IBM Maintenance Library

3145 Processing Unit
Theory — Maintenance

SY24-3581-1

IBM Maintenance Library

3145 Processing Unit
Theory — Maintenance

SY24-3581-1

PREFACE

This manual is for readers who understand the basic concepts of computers and system programming. Introductory information that describes data formats, instruction formats, status switching, and program interrupts is in *IBM System/360 Principles of Operation,* GA22-6821; and *IBM System/370 Principles of Operation,* GA22-7000.

Integrated graphic techniques are used where practical to provide standalone diagrams and modular units of information.

The information ranges from basic system concepts, contained in Chapter 1, to the detailed operations of a functional unit or area contained in Chapters 2-13. Each chapter contains a modular segment of information about a major functional area of the system, such as CPU, storage and console.

This manual also contains information to be used in maintaining or repairing the 3145. It also includes CPU reference data; channel reference data; maintenance techniques; checks, adjustments, and removal procedures, power and cooling maintenance procedures; physical locations; description of panel indicators.

The 370 Microprogram Listing contains the microinstruction sequences required to execute any program instruction. Therefore, the execution of only some of the program instructions is described in this manual. Instead, certain representative examples are described, using flow charts and example operands. The examples show microprogram sequences and data paths required to execute the instruction.

The 3145 Theory-Maintenance manual shipped as part of the maintenance package for the system is contained in four volumes (30 through 33). The contents of each volume is identified on the accompanying red MDM tab.

Because this manual is split into four parts, the index is in the front of the manual. The index, which effectively points to the places where information exists, follows the chart showing the major areas of each chapter. Each chapter contains its own contents. This makes it unnecessary to refer back to Volume 30 whenever you are looking for specific information within a particular chapter.

This manual references diagrams in the *3145 Processing Unit, Maintenance Diagrams,* SY24-3580.

Other related manuals are:

IBM System/370 Model 145 Functional Characteristics, GA24-3557

IBM System/370 Model 145 Operating Procedures, C38-0015

IBM System/370 Model 145 Channel Characteristics, GA24-3573

IBM System/370 Model 145 Installation Information-Physical Planning, GA22-6976

IBM System/370 System Summary, GA22-7001

IBM FE Theory of Operation, Component Circuits, SLT, SLD, ASLT, MST, SY22-2798

IBM FE Theory of Operation, Power Supplies, SLT, SLD, ASLT, MST, SY22-2799

IBM FE Theory of Operation, Monolithic System Technology, Packaging Tools, Wiring Change Procedures, SY22-6739

Second Edition (October 1971)

This edition (SY24-3581-1) is a major revision of, and obsoletes SY24-3581-0. This publication has been revised completely with major additions, deletions, and reformatting of existing material. For this reason, the reader should review this edition in its entirety.

Significant changes and additions to the information contained in this publication are continually being made: any such changes will be in subsequent revisions or Technical Newsletters.

Request for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This manual has been prepared by the IBM Systems Development Division, Product Publications, Dept. K10, P.O. Box 6, Endicott, N.Y. 13760. To help us evaluate this manual, you are encouraged to use the Reader's Comment Form in the back of this manual. Your comments will help to provide better documentation in both this manual and future manuals. If the form has been removed, comments may be sent to the above address. Comments become the property of IBM.

3145 Processing Unit Theory - Maintenance

3145

S/N

MDM

THEORY-MAINT

INTRODUCTION to CHAPTER 3

Vol. 30

- CHAPTER 1 Introduction
  - General Description
  - CPU Configuration
  - Features
  - Microprogram
  - 3145 Data Flow
  - Overall Data Flow
  - Error Handling
  - Compatibility Between the 3145 and Other S/370 Models
  - Standard Interface
  - 3145 Channels
  - IFA
- CHAPTER 2 Functional Units
  - SPTL
  - SDBO Pre-Asm
  - Local Storage
  - Expanded Local Storage
  - External Facilities
  - I-Cycles
  - A-Register B-Register and ALU
  - Z-Register and D-Register
  - Control Register and CPU Clock
  - M, N, and MB Registers
  - Secondary Control Assembly
  - Standard Features
- CHAPTER 3 Main and Control Storage
  - General Information
  - Functional Areas
  - Functional Operations

3145

S/N

MDM

THEORY-MAINT

CHAPTER 4 to CHAPTER 7

Vol. 31

- CHAPTER 4 Microprogram Concepts
  - 270 Micro-program
  - 370 Micro-program Listing
  - Control Words
  - Micro-program Reference Material
- CHAPTER 5 System Control Panel
  - Introduction
  - System Control Panel
  - Fixed Indicators
  - Roller Indicators
  - Rotary Switches
  - Toggle Switches
  - Keys
  - Use Meter
- CHAPTER 6 Console File Adapter Introduction
  - Data Flow
  - Console File Clock
  - Shift Register
  - Section Ready
  - CE Comments
  - CF Error Checks
  - Initial Micro-program Program Load
  - CF Removal and Replacement
- CHAPTER 7 Console Printer-Keyboard
  - Introduction
  - 3210 Console Printer-Keyboard
  - 3215 Console Printer-Keyboard
  - Alter/Display Operations
  - Microprogram Information
  - Programming Information

3145

S/N

MDM

THEORY-MAINT

CHAPTER 8 to CHAPTER 10

Vol. 32

- CHAPTER 8 Channels
  - Introduction
  - Model 145 Channels
  - Byte-Multiplexer Channel
  - Selector Channels
  - Block-Multi-plexer Feature
  - Selector Channel Test Example
- CHAPTER 9 Optional Features
  - IBM 1401/1440/1460 and 1410/7010 Compatibility
  - Direct Control
  - Channel-to Channel Adapter
- CHAPTER 10 Integrated File Adapter
  - Introduction
  - Functional Units
  - Theory of Operations
  - Maintenance Features
  - IFA Power Supplies and Interface

3145

S/N

MDM

THEORY-MAINT

CHAPTER 11 to APPENDIX A

Vol. 33

- CHAPTER 11 Power Control and Distrib-ution
  - General Information
  - Motor Generator
  - Electrical Component Locations
  - Power Distrib-ution
  - Power Sequencing
  - Protection and Checking Circuits
  - Power Service Checks
  - Removal and Replacement
  - Power Trouble-shooting Guides
- CHAPTER 12 Error Handling
  - Introduction
  - Recording Modes
  - Error Check and Correction
  - Machine Checks and Error Retry
  - Machine-Check Interruptions
  - Machine-Check Logout
  - Summary of Extended Machine-Check Handling
  - Stop After Log
  - Software Recovery
- CHAPTER 13 Diagnostic Hardware
  - Testing Philosophy
  - Data Flow
  - Basic Diag-nostic Trouble-shooting
  - CPU Diag-nostic Hard-ware
  - Selector Channel
  - 'GA' Diagnostic Function
- APPENDIX A Maintenance Aids

REMEMBER

Each chapter has its own table of contents.

