS +
6	TXCK -
7	RXCK -
8	Ground
9	TXD -
10	RXD -
11	RTS -
12	CTS -
13	TXCK +
14	RXCK +
15	Test

P6 Power Connector

Pin #	Description
1	- 12 V
2	Unused
3	Ground
4	+ 5 V
5	+ 12 V

5.2 SWITCH SETTINGS

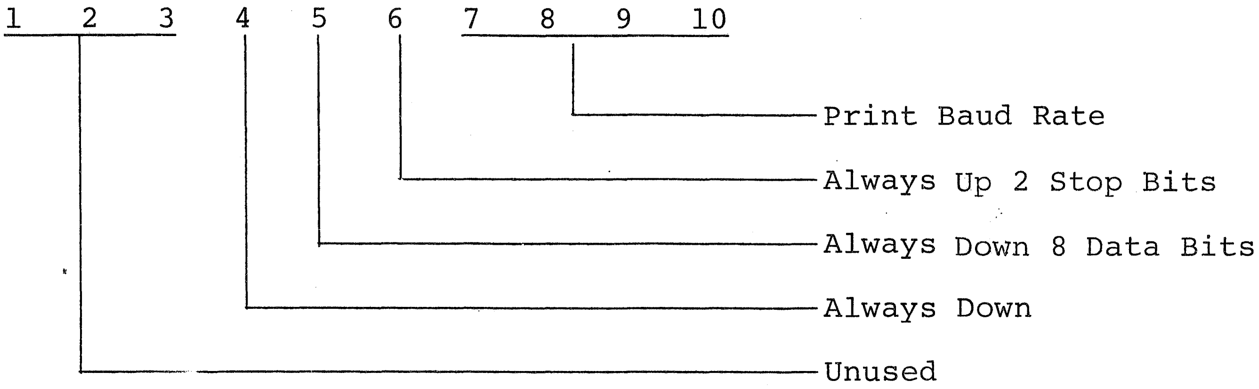
Switch 2 (sw 2)



NOTE:

Use keyboard to enter printer port word length, number of stop bit and parity.

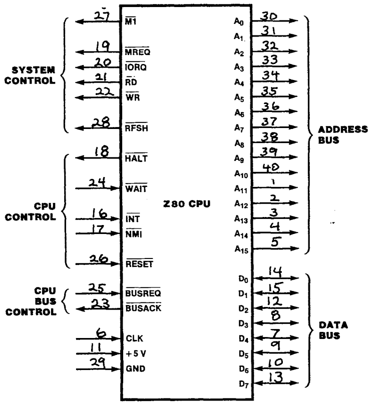
Switch 1 (sw 1)



Switches				Baud Rate (Printer)
7	8	9	10	
0	0	0	0	9600
1	0	0	0	50
0	1	0	0	75
1	1	0	0	110
0	0	1	0	135
1	0	1	0	150
0	1	1	0	300
1	1	1	0	600
0	0	0	1	1200
1	0	0	1	1800
0	1	0	1	2400
1	1	0	1	3600
0	0	1	1	4800
1	0	1	1	7200
0	1	1	1	9600
1	1	1	1	19200

Z8400
Z80® CPU Central Processing Unit

Features


SYSTEM CONTROL
CPU CONTROL
CPU BUS CONTROL
ADDRESS BUS
DATA BUS
Z80 CPU
$\overline{\mathrm{M1}}$ 27
$\overline{\mathrm{MREQ}}$ 19
$\overline{\mathrm{IORQ}}$ 20
$\overline{\mathrm{RD}}$ 21
$\overline{\mathrm{WR}}$ 22
$\overline{\mathrm{RFSH}}$ 28
$\overline{\mathrm{HALT}}$ 18
$\overline{\mathrm{WAIT}}$ 24
$\overline{\mathrm{INT}}$ 16
$\overline{\mathrm{NMI}}$ 17
$\overline{\mathrm{RESET}}$ 26
$\overline{\mathrm{BUSREQ}}$ 25
$\overline{\mathrm{BUSACK}}$ 23
CLK 6
+5V 11
GND 29
A_0 30
A_1 31
A_2 32
A_3 33
A_4 34
A_5 35
A_6 36
A_7 37
A_8 38
A_9 39
A_{10} 40
A_{11} 1
A_{12} 2
A_{13} 3
A_{14} 4
A_{15} 5
D_0 14
D_1 15
D_2 12
D_3 8
D_4 7
D_5 9
D_6 10
D_7 13

Pin Descriptions

A_0-A_{15}. *Address Bus* (output, active High, 3-state). A_0-A_{15} form a 16-bit address bus. The Address Bus provides the address for memory data bus exchanges (up to 64K bytes) and for I/O device exchanges.

$\overline{\mathrm{BUSACK}}$. *Bus Acknowledge* (output, active Low). Bus Acknowledge indicates to the requesting device that the CPU address bus, data bus, and control signals $\overline{\mathrm{MREQ}}$, $\overline{\mathrm{IORQ}}$, $\overline{\mathrm{RD}}$, and $\overline{\mathrm{WR}}$ have entered their high-impedance states. The external circuitry can now control these lines.

$\overline{\mathrm{BUSREQ}}$. *Bus Request* (input, active Low). Bus Request has a higher priority than $\overline{\mathrm{NMI}}$ and is always recognized at the end of the current machine cycle. $\overline{\mathrm{BUSREQ}}$ forces the CPU address bus, data bus, and control signals $\overline{\mathrm{MREQ}}$, $\overline{\mathrm{IORQ}}$, $\overline{\mathrm{RD}}$, and $\overline{\mathrm{WR}}$ to go to a high-impedance state so that other devices can control these lines. $\overline{\mathrm{BUSREQ}}$ is normally wire-ORed and requires an external pullup for these applications. Extended $\overline{\mathrm{BUSREQ}}$ periods due to extensive DMA operations can prevent the CPU from properly refreshing dynamic RAMs.

D_0-D_7. *Data Bus* (input/output, active High, 3-state). D_0-D_7 constitute an 8-bit bidirectional data bus, used for data exchanges with memory and I/O.

$\overline{\mathrm{HALT}}$. *Halt State* (output, active Low). $\overline{\mathrm{HALT}}$ indicates that the CPU has executed a Halt instruction and is awaiting either a nonmaskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOPs to maintain memory refresh.

$\overline{\mathrm{INT}}$. *Interrupt Request* (input, active Low). Interrupt Request is generated by I/O devices. The CPU honors a request at the end of the current instruction if the internal software-controlled interrupt enable flip-flop (IFF) is enabled. $\overline{\mathrm{INT}}$ is normally wire-ORed and requires an external pullup for these applications.

$\overline{\mathrm{IORQ}}$. *Input/Output Request* (output, active Low, 3-state). $\overline{\mathrm{IORQ}}$ indicates that the lower half of the address bus holds a valid I/O address for an I/O read or write operation. $\overline{\mathrm{IORQ}}$ is also generated concurrently with $\overline{\mathrm{M1}}$ during an interrupt acknowledge cycle to indicate that an interrupt response vector can be placed on the data bus.

$\overline{\mathrm{M1}}$. *Machine Cycle One* (output, active Low). $\overline{\mathrm{M1}}$, together with $\overline{\mathrm{MREQ}}$, indicates that the current machine cycle is the opcode fetch cycle of an instruction execution. $\overline{\mathrm{M1}}$, together with $\overline{\mathrm{IORQ}}$, indicates an interrupt acknowledge cycle.

$\overline{\mathrm{MREQ}}$. *Memory Request* (output, active Low, 3-state). $\overline{\mathrm{MREQ}}$ indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{\mathrm{NMI}}$. *Non-Maskable Interrupt* (input, active Low). $\overline{\mathrm{NMI}}$ has a higher priority than $\overline{\mathrm{INT}}$. $\overline{\mathrm{NMI}}$ is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop, and automatically forces the CPU to restart at location 0066H.

$\overline{\mathrm{RD}}$. *Memory Read* (output, active Low, 3-state). $\overline{\mathrm{RD}}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

$\overline{\mathrm{RESET}}$. *Reset* (input, active Low). $\overline{\mathrm{RESET}}$ initializes the CPU as follows: it resets the interrupt enable flip-flop, clears the PC and Registers I and R, and sets the interrupt status to Mode 0. During reset time, the address and data bus go to a high-impedance state, and all control output signals go to the inactive state. Note that $\overline{\mathrm{RESET}}$ must be active for a minimum of three full clock cycles before the reset operation is complete.

$\overline{\mathrm{RFSH}}$. *Refresh* (output, active Low). $\overline{\mathrm{RFSH}}$, together with $\overline{\mathrm{MREQ}}$, indicates that the lower seven bits of the system's address bus can be used as a refresh address to the system's dynamic memories.

$\overline{\mathrm{WAIT}}$. *Wait* (input, active Low). $\overline{\mathrm{WAIT}}$ indicates to the CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter a Wait state as long as this signal is active. Extended $\overline{\mathrm{WAIT}}$ periods can prevent the CPU from refreshing dynamic memory properly.

$\overline{\mathrm{WR}}$. *Memory Write* (output, active Low, 3-state). $\overline{\mathrm{WR}}$ indicates that the CPU data bus holds valid data to be stored at the addressed memory or I/O location.

"Z80 is a trademark of Zilog, Inc.
with whom the publisher is not associated."

Instruction Set

The Z80 microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor. It includes such unique operations as a block move for fast, efficient data transfers within memory or between memory and I/O. It also allows operations on any bit in any location in memory.

The following is a summary of the Z80 instruction set and shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. The *Z80 CPU Technical Manual* (03-0029-01) and *Assembly Language Programming Manual* (03-0002-01) contain significantly more details for programming use.

The instructions are divided into the following categories:

□ 8-bit loads

□ 16-bit loads

□ Exchanges, block transfers, and searches

□ 8-bit arithmetic and logic operations

□ General-purpose arithmetic and CPU control

□ 16-bit arithmetic operations

□ Rotates and shifts

□ Bit set, reset, and test operations

□ Jumps

□ Calls, returns, and restarts

□ Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

□ Immediate

□ Immediate extended

□ Modified page zero

□ Relative

□ Extended

□ Indexed

□ Register

□ Register indirect

□ Implied

□ Bit

8-Bit Load Group

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
LD r, r'	r ← r'	•	•	X	•	X	•	•	•	01 r r'		1	1	4	r, r' Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
LD r, n	r ← n	•	•	X	•	X	•	•	•	00 r 110 ← n →		2	2	7	
LD r, (HL)	r ← (HL)	•	•	X	•	X	•	•	•	01 r 110		1	2	7	
LD r, (IX+d)	r ← (IX+d)	•	•	X	•	X	•	•	•	11 011 101 01 r 101 ← d →	DD	3	5	19	
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	X	•	•	•	11 111 101 01 r 110 ← d →	FD	3	5	19	
LD (HL), r	(HL) ← r	•	•	X	•	X	•	•	•	01 110 r		1	2	7	
LD (IX+d), r	(IX+d) ← r	•	•	X	•	X	•	•	•	11 011 101 01 110 r ← d →	DD	3	5	19	
LD (IY+d), r	(IY+d) ← r	•	•	X	•	X	•	•	•	11 111 101 01 110 r ← d →	FD.	3	5	19	
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00 110 110 ← n →	36	2	3	10	
LD (IX+d), n	(IX+d) ← n	•	•	X	•	X	•	•	•	11 011 101 00 110 110 ← d → ← n →	DD 36	4	5	19	
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	•	11 111 101 00 110 110 ← d → ← n →	FD 36	4	5	19	
LD A, (BC)	A ← (BC)	•	•	X	•	X	•	•	•	00 001 010	0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	•	X	•	•	•	00 011 010	1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	•	X	•	•	•	00 111 010 ← n → ← n →	3A	3	4	13	
LD (BC), A	(BC) ← A	•	•	X	•	X	•	•	•	00 000 010	02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	•	X	•	•	•	00 010 010	12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	•	X	•	•	•	00 110 010 ← n → ← n →	32	3	4	13	
LD A, I	A ← I	↕	↕	X	0	X	IFF	0	•	11 101 101 01 010 111	ED 57	2	2	9	
LD A, R	A ← R	↕	↕	X	0	X	IFF	0	•	11 101 101 01 011 111	ED 5F	2	2	9	
LD I, A	I ← A	•	•	X	•	X	•	•	•	11 101 101 01 000 111	ED 47	2	2	9	
LD R, A	R ← A	•	•	X	•	X	•	•	•	11 101 101 01 001 111	ED 4F	2	2	9	

NOTES: r, r' means any of the registers A, B, C, D, E, H, L.
IFF the content of the interrupt enable flip-flop, (IFF) is copied into the P/V flag.

For an explanation of flag notation and symbols for mnemonic tables, see Symbolic Notation section following tables.

16-Bit Load Group

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00 dd0 001 ← n → ← n →		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11 011 101 00 100 001 ← n → ← n →	DD 21	4	4	14	
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11 111 101 00 100 001 ← n → ← n →	FD 21	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	•	00 101 010 ← n → ← n →	2A	3	5	16	
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	•	•	X	•	X	•	•	•	11 101 101 01 dd1 011 ← n → ← n →	ED	4	6	20	
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	•	•	X	•	X	•	•	•	11 011 101 00 101 010 ← n → ← n →	DD 2A	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	•	•	X	•	X	•	•	•	11 111 101 00 101 010 ← n → ← n →	FD 2A	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00 100 010 ← n → ← n →	22	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	X	•	X	•	•	•	11 101 101 01 dd0 011 ← n → ← n →	ED	4	6	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	X	•	X	•	•	•	11 011 101 00 100 010 ← n → ← n →	DD 22	4	6	20	
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	X	•	X	•	•	•	11 111 101 00 100 010 ← n → ← n →	FD 22	4	6	20	
LD SP, HL	SP ← HL	•	•	X	•	X	•	•	•	11 111 001	F9	1	1	6	
LD SP, IX	SP ← IX	•	•	X	•	X	•	•	•	11 011 101 11 111 001	DD F9	2	2	10	
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11 111 101 11 111 001	FD F9	2	2	10	
PUSH qq	(SP−2) ← qq_L (SP−1) ← qq_H SP ← SP −2	•	•	X	•	X	•	•	•	11 qq0 101		1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP−2) ← IX_L (SP−1) ← IX_H SP ← SP −2	•	•	X	•	X	•	•	•	11 011 101 11 100 101	DD E5	2	4	15	
PUSH IY	(SP−2) ← IY_L (SP−1) ← IY_H SP ← SP −2	•	•	X	•	X	•	•	•	11 111 101 11 100 101	FD E5	2	4	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP) SP ← SP +2	•	•	X	•	X	•	•	•	11 qq0 001		1	3	10	
POP IX	IX_H ← (SP+1) IX_L ← (SP) SP ← SP +2	•	•	X	•	X	•	•	•	11 011 101 11 100 001	DD E1	2	4	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP) SP ← SP +2	•	•	X	•	X	•	•	•	11 111 101 11 100 001	FD E1	2	4	14	

NOTES: dd is any of the register pairs BC, DE, HL, SP.
qq is any of the register pairs AF, BC, DE, HL.
$(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively,
e.g., BC_L = C, AF_H = A.

Exchange, Block Transfer, Block Search Groups

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
EX DE, HL	DE ↔ HL	•	•	X	•	X	•	•	•	11 101 011	EB	1	1	4	
EX AF, AF′	AF ↔ AF′	•	•	X	•	X	•	•	•	00 001 000	08	1	1	4	
EXX	BC ↔ BC′ DE ↔ DE′ HL ↔ HL′	•	•	X	•	X	•	•	•	11 011 001	D9	1	1	4	Register bank and auxiliary register bank exchange
EX (SP), HL	H ↔ (SP+1) L ↔ (SP)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	19	
EX (SP), IX	IX_H ↔ (SP+1) IX_L ↔ (SP)	•	•	X	•	X	•	•	•	11 011 101 11 100 011	DD E3	2	6	23	
EX (SP), IY	IY_H ↔ (SP+1) IY_L ↔ (SP)	•	•	X	•	X	•	•	•	11 111 101 11 100 011	FD E3	2	6	23	
LDI	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC−1	•	•	X	0	X	① ↕	0	•	11 101 101 10 100 000	ED A0	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
LDIR	(DE) ← (HL) DE ← DE+1 HL ← HL+1 BC ← BC−1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 110 000	ED B0	2 2	5 4	21 16	If BC ≠ 0 If BC =0

NOTE: ① P/V flag is 0 if the result of BC−1 = 0, otherwise P/V = 1.

Exchange, Block Transfer, Block Search Groups (Continued)

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
LDD	(DE) ← (HL) DE ← DE−1 HL ← HL−1 BC ← BC−1	•	•	X	0	X	① ↕	0	•	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE−1 HL ← HL−1 BC ← BC−1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16	If BC ≠ 0 If BC = 0
CPI	A − (HL) HL ← HL+1 BC ← BC−1	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A − (HL) HL ← HL+1 BC ← BC−1 Repeat until A = (HL) or BC = 0	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)
CPD	A − (HL) HL ← HL−1 BC ← BC−1	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 101 001	ED A9	2	4	16	
CPDR	A − (HL) HL ← HL−1 BC ← BC−1 Repeat until A = (HL) or BC = 0	↕	② ↕	X	↕	X	① ↕	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16	If BC ≠ 0 and A ≠ (HL) If BC = 0 or A = (HL)

NOTES: ① P/V flag is 0 if the result of BC−1 = 0, otherwise P/V = 1.
② Z flag is 1 if A = (HL), otherwise Z = 0.

8-Bit Arithmetic and Logical Group

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
ADD A, r	A ← A + r	↕	↕	X	↕	X	V	0	↕	10 $\boxed{000}$ r		1	1	4	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
ADD A, n	A ← A + n	↕	↕	X	↕	X	V	0	↕	11 $\boxed{000}$ 110 ← n →		2	2	7	
ADD A, (HL)	A ← A + (HL)	↕	↕	X	↕	X	V	0	↕	10 $\boxed{000}$ 110		1	2	7	
ADD A, (IX+d)	A ← A + (IX+d)	↕	↕	X	↕	X	V	0	↕	11 011 101 10 $\boxed{000}$ 110 ← d →	DD	3	5	19	
ADD A, (IY+d)	A ← A + (IY+d)	↕	↕	X	↕	X	V	0	↕	11 111 101 10 $\boxed{000}$ 110 ← d →	FD	3	5	19	
ADC A, s	A ← A+s+CY	↕	↕	X	↕	X	V	0	↕	$\boxed{001}$					s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction. The indicated bits replace the $\boxed{000}$ in the ADD set above.
SUB s	A ← A−s	↕	↕	X	↕	X	V	1	↕	$\boxed{010}$					
SBC A, s	A ← A−s−CY	↕	↕	X	↕	X	V	1	↕	$\boxed{011}$					
AND s	A ← A ∧ s	↕	↕	X	1	X	P	0	0	$\boxed{100}$					
OR s	A ← A ∨ s	↕	↕	X	0	X	P	0	0	$\boxed{110}$					
XOR s	A ← A ⊕ s	↕	↕	X	0	X	P	0	0	$\boxed{101}$					
CP s	A−s	↕	↕	X	↕	X	V	1	↕	$\boxed{111}$					
INC r	r ← r + 1	↕	↕	X	↕	X	V	0	•	00 r $\boxed{100}$		1	1	4	
INC (HL)	(HL) ←(HL)+1	↕	↕	X	↕	X	V	0	•	00 110 $\boxed{100}$		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	↕	↕	X	↕	X	V	0	•	11 011 101 00 110 $\boxed{100}$ ← d →	DD	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	↕	↕	X	↕	X	V	0	•	11 111 101 00 110 $\boxed{100}$ ← d →	FD	3	6	23	
DEC m	m ← m−1	↕	↕	X	↕	X	V	1	•	$\boxed{101}$					m is any of r, (HL), (IX+d), (IY+d) as shown for INC. DEC same format and states as INC. Replace $\boxed{100}$ with $\boxed{101}$ in opcode.

General-Purpose Arithmetic and CPU Control Groups

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands.	↕	↕	X	↕	X	P	•	↕	00 100 111	27	1	1	4	Decimal adjust accumulator.
CPL	$A \leftarrow \overline{A}$	•	•	X	1	X	•	1	•	00 101 111	2F	1	1	4	Complement accumulator (one's complement).
NEG	A ← 0 − A	↕	↕	X	↕	X	V	1	↕	11 101 101 01 000 100	ED 44	2	2	8	Negate acc. (two's complement).
CCF	$CY \leftarrow \overline{CY}$	•	•	X	X	X	•	0	↕	00 111 111	3F	1	1	4	Complement carry flag.
SCF	CY ← 1	•	•	X	0	X	•	0	1	00 110 111	37	1	1	4	Set carry flag.
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01 110 110	76	1	1	4	
DI ★	IFF ← 0	•	•	X	•	X	•	•	•	11 110 011	F3	1	1	4	
EI ★	IFF ← 1	•	•	X	•	X	•	•	•	11 111 011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 01 000 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 01 010 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

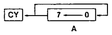
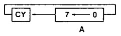
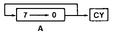
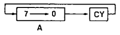
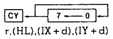
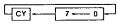
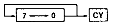
NOTES: IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.
★ indicates interrupts are not sampled at the end of EI or DI.

16-Bit Arithmetic Group

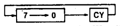
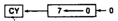
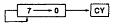
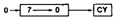
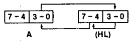
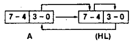
Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
ADD HL, ss	HL ← HL+ss	•	•	X	X	X	•	0	↕	00 ss1 001		1	3	11	ss Reg.
ADC HL, ss	HL ← HL+ss+CY	↕	↕	X	X	X	V	0	↕	11 101 101 01 ss1 010	ED	2	4	15	00 BC 01 DE 10 HL 11 SP
SBC HL, ss	HL ← HL−ss−CY	↕	↕	X	X	X	V	1	↕	11 101 101 01 ss0 010	ED	2	4	15	
ADD IX, pp	IX ← IX + pp	•	•	X	X	X	•	0	↕	11 011 101 01 pp1 001	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	IY ← IY + rr	•	•	X	X	X	•	0	↕	11 111 101 00 rr1 001	FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC ss	ss ← ss + 1	•	•	X	•	X	•	•	•	00 ss0 011		1	1	6	
INC IX	IX ← IX + 1	•	•	X	•	X	•	•	•	11 011 101 00 100 011	DD 23	2	2	10	
INC IY	IY ← IY + 1	•	•	X	•	X	•	•	•	11 111 101 00 100 011	FD 23	2	2	10	
DEC ss	ss ← ss − 1	•	•	X	•	X	•	•	•	00 ss1 011		1	1	6	
DEC IX	IX ← IX − 1	•	•	X	•	X	•	•	•	11 011 101 00 101 011	DD 2B	2	2	10	
DEC IY	IY ← IY − 1	•	•	X	•	X	•	•	•	11 111 101 00 101 011	FD 2B	2	2	10	

NOTES: ss is any of the register pairs BC, DE, HL, SP.
pp is any of the register pairs BC, DE, IX, SP.
rr is any of the register pairs BC, DE, IY, SP.

Rotate and Shift Group

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
RLCA		•	•	X	0	X	•	0	↕	00 000 111	07	1	1	4	Rotate left circular accumulator.
RLA		•	•	X	0	X	•	0	↕	00 010 111	17	1	1	4	Rotate left accumulator.
RRCA		•	•	X	0	X	•	0	↕	00 001 111	0F	1	1	4	Rotate right circular accumulator.
RRA		•	•	X	0	X	•	0	↕	00 011 111	1F	1	1	4	Rotate right accumulator.
RLC r		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 r	CB	2	2	8	Rotate left circular register r.
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 110	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	 r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	11 011 101 11 001 011 ← d → 00 000 110	DD CB	4	6	23	
RLC (IY+d)		↕	↕	X	0	X	P	0	↕	11 111 101 11 001 011 ← d → 00 000 110	FD CB	4	6	23	Instruction format and states are as shown for RLC's. To form new opcode replace 000 or RLC's with shown code.
RL m	 m ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	010					
RRC m	 m ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	001					

Rotate and Shift Group (Continued)

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
RR m	 m ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{011}$					
SLA m	 m ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{100}$					
SRA m	 m ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{101}$					
SRL m	 m ≡ r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	$\boxed{111}$					
RLD		↕	↕	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL).
RRD		↕	↕	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	The content of the upper half of the accumulator is unaffected.

Bit Set, Reset and Test Group

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
BIT b, r	$Z \leftarrow \overline{r}_b$	X	↕	X	1	X	X	0	•	11 001 011 01 b r	CB	2	2	8	r Reg. 000 B
BIT b, (HL)	$Z \leftarrow \overline{(HL)}_b$	X	↕	X	1	X	X	0	•	11 001 011 01 b 110	CB	2	3	12	001 C 010 D
BIT b, $(IX+d)_b$	$Z \leftarrow \overline{(IX+d)}_b$	X	↕	X	1	X	X	0	•	11 011 101 11 001 011 ← d → 01 b 110	DD CB	4	5	20	011 E 100 H 101 L 111 A b Bit Tested
BIT b, $(IY+d)_b$	$Z \leftarrow \overline{(IY+d)}_b$	X	↕	X	1	X	X	0	•	11 111 101 11 001 011 ← d → 01 b 110	FD CB	4	5	20	000 0 001 1 010 2 011 3 100 4 101 5 110 6 111 7
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 $\boxed{11}$ b r	CB	2	2	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 $\boxed{11}$ b 110	CB	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 011 101 11 001 011 ← d → $\boxed{11}$ b 110	DD CB	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 111 101 11 001 011 ← d → $\boxed{11}$ b 110	FD CB	4	6	23	
RES b, m	$m_b \leftarrow 0$ m ≡ r, (HL), (IX+d), (IY+d)	•	•	X	•	X	•	•	•	$\boxed{10}$					To form new opcode replace $\boxed{11}$ of SET b, s with $\boxed{10}$. Flags and time states for SET instruction.

NOTES: The notation m_b indicates bit b (0 to 7) or location m.