ABBREVIATIONS

A

A	and
AAR	A-address register
ABRTY	A and B retry register
AC	alternating current
ACB	address check boundary
ACBR	address check boundary register
ACR	automatic carrier return
ADDR	address
adj	adjust
Adr-I	address-in
Adr-O	address-out
adv	advance
ALD	automated logic diagram
ALS	A-local storage
alt	alter
ALU	arithmetic logic unit
AM	address mark
amp	amplifier, ampere
ANUM	add numeric
appndg	appendage
APR	alternate path retry
ASCII	american standard code II
ASCP	automatic system control program
asm	assembler
AT	attention (file)
ATTN	attention
avl	available

B

BAL	branch and link
BAR	B-address register
BBE	branch on bit equal
BC	basic control
BCA	bit count appendage
BCA	basic channel adapter
BCAI	basic channel adapter interface
BCD	binary coded decimal
BCE	branch on character equal
BDIL	branch and do interpretive loop
bfr	buffer
BI FLAG	branch on invalid flag
B LS	B-local storage
BMF	block multiplexer feature
BR	bit ring
BR	branch
brd	board
BS	byte source
BSM	basic storage module
bwd	backward
BWF	branch if wordmark or zone equal
BYTDST	byte destination

C

C	count
CA	control address
CAR	cylinder address register
CAW	channel address word
CB	circuit breaker
CC	condition code, chain command, cyclic code
CCC	channel control check
CCH	channel check handler
CCW	channel command word
CD	chain data time of day clock
CDC	channel data check
CE	channel end
CF	console file
CFC	console file checking
CFDA	console file disk address
CFDR	console file data register
chan	channel
char	character
chk	check
chng	change
chnl	channel
CKD	count, key, and data
clk	clock
CM	current module
cmd	command
Cmd-O	command out
cmnd	command
cncl	cancel
cnd	condition
cnsl	console
cnt	count
cntr	counter
coax	coaxial cable
comp	compare
con-con	contingent connection
cond	condition
corr	correction
CP	circuit protector
cpmt	complement
CPU	central processing unit
CPURTY	central processing unit retry register
CR	control register
C-Reg	control register
CS	control storage
CSW	channel status word
CTCA	channel to channel adapter
CTCAX	channel to channel adapter X system
CTCAY	channel to channel adapter Y system
Ctrl	control
CU	control unit
CUA	control unit address
CUB	control unit busy
CUE	control unit end
cyc	cycle
cyl	cylinder

D

D	data
DA	data address
DASD	direct access storage device
dbl	double
DC	direct control
DC	direct current
DCBI	direct control bus-in
DCBO	direct control bus-out
DCC	disconnect command chaining
DCHI	direct control hold-in
DDR	dynamic device reconfiguration
DE	device end
dec	decode, decimal, decrement
DED	double error detect
dest	destination
det	detect
DF	disk file
diag	diagnostic
diff	difference
DIL	do interpretive loop
Dir-In	direct control bus-in
Dir-Out	direct control bus-out
Disc-I	disconnect in
dist	distribution
DL	data length
dly	delay
D-Mod	D-modifier
Doc	documentary console
DOS	disk operating system
dply	display
dsbld	disabled
dup	duplicate

E

EBCDIC	extended binary coded decimal interchange code
EBI	external bus-in
EBO	external bus-out
EC	external control
ECC	error checking and correction
ECCL	error checking and correction logic
ECM	extended control mode
ECNT	error count register
ECSW	extended channel status word
ED	external damage
EDBI	external data bus-in
EDBO	external data bus-out
EM	external damage report mask
env	envelope
EOF	end of file
EPO	emergency power off
eq	equal
ERDS	environmental recording data set
EREP	environment recording edit and print program
ERP	error recovery procedure
err	error
EXCA	external control assembler
EXE CPLT	execute complete
exp	expanded
EXPLS	expanded local storage
ext	external
ext asm	external assembler
ext dst	external register destination
extint	external interrupt

F

FBAK	file backup external word
FBO	file bus-out
FCH	file count register high
FCL	file count register low
FCND	file conditions external word
fdbck	feed back
FDR	file data register
FERR	file error external word
FM	file mask
FOP	file operation register
F Stat	file status external word
FTAG	file tags external word
FTC	flush through checking
fwd	forward

G

GBS	selector channel buffer status external word
GBUF	selector channel buffer external word
gen	generate
GM	group mark
gnd	ground
grp	group
GSTAT	selector channel status external word

H

HA	home address
HDV	halt device
hdwr	hardware
hi	high
HIO	halt input/output
HMRTY	H and M retry registers
HS	hard stop
hz	hertz

I

IAR	instruction address register
IB	interrupt buffer
IBU	I-register backup
IC	instruction counter
ICC	interface control check
icplt	incomplete
I-cy	instruction cycle
id	identifier
IFA	integrated file attachment
IFCC	interface control check
IFCU	integrated file control unit
IL	incorrect length
ILC	instruction length code
IM	input/output extended logout mask
IMPL	initial microprogram program loading
inc	increment
ind	indicator
inh	inhibit
inst	instruction
intf	interface
intlk	interlock
intr	interrupt
intv	interval
intvn	intervention
invld	invalid
I/O	input or output
IOCA	input/output communications area
IOEL	input/output extended logout
IPL	initial program load
IPM	in process register
ISK	insert storage key

J

JCL	job control language

K

K	key
KD	key and data
KL	key length

L

L	length
LC	lower case
LD	line driver
LEX	local execute mode
LO	low
Logl	logical
LH	L-register high half
LHM	left hand margin
LL	L-register low
LR	line receiver
LRU	least recently used
LS	local storage
LSCA	local storage control assembler
LSCS	local storage control storage
LSDST	local store destination
lth	latch

M

mach	machine
MB	M-register back up
MBO	multiplexer bus-out
MC	machine check
MCAR	machine-check analysis and recording
MCEL	machine-check extended logout
MCH	machine-check handler
MCIC	machine-check interruption code
MCKA	machine-check A register
MCPU	move data in CPU
MCRR	machine-check recovery recorder
MFT	multiple fixed tasks
MG	motor generator
Mid-Pac	middle power package regulator
MIO	move data for I/O
misc	miscellaneous
MLC	machine level control
MLS	micro listings
mod	module
mono	monolithic
MOP	mini operation register
MPF	main power frame
MPX	multiplexer
MRTY	M retry register
ms	millisecond
us.	microsecond
MS	main storage
MSF	main storage frame
MST	monolithic systems technology
MTO	multiplexer tags-out
MTI	multiplexer tags-in
MUA	multiple unit address
MVT	multiple variable task

N

NA	next address
NOP	no operation
norm	normal
NPL	new product language
ns	nanoseconds

O

OBR	outboard recorder
oc	overcurrent
OE	exclusive OR
OP	operation
OP-I	operational in
OPL	operational
OP-O	operational-out
OS	operating system
osc	oscillator
OV	over voltage

P

PAA	pre-address assembler
PCI	program-controlled interrupt
PD	instruction processing damage
PDAR	program damage assessment and repair
PE	print emitter
PF	power frame
PGA	power gate A
PGB	power gate B
PI DEC	priority interrupt extended control
PIR	priority interrupt register
PIRM	priority interrupt register mask
POH	power on hours
pos	position
PR-KB	printer keyboard
proc	process
prog	program
prot	protect
PSW	program status word
p.y	parity
pwr	power