Jump Group

Mnemonic	Symbolic Operation	S	Z		Flags H		P/V	N	C	Opcode 76 543 210	Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
JP nn	PC ← nn	•	•	X	•	X	•	•	•	11 000 011 ← n → ← n →	C3	3	3	10	
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010 ← n → ← n →		3	3	10	cc Condition 000 NZ non-zero 001 Z zero 010 NC non-carry 011 C carry 100 PO parity odd 101 PE parity even 110 P sign positive 111 M sign negative
JR e	PC ← PC+e	•	•	X	•	X	•	•	•	00 011 000 ← e−2 →	18	2	3	12	
JR C, e	If C = 0, continue If C = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00 111 000 ← e−2 →	38	2 2	2 3	7 12	If condition not met. If condition is met.
JR NC, e	If C = 1, continue If C = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 110 000 ← e−2 →	30	2 2	2 3	7 12	If condition not met. If condition is met.
JP Z, e	If Z = 0 continue If Z = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00 101 000 ← e−2 →	28	2 2	2 3	7 12	If condition not met. If condition is met.
JR NZ, e	If Z = 1, continue If Z = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 100 000 ← e−2 →	20	2 2	2 3	7 12	If condition not met. If condition is met.
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11 101 001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11 011 101 11 101 001	DD E9	2	2	8	

Jump Group (Continued)

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210 Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11 111 101 FD 11 101 001 E9	2	2	8	
DJNZ, e	B ← B−1 If B = 0, continue	•	•	X	•	X	•	•	•	00 010 000 10 ← e−2 →	2	2	8	If B = 0.
	If B ≠ 0, PC ← PC+e										2	3	13	If B ≠ 0.

NOTES: e represents the extension in the relative addressing mode.
e is a signed two's complement number in the range < −126, 129 >.
e−2 in the opcode provides an effective address of pc+e as PC is incremented by 2 prior to the addition of e.

Call and Return Group

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210 Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
CALL nn	(SP−1) ← PC_H (SP−2) ← PC_L PC ← nn	•	•	X	•	X	•	•	•	11 001 101 CD ← n → ← n →	3	5	17	
CALL cc, nn	If condition cc is false	•	•	X	•	X	•	•	•	11 cc 100 ← n →	3	3	10	If cc is false.
	continue, otherwise same as CALL nn									← n →	3	5	17	If cc is true.
RET	PC_L ← (SP) PC_H ← (SP+1)	•	•	X	•	X	•	•	•	11 001 001 C9	1	3	10	
RET cc	If condition cc is false	•	•	X	•	X	•	•	•	11 cc 000	1	1	5	If cc is false.
	continue, otherwise same as RET										1	3	11	If cc is true.
RETI	Return from interrupt	•	•	X	•	X	•	•	•	11 101 101 ED 01 001 101 4D	2	4	14	
RETN[1]	Return from non-maskable interrupt	•	•	X	•	X	•	•	•	11 101 101 ED 01 000 101 45	2	4	14	
RST p	(SP−1) ← PC_H (SP−2) ← PC_L PC_H ← 0 PC_L ← p	•	•	X	•	X	•	•	•	11 t 111	1	3	11	

cc		Condition
000	NZ	non-zero
001	Z	zero
010	NC	non-carry
011	C	carry
100	PO	parity odd
101	PE	parity even
110	P	sign positive
111	M	sign negative

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

NOTE: [1]RETN loads IFF_2 ← IFF_1

Input and Output Group

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210 Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011 DB ← n →	2	3	11	n to A_0 ~ A_7 Acc. to A_8 ~ A_{15}
IN r, (C)	r ← (C) if r = 110 only the flags will be affected	↕	↕	X	↕	X	P	0	•	11 101 101 ED 01 r 000	2	3	12	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INI	(HL) ← (C) B ← B−1 HL ← HL + 1	X	① ↕	X	X	X	X	1	•	11 101 101 ED 10 100 010 A2	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INIR	(HL) ← (C) B ← B−1	X	1	X	X	X	X	1	•	11 101 101 ED 10 110 010 B2	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
	HL ← HL + 1 Repeat until B = 0										2	4 (If B=0)	16	
IND	(HL) ← (C) B ← B −1 HL ← HL−1	X	① ↕	X	X	X	X	1	•	11 101 101 ED 10 101 010 AA	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
INDR	(HL) ← (C) B ← B−1	X	1	X	X	X	X	1	•	11 101 101 ED 10 111 010 BA	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
	HL ← HL−1 Repeat until B = 0										2	4 (If B=0)	16	
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11 010 011 D3 ← n →	2	3	11	n to A_0 ~ A_7 Acc. to A_8 ~ A_{15}
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 101 101 ED 01 r 001	2	3	12	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OUTI	(C) ← (HL) B ← B−1 HL ← HL + 1	X	① ↕	X	X	X	X	1	•	11 101 101 ED 10 100 011 A3	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}
OTIR	(C) ← (HL) B ← B−1	X	1	X	X	X	X	1	•	11 101 101 ED 10 110 011 B3	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
	HL ← HL + 1 Repeat until B = 0										2	4 (If B=0)	16	
OUTD	(C) ← (HL) B ← B−1 HL ← HL−1	X	① ↕	X	X	X	X	1	•	11 101 101 ED 10 101 011 AB	2	4	16	C to A_0 ~ A_7 B to A_8 ~ A_{15}

NOTE: ① If the result of B−1 is zero the Z flag is set, otherwise it is reset.

Input and Output Group (Continued)

Mnemonic	Symbolic Operation	S	Z		H		P/V	N	C	Opcode 76 543 210 Hex	No.of Bytes	No.of M Cycles	No.of T States	Comments
OTDR	(C) ← (HL) B ← B−1	X	1	X	X	X	X	1	•	11 101 101 ED 10 111 011	2	5 (If B≠0)	21	C to A_0 ~ A_7 B to A_8 ~ A_{15}
	HL ← HL−1 Repeat until B = 0										2	4 (If B=0)	16	

Summary of Flag Operation

Instruction	D_7 S	Z		H		P/V	N	D_0 C	Comments
ADD A, s; ADC A, s	↕	↕	X	↕	X	V	0	↕	8-bit add or add with carry.
SUB s; SBC A, s; CP s; NEG	↕	↕	X	↕	X	V	1	↕	8-bit subtract, subtract with carry, compare and negate accumulator.
AND s	↕	↕	X	1	X	P	0	0	Logical operations.
OR s, XOR s	↕	↕	X	0	X	P	0	0	
INC s	↕	↕	X	↕	X	V	0	•	8-bit increment.
DEC s	↕	↕	X	↕	X	V	1	•	8-bit decrement.
ADD DD, ss	•	•	X	X	X	•	0	↕	16-bit add.
ADC HL, ss	↕	↕	X	X	X	V	0	↕	16-bit add with carry.
SBC HL, ss	↕	↕	X	X	X	V	1	↕	16-bit subtract with carry.
RLA, RLCA, RRA; RRCA	•	•	X	0	X	•	0	↕	Rotate accumulator.

RL m; RLC m; RR m;
RRC m; SLA m;
SRA m; SRL m
	↕	↕	X	0	X	P	0	↕	Rotate and shift locations.

RLD; RRD	↕	↕	X	0	X	P	0	•	Rotate digit left and right.
DAA	↕	↕	X	↕	X	P	•	↕	Decimal adjust accumulator.
CPL	•	•	X	1	X	•	1	•	Complement accumulator.
SCF	•	•	X	0	X	•	0	1	Set carry.
CCF	•	•	X	X	X	•	0	↕	Complement carry.
IN r (C)	↕	↕	X	0	X	P	0	•	Input register indirect.
INI, IND, OUTI; OUTD	X	↕	X	X	X	X	1	•	Block input and output. Z = 0 if B ≠ 0 otherwise Z = 0.
INIR; INDR; OTIR; OTDR	X	1	X	X	X	X	1	•	
LDI; LDD	X	X	X	0	X	↕	0	•	Block transfer instructions. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LDIR; LDDR	X	X	X	0	X	0	0	•	
CPI; CPIR; CPD; CPDR	X	↕	X	X	X	↕	1	•	Block search instructions. Z = 1 if A = (HL), otherwise Z = 0. P/V = 1 if BC ≠ 0, otherwise P/V = 0.
LD A, I, LD A, R	↕	↕	X	0	X	IFF	0	•	The content of the interrupt enable flip-flop (IFF) is copied into the P/V flag.
BIT b, s	X	↕	X	1	X	X	0	•	The state of bit b of location s is copied into the Z flag.

Symbolic Notation

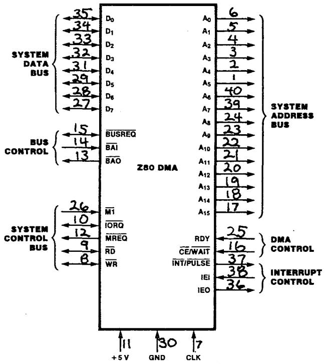
Symbol	Operation
S	Sign flag. S = 1 if the MSB of the result is 1.
Z	Zero flag. Z = 1 if the result of the operation is 0.
P/V	Parity or overflow flag. Parity (P) and overflow (V) share the same flag. Logical operations affect this flag with the parity of the result while arithmetic operations affect this flag with the overflow of the result. If P/V holds parity, P/V = 1 if the result of the operation is even, P/V = 0 if result is odd. If P/V holds overflow, P/V = 1 if the result of the operation produced an overflow.
H	Half-carry flag. H = 1 if the add or subtract operation produced a carry into or borrow from bit 4 of the accumulator.
N	Add/Subtract flag. N = 1 if the previous operation was a subtract.
H & N	H and N flags are used in conjunction with the decimal adjust instruction (DAA) to properly correct the result into packed BCD format following addition or subtraction using operands with packed BCD format.
C	Carry/Link flag. C = 1 if the operation produced a carry from the MSB of the operand or result.

Symbol	Operation
↕	The flag is affected according to the result of the operation.
•	The flag is unchanged by the operation.
0	The flag is reset by the operation.
1	The flag is set by the operation.
X	The flag is a "don't care."
V	P/V flag affected according to the overflow result of the operation.
P	P/V flag affected according to the parity result of the operation.
r	Any one of the CPU registers A, B, C, D, E, H, L.
s	Any 8-bit location for all the addressing modes allowed for the particular instruction.
ss	Any 16-bit location for all the addressing modes allowed for that instruction.
ii	Any one of the two index registers IX or IY.
R	Refresh counter.
n	8-bit value in range < 0, 255 >.
nn	16-bit value in range < 0, 65535 >.

Zilog

Z8410
Z80® DMA Direct Memory Access Controller

Features



Pin Description

A_0-A_{15}. *System Address Bus* (output, 3-state). Addresses generated by the DMA are sent to both source and destination ports (main memory or I/O peripherals) on these lines.

$\overline{\text{BAI}}$. *Bus Acknowledge In* (input, active Low). Signals that the system buses have been released for DMA control. In multiple-DMA configurations, the $\overline{\text{BAI}}$ pin of the highest priority DMA is normally connected to the Bus Acknowledge pin of the CPU. Lower-priority DMAs have their $\overline{\text{BAI}}$ connected to the $\overline{\text{BAO}}$ of a higher-priority DMA.

$\overline{\text{BAO}}$. *Bus Acknowledge Out* (output, active Low). In a multiple-DMA configuration, this pin signals that no other higher-priority DMA has requested the system buses. $\overline{\text{BAI}}$ and $\overline{\text{BAO}}$ form a daisy chain for multiple-DMA priority resolution over bus control.

$\overline{\text{BUSREQ}}$. *Bus Request* (bidirectional, active Low, open drain). As an output, it sends requests for control of the system address bus, data bus and control bus to the CPU. As an input, when multiple DMAs are strung together in a priority daisy chain via $\overline{\text{BAI}}$ and $\overline{\text{BAO}}$, it senses when another DMA has requested the buses and causes this DMA to refrain from bus requesting until the other DMA is finished. Because it is a bidirectional pin, there cannot be any buffers between this DMA and any other DMA. It can, however, have a buffer between it and the CPU because it is unidirectional into the CPU. A pull-up resistor is connected to this pin.

$\overline{\text{CE}}/\overline{\text{WAIT}}$. *Chip Enable and Wait* (input, active Low). Normally this functions only as a $\overline{\text{CE}}$ line, but it can also be programmed to serve a $\overline{\text{WAIT}}$ function. As a $\overline{\text{CE}}$ line from the CPU, it becomes active when $\overline{\text{WR}}$ and $\overline{\text{IORQ}}$ are active and the I/O port address on the system address bus is the DMA's address, thereby allowing a transfer of control or command bytes from the CPU to the DMA. As a $\overline{\text{WAIT}}$ line from memory or I/O devices, after the DMA has received a bus-request acknowledge from the CPU, it causes wait states to be inserted in the DMA's operation cycles thereby slowing the DMA to a speed that matches the memory or I/O device.

CLK. *System Clock* (input). Standard Z-80 single-phase clock at 2.5 MHz (Z-80 DMA) or 4.0 MHz (Z-80A DMA). For slower system clocks, a TTL gate with a pullup resistor may be adequate to meet the timing and voltage level specification. For higher-speed systems, use a clock driver with an active pullup to meet the V_{IH} specification and risetime requirements. In all cases there should be a resistive pullup to the power supply of 10K ohms (max) to ensure proper power when the DMA is reset.

D_0-D_7. *System Data Bus* (bidirectional, 3-state). Commands from the CPU, DMA status, and data from memory or I/O peripherals are transferred on these lines.

IEI. *Interrupt Enable In* (input, active High). This is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this DMA. Thus, this signal blocks lower-priority devices from interrupting while a higher-priority device is being serviced by its CPU interrupt service routine.

$\overline{\text{INT}}$/PULSE. *Interrupt Request* (output, active Low, open drain). This requests a CPU interrupt. The CPU acknowledges the interrupt by pulling its $\overline{\text{IORQ}}$ output Low during an $\overline{\text{M1}}$ cycle. It is typically connected to the $\overline{\text{INT}}$ pin of the CPU with a pullup resistor and tied to all other $\overline{\text{INT}}$ pins in the system. This pin can also be used to generate periodic pulses to an external device. It can be used this way only when the DMA is bus master (i.e., the CPU's $\overline{\text{BUSREQ}}$ and $\overline{\text{BUSACK}}$ lines are both Low and the CPU cannot see interrupts).

$\overline{\text{IORQ}}$. *Input/Output Request* (bidirectional, active Low, 3-state). As an input, this indicates that the lower half of the address bus holds a valid I/O port address for transfer of control or status bytes from or to the CPU, respectively; this DMA is the addressed port if its $\overline{\text{CE}}$ pin and its $\overline{\text{WR}}$ or $\overline{\text{RD}}$ pins are simultaneously active. As an output, after the DMA has taken control of the system buses, it indicates that the 8-bit or 16-bit address bus holds a valid port address for another I/O device involved in a DMA transfer of data. When $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are both active simultaneously, an interrupt acknowledge is indicated.

$\overline{\text{M1}}$. *Machine Cycle One* (input, active Low). Indicates that the current CPU machine cycle is an instruction fetch. It is used by the DMA to decode the return-from-interrupt instruction (RETI) (ED-4D) sent by the CPU. During two-byte instruction fetches, $\overline{\text{M1}}$ is active as each opcode byte is fetched. An interrupt acknowledge is indicated when both $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ are active.

$\overline{\text{MREQ}}$. *Memory Request* (output, active Low, 3-state). This indicates that the address bus holds a valid address for a memory read or write operation. After the DMA has taken control of the system buses, it indicates a DMA transfer request from or to memory.

$\overline{\text{RD}}$. *Read* (bidirectional, active Low, 3-state). As an input, this indicates that the CPU wants to read status bytes from the DMA's read registers. As an output, after the DMA has taken control of the system buses, it indicates a DMA-controlled read from a memory or I/O port address.

RDY. *Ready* (input, programmable active Low or High). This is monitored by the DMA to determine when a peripheral device associated with a DMA port is ready for a read or write operation. Depending on the mode of DMA operation (Byte, Burst or Continuous), the RDY line indirectly controls DMA activity by causing the $\overline{\text{BUSREQ}}$ line to go Low or High.

$\overline{\text{WR}}$. *Write* (bidirectional, active Low, 3-state). As an input, this indicates that the CPU wants to write control or command bytes to the DMA write registers. As an output, after the DMA has taken control of the system buses, it indicates a DMA-controlled write to a memory or I/O port address.

Programming

The Z-80 DMA has two programmable fundamental states: (1) an enabled state, in which it can gain control of the system buses and direct the transfer of data between ports, and (2) a disabled state, in which it can initiate neither bus requests nor data transfers. When the DMA is powered up or reset by any means, it is automatically placed into the disabled state. Program commands can be written to it by the CPU in either state, but this automatically puts the DMA in the disabled state, which is maintained until an enable command is issued by the CPU. The CPU must program the DMA in advance of any data search or transfer by addressing it as an I/O port and sending a sequence of control bytes using an Output instruction (such as OTIR for the Z-80 CPU).

Writing. Control or command bytes are written into one or more of the Write Register groups (WR0-WR6) by first writing to the base register byte in that group. All groups have base registers and most groups have additional associated registers. The associated registers in a group are sequentially accessed by first writing a byte to the base register containing register-group identification and pointer bits (1's) to one or more of that base register's associated registers.

This is illustrated in Figure 8b. In this figure, the sequence in which associated registers within a group can be written to is shown by the vertical position of the associated registers. For example, if a byte written to the DMA contains the bits that identify WR0 (bits D0, D1 and D7), and also contains 1's in the bit positions that point to the associated "Port A Starting Address (low byte)" and "Port A Starting Address (high byte)," then the next two bytes written to the DMA will be stored in these two registers, in that order.

Reading. The Read Registers (RR0-RR6) are read by the CPU by addressing the DMA as an I/O port using an Input instruction (such as INIR for the Z-80 CPU). The readable bytes contain DMA status, byte-counter values, and port addresses since the last DMA reset. The registers are always read in a fixed sequence beginning with RR0 and ending with RR6. However, the register read in this sequence is determined by programming the Read Mask in WR6. The sequence of reading is initialized by writing an Initiate Read Sequence or Set Read Status command to WR6. After a Reset DMA, the sequence must be initialized with the Initiate Read Sequence command or a Read Status command. The sequence of reading all registers that are not excluded by the Read Mask register must be completed before a new Initiate Read Sequence or Read Status command.

Fixed-Address Programming. A special circumstance arises when programming a destination port to have a fixed address. The load command in WR6 only loads a fixed address to a port selected as the source, not to a port selected as the destination. Therefore, a fixed destination address must be loaded by temporarily declaring it a fixed-source address and subsequently declaring the true source as such, thereby implicitly making the other a destination.

The following example illustrates the steps in this procedure, assuming that transfers are to occur from a variable-address source (Port A) to a fixed-address destination (Port B):

1. Temporarily declare Port B as source in WR0.
2. Load Port B address in WR6.
3. Declare Port A as source in WR0.

Programming (Continued)

4. Load Port A address in WR6.
5. Enable DMA in WR6.

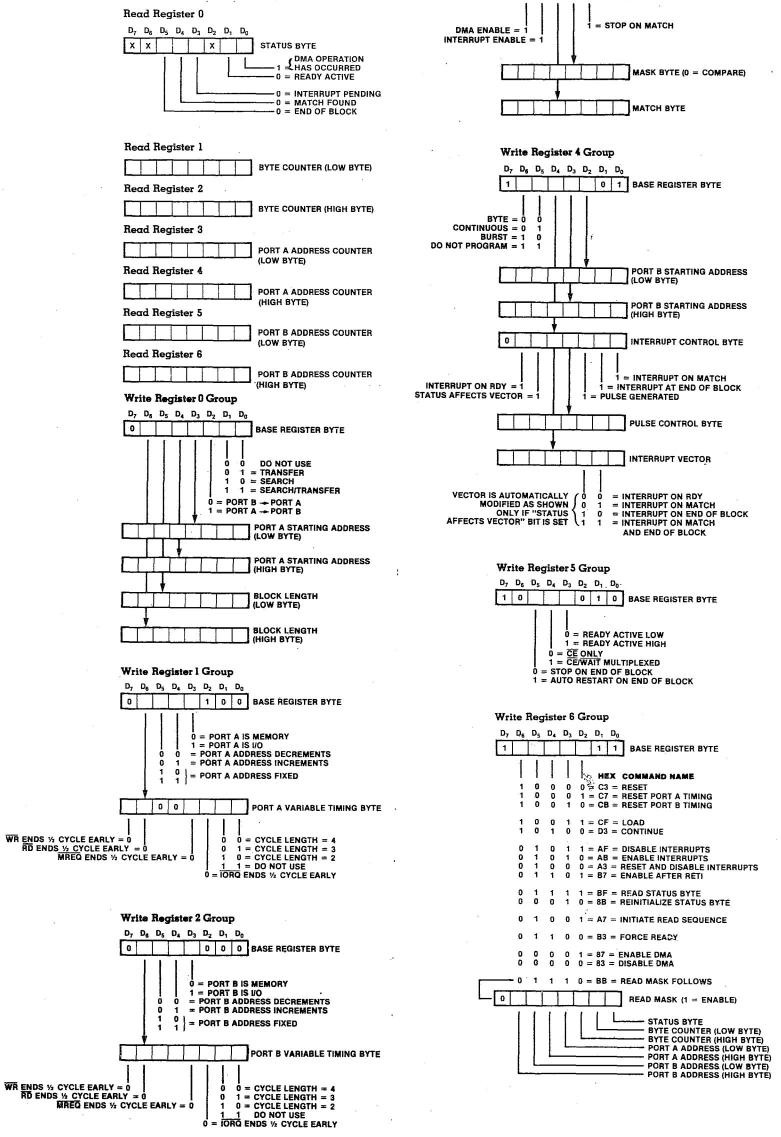
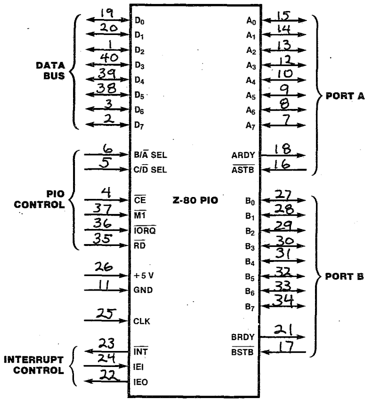


Figure 8b. Write Registers

Z8420 Z80® PIO Parallel Input/Output Controller

Features



Pin Description

A_0-A_7. *Port A Bus* (bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port A of the PIO and a peripheral device. A_0 is the least significant bit of the Port A data bus.

ARDY. *Register A Ready* (output, active High). The meaning of this signal depends on the mode of operation selected for Port A as follows:

Output Mode. This signal goes active to indicate that the Port A output register has been loaded and the peripheral data bus is stable and ready for transfer to the peripheral device.

Input Mode. This signal is active when the Port A input register is empty and ready to accept data from the peripheral device.

Bidirectional Mode. This signal is active when data is available in the Port A output register for transfer to the peripheral device. In this mode, data is not placed on the Port A data bus, unless $\overline{\text{ASTB}}$ is active.

Control Mode. This signal is disabled and forced to a Low state.

$\overline{\text{ASTB}}$. *Port A Strobe Pulse From Peripheral Device* (input, active Low). The meaning of this signal depends on the mode of operation selected for Port A as follows:

Output Mode. The positive edge of this strobe is issued by the peripheral to acknowledge the receipt of data made available by the PIO.

Input Mode. The strobe is issued by the peripheral to load data from the peripheral into the Port A input register. Data is loaded into the PIO when this signal is active.

Bidirectional Mode. When this signal is active, data from the Port A output register is gated onto the Port A bidirectional data bus. The positive edge of the strobe acknowledges the receipt of the data.

Control Mode. The strobe is inhibited internally.

B_0-B_7. *Port B Bus* (bidirectional, 3-state). This 8-bit bus transfers data, status, or control information between Port B and a peripheral device. The Port B data bus can supply 1.5 mA at 1.5 V to drive Darlington transistors. B_0 is the least significant bit of the bus.

B/$\overline{\text{A}}$. *Port B Or A Select* (input, High = B). This pin defines which port is accessed during a data transfer between the CPU and the PIO. A Low on this pin selects Port A; a High selects Port B. Often address bit A_0 from the CPU is used for this selection function.

BRDY. *Register B Ready* (output, active High). This signal is similar to ARDY, except that in the Port A bidirectional mode this signal is High when the Port A input register is empty and ready to accept data from the peripheral device.

$\overline{\text{BSTB}}$. *Port B Strobe Pulse From Peripheral Device* (input, active Low). This signal is similar to $\overline{\text{ASTB}}$, except that in the Port A bidirectional mode this signal strobes data from the peripheral device into the Port A input register.

C/$\overline{\text{D}}$. *Control Or Data Select* (input, High = C). This pin defines the type of data transfer to be performed between the CPU and the PIO. A High on this pin during a CPU write to the PIO causes the Z-80 data bus to be interpreted as a *command* for the port selected by the B/$\overline{\text{A}}$ Select line. A Low on this pin means that the Z-80 data bus is being used to transfer data between the CPU and the PIO. Often address bit A_1 from the CPU is used for this function.

$\overline{\text{CE}}$. *Chip Enable* (input, active Low). A Low on this pin enables the PIO to accept command or data inputs from the CPU during a write cycle or to transmit data to the CPU during a read cycle. This signal is generally decoded from four I/O port numbers for Ports A and B, data, and control.

CLK. *System Clock* (input). The Z-80 PIO uses the standard single-phase Z-80 system clock.

D_0-D_7. *Z-80 CPU Data Bus* (bidirectional, 3-state). This bus is used to transfer all data and commands between the Z-80 CPU and the Z-80 PIO. D_0 is the least significant bit.

IEI. *Interrupt Enable In* (input, active High). This signal is used to form a priority-interrupt daisy chain when more than one interrupt-driven device is being used. A High level on this pin indicates that no other devices of higher priority are being serviced by a CPU interrupt service routine.

Pin Description (Continued)

IEO. *Interrupt Enable Out* (output, active High). The IEO signal is the other signal required to form a daisy chain priority scheme. It is High only if IEI is High and the CPU is not servicing an interrupt from this PIO. Thus this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{\text{INT}}$. *Interrupt Request* (output, open drain, active Low). When $\overline{\text{INT}}$ is active the Z-80 PIO is requesting an interrupt from the Z-80 CPU.