R

R0	record zero
R1	record 1
RAC	remote analysis center
RAS	reliability and serviceability
RCNT	retry count register
RCS	reloadable control storage
RD	read
RDK	reset diagnostic key
rdy	ready
recal	recalibrate
ref	reference
reg	regulator, register
req	request
reqd	required
Req-I	request-in
rev	reverse
RHM	right-hand margin
RM	record mark
RM	recovery report
RMS	recovery management system
RR	record ready
Rst	reset, restart
rtn	return
rty	retry
rty flg	retry flag

S

SAR	storage address register
sch	search
SCR	silicon controlled rectifier
SD	system damage
SDBI	storage data bus-in
SDBO	storage data bus-out
SDC	suppress data check
SDK	set diagnostic key
SDR	storage data register
SDR	statistical data recorder
sec	secondary
sect	sector
sel	select
seld	selected
Sel-I	select-in
Sel-O	select-out
selr	selector
seq	sequence
ser des	serializer/deserializer
SEREP	system error recording edited printout
Serv-I	service-in
Serv-O	service-out
SI	system incidents
SIO	start input/output
SIOF	start I/O fast release
SLI	suppress length indication
SM	synchronous mask
sng	single
SPTL	S, P, T and L registers
SPTLB	SPTL back-up word for SPTL registers
SR	system recovery

S/R	set or reset
SS	singleshot
SSK	set storage key
Stat-I	status-in
std	standard
stg	storage
STI	status-in
STIDC	store identification channel
STIDP	store identification processor
stkd	stacked
stor	storage
stp	stop
SUA	single unit address
supp	suppress
Sup-O	suppress-out
SUT	system unavailable time
SVC	supervisor call
SVI	service-in
sw	switch
SX	selector/block multiplexer channel
sync	synchronize
sys	system

T

TB	terminal block
TCH	test channel
TD	timer damage
tfrm	transformer
TH	T-register high
therm	thermal
thld	threshold
thru	through
TIC	transfer in channel
TIO	test input/output
TOD	time of day
TODH	time of day high word
TODL	time of day low word
T/R	tilt or rotate, transformer/rectifier
tgr	trigger
TSBO	timing signal bus-out

U

UCW	unit control word

V

vac	volts alternating current
val	validate
vdc	volts direct current
VFO	variable frequency oscillator
VOM	volt/ohm meter

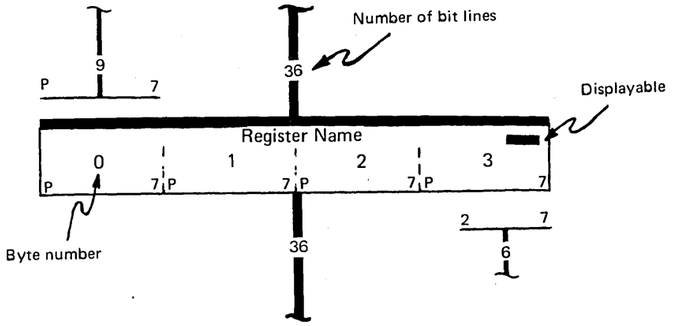
W

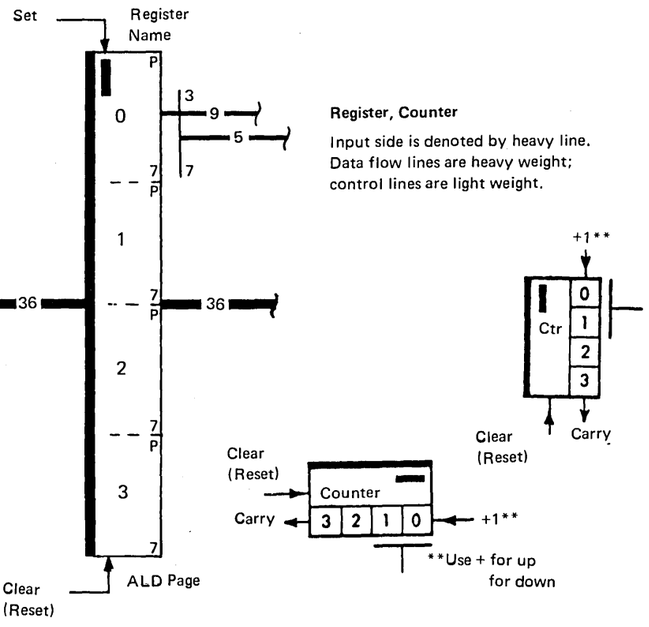
wd	word
WLR	wrong length record
WM	word mark
Wr	write
WS	word separator, word source

X

XFer	transfer
X-Late	translate

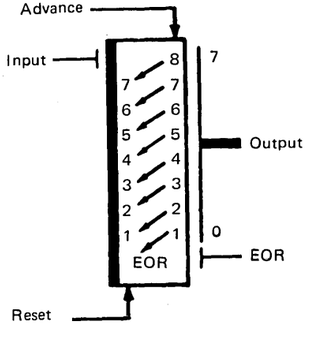
LEGEND

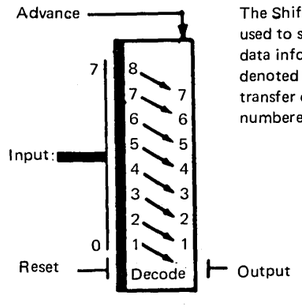




Register, Counter

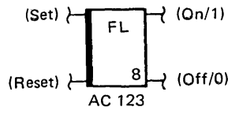
Input side is denoted by heavy line. Data flow lines are heavy weight; control lines are light weight.





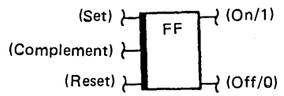
Shift Registers

The Shift Registers are commonly used to serialize and deserialize data information. Input side is denoted by heavy line. A partial transfer of contents is shown by numbered input/output lines.



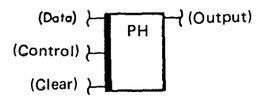
Flip Latch

Input side is denoted by heavy line. Circuit multiples shown by numerals in lower right corner. ALD reference page may be shown beneath.



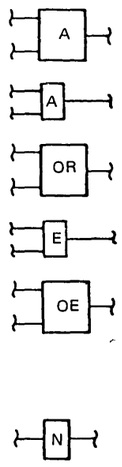
Flio-Flop

Input side is denoted by heavy line. Shift signal is shown by a P or N.



Polarity Hold

Input side is denoted by heavy line.



And

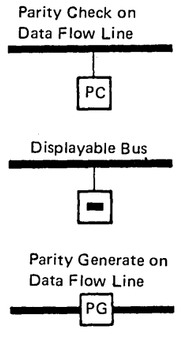
And

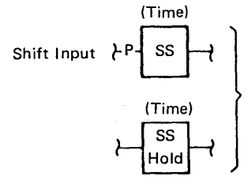
Or

Even

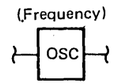
Exclusive OR

Negator (Inverter)

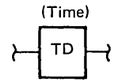




Singleshot



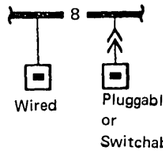
Oscillator



Time Delay

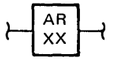
Indicator Lamp

Identifies indicatable bus,register, latch, etc. such as:



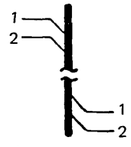
Indicatable bus with number of bus lines indicated.