$\overline{\text{IORQ}}$. *Input/Output Request* (input from Z-80 CPU, active Low). $\overline{\text{IORQ}}$ is used in conjunction with B/$\overline{\text{A}}$, C/$\overline{\text{D}}$, $\overline{\text{CE}}$, and $\overline{\text{RD}}$ to transfer commands and data between the Z-80 CPU and the Z-80 PIO. When $\overline{\text{CE}}$, $\overline{\text{RD}}$, and $\overline{\text{IORQ}}$ are active, the port addressed by B/$\overline{\text{A}}$ transfers data to the CPU (a read operation). Conversely, when $\overline{\text{CE}}$ and $\overline{\text{IORQ}}$ are active but $\overline{\text{RD}}$ is not, the port addressed by B/$\overline{\text{A}}$ is written into from the CPU with either data or control information, as specified by C/$\overline{\text{D}}$. Also, if $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are active simultaneously, the CPU is acknowledging an interrupt; the interrupting port automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

$\overline{\text{M1}}$. *Machine Cycle* (input from CPU, active Low). This signal is used as a sync pulse to control several internal PIO operations. When both the $\overline{\text{M1}}$ and $\overline{\text{RD}}$ signals are active, the Z-80 CPU is fetching an instruction from memory. Conversely, when both $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ are active, the CPU is acknowledging an interrupt. In addition, $\overline{\text{M1}}$ has two other functions within the Z-80 PIO: it synchronizes the PIO interrupt logic; when $\overline{\text{M1}}$ occurs without an active $\overline{\text{RD}}$ or $\overline{\text{IORQ}}$ signal, the PIO is reset.

$\overline{\text{RD}}$. *Read Cycle Status* (input from Z-80 CPU, active Low). If $\overline{\text{RD}}$ is active, or an I/O operation is in progress, $\overline{\text{RD}}$ is used with B/$\overline{\text{A}}$, C/$\overline{\text{D}}$, $\overline{\text{CE}}$, and $\overline{\text{IORQ}}$ to transfer data from the Z-80 PIO to the Z-80 CPU.

Programming

Mode 0, 1, or 2. *(Byte Input, Output, or Bidirectional).* Programming a port for Mode 0, 1, or 2 requires two words per port. These words are:

A Mode Control Word. Selects the port operating mode (Figure 6). This word may be written any time.

An Interrupt Vector. The Z-80 PIO is designed for use with the Z-80 CPU in interrupt Mode 2 (Figure 7). When interrupts are enabled, the PIO must provide an interrupt vector.

Mode 3. *(Bit Input/Output).* Programming a port for Mode 3 operation requires a control word, a vector (if interrupts are enabled), and three additional words, described as follows:

I/O Register Control. When Mode 3 is selected, the mode control word must be followed by another control word that sets the I/O control register, which in turn defines which port lines are inputs and which are outputs (Figure 8).

Interrupt Control Word. In Mode 3, handshake is not used. Interrupts are generated as a logic function of the input signal levels. The interrupt control word sets the logic conditions and the logic levels required for generating an interrupt. Two logic conditions or functions are available: AND (if all input bits change to the active level, an interrupt is triggered), and OR (if any one of the input bits changes to the active level, an interrupt is triggered). Bit D_6 sets the logic function, as shown in Figure 9. The active level of the input bits can be set either High or Low. The active level is controlled by Bit D_5.

Mask Control Word. This word sets the mask control register, allowing any unused bits to be masked off. If any bits are to be masked, then D_4 must be set. When D_4 is set, the next word written to the port must be a mask control word (Figure 10).

Interrupt Disable. There is one other control word which can be used to enable or disable a port interrupt. It can be used without changing the rest of the interrupt control word (Figure 11).

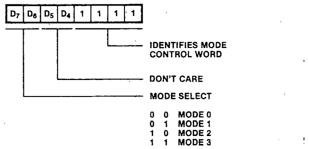


Figure 6. Mode Control Word

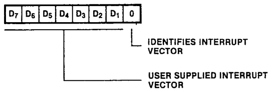


Figure 7. Interrupt Vector Word

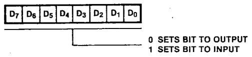


Figure 8. I/O Register Control Word

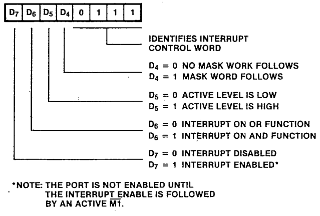

D7 D6 D5 D4 0 1 1 1
IDENTIFIES INTERRUPT CONTROL WORD
D4 = 0 NO MASK WORK FOLLOWS
D4 = 1 MASK WORD FOLLOWS
D5 = 0 ACTIVE LEVEL IS LOW
D5 = 1 ACTIVE LEVEL IS HIGH
D6 = 0 INTERRUPT ON OR FUNCTION
D6 = 1 INTERRUPT ON AND FUNCTION
D7 = 0 INTERRUPT DISABLED
D7 = 1 INTERRUPT ENABLED*
*NOTE: THE PORT IS NOT ENABLED UNTIL THE INTERRUPT ENABLE IS FOLLOWED BY AN ACTIVE M1.

Figure 9. Interrupt Control Word

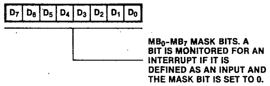

D7 D6 D5 D4 D3 D2 D1 D0
MB0-MB7 MASK BITS. A BIT IS MONITORED FOR AN INTERRUPT IF IT IS DEFINED AS AN INPUT AND THE MASK BIT IS SET TO 0.

Figure 10. Mask Control Word

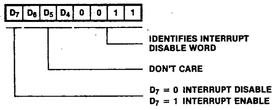
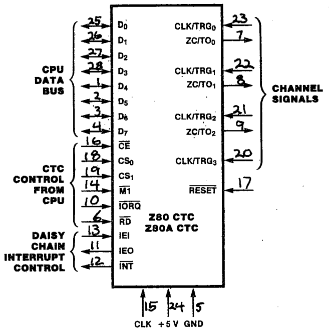


Figure 11. Interrupt Disable Word

Zilog

Z8430
Z80® CTC Counter/ Timer Circuit

Features



Pin Description

$\overline{\text{CE}}$. *Chip Enable* (input, active Low). When enabled the CTC accepts control words, interrupt vectors, or time constant data words from the data bus during an I/O write cycle; or transmits the contents of the down-counter to the CPU during an I/O read cycle. In most applications this signal is decoded from the eight least significant bits of the address bus for any of the four I/O port addresses that are mapped to the four counter-timer channels.

CLK. *System Clock* (input). Standard single-phase Z-80 system clock.

CLK/TRG$_0$-CLK/TRG$_3$. *External Clock/Timer Trigger* (input, user-selectable active High or Low). Four pins corresponding to the four Z-80 CTC channels. In counter mode, every active edge on this pin decrements the down-counter. In timer mode, an active edge starts the timer.

CS$_0$-CS$_1$. *Channel Select* (inputs active High). Two-bit binary address code selects one of the four CTC channels for an I/O write or read (usually connected to A_0 and A_1).

D_0-D_7. *System Data Bus* (bidirectional, 3-state). Transfers all data and commands between the Z-80 CPU and the Z-80 CTC.

IEI. *Interrupt Enable In* (input, active High). A High indicates that no other interrupting devices of higher priority in the daisy chain are being serviced by the Z-80 CPU.

IEO. *Interrupt Enable Out* (output, active High). High only if IEI is High and the Z-80 CPU is not servicing an interrupt from any Z-80 CTC channel. IEO blocks lower priority devices from interrupting while a higher priority interrupting device is being serviced.

$\overline{\text{INT}}$. *Interrupt Request* (output, open drain, active Low). Low when any Z-80 CTC channel that has been programmed to enable interrupts has a zero-count condition in its down-counter.

$\overline{\text{IORQ}}$. *Input/Output Request* (input from CPU, active Low). Used with $\overline{\text{CE}}$ and $\overline{\text{RD}}$ to transfer data and channel control words between the Z-80 CPU and the Z-80 CTC. During a write cycle, $\overline{\text{IORQ}}$ and $\overline{\text{CE}}$ are active and $\overline{\text{RD}}$ inactive. The Z-80 CTC does not receive a specific write signal; rather, it internally generates its own from the inverse of an active $\overline{\text{RD}}$ signal. In a read cycle, $\overline{\text{IORQ}}$, $\overline{\text{CE}}$ and $\overline{\text{RD}}$ are active; the contents of the down-counter are read by the Z-80 CPU. If $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are both true, the CPU is acknowledging an interrupt request, and the highest priority interrupting channel places its interrupt vector on the Z-80 data bus.

$\overline{\text{M1}}$. *Machine Cycle One* (input from CPU, active Low). When $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ are active, the Z-80 CPU is acknowledging an interrupt. The Z-80 CTC then places an interrupt vector on the data bus if it has highest priority, and if a channel has requested an interrupt ($\overline{\text{INT}}$).

$\overline{\text{RD}}$. *Read Cycle Status* (input, active Low). Used in conjunction with $\overline{\text{IORQ}}$ and $\overline{\text{CE}}$ to transfer data and channel control words between the Z-80 CPU and the Z-80 CTC.

$\overline{\text{RESET}}$. *Reset* (input active Low). Terminates all down-counts and disables all interrupts by resetting the interrupt bits in all control registers; the ZC/TO and the Interrupt outputs go inactive; IEO reflects IEI; D_0-D_7 go to the high-impedance state.

ZC/TO$_0$-ZC/TO$_2$. *Zero Count/Timeout* (output, active High). Three ZC/TO pins corresponding to Z-80 CTC channels 2 through 0 (Channel 3 has no ZC/TO pin). In both counter and timer modes the output is an active High pulse when the down-counter decrements to zero.

Programming

Each Z-80 CTC channel must be programmed prior to operation. Programming consists of writing two words to the I/O port that corresponds to the desired channel. The first word is a control word that selects the operating mode and other parameters; the second word is a time constant, which is a binary data word with a value from 1 to 256. A time constant word must be preceded by a channel control word.

After initialization, channels may be reprogrammed at any time. If updated control and time constant words are written to a channel during the count operation, the count continues to zero before the new time constant is loaded into the counter.

If the interrupt on any Z-80 CTC channel is enabled, the programming procedure should also include an interrupt vector. Only one vector is required for all four channels, because the interrupt logic automatically modifies the vector for the channel requesting service.

A control word is identified by a 1 in bit 0. A 0 in bit 2 indicates a time constant word is to follow. Interrupt vectors are always addressed to Channel 0, and identified by a 0 in bit 0.

Addressing. During programming, channels are addressed with the channel select pins CS_1 and CS_2. A 2-bit binary code selects the appropriate channel as shown in the following table.

Channel	CS_1	CS_0
0	0	0
1	0	1
2	1	0
3	1	1

Reset. The CTC has both hardware and software resets. The hardware reset terminates all down-counts and disables all CTC interrupts by resetting the interrupt bits in the control registers. In addition, the ZC/TO and Interrupt outputs go inactive, IEO reflects IEI, and D_0-D_7 go to the high-impedance state. All channels must be completely reprogrammed after a hardware reset.

The software reset is controlled by bit 1 in the channel control word. When a channel receives a software reset, it stops counting. When a software reset is used, the other bits in the control word also change the contents of the channel control register. After a software reset a new time constant word must be written to the same channel.

If the channel control word has both bits D_1 and D_2 set to 1, the addressed channel stops operating, pending a new time constant word. The channel is ready to resume after the new constant is programmed. In timer mode, if D_3 = 0, operation is triggered automatically when the time constant word is loaded.

Channel Control Word Programming. The channel control word is shown in Figure 5. It sets the modes and parameters described below.

Interrupt Enable. D_7 enables the interrupt, so that an interrupt output ($\overline{INT}$) is generated at zero count. Interrupts may be programmed in either mode and may be enabled or disabled at any time.

Operating Mode. D_6 selects either timer or counter mode.

Prescaler Factor. (Timer Mode Only). D_5 selects factor—either 16 or 256.

Trigger Slope. D_4 selects the active edge or slope of the CLK/TRG input pulses. Note that reprogramming the CLK/TRG slope during operation is equivalent to issuing an active edge. If the trigger slope is changed by a control word update while a channel is pending operation in timer mode, the result is the same as a CLK/TRG pulse and the timer starts. Similarly, if the channel is in counter mode, the counter decrements.

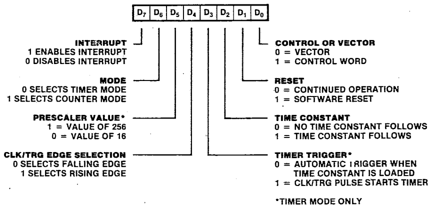


Figure 5. Channel Control Word

Programming (Continued)

Trigger Mode (Timer Mode Only). D_3 selects the trigger mode for timer operation. When D_3 is reset to 0, the timer is triggered automatically. The time constant word is programmed during an I/O write operation, which takes one machine cycle. At the end of the write operation there is a setup delay of one clock period. The timer starts automatically (decrements) on the rising edge of the second clock pulse (T_2) of the machine cycle following the write operation. Once started, the timer runs continuously. At zero count the timer reloads automatically and continues counting without interruption or delay, until stopped by a reset.

When D_3 is set to 1, the timer is triggered externally through the CLK/TRG input. The time constant word is programmed during an I/O write operation, which takes one machine cycle. The timer is ready for operation on the rising edge of the second clock pulse (T_2) of the following machine cycle. Note that the first timer decrement follows the active edge of the CLK/TRG pulse by a delay time of one clock cycle if a minimum setup time to the rising edge of clock is met. If this minimum is not met, the delay is extended by another clock period. Consequently, for immediate triggering, the CLK/TRG input must precede T_2 by one clock cycle plus its minimum setup time. If the minimum time is not met, the timer will start on the third clock cycle (T_3).

Once started the timer operates continuously, without interruption or delay, until stopped by a reset.

Time Constant to Follow. A 1 in D_2 indicates that the next word addressed to the selected channel is a time constant data word for the time constant register. The time constant word may be written at any time.

A 0 in D_2 indicates no time constant word is to follow. This is ordinarily used when the channel is already in operation and the new channel control word is an update. A channel will not operate without a time constant value. The only way to write a time constant value is to write a control word with D_2 set.

Software Reset. Setting D_1 to 1 causes a software reset, which is described in the Reset section.

Control Word. Setting D_0 to 1 identifies the word as a control word.

Time Constant Programming. Before a channel can start counting it must receive a time constant word from the CPU. During programming or reprogramming, a channel control word in which bit 2 is set must precede the time constant word to indicate that the next word is a time constant. The time constant word can be any value from 1 to 256 (Figure 6). Note that 00_{16} is interpreted as 256.

In timer mode, the time interval is controlled by three factors:

- The system clock period (ϕ)
- The prescaler factor (P), which multiplies the interval by either 16 or 256
- The time constant (T), which is programmed into the time constant register

Consequently, the time interval is the product of $\phi \times P \times T$. The minimum timer resolution is $16 \times \phi$ (4 μs with a 4 MHz clock). The maximum timer interval is $256 \times \phi \times 256$ (16.4 ms with a 4 MHz clock). For longer intervals timers may be cascaded.

Interrupt Vector Programming. If the Z-80 CTC has one or more interrupts enabled, it can supply interrupt vectors to the Z-80 CPU. To do so, the Z-80 CTC must be pre-programmed with the most-significant five bits of the interrupt vector. Programming consists of writing a vector word to the I/O port corresponding to the Z-80 CTC Channel 0. Note that D_0 of the vector word is always zero, to distinguish the vector from a channel control word. D_1 and D_2 are not used in programming the vector word. These bits are supplied by the interrupt logic to identify the channel requesting interrupt service with a unique interrupt vector (Figure 7). Channel 0 has the highest priority.

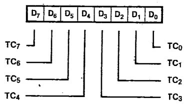


Figure 6. Time Constant Word

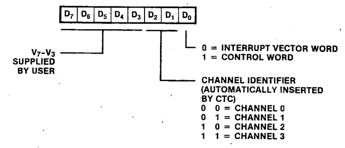
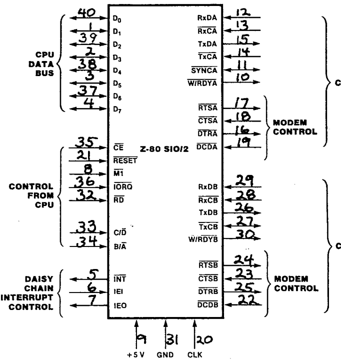


Figure 7. Interrupt Vector Word

Zilog

Z8440
Z80® SIO Serial Input/Output Controller

Features



Pin Description

Figures 1 through 6 illustrate the three pin configurations (bonding options) available in the SIO. The constraints of a 40-pin package make it impossible to bring out the Receive Clock ($\overline{\text{RxC}}$), Transmit Clock ($\overline{\text{TxC}}$), Data Terminal Ready ($\overline{\text{DTR}}$) and Sync ($\overline{\text{SYNC}}$) signals for both channels. Therefore, either Channel B lacks a signal or two signals are bonded together in the three bonding options offered:

- Z-80 SIO/2 lacks $\overline{\text{SYNCB}}$
- Z-80 SIO/1 lacks $\overline{\text{DTRB}}$
- Z-80 SIO/0 has all four signals, but $\overline{\text{TxCB}}$ and $\overline{\text{RxCB}}$ are bonded together

The first bonding option above (SIO/2) is the preferred version for most applications. The pin descriptions are as follows:

B/$\overline{\text{A}}$. *Channel A Or B Select* (input, High selects Channel B). This input defines which channel is accessed during a data transfer between the CPU and the SIO. Address bit A_0 from the CPU is often used for the selection function.

C/$\overline{\text{D}}$. *Control Or Data Select* (input, High selects Control). This input defines the type of information transfer performed between the CPU and the SIO. A High at this input during a CPU write to the SIO causes the information on the data bus to be interpreted as a command for the channel selected by B/$\overline{\text{A}}$. A Low at C/$\overline{\text{D}}$ means that the information on the data bus is data. Address bit A_1 is often used for this function.

$\overline{\text{CE}}$. *Chip Enable* (input, active Low). A Low level at this input enables the SIO to accept command or data input from the CPU during a write cycle or to transmit data to the CPU during a read cycle.

CLK. *System Clock* (input). The SIO uses the standard Z-80 System Clock to synchronize internal signals. This is a single-phase clock.

$\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$. *Clear To Send* (inputs, active Low). When programmed as Auto Enables, a Low on these inputs enables the respective transmitter. If not programmed as Auto Enables, these inputs may be programmed as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these inputs and interrupts the CPU on both logic level transitions. The Schmitt-trigger buffering does not guarantee a specified noise-level margin.

D_0-D_7. *System Data Bus* (bidirectional, 3-state). The system data bus transfers data and commands between the CPU and the Z-80 SIO. D_0 is the least significant bit.

$\overline{\text{DCDA}}$, $\overline{\text{DCDB}}$. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if the SIO is programmed for Auto Enables; otherwise they may be used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow-risetime signals. The SIO detects pulses on these pins and interrupts the CPU on both logic level transitions. Schmitt-trigger buffering does not guarantee a specific noise-level margin.

$\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$. *Data Terminal Ready* (outputs, active Low). These outputs follow the state programmed into Z-80 SIO. They can also be programmed as general-purpose outputs.

In the Z-80 SIO/1 bonding option, $\overline{\text{DTRB}}$ is omitted.

IEI. *Interrupt Enable In* (input, active High). This signal is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this SIO. Thus, this signal blocks lower priority devices from interrupting while a higher priority device is being serviced by its CPU interrupt service routine.

$\overline{\text{INT}}$. *Interrupt Request* (output, open drain, active Low). When the SIO is requesting an interrupt, it pulls $\overline{\text{INT}}$ Low.

$\overline{\text{IORQ}}$. *Input/Output Request* (input from CPU, active Low). $\overline{\text{IORQ}}$ is used in conjunction with B/$\overline{\text{A}}$, C/$\overline{\text{D}}$, $\overline{\text{CE}}$ and $\overline{\text{RD}}$ to transfer commands and data between the CPU and the SIO. When $\overline{\text{CE}}$, $\overline{\text{RD}}$ and $\overline{\text{IORQ}}$ are all active, the channel selected by B/$\overline{\text{A}}$ transfers data to the CPU (a read operation). When $\overline{\text{CE}}$ and $\overline{\text{IORQ}}$ are active but $\overline{\text{RD}}$ is inactive, the channel selected by B/$\overline{\text{A}}$ is written to by the CPU with either data or control information as specified by C/$\overline{\text{D}}$. If $\overline{\text{IORQ}}$ and $\overline{\text{M1}}$ are active simultaneously, the CPU is acknowledging an interrupt and the SIO automatically places its interrupt vector on the CPU data bus if it is the highest priority device requesting an interrupt.

Pin Description (Continued)

$\overline{\text{M1}}$. *Machine Cycle* (input from Z-80 CPU, active Low). When $\overline{\text{M1}}$ is active and $\overline{\text{RD}}$ is also active, the Z-80 CPU is fetching an instruction from memory; when $\overline{\text{M1}}$ is active while $\overline{\text{IORQ}}$ is active, the SIO accepts $\overline{\text{M1}}$ and $\overline{\text{IORQ}}$ as an interrupt acknowledge if the SIO is the highest priority device that has interrupted the Z-80 CPU.

$\overline{\text{RxCA}}$, $\overline{\text{RxCB}}$. *Receiver Clocks* (inputs). Receive data is sampled on the rising edge of $\overline{\text{RxC}}$. The Receive Clocks may be 1, 16, 32 or 64 times the data rate in asynchronous modes. These clocks may be driven by the Z-80 CTC Counter Timer Circuit for programmable baud rate generation. Both inputs are Schmitt-trigger buffered (no noise level margin is specified).

In the Z-80 SIO/0 bonding option, $\overline{\text{RxCB}}$ is bonded together with $\overline{\text{TxCB}}$.

$\overline{\text{RD}}$. *Read Cycle Status* (input from CPU, active Low). If $\overline{\text{RD}}$ is active, a memory or I/O read operation is in progress. $\overline{\text{RD}}$ is used with B/$\overline{\text{A}}$, $\overline{\text{CE}}$ and $\overline{\text{IORQ}}$ to transfer data from the SIO to the CPU.

RxDA, RxDB. *Receive Data* (inputs, active High). Serial data at TTL levels.

$\overline{\text{RESET}}$. *Reset* (input, active Low). A Low $\overline{\text{RESET}}$ disables both receivers and transmitters, forces TxDA and TxDB marking, forces the modem controls High and disables all interrupts. The control registers must be rewritten after the SIO is reset and before data is transmitted or received.

$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$. *Request To Send* (outputs, active Low). When the RTS bit in Write Register 5 (Figure 14) is set, the $\overline{\text{RTS}}$ output goes Low. When the RTS bit is reset in the Asynchronous mode, the output goes High after the transmitter is empty. In Synchronous modes, the $\overline{\text{RTS}}$ pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

$\overline{\text{SYNCA}}$, $\overline{\text{SYNCB}}$. *Synchronization* (inputs/outputs, active Low). These pins can act either as inputs or outputs. In the asynchronous receive mode, they are inputs similar to $\overline{\text{CTS}}$ and $\overline{\text{DCD}}$. In this mode, the transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 (Figure 13), but have no other function. In the External Sync mode, these lines also act as inputs. When external synchronization is achieved, $\overline{\text{SYNC}}$ must be driven Low on the second rising edge of $\overline{\text{RxC}}$ after that rising edge of $\overline{\text{RxC}}$ on which the last bit of the sync character was received. In other words, after the sync pattern is detected, the external logic must wait for two full Receive Clock cycles to activate the $\overline{\text{SYNC}}$ input. Once $\overline{\text{SYNC}}$ is forced Low, it should be kept Low until the CPU informs the external synchronization detect logic that synchronization has been lost or a new message is about to start. Character assembly begins on the rising edge of $\overline{\text{RxC}}$ that immediately precedes the falling edge of $\overline{\text{SYNC}}$ in the External Sync mode.

In the internal synchronization mode (Monosync and Bisync), these pins act as outputs that are active during the part of the receive clock ($\overline{\text{RxC}}$) cycle in which sync characters are recognized. The sync condition is not latched, so these outputs are active each time a sync pattern is recognized, regardless of character boundaries.

In the Z-80 SIO/2 bonding option, $\overline{\text{SYNCB}}$ is omitted.

$\overline{\text{TxCA}}$, $\overline{\text{TxCB}}$. *Transmitter Clocks* (inputs). In asynchronous modes, the Transmitter Clocks may be 1, 16, 32 or 64 times the data rate; however, the clock multiplier for the transmitter and the receiver must be the same. The Transmit Clock inputs are Schmitt-trigger buffered for relaxed rise- and fall-time requirements (no noise level margin is specified). Transmitter Clocks may be driven by the Z-80 CTC Counter Timer Circuit for programmable baud rate generation.

In the Z-80 SIO/0 bonding option, $\overline{\text{TxCB}}$ is bonded together with $\overline{\text{RxCB}}$.

TxDA, TxDB. *Transmit Data* (outputs, active High). Serial data at TTL levels. TxD changes from the falling edge of $\overline{\text{TxC}}$.

$\overline{\text{W}}$/$\overline{\text{RDYA}}$, $\overline{\text{W}}$/$\overline{\text{RDYB}}$. *Wait/Ready A, Wait/Ready B* (outputs, open drain when programmed for Wait function, driven High and Low when programmed for Ready function). These dual-purpose outputs may be programmed as Ready lines for a DMA controller or as Wait lines that synchronize the CPU to the SIO data rate. The reset state is open drain.

Programming

The system program first issues a series of commands that initialize the basic mode of operation and then other commands that qualify conditions within the selected mode. For example, the asynchronous mode, character length, clock rate, number of stop bits, even or odd parity might be set first; then the interrupt mode; and finally, receiver or transmitter enable.

Both channels contain registers that must be programmed via the system program prior to operation. The channel-select input ($B/\overline{A}$) and the control/data input ($C/\overline{D}$) are the command-structure addressing controls, and are normally controlled by the CPU address bus. Figures 15 and 16 illustrate the timing relationships for programming the write registers and transferring data and status.

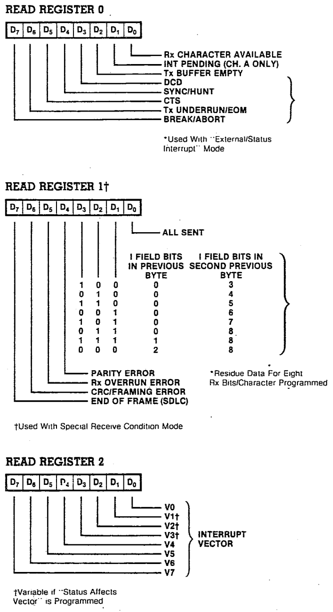
Read Registers. The SIO contains three read registers for Channel B and two read registers for Channel A (RR0-RR2 in Figure 13) that can be read to obtain the status information; RR2 contains the internally-modifiable interrupt vector and is only in the Channel B register set. The status information includes error conditions, interrupt vector and standard communications-interface signals.

To read the contents of a selected read register other than RR0, the system program must first write the pointer byte to WR0 in exactly the same way as a write register operation. Then, by executing a read instruction, the contents of the addressed read register can be read by the CPU.

The status bits of RR0 and RR1 are carefully grouped to simplify status monitoring. For example, when the interrupt vector indicates that a Special Receive Condition interrupt has occurred, all the appropriate error bits can be read from a single register (RR1).

Write Registers. The SIO contains eight write registers for Channel B and seven write registers for Channel A (WR0-WR7 in Figure 14) that are programmed separately to configure the functional personality of the channels; WR2 contains the interrupt vector for both channels and is only in the Channel B register set. With the exception of WR0, programming the write registers requires two bytes. The first byte is to WR0 and contains three bits (D_0–D_2) that point to the selected register; the second byte is the actual control word that is written into the register to cor figure the SIO.

WR0 is a special case in that all of the basic commands can be written to it with a single byte. Reset (internal or external) initializes the pointer bits D_0–D_2 to point to WR0. This implies that a channel reset must not be combined with the pointing to any register.



READ REGISTER 0

*Used With "External/Status Interrupt" Mode

READ REGISTER 1†

*Residue Data For Eight Rx Bits/Character Programmed

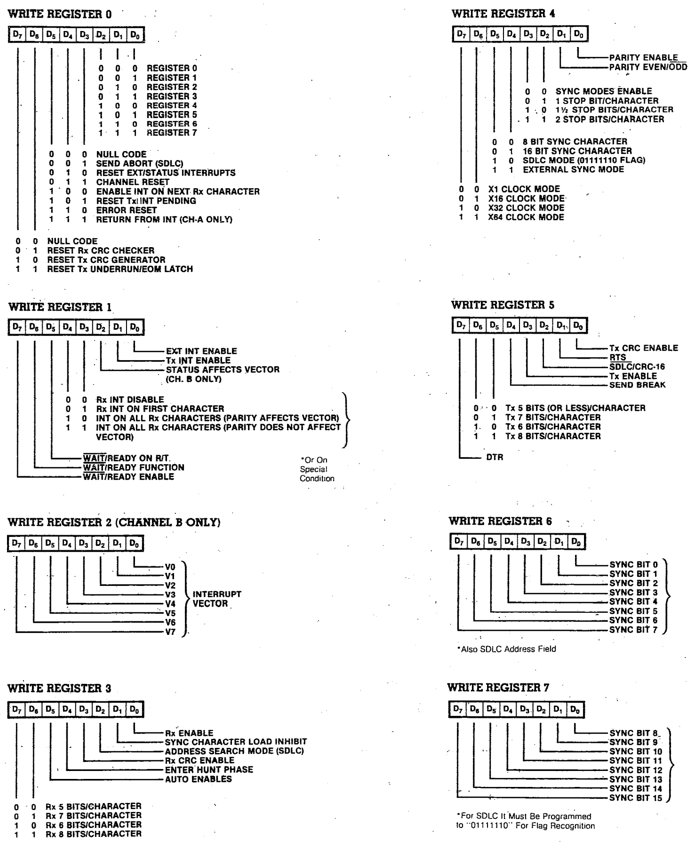
†Used With Special Receive Condition Mode

READ REGISTER 2

†Variable if "Status Affects Vector" is Programmed

Figure 13. Read Register Bit Functions

Programming (Continued)



WRITE REGISTER 0

WRITE REGISTER 1

D7 D6 D5 D4 D3 D2 D1 D0
EXT INT ENABLE
Tx INT ENABLE
STATUS AFFECTS VECTOR (CH. B ONLY)
0 0 Rx INT DISABLE
0 1 Rx INT ON FIRST CHARACTER
1 0 INT ON ALL Rx CHARACTERS (PARITY AFFECTS VECTOR)
1 1 INT ON ALL Rx CHARACTERS (PARITY DOES NOT AFFECT VECTOR)
$\overline{\text{WAIT}}$/READY ON R/T
$\overline{\text{WAIT}}$/READY FUNCTION
$\overline{\text{WAIT}}$/READY ENABLE

*Or On Special Condition

WRITE REGISTER 2 (CHANNEL B ONLY)

WRITE REGISTER 3

WRITE REGISTER 4

WRITE REGISTER 5

WRITE REGISTER 6

*Also SDLC Address Field

WRITE REGISTER 7

*For SDLC It Must Be Programmed to "01111110" For Flag Recognition

Figure 14. Write Register Bit Functions

6.0 TERMINAL PORTION OPERATION

CPU, Timing and Control

(Refer to Figure 1) The 23.814 Mhz oscillator (Osc 1) is used to generate all timing for the terminal. It is used directly as the dot clock (Shift Clock), divided by 13 to drive the UARTs, and divided by 14 (1.701 Mhz) to drive the CRT controller (CCLK) and the CPU (via the clock stretch circuit).

The clock stretch circuit is capable, upon command, of generating clock periods twice the normal length (588 ns versus 1175 ns) for accessing slow memory or peripheral devices. Its output drives the I_0 input of the 6502 CPU. The CPU then outputs I_2, which controls the timing of the CPU bus. I_2 is a slightly delayed version of I_0.

The result of these circuits are I_2 and CCLK, two signals of identical frequency but opposite phase, (except during clock stretched cycles). The importance of this will be made clear later in our discussion of the display controller.

The CPU fetches its program from the ROMs (Read Only Memory) A41-43. It uses the 6522 (A54) to sense switches S1 and S2 and to generate control signals for the rest of the terminal.

Display Controller

(Refer to Figure 2) Timer T2, part of the 6522, and the 6545 (A55) are used to generate the memory address, in Display RAM, of each character as it is about to be displayed, and the horizontal and vertical synchronization pulses necessary to control the deflection circuits of the monitor.

Timer T2 is used to count horizontal scan lines and interrupt the processor (via NMI) when a specified number of scans has occurred. The processor then loads the memory address of the next data row into the CRT Controller and "sets" this address by generating a carefully-timed reset to the 6545.

At this same time the processor loads a 4 bit value into latch, A61. At the time of the CRT reset this value is transferred to counter A60 and becomes the Row Address of the next data row. This value is then incremented by each horizontal sync pulse until the start of the next data row when it is again preset to a value determined by the CPU.

The CPU and the display controller share access to the System and Display RAM (Random Access Memory). This is done during alternate phases of the I_2 clock. During the positive portion of I_2 the CPU address may be gated onto the RAM address bus by Multiplexers A43-46, and bidirectional transceiver A14 is enabled to pass data between the CPU data bus and the RAM data bus.

During the negative portion of I_2 the 6545 address bus is gated onto the RAM address bus allowing the video data to be latched by A24 and held for the display generator.

This alternating access or "interleaved" access allows the processor to operate at normal speed, without waits of any kind, yet prevents degradation of the display quality that could be caused by inadvertant appropriation of the display bus by the processor to access data.

The only penalty for this scheme is the necessity for fast RAM (150 ns or faster).

Video Generation

(Refer to Figure 3) This Display Data and the Row Address (or scan address) are used to obtain the dots for the next character to be displayed from the character generator ROMs A32 and A33.

These dots are then fed in parallel to shift registers A22 and A23 and emerge serially as raw video.

Additionally, bits 0-3 of Display data and bit 7 of A33 are combined to generate the attribute signals Underline, Blink, Blank, and Reverse. ICs A19, 20, 21 and 30 latch and delay the decoded attributes from the previous data row for carry-over into the next.

Bit 6 of A33 controls the intensity of the character to be displayed.

Gates A1, 2, 10 and 11 are used to modify the raw video to the proper intensity and polarity, and gate it on or off in response to the attribute signals and control signals BOW (used to reverse the entire display), cursor, BLI-RATE (used to blink the video) and FORCE BLANK (used to blank the entire screen).

Transistor Q1 is used to drive the video to the proper voltage and current levels to drive the video module and/or an external monitor (using the composite video jumpers).

I/O Circuits

(Refer to Figure 4) UART A49 is used to receive (and optionally transmit) serial data from (and to) the keyboard. The transmit path to the keyboard is normally used to conduct the bell tone from the 6522 (via driver Q4) to the speaker in the keyboard.

UARTs A50 (Main Port, P3) and A51 (Printer Port, P4) are used to send and receive serial data from P3 and P4 via the drivers, receivers and switching circuits A39, 40, 47, 48, 56, 57, 58 and 59.

The UARTs A49, 50 and A51 (6551s) are connected to the CPU Bus and generate IRQ interrrupts when commanded by the CPU to send or receive data. Additionally these parts contain internal baud rate generators that must be programmed by the CPU to control the baud rates.

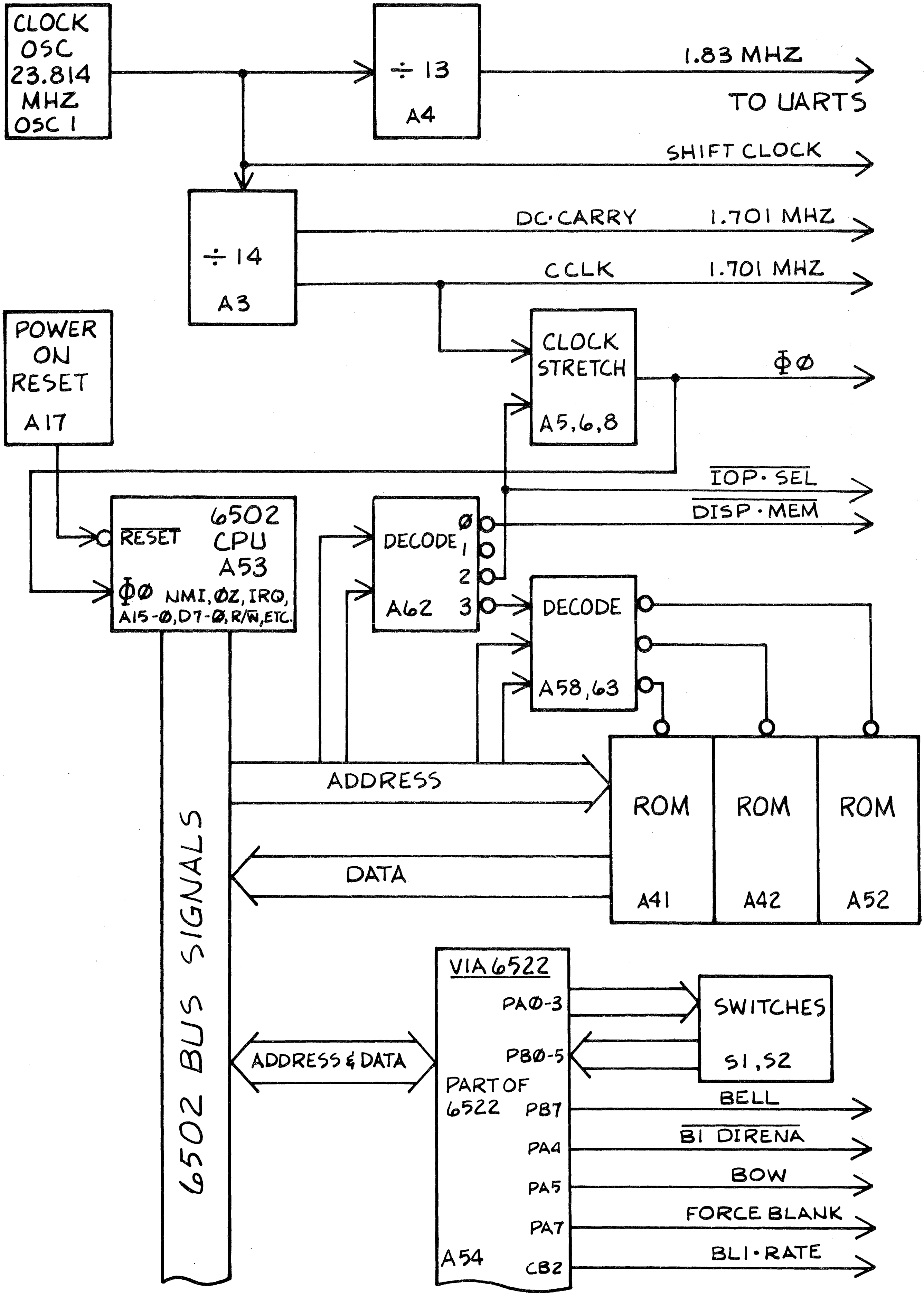


FIG. 1 CPU, TIMING, and CONTROL

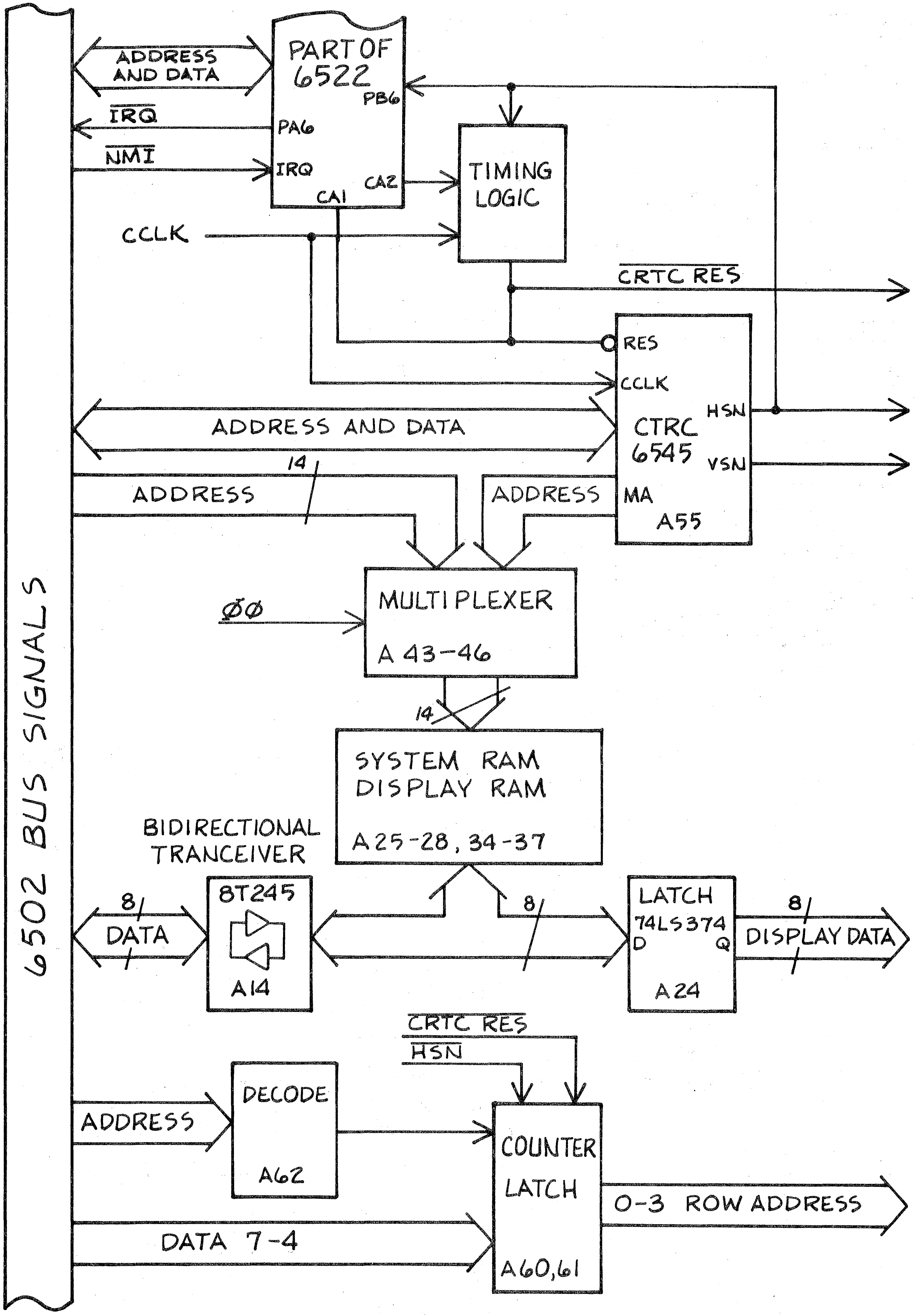


FIG.2 DISPLAY CONTROLLER

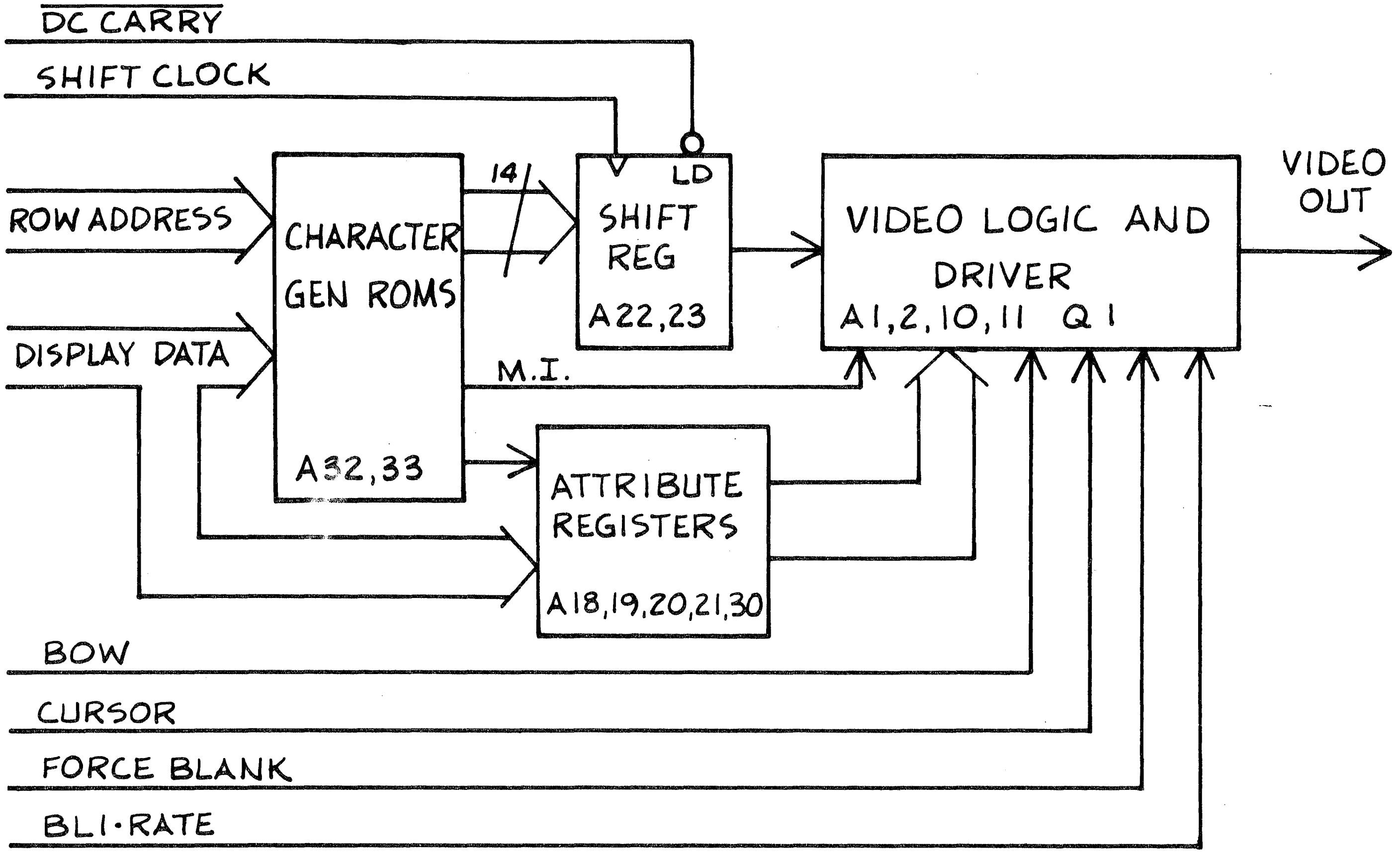


FIG.3 VIDEO GENERATION

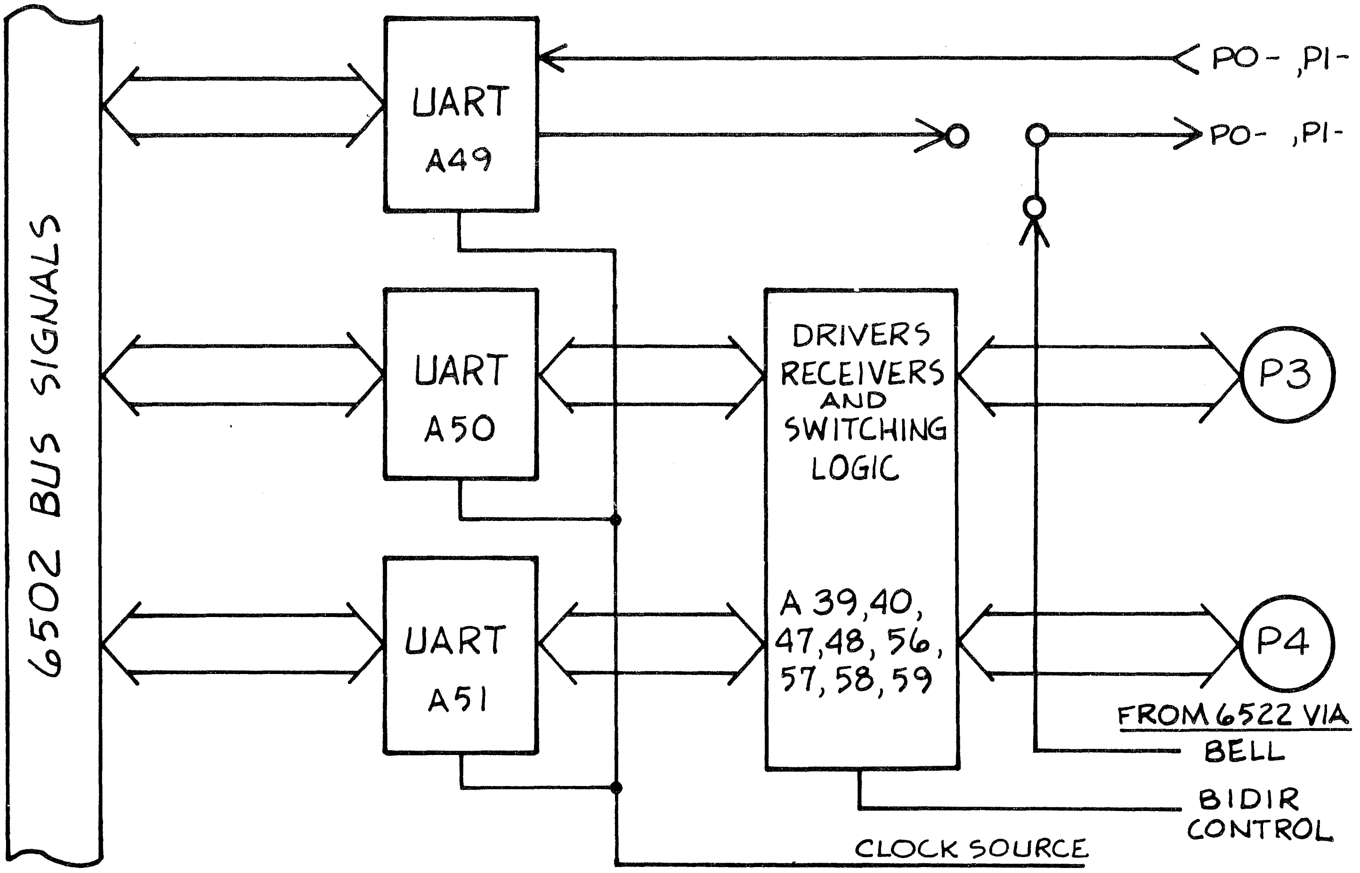


FIG.4 I/O CIRCUITS

Rockwell

R6500 Microcomputer System DATA SHEET

Asynchronous Communication Interface Adapter (ACIA)

The R6551 Asynchronous Communication Interface Adapter (ACIA) provides a program-controlled interface between 8-bit microprocessor-based systems and serial communication data sets and modems.

With its on-chip baud rate generator, the R6551 is capable of transmitting at 15 different program-selectable rates between 50 baud and 19,200 baud, and receiving at either the transmit rate or at 16 times an external clock rate. The R6551 has programmable word lengths of 5, 6, 7, or 8 bits; even, odd or no parity; 1, 1-1/2 or 2 stop bits.

With the R6551, a crystal is the only required external support component — eliminating the multiple-component support that is typically needed.

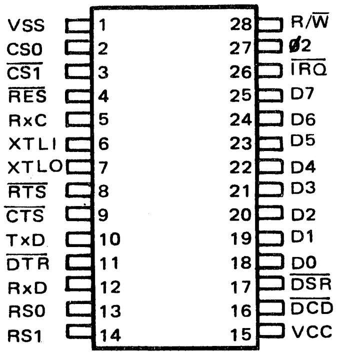
In addition, the R6551 is designed for maximum programmed control from the CPU, to simplify hardware implementation. A control register and a separate command register permit the CPU to easily select the R6551's operating modes and check data, parameters and status.