Indicatable Flip Latch

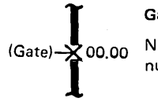


Amplifier

XX Abbreviations
CD = Core Driver
DF = Differential
HD = Head Driver
ID = Indicator Driver
LD = Line Driver
LT = Line Terminator
MD = Magnet Driver
V = Voltage



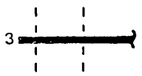
Multiple Line Transfer



Gate

Numerals against gate symbol give page or diagram number of gating circuit.

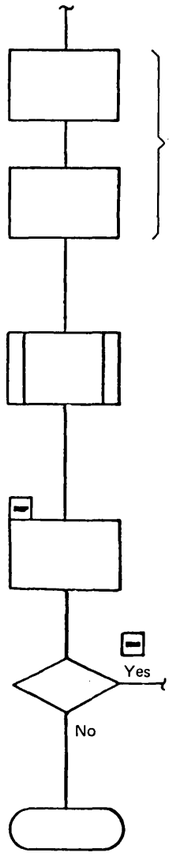
TIMING CHARTS



Active State

Numerals at beginning and end of the bar identify the signal(s) on the same chart that activate and deactivate this line. "(Not)" with the number indicates that lock of the signal conditions the line.

Flowcharts



Process

Major Functions or Events

Function or Process Detailed Elsewhere

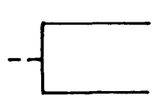
Indicate in block where the detailed flow chart of the process is located.

Flow chart block where action is indicatable.
(Use and placement of indicator is optional)

Flow chart block where condition branched on is indicatable.
(Use and placement of indicator is optional)

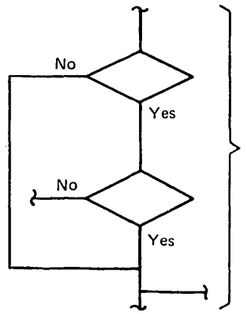
Terminal

Indicates beginning or end of flow path.



Annotation, comment block.

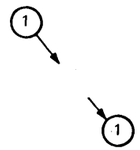
Gives descriptive comment or explanatory note.



Decision

Indicates a point in a flow chart where a branch to alternate paths is possible.

GENERAL



On-Page Connector

Indicates connection between two parts of the same diagram. Arrow leaving symbol points (line-of-sight) to correspondingly numbered symbol.

On-Page Connector

Indicates connection between two parts of the same diagram. Alphameric grid coordinate of complementary connector shown beneath.

Off-Page Connector

Indicates connection between diagrams located on separate pages. Location of correspondingly - lettered symbol shown adjacent. Alphameric grid coordinate may be included.

INDEX

T

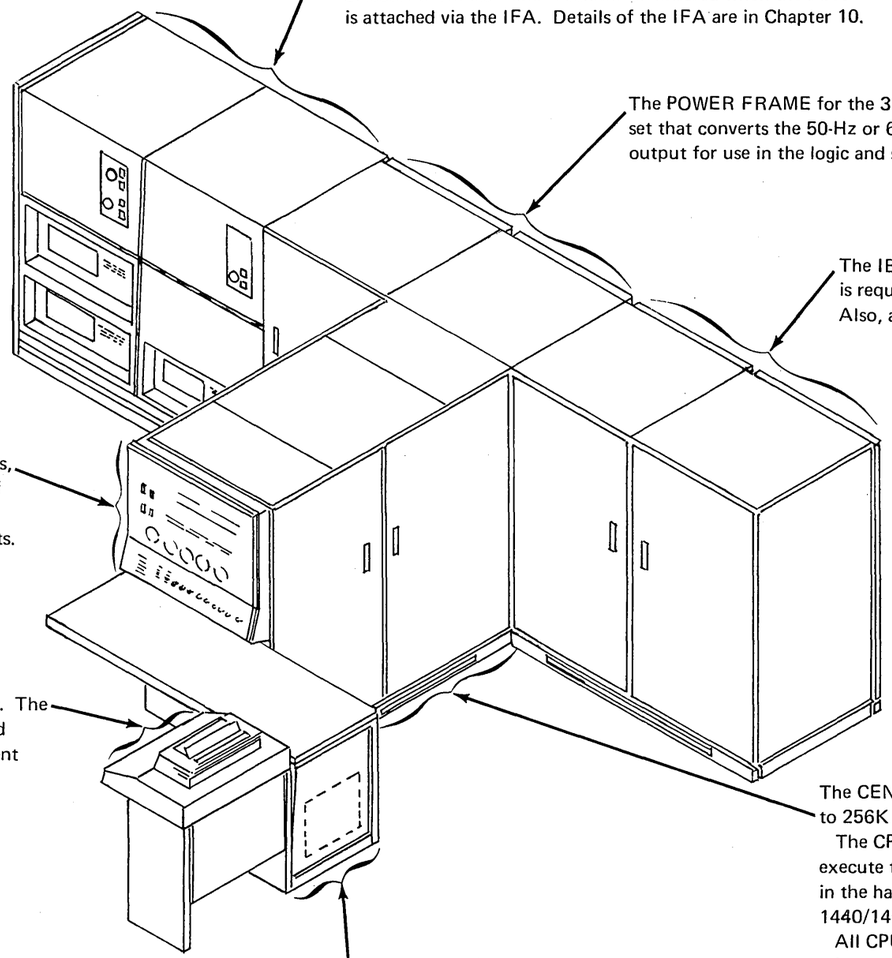
CHAPTER 1. INTRODUCTION

3145 PROCESSING UNIT--GENERAL DESCRIPTION

The System/370 Model 145 (3145) is a high-availability general-purpose data processing system that provides reliability, availability, performance, and convenience demanded by both business and scientific users.

This is achieved by:

- Using monolithic system technology (MST) circuitry and monolithic storage.
- Providing logout information. Hard copy is available under console switch control. Programming determines whether the logout information is to be written on some I/O device (disk, tape, etc).
- Providing microprogram retry. Certain kinds of machine errors cause a portion of the microprogram to repeat the same function.
- Providing Error Checking and Correction (ECC). Correction circuitry for both main and control storage automatically corrects single-bit failures. Detects double-bit failures.
- Using extensive internal checking.



The 2319 DISK STORAGE FACILITY attached via the INTEGRATED FILE ATTACHMENT (IFA) is an optional feature. When installed, a portion of the control logic is contained within the CPU and is addressed as channel 1. Selector channels 1 and 4 cannot be ordered when the 2319 is attached via the IFA. Details of the IFA are in Chapter 10.

The POWER FRAME for the 3145 contains a motor generator set that converts the 50-Hz or 60-Hz input supply to 400-Hz output for use in the logic and storage. Details are in Chapter 11.