FEATURES

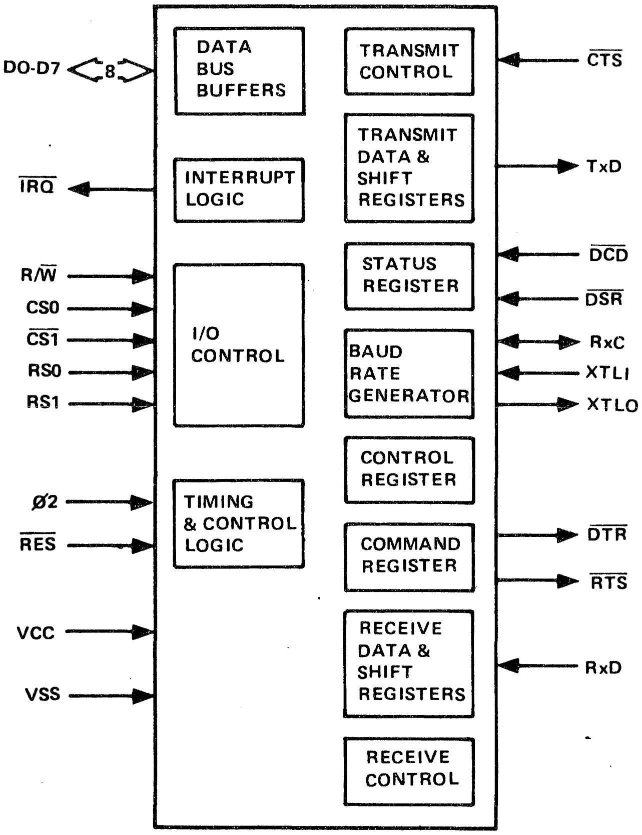
- Compatible with 8-bit microprocessors
- Full duplex or half duplex operation with buffered receiver and transmitter
- 15 programmable Baud Rates (50 to 19,200)
- Receiver data rate may be identical to baud rate or may be 16 times the external clock input
- Data set/modem control functions
- Programmable word lengths, number of stop bits, and parity bit generation and detection
- Programmable interrupt control
- Software reset
- Program-selectable serial echo mode
- Two chip selects
- 2 MHz or 1 MHz clock rate
- Single +5V ±5% power supply
- 28-pin plastic or ceramic DIP
- Full TTL compatibility

Ordering Information

Order Number	Package Type	Frequency	Temperature Range
R6551P	Plastic	1 MHz	0°C to +70°C
R6551AP	Plastic	2 MHz	0°C to +70°C
R6551C	Ceramic	1 MHz	0°C to +70°C
R6551AC	Ceramic	2 MHz	0°C to +70°C



R6551 Pin Configuration

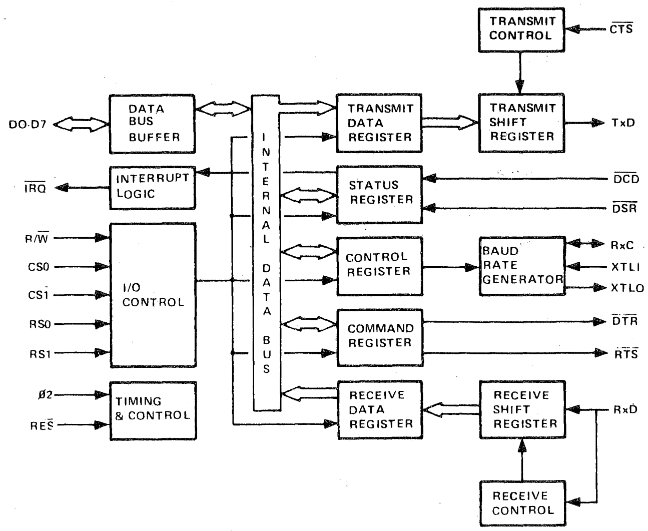


R6551 Interface Diagram

Specifications subject to change without notice

Document No. 29000 D53
Rev. 1, January 1981

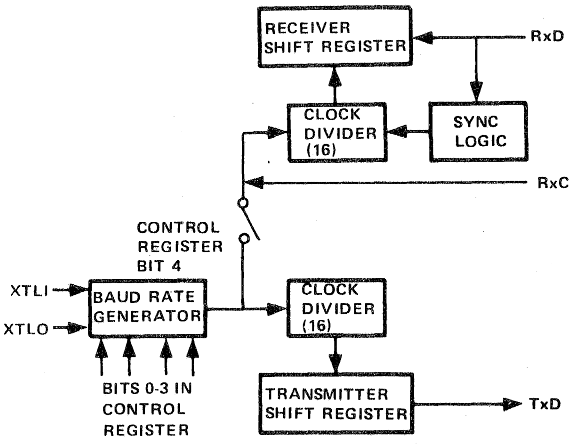
INTERNAL ORGANIZATION



R6551 Block Diagram

Transmitter/Receiver

Bits 0-3 of the Control Register select the divisor used to generate the baud rate for the Transmitter. If the Receiver clock is to use the same baud rate as the Transmitter, then RxC becomes an output pin and can be used to slave other circuits to the R6551.



Transmitter/Receiver Clock Circuits

Transmit and Receive Data Registers

These registers are used as temporary data storage for the 6551 Transmit and Receive circuits. The Transmit Data Register is characterized as follows:

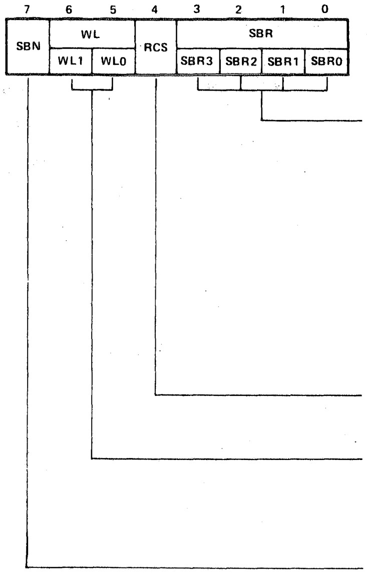
- Bit 0 is the leading bit to be transmitted.
- Unused data bits are the high-order bits and are "don't care" for transmission.

The Receive Data Register is characterized in a similar fashion:

- Bit 0 is the leading bit received.
- Unused data bits are the high-order bits and are "0" for the receiver.
- Parity bits are not contained in the Receive Data Register, but are stripped-off after being used for external parity checking. Parity and all unused high-order bits are "0".

Control Register

The Control Register selects the desired baud rate, frequency source, word length, and the number of stop bits.



SELECTED BAUD RATE (SBR)

3	2	1	0		
0	0	0	0	16x External Clock	
0	0	0	1	50	Baud
0	0	1	0	75	Baud
0	0	1	1	109.92	Baud
0	1	0	0	134.58	Baud
0	1	0	1	150	Baud
0	1	1	0	300	Baud
0	1	1	1	600	Baud
1	0	0	0	1200	Baud
1	0	0	1	1800	Baud
1	0	1	0	2400	Baud
1	0	1	1	3600	Baud
1	1	0	0	4800	Baud
1	1	0	1	7200	Baud
1	1	1	0	9600	Baud
1	1	1	1	19,200	Baud

RECEIVER CLOCK SOURCE (RCS)

0 = External Receiver Clock
1 = Baud Rate

WORD LENGTH (WL)

6	5	
0	0	8 Bits
0	1	7 Bits
1	0	6 Bits
1	1	5 Bits

STOP BIT NUMBER (SBN)

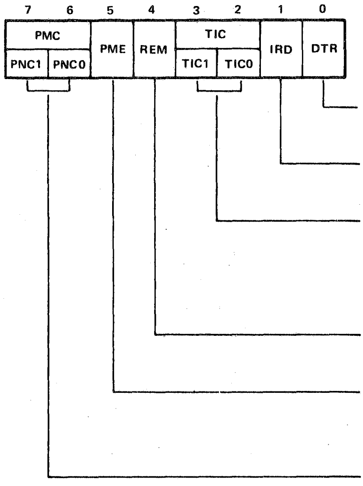
0 = 1 Stop Bit
1 = 2 Stop Bits
= 1½ Stop Bits
(For WL = 5 and No Parity)
= 1 Stop Bit
(For WL = 8 and Parity)

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Hardware Reset (RES)
–	–	–	–	–	–	–	–	Program Reset

R6551 Control Register

Command Register

The Command Register controls specific modes and functions.



DATA TERMINAL READY (DTR)

0 = Data Terminal Not Ready (DTR High)
1 = Data Terminal Ready (DTR Low)

INTERRUPT REQUEST DISABLED (IRD)

0 = IRQ Enabled
1 = IRQ Disabled

TRANSMITTER INTERRUPT CONTROL (TIC)

3	2	
0	0	RTS = High, Transmit Interrupt Disabled
0	1	RTS = Low, Transmit Interrupt Enabled
1	0	RTS = Low, Transmit Interrupt Enabled
1	1	RTS = Low, Transmit Interrupt Disabled Transmit Break on TxD

RECEIVER ECHO MODE (REM)

0 = Receiver Normal Mode
1 = Receiver Echo Mode

PARITY MODE ENABLED (PME)

0 = Parity Mode Disabled
No Parity Bit Generated
Parity Check Disabled
1 = Parity Mode Enabled

PARITY MODE CONTROL (PMC)

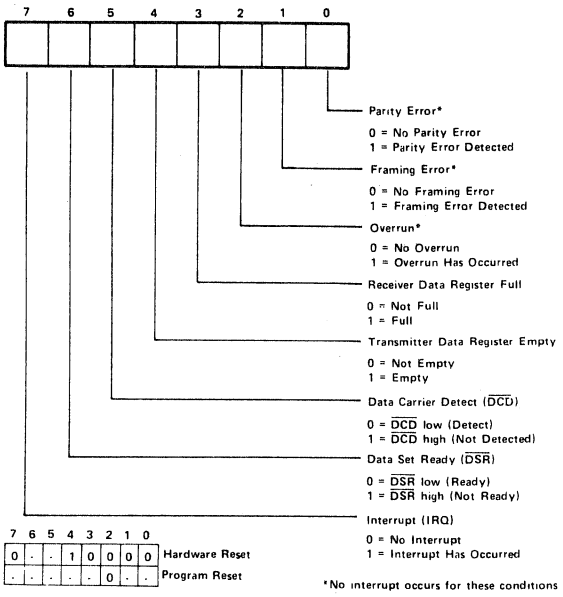
7	6	
0	0	Odd Parity Transmitted/Received
0	1	Even Parity Transmitted/Received
1	0	Mark Parity Bit Transmitted Parity Check Disabled
1	1	Space Parity Bit Transmitted Parity Check Disabled

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	Hardware Reset (RES)
–	–	–	0	0	0	0	0	Program Reset

R6551 Command Register

Status Register

The Status Register reports the status of various R6551 functions


7 6 5 4 3 2 1 0
Parity Error*
0 = No Parity Error
1 = Parity Error Detected
Framing Error*
0 = No Framing Error
1 = Framing Error Detected
Overrun*
0 = No Overrun
1 = Overrun Has Occurred
Receiver Data Register Full
0 = Not Full
1 = Full
Transmitter Data Register Empty
0 = Not Empty
1 = Empty
Data Carrier Detect ($\overline{DCD}$)
0 = $\overline{DCD}$ low (Detect)
1 = $\overline{DCD}$ high (Not Detected)
Data Set Ready ($\overline{DSR}$)
0 = $\overline{DSR}$ low (Ready)
1 = $\overline{DSR}$ high (Not Ready)
Interrupt (IRQ)
0 = No Interrupt
1 = Interrupt Has Occurred
7 6 5 4 3 2 1 0
0 · · 1 0 0 0 0 Hardware Reset
· · · · · 0 · · Program Reset
*No interrupt occurs for these conditions

R6551 Status Register

INTERFACE SIGNAL DESCRIPTION

$\overline{RES}$ (Reset)

During system initialization a low on the $\overline{RES}$ input will cause internal registers to be cleared.

Ø2 (Input Clock)

The input clock is the system Ø2 clock and is used to synchronize all data transfers between the system microprocessor and the R6551.

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

The R/$\overline{W}$ is generated by the microprocessor and is used to control the direction of data transfers. A high on the R/$\overline{W}$ pin allows the processor to read the data supplied by the R6551. A low on the R/$\overline{W}$ pin allows a write to the R6551.

$\overline{IRQ}$ (Interrupt Request)

The $\overline{IRQ}$ pin is an interrupt output from the interrupt control logic. It is an open drain output, permitting several devices to be connected to the common $\overline{IRQ}$ microprocessor input. Normally a high level, $\overline{IRQ}$ goes low when an interrupt occurs.

D0-D7 (Data Bus)

The D0-D7 pins are the eight data lines used to transfer data between the processor and the R6551. These lines are bi-directional and are normally high-impedance, except during Read cycles when the R6551 is selected.

CS0, $\overline{CS1}$ (Chip Selects)

The two chip select inputs are normally connected to the processor address lines either directly or through decoders. The R6551 is selected when CS0 is high and $\overline{CS1}$ is low.

RS0, RS1 (Register Selects)

The two register select lines are normally connected to the processor address lines to allow the processor to select the various R6551 internal registers. The following table indicates the internal register select coding:

RS1	RS0	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	

Note that only the Command and Control registers are read/write. The Programmed Reset operation does not cause any data transfer, but is used to clear Bits 0 through 4 in the Command Register and Bit 2 in the Status Register. The Programmed Reset is slightly different from the Hardware Reset ($\overline{RES}$); these differences are described in the individual register definitions.

ACIA/Modem Interface Signal Description

XTLI, XTLO (Crystal Pins)

These pins are normally directly connected to the external crystal (1.8432 MHz) used to derive the various baud rates. Alternatively, an externally generated clock may be used to drive the XTLI pin, in which case the XTLO pin must float. XTLI is the input pin for the transmit clock.

TxD (Transmit Data)

The TxD output line is used to transfer serial NRZ (non-return-to-zero) data to the modem. The LSB (least significant bit) of the Transmit Data Register is the first data bit transmitted and the rate of data transmission is determined by the baud rate selected, or under control of an external clock (as selected by the Control Register).

RxD (Receive Data)

The RxD input line is used to transfer serial NRZ data into the ACIA from the modem, LSB first. The receiver data rate is either the programmed baud rate or the rate of an externally generated receiver clock (as selected by the Control Register).

RxC (Receive Clock)

The RxC is a bi-directional pin which serves as either the receiver 16x clock input or the receiver 16x clock output. The latter mode results if the internal baud rate generator is selected for receiver data clocking.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output pin is used to control the modem from the processor. The state of the $\overline{RTS}$ pin is determined by the contents of the Command Register.

$\overline{CTS}$ (Clear to Send)

The $\overline{CTS}$ input pin is used to control the transmitter operation. The enable state is with $\overline{CTS}$ low. The transmitter is automatically disabled if $\overline{CTS}$ is high.

$\overline{DTR}$ (Data Terminal Ready)

This output pin is used to indicate the status of the R6551 to the modem. A low on $\overline{DTR}$ indicates the R6551 is enabled and a high indicates it is disabled. The processor controls this pin via bit 0 of the Command Register.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input pin is used to indicate to the R6551 the status of the modem. A low indicates the "ready" state and a high, "not-ready". $\overline{DSR}$ is a high-impedance input, and must be connected. If unused, it should be driven high or low, but not switched.

$\overline{DCD}$ (Data Carrier Detect)

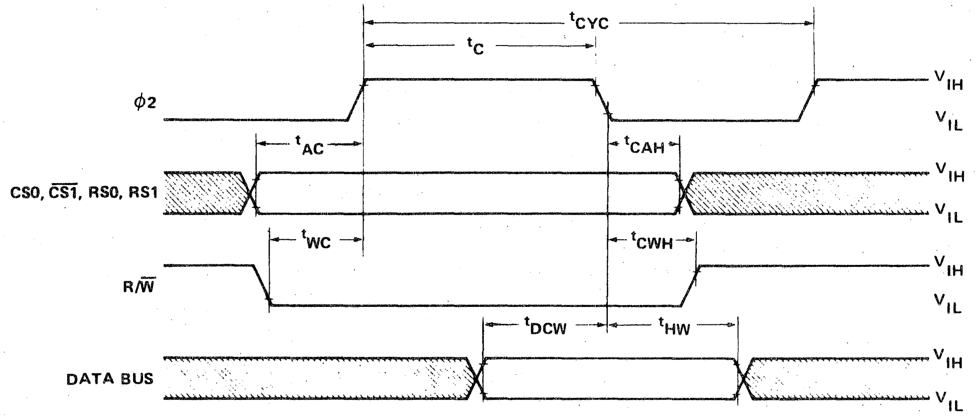
The $\overline{DCD}$ input pin is used to indicate to the R6551 the status of the carrier-detect output of the modem. A low indicates that the modem carrier signal is present and a high, that it is not. Like $\overline{DSR}$, $\overline{DCD}$ is a high-impedance input, and must be connected.

READ/WRITE CYCLE CHARACTERISTICS

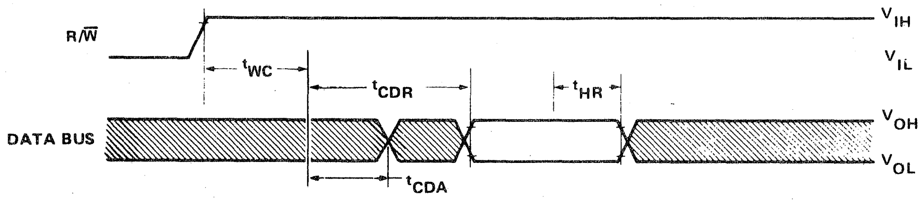
(V_{CC} = 5.0V ±5%, T_A = 0 to 70°C, unless otherwise noted)

Characteristic	Symbol	1 MHz		2 MHz		Unit
		Min	Max	Min	Max	
Cycle Time	t_{CYC}	1.0	40	0.5	40	µs
Ø2 Pulse Width	t_C	400	–	200	–	ns
Address Set-Up Time	t_{AC}	120	–	70	–	ns
Address Hold Time	t_{CAH}	0	–	0	–	ns
R/$\overline{W}$ Set-Up Time	t_{WC}	120	–	70	–	ns
R/$\overline{W}$ Hold Time	t_{CWH}	0	–	0	–	ns
Data Bus Set-Up Time	t_{DCW}	150	–	60	–	ns
Data Bus Hold Time	t_{HW}	20	–	20	–	ns
Read Access Time (Valid Data)	t_{CDR}	–	200	–	150	ns
Read Hold Time	t_{HR}	20	–	20	–	ns
Bus Active Time (Invalid Data)	t_{CDA}	40	–	40	–	ns

(t_r and t_f = 10 to 30 ns)



Write Timing Characteristics



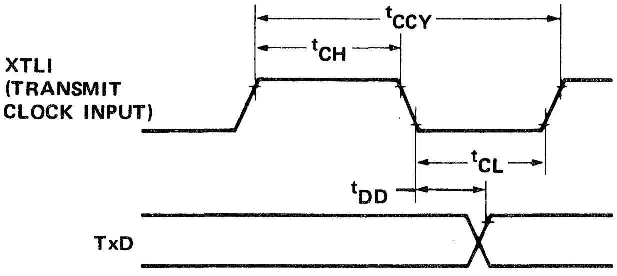
Read Timing Characteristics

TRANSMIT/RECEIVE CHARACTERISTICS

Characteristic	Symbol	1 MHz		2 MHz		Unit
		Min	Max	Min	Max	
Transmit/Receive Clock Rate	t_{CCY}	400*	–	400*	–	ns
Transmit/Receive Clock High Time	t_{CH}	175	–	175	–	ns
Transmit/Receive Clock Low Time	t_{CL}	175	–	175	–	ns
XTLI to TxD Propagation Delay	t_{DD}	–	500	–	500	ns
$\overline{\text{RTS}}$ Propagation Delay	t_{DLY}	–	500	–	500	ns
$\overline{\text{IRQ}}$ Propagation Delay (Clear)	t_{IRQ}	–	500	–	500	ns

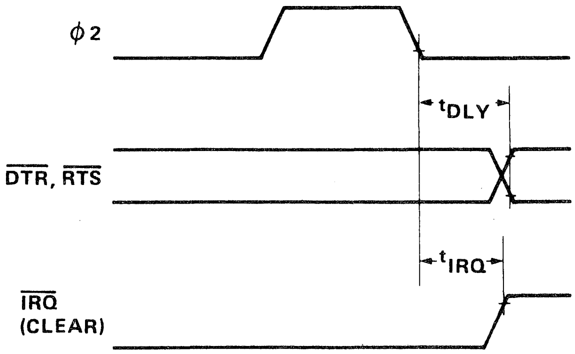
(t_r, t_f = 10 to 30 ns)

*The baud rate with external clocking is: Baud Rate = $\frac{1}{16 \times T_{CCY}}$



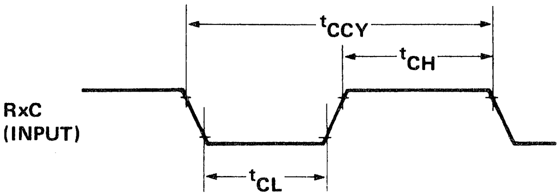
NOTE: TxD rate is 1/16 TxC rate

Transmit Timing with External Clock



$\phi 2$
t_{DLY}
$\overline{\text{DTR}}$, $\overline{\text{RTS}}$
t_{IRQ}
$\overline{\text{IRQ}}$ (CLEAR)

Interrupt and Output Timing

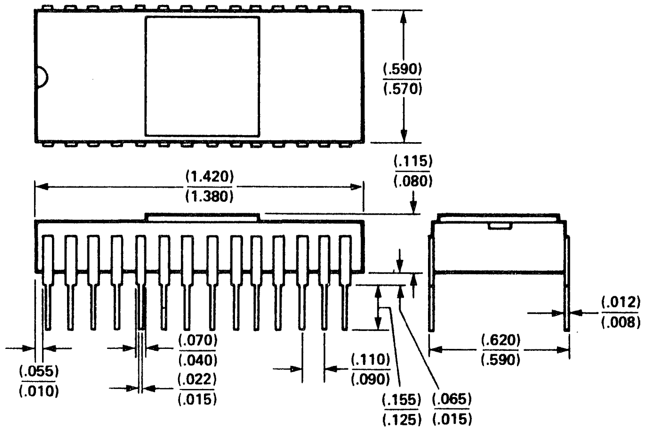


NOTE: RxD rate is 1/16 RxC rate

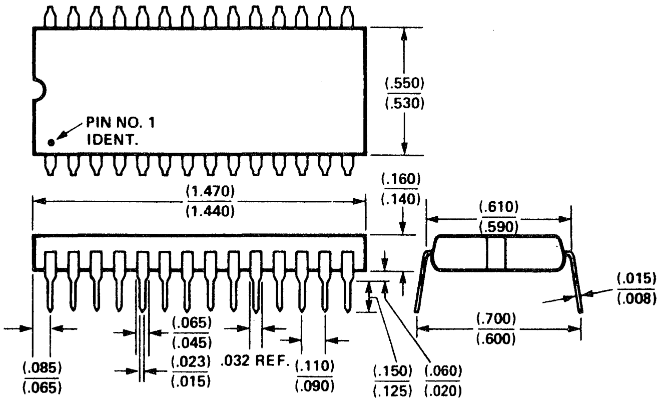
Receive External Clock Timing

PACKAGE OUTLINES

28 LEAD CERAMIC



28 LEAD PLASTIC



PART NUMBER
R6545-1

Rockwell

R6500 Microcomputer System
DATA SHEET

CRT CONTROLLER (CRTC)

DESCRIPTION

The R6545-1 CRT Controller (CRTC) is designed to interface an 8-bit microprocessor to CRT raster scan video displays, and adds an advanced CRT controller to the established and expanding line of R6500 products.

The R6545-1 provides refresh memory addresses and character generator row addresses which allow up to 16K characters with 32 scan lines per character to be addressed. A major advantage of the R6545-1 is that the refresh memory may be addressed in either straight binary or by row/column.

Other functions in the R6545-1 include an internal cursor register which generates a cursor output when its contents are equal to the current refresh address. Programmable cursor start and end registers allow a cursor of up to the full character scan in height to be placed on any scan lines of the character. Variable cursor display blink rates are provided. A light pen strobe input allows capture of the current refresh address in an internal light pen register. The refresh address lines are configured to provide direct dynamic memory refresh.

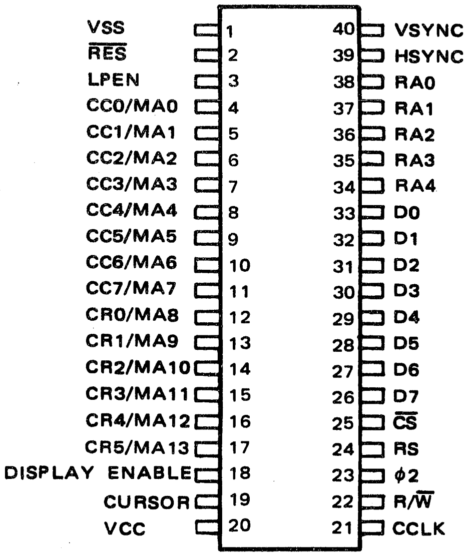
All timing for the video refresh memory signals is derived from the character clock input. Shift register, latch, and multiplex control signals (when needed) are provided by external high-speed timing. The mode control register allows non-interlaced video display modes at 50 or 60 Hz refresh rate. The internal status register may be used to monitor the R6545-1 operation. The $\overline{\text{RES}}$ input allows the CRTC-generated field rate to be dynamically-synchronized with line frequency jitter.

ORDERING INFORMATION

Part Number	Package Type	Frequency	Temperature Range
R6545-1P	Plastic	1 MHz	0°C to +70°C
R6545-1AP	Plastic	2 MHz	0°C to +70°C
R6545-1C	Ceramic	1 MHz	0°C to +70°C
R6545-1AC	Ceramic	2 MHz	0°C to +70°C

FEATURES

- Compatible with 8-bit microprocessors
- Up to 2.5 MHz character clock operation
- Refresh RAM may be configured in row/column or straight binary addressing
- Alphanumeric and limited graphics capability
- Up and down scrolling by page, line, or character
- Programmable Vertical Sync Width
- Fully programmable display (rows, columns, character matrix)
- Non-interlaced scan
- 50/60 Hz operation
- Fully programmable cursor
- Light pen register
- Addresses refresh RAM to 16K characters
- No external DMA required
- Internal status register
- 40-Pin ceramic or plastic DIP
- Pin-compatible with MC6845
- Single +5 ±5% Volt Power Supply



R6545-1 Pin Configuration

CRT CONTROLLER (CRTC)

Specifications subject to change without notice
Document No. 29000 D67
December 1980

INTERFACE SIGNAL DESCRIPTION

CPU INTERFACE

ϕ2 (Phase 2 Clock)

The input clock is the system Phase 2 (ϕ2) clock and is used to trigger all data transfers between the system processor (CPU) and the R6545-1. Since there is no maximum limit to the allowable ϕ2 clock time, it is not necessary for it to be a continuous clock. This capability permits the R6545-1 to be easily interfaced to non-6500 compatible microprocessors.

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

The R/$\overline{W}$ input signal generated by the processor is used to control the direction of data transfers. A high on the R/$\overline{W}$ pin allows the processor to read the data supplied by the R6545-1, a low on the R/$\overline{W}$ pin allows data on data lines D0-D7 to be written into the R6545-1.

$\overline{CS}$ (Chip Select)

The Chip Select input is normally connected to the processor address bus either directly or through a decoder. The R6545-1 is selected when $\overline{CS}$ is low.

RS (Register Select)

The Register Select input is used to access internal registers. A low on this pin permits writes (R/$\overline{W}$ = low) into the Address Register and reads (R/$\overline{W}$ = high) from the Status Register. The contents of the Address Register is the identity of the register accessed when RS is high.