The IBM 3345 MAIN STORAGE FRAME Model 1 or Model 2 is required for systems that exceed 256K bytes of main storage. Also, a 3046 Power Unit (not shown) is required with the 3345.

SYSTEM CONSOLE contains the operator panels, the lights, and the switches used during check-out and maintenance of the system. This console contains roll bars so that you can check the status of different conditions using the same lights. Details concerning the console and its use are in Chapter 5.

ALTER/DISPLAY functions are manually initiated at the console printer-keyboard through an integrated attachment. The printer provides hard copy (printed copy) of the altered and displayed data. Details concerning this integrated attachment are in Chapter 7.

CONSOLE FILE (23FD) through an integrated attachment is used to load control storage with either the 3145 microprogram for customer operations or with the microdiagnostic you use to check out the CPU. The customer may use both sets if he is engaged with Customer Involvement of failures. Details concerning this integrated attachment are covered in Chapter 6.

The CENTRAL PROCESSING UNIT (CPU) contains from 112K to 256K bytes of main storage, plus 32K for control storage.

The CPU contains all the elements necessary to decode and execute the System/370 instruction set and, optionally those in the hardware compatibility features required by the 1401/1440/1460 and 1410/7010 emulator programs.

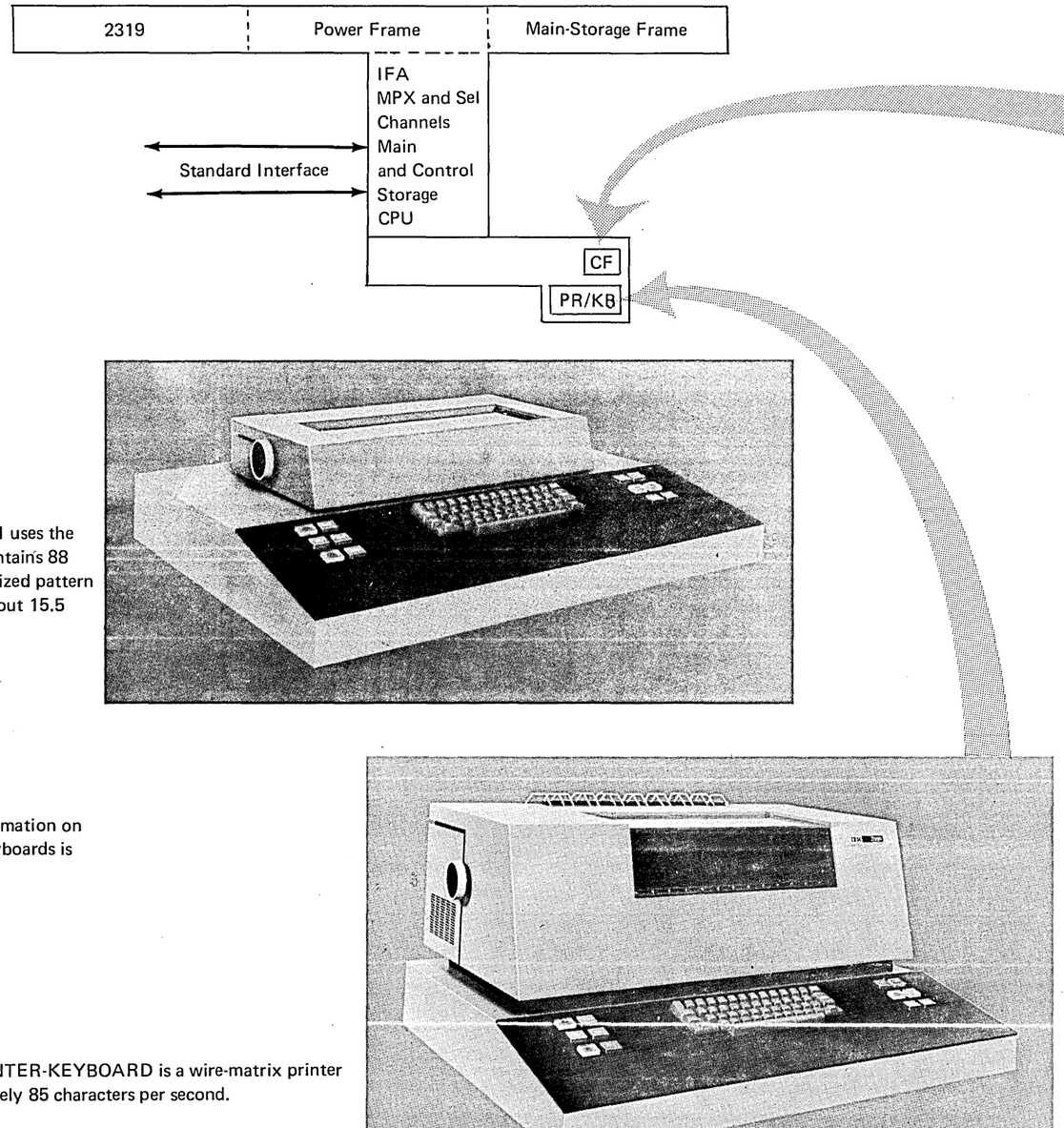
All CPU and channel operations are controlled by the microprogram contained in control storage loaded from the console file.

The CPU and channel operations are controlled by the microprogram contained in control storage loaded from the console file.

The CPU has several cycle times: 202.5, 247.5, 292.5, and 315 nanoseconds. Instruction times are composed from combinations of all of them.

The 3145 is known as a word machine because four bytes (one word) are selected to be operated on each time storage is read out. However, the data path varies in size up to eight bytes wide (doubleword).

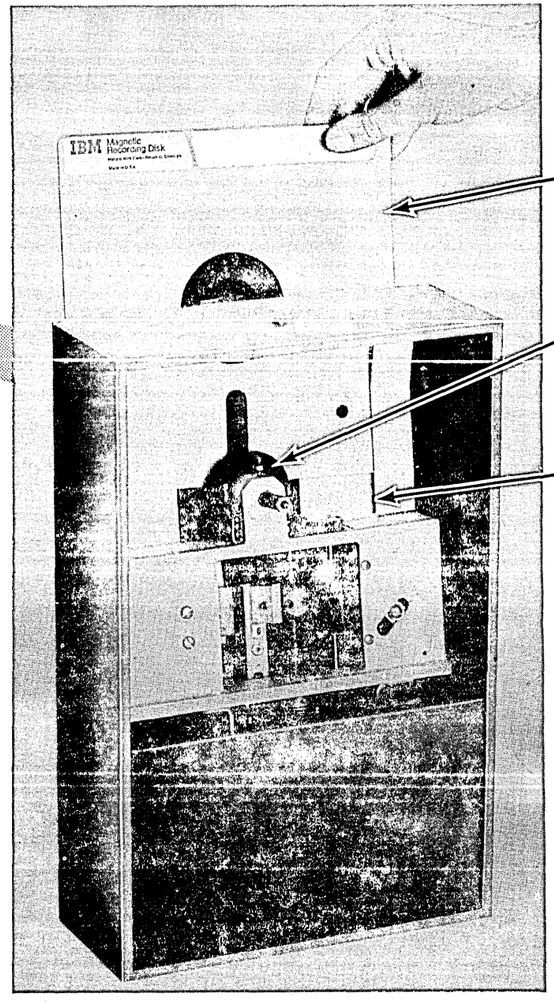
CPU CONFIGURATION



3210 CONSOLE PRINTER-KEYBOARD MODEL 1 uses the Selectric I/O II printing unit. The print element contains 88 characters. The characters are arranged in an optimized pattern to provide faster response. The printing speed is about 15.5 characters per second.

Either Console Printer-Keyboard can be used. Information on integrated attachment for these Console Printer Keyboards is in Chapter 7.

3215 CONSOLE PRINTER-KEYBOARD is a wire-matrix printer and prints approximately 85 characters per second.



CONSOLE FILE (23FD) is the unit that does the Initial Microprogram Load (IMPL). Removable disks are provided for each microprogram to be loaded into control storage.

An IMPL occurs automatically when system power is applied. Also, after power is up, an IMPL can be initiated by using the console-file start key on the console. When the microprogram is loaded, the power to the console file is turned off.

The removable disks shipped with each CPU are:

- 370 Microprogram (two identical disks)
- BAS Basic diagnostic
- EX1 Extended diagnostic
- EX2 Extended diagnostic
- STF System Test File (ASCP)
- SAO IFA Feature Test
- CE Disk

FEATURES

Standard Features

3145 Processing Unit (with main power frame)
112K Bytes minimum main-storage configuration with Error Checking and Correction (ECC)
Optional main-storage sizes available are: 160K, 208K, 256K, 384K, 512K
32K Bytes of control storage (with ECC)
Audible alarm
Byte-multiplexer channel
Byte-oriented operand
Channel retry information
Command retry
Console file
Extended channel logout
Extended external masking
Extended I/O masking
Interval timer
Limited channel logout
Machine-check handling
Microprogram instruction retry
OS/DOS Compatibility
Selector channel 1 if the integrated file adapter (IFA) is not installed or selector channel 2 if IFA is installed
Storage protection (store and fetch)
System/370 commercial instruction set
Time-of-day clock

System/370 commercial instruction set, which includes the System/360 commercial instruction set, modified instruction formats for start I/O and halt I/O, and these enhancement instructions:

Compare logical characters under mask
Compare logical long
Insert characters under mask
Load control
Move long
Set clock
Shift and round decimal
Start I/O fast release (executed as start I/O on the Model 145)
Store channel ID
Store characters under mask
Store clock
Store CPU ID
Store control

see Appendix A
For details of the instructions refer to the System/370 Principle of Operations (GA22-7000).

Automatic correction of all single-bit main-storage errors, and detection of double-bit errors. This is done through the use of a new internal storage code instead of byte parity.

ECC minimizes the effect of the majority of intermittent memory failures that could cause a machine check. It also allows maintenance to be deferred on solid single-bit errors.

Intermittent double-bit errors may also be corrected by ECC. The error-correction hardware reads out the offending storage position multiple times. Thus, if an intermittent error exists, the chances are that during one of the retries one of the bits in error will correct itself. The other bit in error then can be corrected as a single-bit error.

This feature makes it unnecessary to align operands of non-privileged operations on halfword or doubleword boundaries as with current System/360. It affects storage references made by the CPU using RX and RS formats and applies to fixed-point, floating-point, and logical operations.

It does not apply to the following:

Instruction addresses
Branch addresses
Subject of an execute instruction
Any privileged instruction such as:

Load PSW
Set and insert storage key

Channel operations

Significant performance degradation is possible when this feature is used. To assure optimum performance, storage operands should be aligned on integral boundaries, and use of this feature should be reserved for exceptional cases.

Optional Features

Additional byte-multiplexer-channel subchannels (Note 1)
Block-multiplexer channels (up to four) (Note 2)
Channel-to-channel adapter (same as System/360)
Direct control (with external interrupt) same as S/360
Extended precision floating-point feature
Integrated file adapter (IFA) (Note 3)
Selector channels 2, 3, and 4 (without IFA) or selector channel 3 (with IFA) (Note 3)
Word buffer (Note 4)
1401/1460, 1440 Emulation (Note 5)
1401/1460, 1440 and 1410/7010 Emulation (Note 5)
3210 Console Printer-Keyboard Model 1 (Note 6)
3210 Console Printer-Keyboard Model 2 (Note 6)
3215 Console Printer-Keyboard Model 1 (Note 6)
3345 Main Storage Frame Model 1 (Note 7)
3346 Main Storage Frame Model 1 (Note 7)

Note 1: The byte-multiplexer channel has 16 subchannels that address up to 136 I/O devices (eight shared UCWs can address up to 16 devices each; eight unshared UCWs can address one device each). Up to eight control units can be attached. Configurations with 32, 64, 128, or 256 subchannels are available.

Note 2: A block-multiplexer channel can be ordered to modify any selector channel. Block-multiplexer channels cannot be ordered for selector channels 1 and 4 when the integrated file adapter is installed.

Note 3: The integrated file adapter, and selector channels 1 and 4, are mutually exclusive. When IFA is attached, selector channel 2 becomes standard. Selector channel 3 is optional. Each selector channel address up to 256 I/O devices. Up to eight control units can be attached.

Note 4: The word buffer feature is installed on all selector- or block-multiplexer channels, or none. The work buffer feature is not available for the integrated file adapter feature.

Note 5: This is a microprogram-controlled feature that allows 1401/1440/1460 or 1410/7010 object programs to be run on the Model 145 without change.

Emulators will run under DOS or OS. These emulators will run in a multiprogramming environment and allow integrated and System/370 data sets to coexist.

The emulators do not require dedicated devices.

Note 6: The 3210 Model 1 and the 3215 are mutually exclusive: one is required. The 3210 Model 2 can be used as an auxiliary printer-keyboard (except for alter/display functions).

Note 7: Main storage above 256K bytes is contained in the 3345 Main Storage Frame Model 1 (128K additional bytes of 384K bytes total) or the 3346 Main Storage Frame Model 1 (256K additional bytes for 512K total). When either of these units is included, it contains the low-order storage addresses. The 3046 Power Unit is required for the 3345 or 3346 Main Storage Frame.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

MICROPROGRAMS

Details concerning the control word are in Chapter 4.

- All functions performed by the 3145 are controlled by a microprogram.
- Before any processing may begin, the microprogram must be loaded into the control-storage area. J
- The microprogram is loaded into control storage from a disk that is read by the console file.
- This loading process is called Initial Microprogram Program Load (IMPL).
- The microprogram is composed of microroutines of varying sizes, each having a specific task to perform.
- The microprogram handles the processing of the instructions and data that are read into the main-storage area.
- Channel operations and the operations of the integrated devices are also handled by the microprogram.
- Each microroutine is composed of bit-significant control words that handle particular functions that result in the execution of the specified task of the microroutine.