D0-D7 (Data Bus)

D0-D7 are the eight data lines used to transfer data between the processor and the R6545-1. These lines are bidirectional and are normally high-impedance except during read cycles when the chip is selected ($\overline{CS}$ = low).

VIDEO INTERFACE

HSYNC (Horizontal Sync)

The HSYNC signal is an active-high output used to determine the horizontal position of displayed text. It may drive a CRT monitor directly or may be used for composite video generation. HSYNC time position and width are fully programmable.

VSYNC (Vertical Sync)

The VSYNC signal is an active high output used to determine the vertical position of displayed text. Like HSYNC, VSYNC may be used to drive a CRT monitor or composite video generation circuits. VSYNC time position and width are both programmable.

DISPLAY ENABLE (Display Enable)

The DISPLAY ENABLE signal is an active-high output used to indicate when the R6545-1 is generating active display information. The number of horizontal display characters per row and the number of vertical display rows are both fully programmable and together are used to generate the DISPLAY ENABLE signal. DISPLAY ENABLE can be delayed one character time by setting bit 4 of R8 equal to 1.

CURSOR (Cursor Coincidence)

The CURSOR signal is an active-high output used to indicate when the scan coincides with the programmed cursor position The cursor position may be programmed to be any character in the address field. Furthermore, within the character, the cursor may be programmed to be any block of scan lines, since the start scan line and the end scan line are both programmable. The cursor position may be delayed by one character time by setting Bit 5 of R8 to A "1".

LPEN (Light Pen Strobe)

The LPEN signal is an edge-sensitive input used to load the internal Light Pen Register with the contents of the Refresh Scan Counter at the time the active edge occurs. The active edge of LPEN is the low-to-high transition.

CCLK (Clock)

The CCLK signal is the character timing clock input and is used as the time base for all internal count/control functions.

$\overline{RES}$

The $\overline{RES}$ signal is an active-low input used to initialize all internal scan counter circuits. When $\overline{RES}$ is low, all internal counters are stopped and cleared, all scan and video outputs are low, and control registers are unaffected. $\overline{RES}$ must stay low for at least one CCLK period. All scan timing is initiated when $\overline{RES}$ goes high. In this way, $\overline{RES}$ can be used to synchronize display frame timing with line frequency. $\overline{RES}$ may also be used to synchronize multiple CRTC's in horizontal and/or vertical split screen operation.

REFRESH RAM AND CHARACTER ROM INTERFACE

MA0-MA13 (Refresh RAM Address Lines)

These 14 signals are active-high outputs used to address the Refresh RAM for character storage and display operations. The starting scan address is fully programmable and the ending scan address is determined by the total number of characters displayed, which is also programmable, in terms of characters/line and lines/frame.

There are two selectable address modes for MA0-MA13:

In the straight binary mode (R8, Mode Control, bit 2 = "0"), characters are stored in successive memory locations. Thus, the software must be designed such that row and column character coordinates are translated into sequentially-numbered addresses. In the row/column mode (R8, Mode Control, bit 2 = "1"), MA0-MA7 become column addresses CC0-CC7 and MA8-MA13 become row addresses CR0-CR5. In this case, the software can manipulate characters in terms of row and column locations, but additional address compression circuits are needed to convert the CC0-CC7 and CR0-CR5 addresses into a memory-efficient binary address scheme.

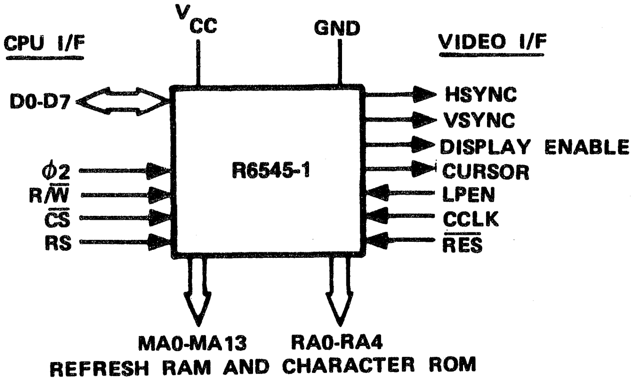
RA0-RA4 (Raster Address Lines)

These 5 signals are active-high outputs used to select each raster scan within an individual character row. The number of raster scan lines is programmable and determines the character height, including spaces between character rows.

INTERNAL REGISTER ORGANIZATION

		Address Register					Reg.			Read	Write	Register Bit							
$\overline{CS}$	RS	4	3	2	1	0	No.	Register Name	Register Units	(R/$\overline{W}$ = High)	(R/$\overline{W}$ = Low)	7	6	5	4	3	2	1	0
1	X	X	X	X	X	X	X					/	/	/	/	/	/	/	
0	0	X	X	X	X	X	X	Address Register	Register No.		✓	/	/	/	4	3	2	1	0
0	0	X	X	X	X	X	X	Status Register	–	✓		/	6	5	/	/	/	/	
0	1	0	0	0	0	0	R0	Horizontal Total Char	No. of Characters/Row		✓	7	6	5	4	3	2	1	0
0	1	0	0	0	0	1	R1	Horizontal Displayed Char	No. of Characters/Row		✓	7	6	5	4	3	2	1	0
0	1	0	0	0	1	0	R2	Horizontal Sync Position	Character Position		✓	7	6	5	4	3	2	1	0
0	1	0	0	0	1	1	R3	YSYNC, HSYNC Widths	No. of Scan Lines, Characters		✓	7	6	5	4	3	2	1	0
0	1	0	0	1	0	0	R4	Vertical Total Rows	No. of Character Rows		✓	/	6	5	4	3	2	1	0
0	1	0	0	1	0	1	R5	Vertical Total Adjust Lines	No. of Scan Lines		✓	/	/	/	4	3	2	1	0
0	1	0	0	1	1	0	R6	Vertical Displayed Rows	No. of Character Rows		✓	/	6	5	4	3	2	1	0
0	1	0	0	1	1	1	R7	Vertical Sync Position	No. of Character Rows		✓	/	6	5	4	3	2	1	0
0	1	0	1	0	0	0	R8	Mode Control	–		✓	7	6	5	4	3	2	1	0
0	1	0	1	0	0	1	R9	Scan Line	No. of Scan Lines		✓	/	/	/	4	3	2	1	0
0	1	0	1	0	1	0	R10	Cursor Start Line	Scan Line No.		✓	/	6	5	4	3	2	1	0
0	1	0	1	0	1	1	R11	Cursor End Line	Scan Line No.		✓	/	/	/	4	3	2	1	0
0	1	0	1	1	0	0	R12	Display Start Address (H)	–		✓	/	/	5	4	3	2	1	0
0	1	0	1	1	0	1	R13	Display Start Address (L)	–		✓	7	6	5	4	3	2	1	0
0	1	0	1	1	1	0	R14	Cursor Position Address (H)	–	✓	✓	/	/	5	4	3	2	1	0
0	1	0	1	1	1	1	R15	Cursor Position Address (L)	–	✓	✓	7	6	5	4	3	2	1	0
0	1	1	0	0	0	0	R16	Light Pen Register (H)	–	✓		/	/	5	4	3	2	1	0
0	1	1	0	0	0	1	R17	Light Pen Register (L)	–	✓		7	6	5	4	3	2	1	0

Table 1. Overall Register Structure and Addressing



R6545-1 Interface Diagram

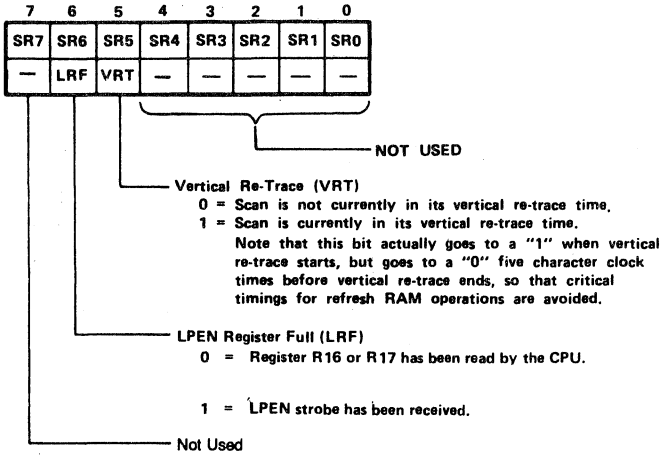
INTERNAL REGISTER DESCRIPTION

ADDRESS REGISTER

This 5-bit write-only register is used as a "pointer" to direct CRTC/CPU data transfers within the CRTC. Its contents is the number of the desired register (0-17). When $\overline{CS}$ and RS are low, then this register may be loaded; when $\overline{CS}$ is low and RS is high, then the register selected is the one whose identity is stored in this address register.

STATUS REGISTER (SR)

This 8-bit register contains the status of the CRTC. Only two bits are assigned, as follows:



7	6	5	4	3	2	1	0
SR7	SR6	SR5	SR4	SR3	SR2	SR1	SR0
—	LRF	VRT	—	—	—	—	—

NOT USED

Vertical Re-Trace (VRT)
0 = Scan is not currently in its vertical re-trace time.
1 = Scan is currently in its vertical re-trace time.
Note that this bit actually goes to a "1" when vertical re-trace starts, but goes to a "0" five character clock times before vertical re-trace ends, so that critical timings for refresh RAM operations are avoided.

LPEN Register Full (LRF)
0 = Register R16 or R17 has been read by the CPU.
1 = LPEN strobe has been received.

Not Used

NOTE: The Status Register takes the State, | – | 0 | 1 | – | – | – | – | – | immediately after power (V$_{CC}$) turn-on.

R0—HORIZONTAL TOTAL CHARACTERS

This 8-bit write-only register contains the total of displayed and non-displayed characters, minus one, per horizontal line. The frequency of HSYNC is thus determined by this register.

R1—HORIZONTAL DISPLAYED CHARACTERS

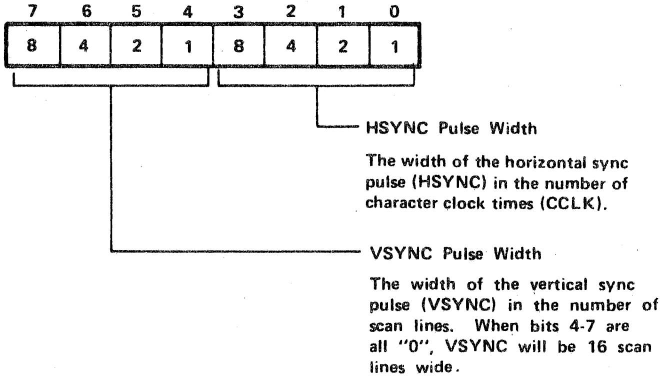
This 8-bit write-only register contains the number of displayed characters per horizontal line.

R2—HORIZONTAL SYNC POSITION

This 8-bit write-only register contains the position of the horizontal SYNC on the horizontal line, in terms of the character location number on the line. The position of the HSYNC determines the left to right location of the displayed text on the video screen. In this way, the side margins are adjusted.

R3—HORIZONTAL AND VERTICAL SYNC WIDTHS

This 8-bit write-only register contains the widths of both HSYNC and VSYNC, as follows:


7 6 5 4 3 2 1 0
8 4 2 1 8 4 2 1
HSYNC Pulse Width
The width of the horizontal sync pulse (HSYNC) in the number of character clock times (CCLK).
VSYNC Pulse Width
The width of the vertical sync pulse (VSYNC) in the number of scan lines. When bits 4-7 are all "0", VSYNC will be 16 scan lines wide.

Control of these parameters allows the R6545-1 to be interfaced to a variety of CRT monitors, since the HSYNC and VSYNC timing signals may be accommodated without the use of external one shot timing.

R4—VERTICAL TOTAL ROWS

The Vertical Total Register is a 7-bit register containing the total number of character rows in a frame, minus one. This register, along with R5, determines the overall frame rate, which should be close to the line frequency to ensure flicker-free appearance. If the frame time is adjusted to be longer than the period of the line frequency, then $\overline{\text{RES}}$ may be used to provide absolute synchronism.

R5—VERTICAL TOTAL LINE ADJUST

The Vertical Total Line Adjust Register (R5) is a 5-bit write-only register containing the number of additional scan lines needed to complete an entire frame scan and is intended as a fine adjustment for the video frame time.

R6—VERTICAL DISPLAYED ROWS

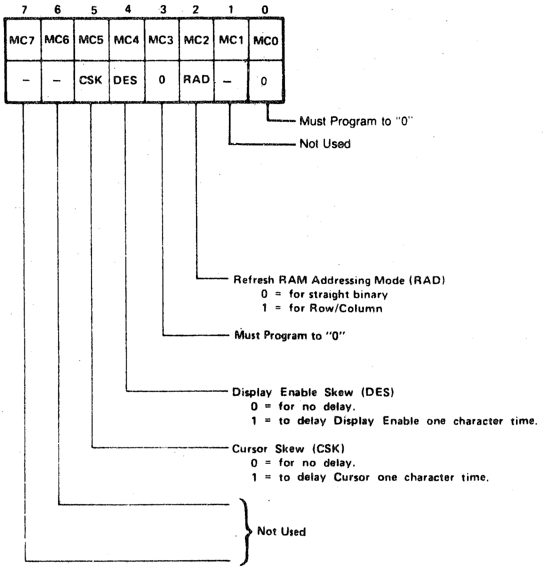
This 7-bit write-only register contains the number of displayed character rows in each frame.

R7—VERTICAL SYNC POSITION

This 7-bit write-only register is used to select the character r time at which the vertical SYNC pulse is desired to occur a thus, is used to position the displayed text in the vertical direction.

R8—MODE CONTROL (MC)

This 8-bit write-only register selects the operating modes of the R6545-1, as follows:



R9—ROW SCAN LINES

This 5-bit write-only register contains the number of scan lines, minus one, per character row, including spacing.

R10—CURSOR START LINE
R11—CURSOR END LINE

These 5-bit write-only registers select the starting and ending scan lines for the cursor. In addition, bits 5 and 6 of R10 are used to select the cursor blink mode, as follows:

Bit 6	Bit 5	Cursor Blink Mode
0	0	Display Cursor Continuously
0	1	Blank Cursor Continuously
1	0	Blink Cursor at 1/16 Field Rate
1	1	Blink Cursor at 1/32 Field Rate

R12—DISPLAY START ADDRESS HIGH
R13—DISPLAY START ADDRESS LOW

These registers form a 14-bit register whose contents is the memory address of the first character of the displayed scan (the character on the top left of the video display, as in Figure 1). Subsequent memory addresses are generated by the R6545-1 as a result of CCLK input pulses. Scrolling of the display is accomplished by changing R12 and R13 to the memory addre associated with the first character of the desired line of text be displayed first. Entire pages of text may be scrolled or changed as well via R12 and R13.

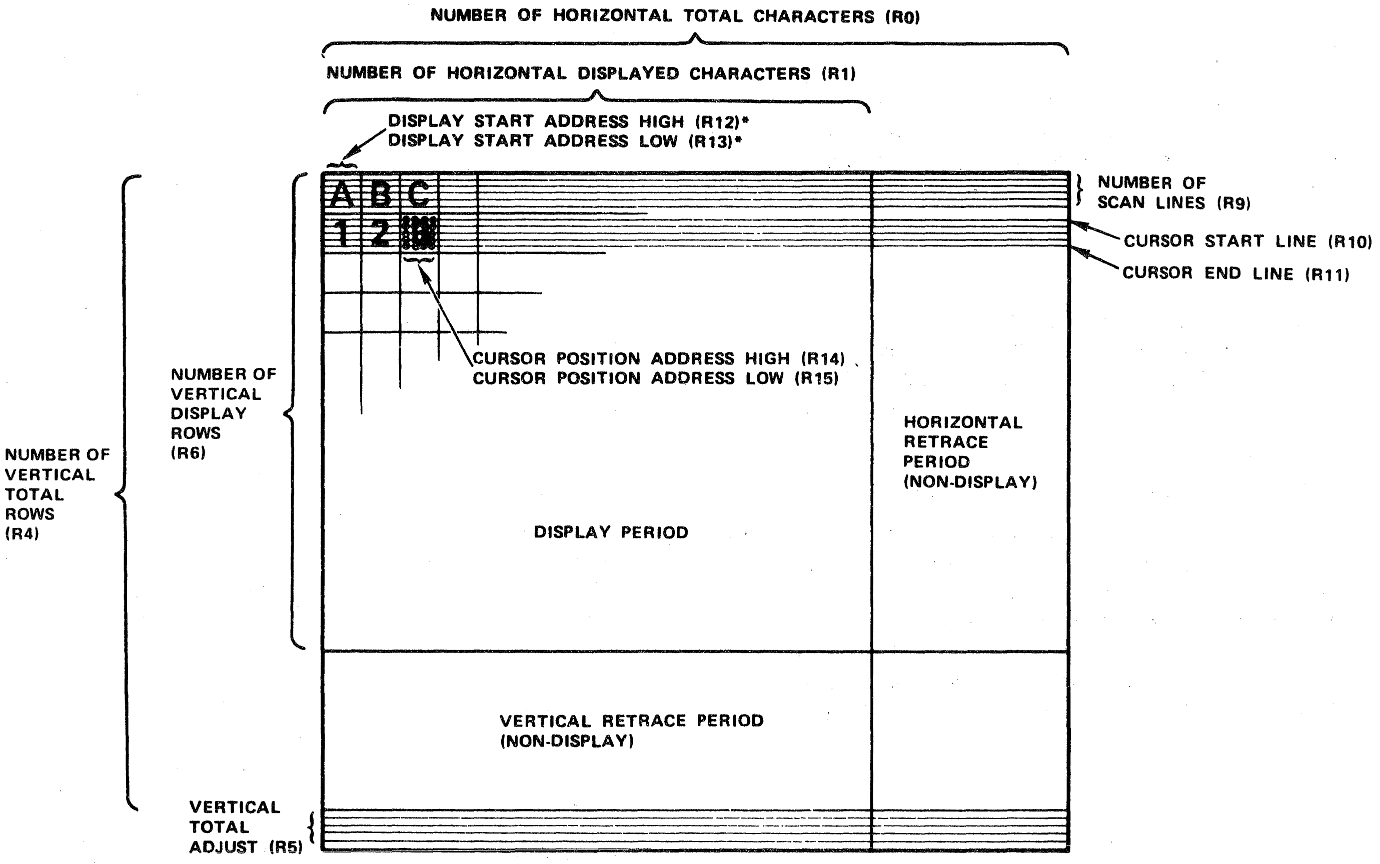

NUMBER OF HORIZONTAL TOTAL CHARACTERS (R0)
NUMBER OF HORIZONTAL DISPLAYED CHARACTERS (R1)
DISPLAY START ADDRESS HIGH (R12)*
DISPLAY START ADDRESS LOW (R13)*
NUMBER OF SCAN LINES (R9)
CURSOR START LINE (R10)
CURSOR END LINE (R11)
A
B
C
1
2
CURSOR POSITION ADDRESS HIGH (R14)
CURSOR POSITION ADDRESS LOW (R15)
NUMBER OF VERTICAL DISPLAY ROWS (R6)
NUMBER OF VERTICAL TOTAL ROWS (R4)
HORIZONTAL RETRACE PERIOD (NON-DISPLAY)
DISPLAY PERIOD
VERTICAL RETRACE PERIOD (NON-DISPLAY)
VERTICAL TOTAL ADJUST (R5)

Figure 1. Video Display Format

R14—CURSOR POSITION HIGH
R15—CURSOR POSITION LOW

These registers form a 14-bit register whose contents is the memory address of the current cursor position. When the video display scan counter (MA lines) matches the contents of this register, and when the scan line counter (RA lines) falls within the bounds set by R10 and R11, then the CURSOR output becomes active. Bit 5 of the Mode Control Register (R8) may be used to delay the CURSOR output by a full CCLK time to accommodate slow access memories.

R16—LIGHT PEN HIGH
R17—LIGHT PEN LOW

These registers form a 14-bit register whose contents is the light pen strobe position, in terms of the video display address at which the strobe occurred. When the LPEN input changes from low to high, then, on the next negative-going edge of CCLK, the contents of the internal scan counter is stored in registers R16 and R17.

REGISTER FORMATS

Register pairs R12/R13, R14/R15, and R16/R17 are formatted in one of two ways:

(1) Straight binary, if register R8, bit 2 = "0".

(2) Row/Column, if register R8, bit 2 = "1". In this case the low byte is the Character Column and the high byte is the Character Row.

DESCRIPTION OF OPERATION

VIDEO DISPLAY

Figure 1 indicates the relationship of the various program registers in the R6545-1 and the resultant video display.

Non-displayed areas of the Video Display are used for horizontal and vertical retrace functions of the CRT monitor. The horizontal and vertical sync signals, HSYNC and VSYNC, are programmed to occur during these intervals and are used to trigger the retrace in the CRT monitor. The pulse widths are constrained by the monitor requirements. The time position of the pulses may be adjusted to vary the display margins (left, right, top, and bottom).

REFRESH RAM ADDRESSING

Shared Memory Mode (R8, bit 3 = "0")

In this mode, the Refresh RAM address lines (MA0-MA13) directly reflect the contents of the internal refresh scan character counter. Multiplex control, to permit addressing and selection of the RAM by both the CPU and the CRTC, must be provided external to the CRTC. In the Row/Column address mode, lines MA0-MA7 become character column addresses (CC0-CC7) and MA8-MA13 become character row addresses (CR0-CR5).

ADDRESSING MODES

Row/Column

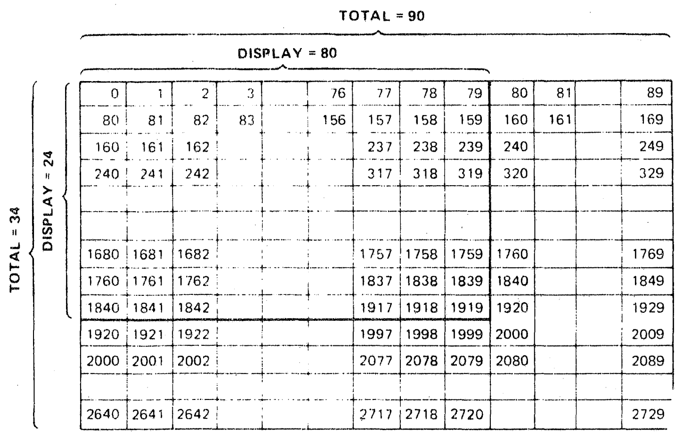
In this mode, the CRTC address lines (MA0-MA13) are generated as 8 column (MA0-MA7) and 6 row (MA8-MA13) addresses. Extra hardware is needed to compress this addressing into a straight binary sequence in order to conserve memory in the refresh RAM.

Binary

In this mode, the CRTC address lines are straight binary and no compression circuits are needed. However, software complexity is increased since the CRT characters cannot be stored in terms of their row and column locations, but must be sequential.

USE OF DYNAMIC RAM FOR REFRESH MEMORY

The R6545-1 permits the use of dynamic RAMS as storage devices for the Refresh RAM by continuing to increment memory addresses in the non-display intervals of the scan. This is a viable technique, since the Display Enable signal controls the actual video display blanking. Figure 2 illustrates Refresh RAM addressing for the case of binary addressing for 80 columns and 24 rows with 10 non-displayed columns and 10 non-displayed rows.


TOTAL = 90
DISPLAY = 80
TOTAL = 34
DISPLAY = 24

0	1	2	3		76	77	78	79	80	81		89
80	81	82	83		156	157	158	159	160	161		169
160	161	162				237	238	239	240			249
240	241	242				317	318	319	320			329
1680	1681	1682				1757	1758	1759	1760			1769
1760	1761	1762				1837	1838	1839	1840			1849
1840	1841	1842				1917	1918	1919	1920			1929
1920	1921	1922				1997	1998	1999	2000			2009
2000	2001	2002				2077	2078	2079	2080			2089
2640	2641	2642				2717	2718	2720				2729

Figure 2. Memory Addressing Example (80 x 24)

CURSOR OPERATION

A one character wide cursor can be controlled by storing values into the Cursor Start Line (R10) and Cursor End Line (R11) registers and into the Cursor Position Address High (R14) and Cursor Position Low (R15) registers.

Bits 5 and 6 in the Cursor Start Line High Register (R10) control the cursor display and blink rate as follows:

Bit 6	Bit 5	Cursor Operating Mode
0	0	Display Cursor Continuously
0	1	Blank Cursor Continuously
1	0	Blink Cursor at 1/16 Field Rate
1	1	Blink Cursor at 1/32 Field Rate

The cursor of up to 32 characters in height can be displayed on and between the scan lines as loaded into the Cursor Start Line (R10) and Cursor End Line (R11) Registers.

The cursor is positioned on the screen by loading the Cursor Position Address High (R14) and Cursor Position Address Low (R15) registers with the desired refresh RAM address. The cursor can be positioned in any of the 16K character positions. Hardware paging and data scrolling is thus allowed without loss of cursor position. Figure 3 is an example of the display cursor scan line.

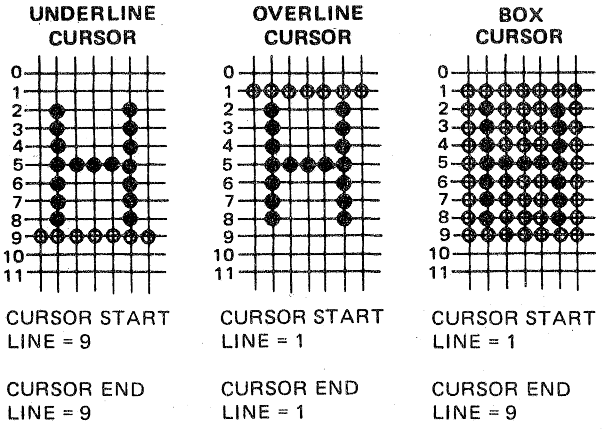


Figure 3. Cursor Display Scan Line Control Examples

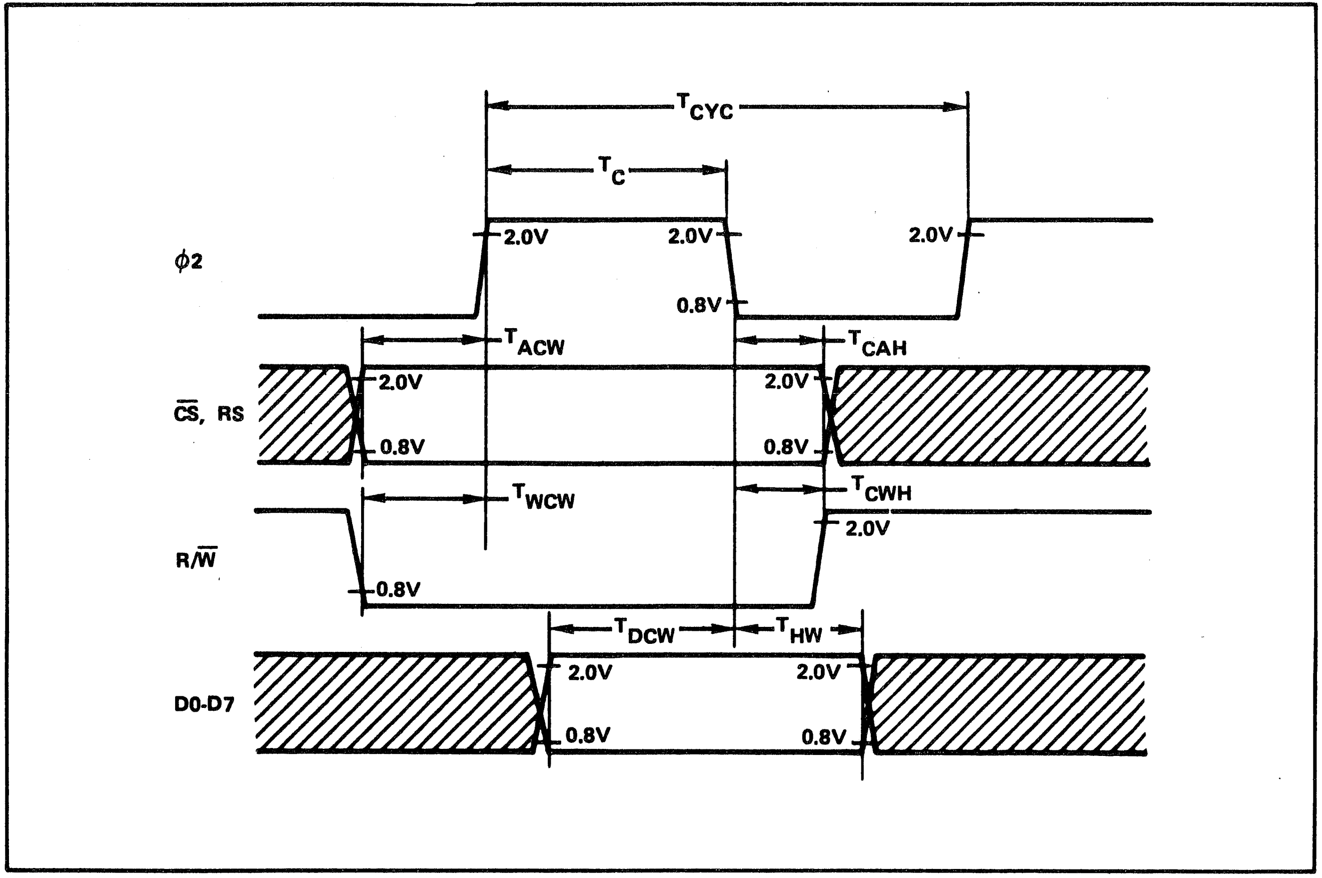
MPU WRITE TIMING CHARACTERISTICS

(V_{CC} = 5.0V ±5%, T_A = 0 to 70°C, unless otherwise noted)

Characteristic	Symbol	1 MHz		2 MHz		Unit
		Min	Max	Min	Max	
Cycle Time	T_{CYC}	1.0	–	0.5	–	μs
Ø2 Pulse Width	T_C	440	–	200	–	ns
Address Set-Up Time	T_{ACW}	180	–	90	–	ns
Address Hold Time	T_{CAH}	0	–	0	–	ns
R/$\overline{W}$ Set-Up Time	T_{WCW}	180	–	90	–	ns
R/$\overline{W}$ Hold Time	T_{CWH}	0	–	0	–	ns
Data Bus Set-Up Time	T_{DCW}	265	–	100	–	ns
Data Bus Hold Time	T_{HW}	10	–	10	–	ns

(t_r and t_f = 10 to 30 ns)

WRITE CYCLE



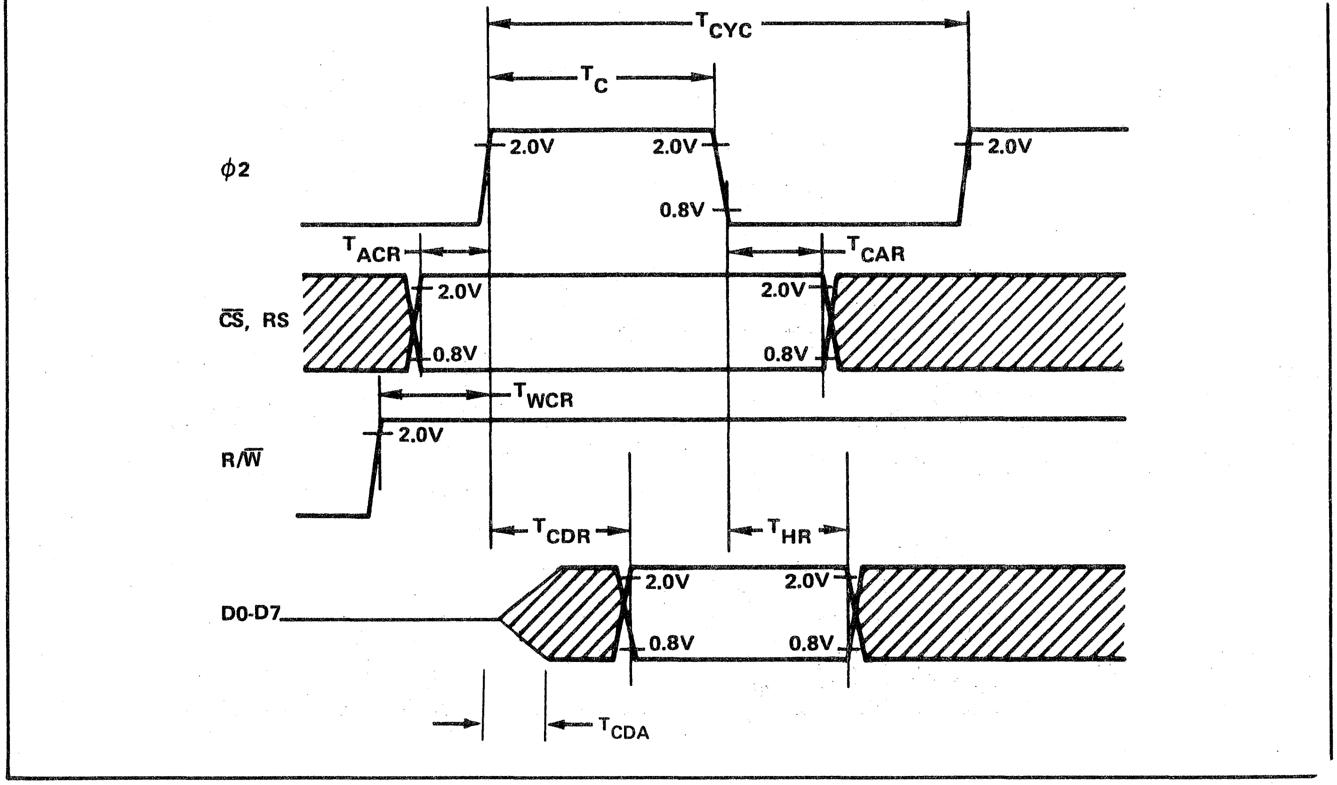
MPU READ TIMING CHARACTERISTICS

(V_{CC} = 5.0V ±5%, T_A = 0 to 70°C, unless otherwise noted)

Characteristic	Symbol	1 MHz		2 MHz		Unit
		Min	Max	Min	Max	
Cycle Time	T_{CYC}	1.0	–	0.5	–	μs
Ø2 Pulse Width	T_C	440	–	200	–	ns
Address Set-Up Time	T_{ACR}	180	–	90	–	ns
Address Hold Time	T_{CAR}	0	–	0	–	ns
R/$\overline{W}$ Set-Up Time	T_{WCR}	180	–	90	–	ns
Read Access Time	T_{CDR}	–	340	–	150	ns
Read Hold Time	T_{HR}	10	–	10	–	ns
Data Bus Active Time (Invalid Data)	T_{CDA}	40	–	40	–	ns

(t_r and t_f = 10 to 30 ns)

READ CYCLE



DOCUMENT NO. 29000 D39
REV. 3, FEBRUARY 1979

PART NUMBER
R650X and R651X

Rockwell

R6500 Microcomputer System DATA SHEET

R6500 MICROPROCESSORS (CPU's)

SYSTEM ABSTRACT

The 8-bit R6500 microcomputer system is produced with N-Channel, Silicon Gate technology. Its performance speeds are enhanced by advanced system architecture. This innovative architecture results in smaller chips — the semiconductor threshold to cost-effectivity. System cost-effectivity is further enhanced by providing a family of 10 software-compatible microprocessor (CPU) devices, described in this document. Rockwell also provides memory and microcomputer system . . . as well as low-cost design aids and documentation.

R6500 MICROPROCESSOR (CPU) CONCEPT

Ten CPU devices are available. All are software-compatible. They provide options of addressable memory, interrupt input, on-chip clock oscillators and drivers. All are bus-compatible with earlier generation microprocessors like the M6800 devices.

The family includes six microprocessors with on-board clock oscillators and drivers and four microprocessors driven by external clocks. The on-chip clock versions are aimed at high performance, low cost applications where single phase inputs, crystal or RC inputs provide the time base. The external clock versions are geared for multiprocessor system applications where maximum timing control is mandatory. All R6500 microprocessors are also available in a variety of packaging (ceramic and plastic), operating frequency (1 MHz and 2 MHz) and temperature (commercial, industrial and military) versions.

MEMBERS OF THE R6500 MICROPROCESSOR (CPU) FAMILY

Microprocessors with On-Chip Clock Oscillator

Model	Addressable Memory
R6502	65K Bytes
R6503	4K Bytes
R6504	8K Bytes
R6505	4K Bytes
R6506	4K Bytes
R6507	8K Bytes

Microprocessors with External Two Phase Clock Output

Model	Addressable Memory
R6512	65K Bytes
R6513	4K Bytes
R6514	8K Bytes
R6515	4K Bytes

FEATURES

- Single +5V supply
- N channel, silicon gate, depletion load technology
- Eight bit parallel processing
- 56 Instructions
- Decimal and binary arithmetic
- Thirteen addressing modes
- True indexing capability
- Programmable stack pointer
- Variable length stack
- Interrupt capability
- Non-maskable interrupt
- Use with any type of speed memory
- 8-bit Bidirectional Data Bus
- Addressable memory range of up to 65K bytes
- "Ready" input
- Direct Memory Access capability
- Bus compatible with M6800
- 1 MHz and 2 MHz operation
- Choice of external or on-chip clocks
- On-the-chip clock options
  - External single clock input
  - RC time base input
  - Crystal time base input
- Commercial, industrial and military temperature versions
- Pipeline architecture

Ordering Information

Order Number: R65XX_ _ _

Temperature Range:
No suffix = $0^{\circ}C$ to $+70^{\circ}C$
E = $-40^{\circ}C$ to $+85^{\circ}C$ (Industrial)
MT = $-55^{\circ}C$ to $+125^{\circ}C$ (Military)
M = MIL-STD-883, Class B

Package:
C = Ceramic
P = Plastic
(Not Avaible for M or MT suffix)

Frequency Range:
No suffix = 1 MHz
A = 2 MHz

Model Designator:
XX = 02, 03, 04, ... 15

NOTE: Contact your local Rockwell Representative concerning availability.

Specifications subject to change without notice

R6500 Signal Description

Clocks (ϕ_1, ϕ_2)

The R651X requires a two phase non-overlapping clock that runs at the V_{CC} voltage level.

The R650X clocks are supplied with an internal clock generator. The frequency of these clocks is externally controlled.

Address Bus (A0-A15)

These outputs are TTL compatible, capable of driving one standard TTL load and 130 pF.

Data Bus (D0-D7)

Eight pins are used for the data bus. This is a bidirectional bus, transferring data to and from the device and peripherals. The outputs are tri-state buffers capable of driving one standard TTL load and 130 pF.

Data Bus Enable (DBE)

This TTL compatible input allows external control of the tri-state data output buffers and will enable the microprocessor bus driver when in the high state. In normal operation DBE would be driven by the phase two (ϕ_2) clock, thus allowing data output from microprocessor only during ϕ_2. During the read cycle, the data bus drivers are internally disabled, becoming essentially an open circuit. To disable data bus drivers externally, DBE should be held low.

Ready (RDY)

This input signal allows the user to halt or single cycle the microprocessor on all cycles except write cycles. A negative transition to the low state during or coincident with phase one (ϕ_1) will halt the microprocessor with the output address lines reflecting the current address being fetched. If Ready is low during a write cycle, it is ignored until the following read operation. This condition will remain through a subsequent phase two (ϕ_2) in which the Ready signal is low. This feature allows microprocessor interfacing with the low speed PROMs as well as fast (max. 2 cycle) Direct Memory Access (DMA).

Interrupt Request ($\overline{\text{IRQ}}$)

This TTL level input requests that an interrupt sequence begin within the microprocessor. The microprocessor will complete the current instruction being executed before recognizing the request. At that time, the interrupt mask bit in the Status Code Register will be examined. If the interrupt mask flag is not set, the microprocessor will begin an interrupt sequence. The Program Counter and Processor Status Register are stored in the stack. The microprocessor will then set the interrupt mask flag high so that no further interrupts may occur. At the end of this cycle, the program counter low will be loaded from address FFFE, and program counter high from location FFFF, therefore transferring program control to the memory vector located at these addresses. The RDY signal must be in the high state for any interrupt to be recognized. A 3KΩ external resistor should be used for proper wire-OR operation.

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

A negative going edge on this input requests that a non-maskable interrupt sequence be generated within the microprocessor.

$\overline{\text{NMI}}$ is an unconditional interrupt. Following completion of the current instruction, the sequence of operations defined for $\overline{\text{IRQ}}$ will be performed, regardless of the state interrupt mask flag. The vector address loaded into the program counter, low and high, are locations FFFA and FFFB respectively, thereby transferring program control to the memory vector located at these addresses. The instructions loaded at these locations cause the microprocessor to branch to a non-maskable interrupt routine in memory.

$\overline{\text{NMI}}$ also requires an external 3K Ω register to V_{CC} for proper wire-OR operations.

Inputs $\overline{\text{IRQ}}$ and $\overline{\text{NMI}}$ are hardware interrupts lines that are sampled during ϕ_2 (phase 2) and will begin the appropriate interrupt routine on the ϕ_1 (phase 1) following the completion of the current instruction.

Set Overflow Flag (S.O.)

A negative going edge on this input sets the overflow bit in the Status Code Register. This signal is sampled on the trailing edge of $\phi 1$ and must be externally synchronized.

SYNC

This output line is provided to identify those cycles in which the microprocessor is doing an OP CODE fetch. The SYNC line goes high during ϕ_1 of an OP CODE fetch and stays high for the remainder of that cycle. If the RDY line is pulled low during the ϕ_1 clock pulse in which SYNC went high, the processor will stop in its current state and will remain in the state until the RDY line goes high. In this manner, the SYNC signal can be used to control RDY to cause single instruction execution.

Reset

This input is used to reset or start the microprocessor from a power down condition. During the time that this line is held low, writing to or from the microprocessor is inhibited. When a positive edge is detected on the input, the microprocessor will immediately begin the reset sequence.

After a system initialization time of six clock cycles, the mask interrupt flag will be set and the microprocessor will load the program counter from the memory vector locations FFFC and FFFD. This is the start location for program control.

After V_{CC} reaches 4.75 volts in a power up routine, reset must be held low for at least two clock cycles. At this time the R/W and (SYNC) signal will become valid.

When the reset signal goes high following these two clock cycles, the microprocessor will proceed with the normal reset procedure detailed above.

ADDRESSING MODES

ACCUMULATOR ADDRESSING — This form of addressing is represented with a one byte instruction, implying an operation on the accumulator.

IMMEDIATE ADDRESSING — In immediate addressing, the operand is contained in the second byte of the instruction, with no further memory addressing required.

ABSOLUTE ADDRESSING — In absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Thus, the absolute addressing mode allows access to the entire 65K bytes of addressable memory.

ZERO PAGE ADDRESSING — The zero page instructions allow for shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. Careful use of the zero page can result in significant increase in code efficiency.

INDEXED ZERO PAGE ADDRESSING — (X, Y indexing) — This form of addressing is used in conjunction with the index register and is referred to as "Zero Page, X" or "Zero Page, Y". The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally due to the "Zero Page" addressing nature of this mode, no carry is added to the high order 8 bits of memory and crossing of page boundaries does not occur.

INDEXED ABSOLUTE ADDRESSING — (X, Y indexing) — This form of addressing is used in conjunction with X and Y index register and is referred to as "Absolute, X", and "Absolute, Y". The effective address is formed by adding the contents of X or Y to the address contained in the second and third bytes of the instruction. This mode allows the index register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

IMPLIED ADDRESSING — In the implied addressing mode, the address containing the operand is implicitly stated in the operation code of the instruction.

RELATIVE ADDRESSING — Relative addressing is used only with branch instructions and establishes a destination for the conditional branch.

The second byte of the instruction becomes the operand which is an "Offset" added to the contents of the lower eight bits of the program counter when the counter is set at the next instruction. The range of the offset is -128 to +127 bytes from the next instruction.

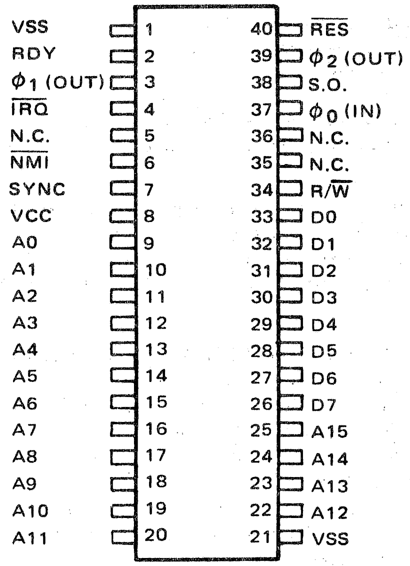
INDEXED INDIRECT ADDRESSING — In indexed indirect addressing (referred to as (Indirect, X)), the second byte of the instruction is added to the contents of the X index register, discarding the carry. The result of this addition points to a memory location on page zero whose contents is the low order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must be in page zero.

INDIRECT INDEXED ADDRESSING — In indirect indexed addressing (referred to as (Indirect), Y), the second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y index register, the result being the low order eight bits of the effective address. The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

ABSOLUTE INDIRECT — The second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits of that memory location is contained in the third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the sixteen bits of the program counter.

INSTRUCTION SET — ALPHABETIC SEQUENCE

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result not Zero
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrement Memory by One
DEX	Decrement Index X by One
DEY	Decrement Index Y by One
EOR	"Exclusive-or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index X by One
INY	Increment Index Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift One Bit Right (Memory or Accumulator)
NOP	No Operation
ORA	"OR" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	Return from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store Index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Register
TYA	Transfer Index Y to Accumulator



R6502 – 40 Pin Package

Features of R6502

- 65K Addressable Bytes of Memory (A0-A15)
- $\overline{\text{IRQ}}$ Interrupt
- On-the-chip Clock
  - TTL Level Single Phase Input
  - RC Time Base Input
  - Crystal Time Base Input
- SYNC Signal
  (can be used for single instruction execution)
- RDY Signal
  (can be used to halt or single cycle execution)
- Two Phase Output Clock for Timing of Support Chips
- $\overline{\text{NMI}}$ Interrupt

INSTRUCTION SET

INSTRUCTIONS		IMMEDIATE			ABSOLUTE			ZERO PAGE			ACCUM			IMPLIED			(IND, X)			(IND), Y			Z. PAGE, X			ABS, X			ABS, Y			RELATIVE			INDIRECT			Z. PAGE, Y			PROCESSOR STATUS CODES	
MNEMONIC	OPERATION	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	OP	n	#	7 6 5 4 3 2 1 0 N V • B D I Z C	MNEMONIC
ADC	A + M + C → A (4) (1)	69	2	2	6D	4	3	65	3	2							61	6	2	71	5	2	75	4	2	7D	4	3	79	4	3										N V • • • • Z C	ADC
AND	A ∧ M → A (1)	29	2	2	2D	4	3	25	3	2							21	6	2	31	5	2	35	4	2	3D	4	3	39	4	3										N • • • • • Z •	AND
ASL	C ← [7 0] ← 0				0E	6	3	06	5	2	0A	2	1										16	6	2	1E	7	3													N • • • • • Z C	ASL
BCC	BRANCH ON C = 0 (2)																															90	2	2							• • • • • • • •	BCC
BCS	BRANCH ON C = 1 (2)																															B0	2	2							• • • • • • • •	BCS
BEQ	BRANCH ON Z = 1 (2)																															F0	2	2							• • • • • • • •	BEQ
BIT	A ∧ M				2C	4	3	24	3	2																															M_7 M_6 • • • • Z •	BIT
BMI	BRANCH ON N = 1 (2)																															30	2	2							• • • • • • • •	BMI
BNE	BRANCH ON Z = 0 (2)																															D0	2	2							• • • • • • • •	BNE
BPL	BRANCH ON N = 0 (2)																															10	2	2							• • • • • • • •	BPL
BRK	BREAK													00	7	1																									• • • 1 • 1 • •	BRK
BVC	BRANCH ON V = 0 (2)																															50	2	2							• • • • • • • •	BVC
BVS	BRANCH ON V = 1 (2)																															70	2	2							• • • • • • • •	BVS
CLC	0 → C													18	2	1																									• • • • • • • 0	CLC
CLD	0 → D													D8	2	1																									• • • • 0 • • •	CLD
CLI	0 → I													58	2	1																									• • • • • 0 • •	CLI
CLV	0 → V													B8	2	1																									• 0 • • • • • •	CLV
CMP	A − M	C9	2	2	CD	4	3	C5	3	2							C1	6	2	D1	5	2	D5	4	2	DD	4	3	D9	4	3										N • • • • • Z C	CMP
CPX	X − M	E0	2	2	EC	4	3	E4	3	2																															N • • • • • Z C	CPX
CPY	Y − M	C0	2	2	CC	4	3	C4	3	2																															N • • • • • Z C	CPY
DEC	M − 1 → M				CE	6	3	C6	5	2													D6	6	2	DE	7	3													N • • • • • Z •	DEC
DEX	X − 1 → X													CA	2	1																									N • • • • • Z •	DEX
DEY	Y − 1 → Y													88	2	1																									N • • • • • Z •	DEY
EOR	A ⊻ M → A (1)	49	2	2	4D	4	3	45	3	2							41	6	2	51	5	2	55	4	2	5D	4	3	59	4	3										N • • • • • Z •	EOR
INC	M + 1 → M				EE	6	3	E6	5	2													F6	6	2	FE	7	3													N • • • • • Z •	INC
INX	X + 1 → X													E8	2	1																									N • • • • • Z •	INX
INY	Y + 1 → Y													C8	2	1																									N • • • • • Z •	INY
JMP	JUMP TO NEW LOC				4C	3	3																												6C	5	3				• • • • • • • •	JMP
JSR	JUMP SUB				20	6	3																																		• • • • • • • •	JSR
LDA	M → A (1)	A9	2	2	AD	4	3	A5	3	2							A1	6	2	B1	5	2	B5	4	2	BD	4	3	B9	4	3										N • • • • • Z •	LDA
LDX	M → X (1)	A2	2	2	AE	4	3	A6	3	2																			BE	4	3							B6	4	2	N • • • • • Z •	LDX
LDY	M → Y (1)	A0	2	2	AC	4	3	A4	3	2													B4	4	2	BC	4	3													N • • • • • Z •	LDY
LSR	0 → [7 0] → C				4E	6	3	46	5	2	4A	2	1										56	6	2	5E	7	3													0 • • • • • Z C	LSR
NOP	NO OPERATION													EA	2	1																									• • • • • • • •	NOP
ORA	A ∨ M → A	09	2	2	0D	4	3	05	3	2							01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3										N • • • • • Z •	ORA
PHA	A → Ms S − 1 → S													48	3	1																									• • • • • • • •	PHA
PHP	P → Ms S − 1 → S													08	3	1																									• • • • • • • •	PHP
PLA	S + 1 → S Ms → A													68	4	1																									N • • • • • Z •	PLA
PLP	S + 1 → S Ms → P													28	4	1																									(RESTORED)	PLP
ROL	← [7 0] ← [C] ←				2E	6	3	26	5	2	2A	2	1										36	6	2	3E	7	3													N • • • • • Z C	ROL
ROR	→ [C] → [7 0] →				6E	6	3	66	5	2	6A	2	1										76	6	2	7E	7	3													N • • • • • Z C	ROR
RTI	RTRN INT													40	6	1																									(RESTORED)	RTI
RTS	RTRN SUB													60	6	1																									• • • • • • • •	RTS
SBC	A − M − $\overline{C}$ → A (1)	E9	2	2	ED	4	3	E5	3	2							E1	6	2	F1	5	2	F5	4	2	FD	4	3	F9	4	3										N V • • • • Z (3)	SBC
SEC	1 → C													38	2	1																									• • • • • • • 1	SEC
SED	1 → D													F8	2	1																									• • • • 1 • • •	SED
SEI	1 → I													78	2	1																									• • • • • 1 • •	SEI
STA	A → M				8D	4	3	85	3	2							81	6	2	91	6	2	95	4	2	9D	5	3	99	5	3										• • • • • • • •	STA
STX	X → M				8E	4	3	86	3	2																												96	4	2	• • • • • • • •	STX
STY	Y → M				8C	4	3	84	3	2													94	4	2																• • • • • • • •	STY
TAX	A → X													AA	2	1																									N • • • • • Z •	TAX
TAY	A → Y													A8	2	1																									N • • • • • Z •	TAY
TSX	S → X													BA	2	1																									N • • • • • Z •	TSX
TXA	X → A													8A	2	1																									N • • • • • Z •	TXA
TXS	X → S													9A	2	1																									• • • • • • • •	TXS
TYA	Y → A													98	2	1																									N • • • • • Z •	TYA