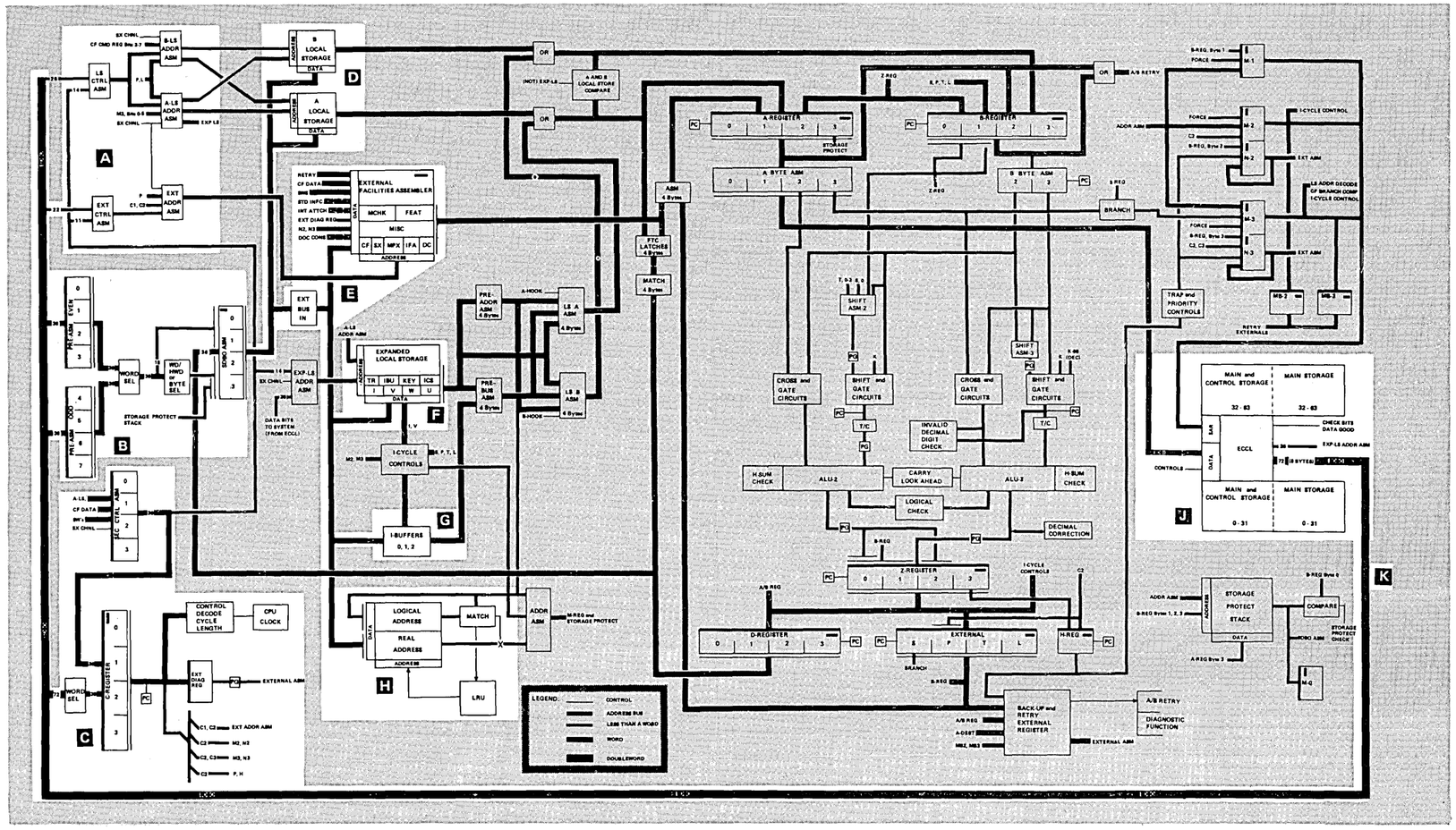
CONTROL WORD READOUT

Before a control word can perform any of its functions, it must be set into the four-byte control register (C-Reg). C The outputs of the C-Reg activate circuitry that causes the execution of specified data-flow functions.

Control words are normally read from control storage and set into the C-Reg. However, control words can be set into the C-Reg directly from the console file, and certain control-word bit combinations may be forced into the C-Reg by circuitry.

Assume that control and main storage have been loaded and that processing has begun;

1. Control words are read out of control storage and gated out on the Storage Data Bus Out (SDBO). K
2. Portions of the control words are gated from the SDBO to either the local storage control assembler A or the external control assembler A. This is done to set up source addresses for these facilities early in the cycle.
3. The control words are gated into the control register (C-Reg) C, where they are decoded. Decoding the control words brings up control and addressing lines that access and execute the programs located in main storage.



INSTRUCTION/DATA READOUT

When a control word performs a read operation on main storage, either instructions or data is accessed. All read operations, for control or main storage, result in a double-word's being accessed from storage.

Assume that a control word is performing a read operation on main storage;

1. The doubleword from main storage is gated out on the SDBO K to the SDBO pre-assembly latches B.
2. The odd or even address word, of the doubleword, is selected and gated to the SDBO assembler.
3. If the word selected is a data word, it is gated to local storage D or some external facility E.
4. If the word is an instruction, it is gated to the I-buffers G, expanded local storage F, and in some cases to the address adjustment circuits H.

CONTROL WORD FUNCTIONS

The control words and their high-level functions are:

Details concerning control words are in Chapter 4.

Branch and Module Switch

Functions:

- Branch
- Module switch
- Destine data to the S, T, or L registers.

Branch Word

Functions:

- Branch
- Module switch (special function)
- Set/reset bits in local storage or external registers

Branch and Link or Return

Functions: Branch and link

- Store S, P, N2, N3 into a link register
- Set P with a value, or, module switch
- Branch

Functions: Return

- Restore S, P, N2, N3
- Reset H-register bits
- Alter the link address in some cases

Word Move

- Move a full word or selected bytes from one local storage/external location to another.
- Branch

Storage Word

Functions:

- Read data from or store data into:
  Local storage
  Main storage
  Control storage
  External registers
  Storage protect stack
- Branch