(1) ADD 1 to "N" IF PAGE BOUNDARY IS CROSSED

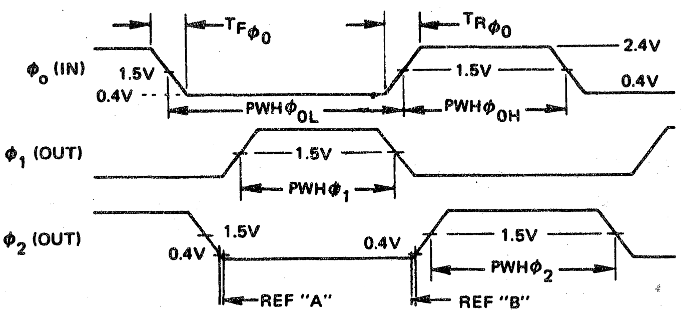
(2) ADD 1 TO "N" IF BRANCH OCCURS TO SAME PAGE
ADD 2 TO "N" IF BRANCH OCCURS TO DIFFERENT PAGE

(3) CARRY NOT = BORROW

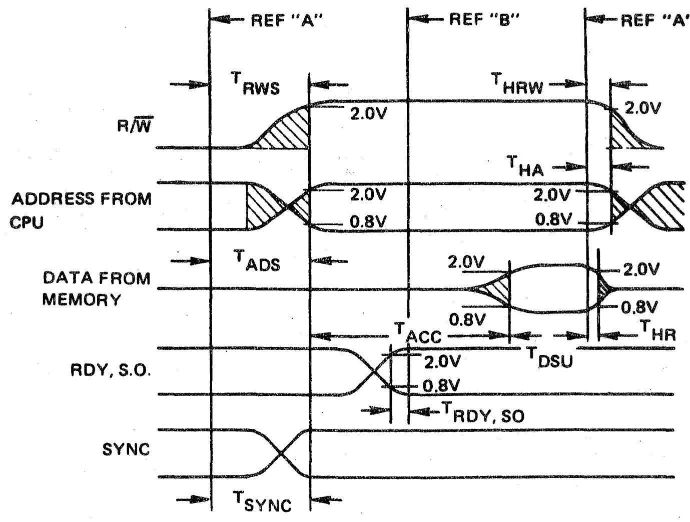
(4) IF IN DECIMAL MODE, Z FLAG IS INVALID
ACCUMULATOR MUST BE CHECKED FOR ZERO RESULT

X	INDEX X	+	ADD	M_7	MEMORY BIT 7
Y	INDEX Y	−	SUBTRACT	M_6	MEMORY BIT 6
A	ACCUMULATOR	∧	AND	n	NO. CYCLES
M	MEMORY PER EFFECTIVE ADDRESS	∨	OR	#	NO. BYTES
Ms	MEMORY PER STACK POINTER	⊻	EXCLUSIVE OR		

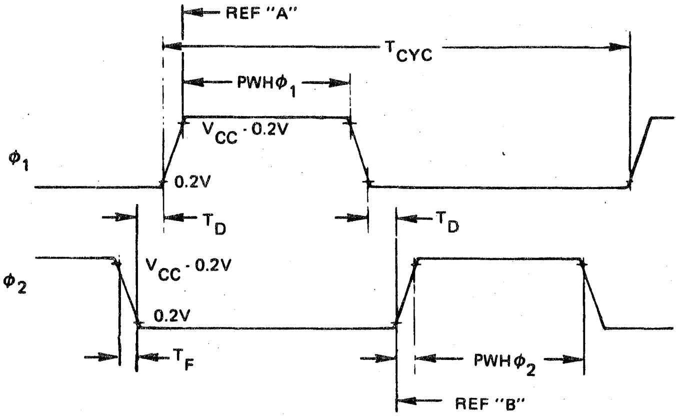
Clock Timing – R6502, 03, 04, 05, 06, 07



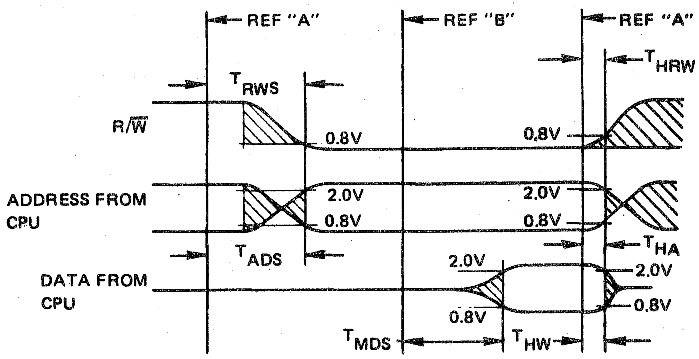
Timing for Reading Data from Memory or Peripherals



Clock Timing – R6512, 13, 14, 15

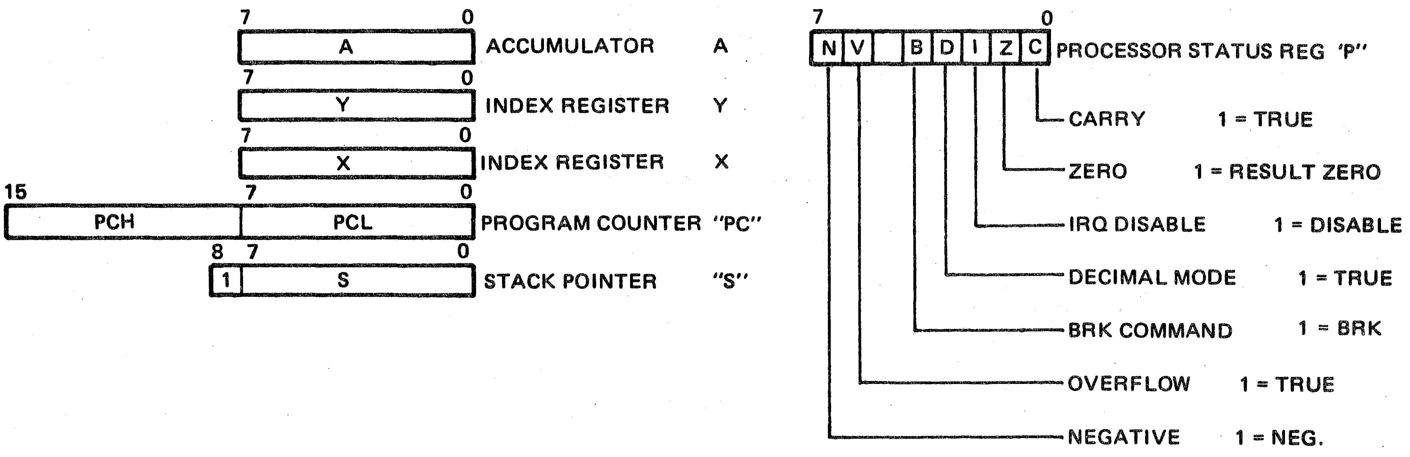


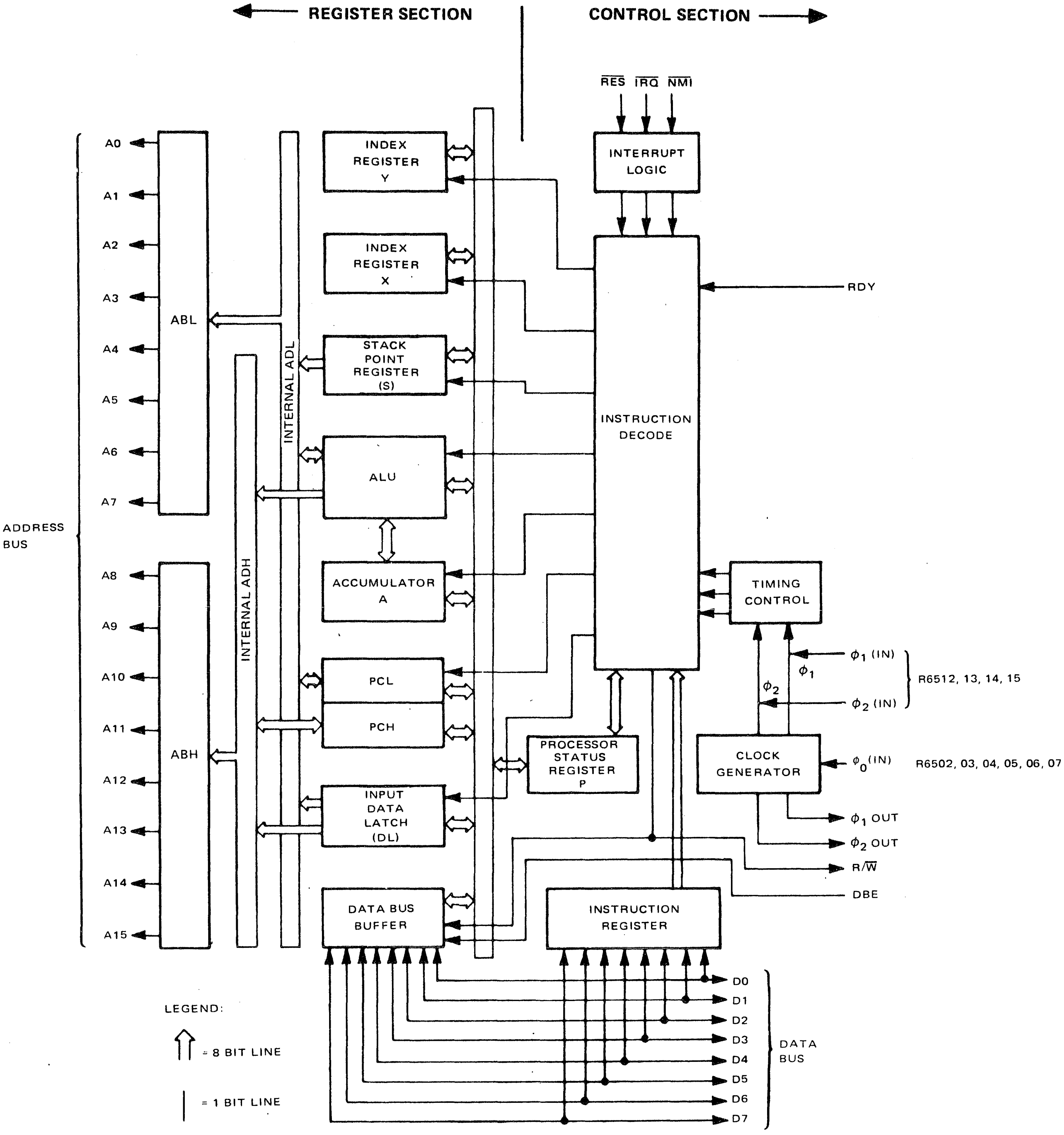
Timing for Writing Data to Memory or Peripherals



Note: "REF." means Reference Points on clocks.

PROGRAMMING MODEL





Note: 1. Clock Generator is not included on R6512, 13, 14, 15
2. Addressing Capability and control options vary with each of the R6500 Products.

R6500 Internal Architecture

DOCUMENT NO. 29000 D47
REVISION 1, OCT. 1978

PART NUMBER
R6522

Rockwell

R6500 Microcomputer System

DATA SHEET

VERSATILE INTERFACE ADAPTER (VIA)

R6522 VERSATILE INTERFACE ADAPTER (VIA)

SYSTEM ABSTRACT

The 8-bit R6500 microcomputer system is produced with N-channel, silicon-gate, depletion-load technology. Its performance speeds are enhanced by advanced system architecture. Its innovative architecture results in smaller chips — the semiconductor threshold to cost-effectivity. System cost-effectivity is further enhanced by providing a family of 10 software-compatible microprocessor (CPU) devices, memory and I/O devices . . . as well as low-cost design aids and documentation.

DESCRIPTION

The R6522 VIA adds two powerful, flexible Interval Timers, a serial-to-parallel/parallel-to-serial shift register and input latching on the peripheral ports to the capabilities of the R6520 Peripheral Interface Adapter (PIA) device. Handshaking capability is expanded to allow control of bidirectional data transfers between VIAs in multiple processor systems and between peripherals.

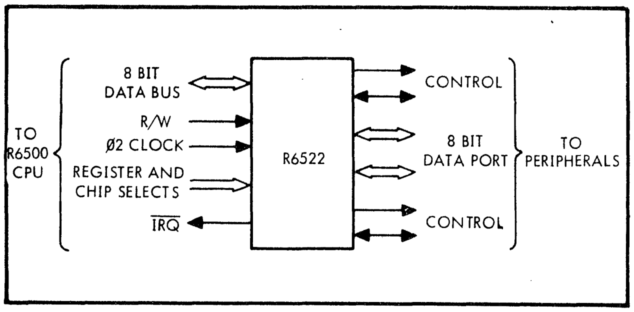
Control of peripherals is primarily through two 8-bit bidirectional ports. Each of these ports can be programmed to act as an input or an output. Peripheral I/O lines can be selectively controlled by the Interval Timers to generate programmable-frequency square waves and/or to count externally generated pulses. Positive control of VIA functions is gained through its internal register organization: Interrupt Flag Register, Interrupt Enable Register, and two Function Control Registers.

FEATURES

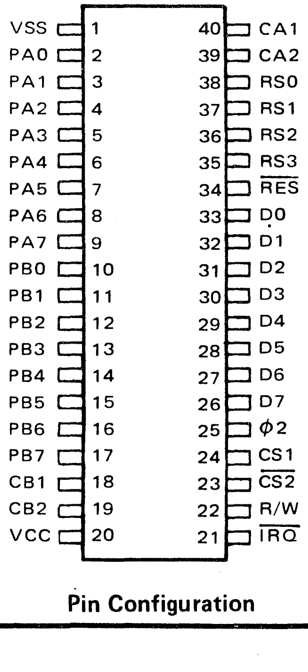
- Organized for simplified software control of many functions
- Compatible with the R650X and R651X family of microprocessors (CPUs)
- Bi-directional, 8-bit data bus for communication with microprocessor
- Two Bi-directional, 8-bit input/output ports for interface with peripheral devices
- CMOS and TTL compatible input/output peripheral ports
- Data Direction Registers allow each peripheral pin to act as either an input or an output
- Interrupt Flag Register allows the microprocessor to readily determine the source of an interrupt and provides convenient control of the interrupts within the chip
- Handshake control logic for input/output peripheral data transfer operations
- Data latching on peripheral input/output ports
- Two fully-programmable interval timers/counters
- Eight-bit Shift Register for serial interface
- Forty-pin plastic or ceramic DIP package.

Ordering Information

Order Number	Package Type	Frequency	Temperature Range
R6522P	Plastic	1 MHz	0°C to +70°C
R6522AP	Plastic	2 MHz	0°C to +70°C
R6522C	Ceramic	1 MHz	0°C to +70°C
R6522AC	Ceramic	2 MHz	0°C to +70°C
R6522PE	Plastic	1 MHz	-40°C to +85°C
R6522APE	Plastic	2 MHz	-40°C to +85°C
R6522CE	Ceramic	1 MHz	-40°C to +85°C
R6522ACE	Ceramic	2 MHz	-40°C to +85°C
R6522CMT	Ceramic	1 MHz	-55°C to +125°C



Basic R6522 Interface Diagram



Pin Configuration

Specifications subject to change without notice

OPERATION SUMMARY

Register Select Lines (RS0, RS1, RS2, RS3)

The four Register select lines are normally connected to the processor address bus lines to allow the processor to select the internal R6522 register which is to be accessed. The sixteen possible combinations access the registers as follows:

RS3	RS2	RS1	RS0	Register	Remarks	RS3	RS2	RS1	RS0	Register	Remarks
L	L	L	L	ORB		H	L	L	L	T2L-L T2C-L	Write Latch Read Counter
L	L	L	H	ORA	Controls Handshake	H	L	L	H	T2C-H	Triggers T2L-L/T2C-L Transfer
L	L	H	L	DDRB		H	L	H		SR	
L	L	H	H	DDRA		H	L	H	H	ACR	
L	H	L	L	T1L-L T1C-L	Write Latch Read Counter	H	H	L	L	PCR	
L	H	L	H	T1C-H	Trigger T1L-L/T1C-L Transfer	H	H	L	H	IFR	
L	H	H	L	T1L-L		H	H	H	L	IER	
L	H	H	H	T1L-H		H	H	H	H	ORA	No Effect on Handshake

Note: L · 0.4V DC, H = 2.4V DC.

Timer 2 Control

RS3	RS2	RS1	RS0	R/W = L	R/W = H
H	L	L	L	Write T2L-L	Read T2C-L Clear Interrupt flag
H	L	L	H	Write T2C-H Transfer T2L-L to T2C-L Clear Interrupt flag	Read T2C-H

Writing the Timer 1 Register

The operations which take place when writing to each of the four T1 addresses are as follows:

RS3	RS2	RS1	RS0	Operation (R/W = L)
L	H	L	L	Write into low order latch
L	H	L	H	Write into high order latch Write into high order counter Transfer low order latch into low order counter Reset T1 interrupt flag
L	H	H	L	Write low order latch
X	H	H	H	Write high order latch Reset T1 interrupt flag

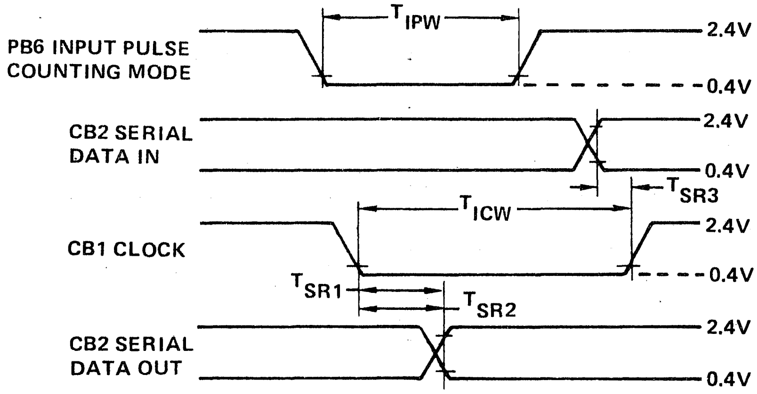
Reading the Timer 1 Registers

For reading the Timer 1 registers, the four addresses relate directly to the four registers as follows:

RS3	RS2	RS1	RS0	Operation (R/W = H)
L	H	L	L	Read T1 low order counter Reset T1 interrupt flag
L	H	L	H	Read T1 high order counter
L	H	H	L	Read T1 low order latch
L	H	H	H	Read T1 high order latch

I/O Timing Characteristics

Characteristic	Symbol	Min	Typ	Max	Unit
Rise and fall time for CA1, CB1, CA2 and CB2 input signals	T_{RF}	–	–	1.0	μs
Delay time, clock negative transition to CA2 negative transition (read handshake or pulse mode)	T_{CA2}	–	–	1.0	μs
Delay time, clock negative transition to CA2 positive transition (pulse mode)	T_{RS1}	–	–	1.0	μs
Delay time, CA1 active transition to CA2 positive transition (handshake mode)	T_{RS2}	–	–	2.0	μs
Delay time, clock positive transition to CA2 or CB2 negative transition (write handshake)	T_{WHS}	–	–	1.0	μs
Delay time, peripheral data valid to CB2 negative transition	T_{DC}	0	–	1.5	μs
Delay time, clock positive transition to CA2 or CB2 positive transition (pulse mode)	T_{RS3}	–	–	1.0	μs
Delay time, CB1 active transition to CA2 or CB2 positive transition (handshake mode)	T_{RS4}	–	–	2.0	μs
Delay time, peripheral data valid to CA1 or CB1 active transition (input latching)	T_{IL}	300	–	–	ns
Delay time CB1 negative transition to CB2 data valid (internal SR clock, shift out)	T_{SR1}	–	–	300	ns
Delay time, negative transition of CB1 input clock to CB2 data valid (external clock, shift out)	T_{SR2}	–	–	300	ns
Delay time, CB2 data valid to positive transition of CB1 clock (shift in, internal or external clock)	T_{SR3}	–	–	300	ns
Pulse Width -- PB6 Input Pulse	T_{IPW}	2	–	–	μs
Pulse Width – CB1 Input Clock	T_{ICW}	2	–	–	μs
Pulse Spacing - PB6 Input Pulse	I_{IPS}	2	–	–	μs
Pulse Spacing -- CB1 Input Pulse	I_{ICS}	2	–	–	μs

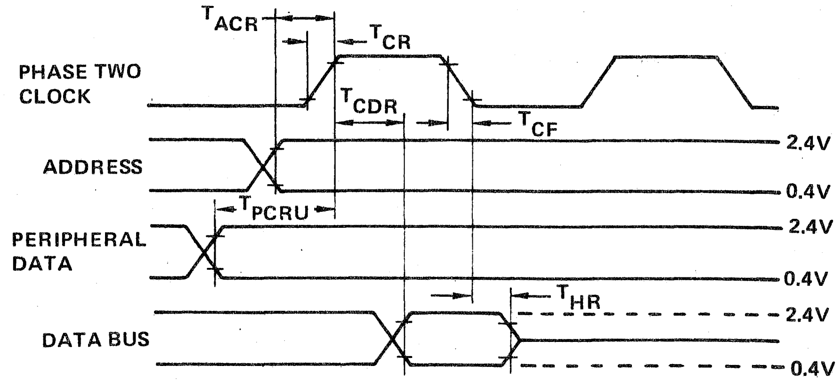


I/O Timing Characteristics

TIMING CHARACTERISTICS

Read Timing Characteristics (loading 130 pF and one TTL load)

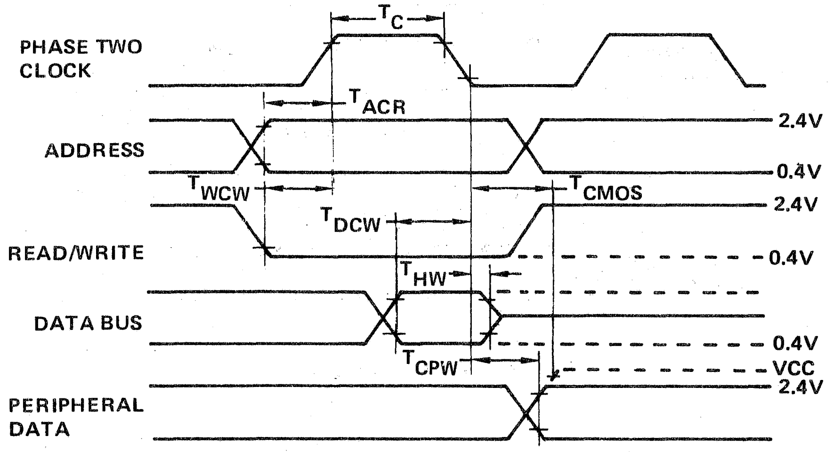
Parameter	Symbol	Min	Typ	Max	Unit
Delay time, address valid to clock positive transition	T_{ACR}	180	–	–	nS
Delay time, clock positive transition to data valid on bus	T_{CDR}	–	–	395	nS
Peripheral data setup time	T_{PCR}	300	–	–	nS
Data bus hold time	T_{HR}	10	–	–	nS
Rise and fall time for clock input	T_{RC} T_{RF}	–	–	25	nS



Read Timing Characteristics

Write Timing Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
Enable pulse width	T_C	0.47	–	25	µS
Delay time, address valid to clock positive transition	T_{ACW}	180	–	–	nS
Delay time, data valid to clock negative transition	T_{DCW}	300	–	–	nS
Delay time, read/write negative transition to clock positive transition	T_{WCW}	180	–	–	nS
Data bus hold time	T_{HW}	10	–	–	nS
Delay time, Enable negative transition to peripheral data valid	T_{CPW}	–	–	1.0	µS
Delay time, clock negative transition to peripheral data valid CMOS (VCC - 30%)	T_{CMOS}	–	–	2.0	µS



Write Timing Characteristics

Timer 1 Operating Modes

Two bits are provided in the Auxiliary Control Register to allow selection of the T1 operating modes. These bits and the four possible modes are as follows:

ACR7 Output Enable	ACR6 "Free-Run" Enable	Mode
0	0	Generate a single time-out interrupt each time T1 is loaded
0	1	Generate continuous interrupts
1	0	Generate a single interrupt and an output pulse on PB7 for each T1 load operation
1	1	Generate continuous interrupts and a square wave output on PB7

FUNCTION CONTROL

Control of the various functions and operating modes within the R6522 is accomplished primarily through two registers, the Peripheral Control Register (PCR), and the Auxiliary Control Register (ACR). The PCR is used primarily to select the operating mode for the four peripheral control pins. The Auxiliary Control Register selects the operating mode for the Interval Timers (T1, T2), and the Serial Port (SR).

Peripheral Control Register

The Peripheral Control Register is organized as follows:

Bit #	7	6	5	4	3	2	1	0
Function	CB2 Control			CB1 Control	CA2 Control			CA1 Control

Typical functions are shown below:

PCR3	PCR2	PCR1	Mode
0	0	0	Input mode — Set CA2 interrupt flag (IFR0) on a negative transition of the input signal. Clear IFR0 on a read or write of the Peripheral A Output Register.
0	0	1	Independent interrupt input mode — Set IFR0 on a negative transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 interrupt flag.
0	1	0	Input mode — Set CA2 interrupt flag on a positive transition of the CA2 input signal. Clear IFR0 with a read or write of the Peripheral A Output Register.
0	1	1	Independent interrupt input mode — Set IFR0 on a positive transition of the CA2 input signal. Reading or writing ORA does not clear the CA2 interrupt flag.
1	0	0	Handshake output mode — Set CA2 output low on a read or write of the Peripheral A Output Register. Reset CA2 high with an active transition on CA1.
1	0	1	Pulse output mode — CA2 goes low for one cycle following a read or write of the Peripheral A Output Register.
1	1	0	Manual output mode — The CA2 output is held low in this mode.
1	1	1	Manual output mode — The CA2 output is held high in this mode.

Auxiliary Control Register

Many of the functions in the Auxiliary Control Register have been discussed previously. However, a summary of this register is presented here as a convenient reference for the R6522 user. The Auxiliary Control Register is organized as follows:

Bit #	7	6	5	4	3	2	1	0
Function	T1 Control		T2 Control	Shift Register Control			PB Latch Enable	PA Latch Enable

Shift Register Control

The Shift Register operating mode is selected as follows:

ACR4	ACR3	ACR2	Mode
0	0	0	Shift Register Disabled.
0	0	1	Shift in under control of Timer 2.
0	1	0	Shift in under control of system clock.
0	1	1	Shift in under control of external clock pulses.
1	0	0	Free-running output at rate determined by Timer 2.
1	0	1	Shift out under control of Timer 2.
1	1	0	Shift out under control of the system clock.
1	1	1	Shift out under control of external clock pulses.

T2 Control

Timer 2 operates in two modes. If ACR5 = 0, T2 acts as an interval timer in the one-shot mode. If ACR5 = 1, Timer 2 acts to count a predetermined number of pulses on pin PB6.