Arithmetic Word

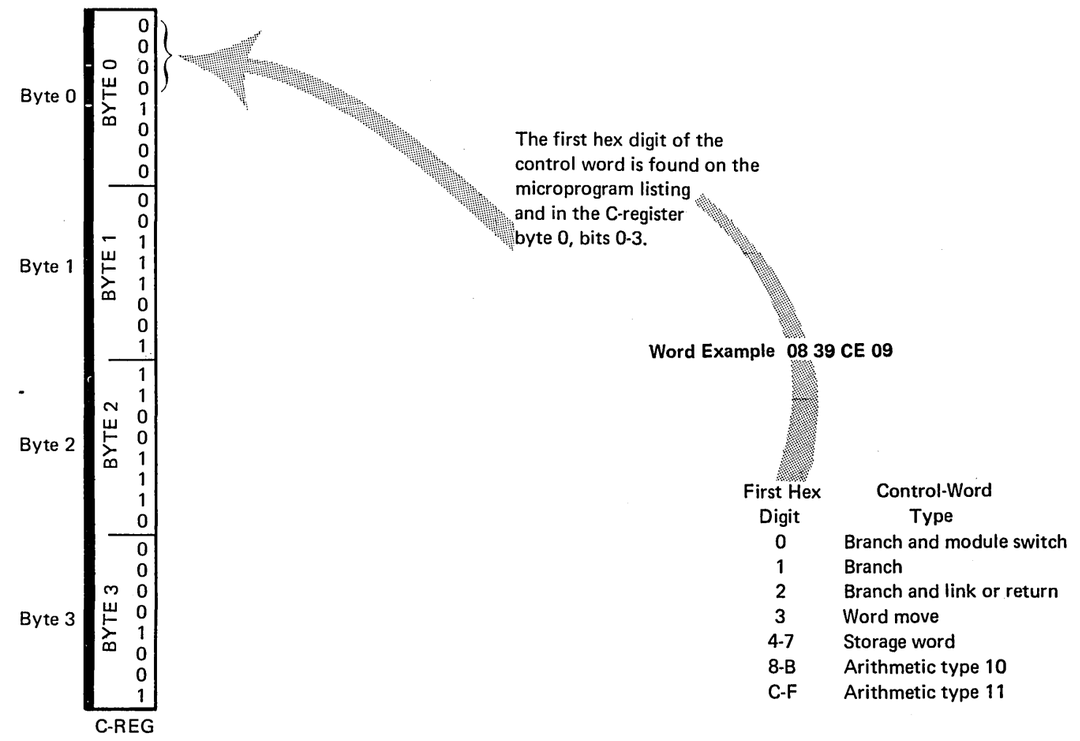
Functions: Type 10:

- Perform a variety of arithmetic and logical operations.
- Operate on fullwords for arithmetic or shifting operations.

Function: Type 11

- Operates on bytes only
- Exclusive OR, or, true ADD only
- A-register input crossing provided

Word-Type Definition



First Hex Digit	Control-Word Type
0	Branch and module switch
1	Branch
2	Branch and link or return
3	Word move
4-7	Storage word
8-B	Arithmetic type 10
C-F	Arithmetic type 11

3145 DATA FLOW

The facing page illustrates the overall data flow for the 3145 CPU. This section provides you with a brief description of the 3145 functional units illustrated in the data flow.

The number of bits entering or leaving a function or register are identified either by the weight (thickness) of the line (see the legend block at the lower center of the overall data flow) or by placing the number of bits in the flow line.

This is a high-level data flow of the 3145 CPU. The general layout is such that you can easily reference from any area of this high-level data flow to the same area of the overall data flow.

Local Storage

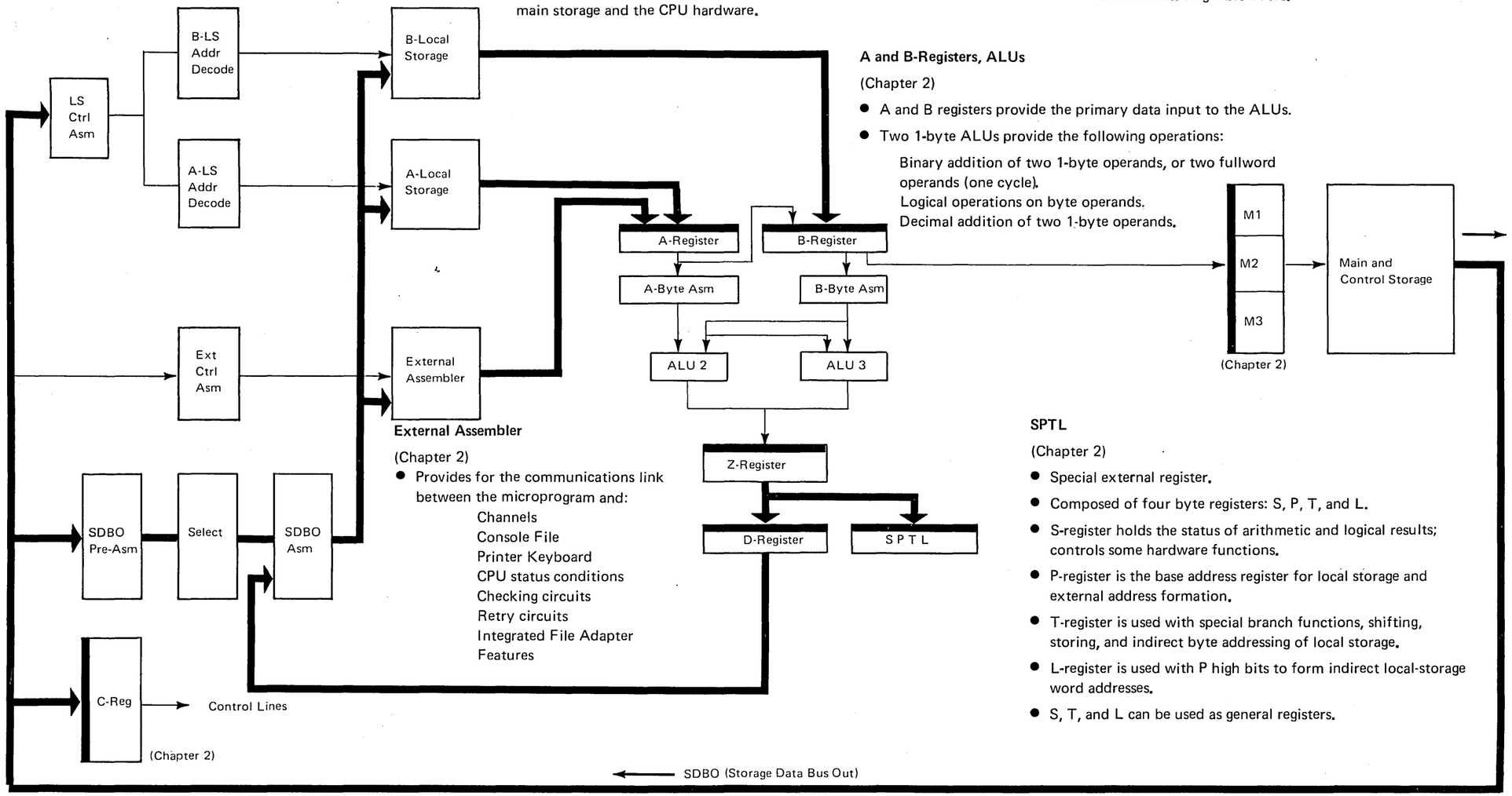
(Chapter 2)

- Local storage A and B are identical monolithic stacks of 64 words each.
- Both stacks contain the same information at the corresponding address.
- The microprogram uses the local storage as a buffer between main storage and the CPU hardware.

Main and Control Storage

(Chapter 3)

- Contains the main-storage area and the control-storage area.
- Control storage is located in the high-address range.
- Monolithic circuitry provides nondestructive readout.
- Contains the ECC feature, which detects double-bit errors and corrects single-bit errors.



A and B-Registers, ALUs

(Chapter 2)

- A and B registers provide the primary data input to the ALUs.
- Two 1-byte ALUs provide the following operations:

  Binary addition of two 1-byte operands, or two fullword operands (one cycle).
  Logical operations on byte operands.
  Decimal addition of two 1-byte operands.

External Assembler

(Chapter 2)

- Provides for the communications link between the microprogram and:

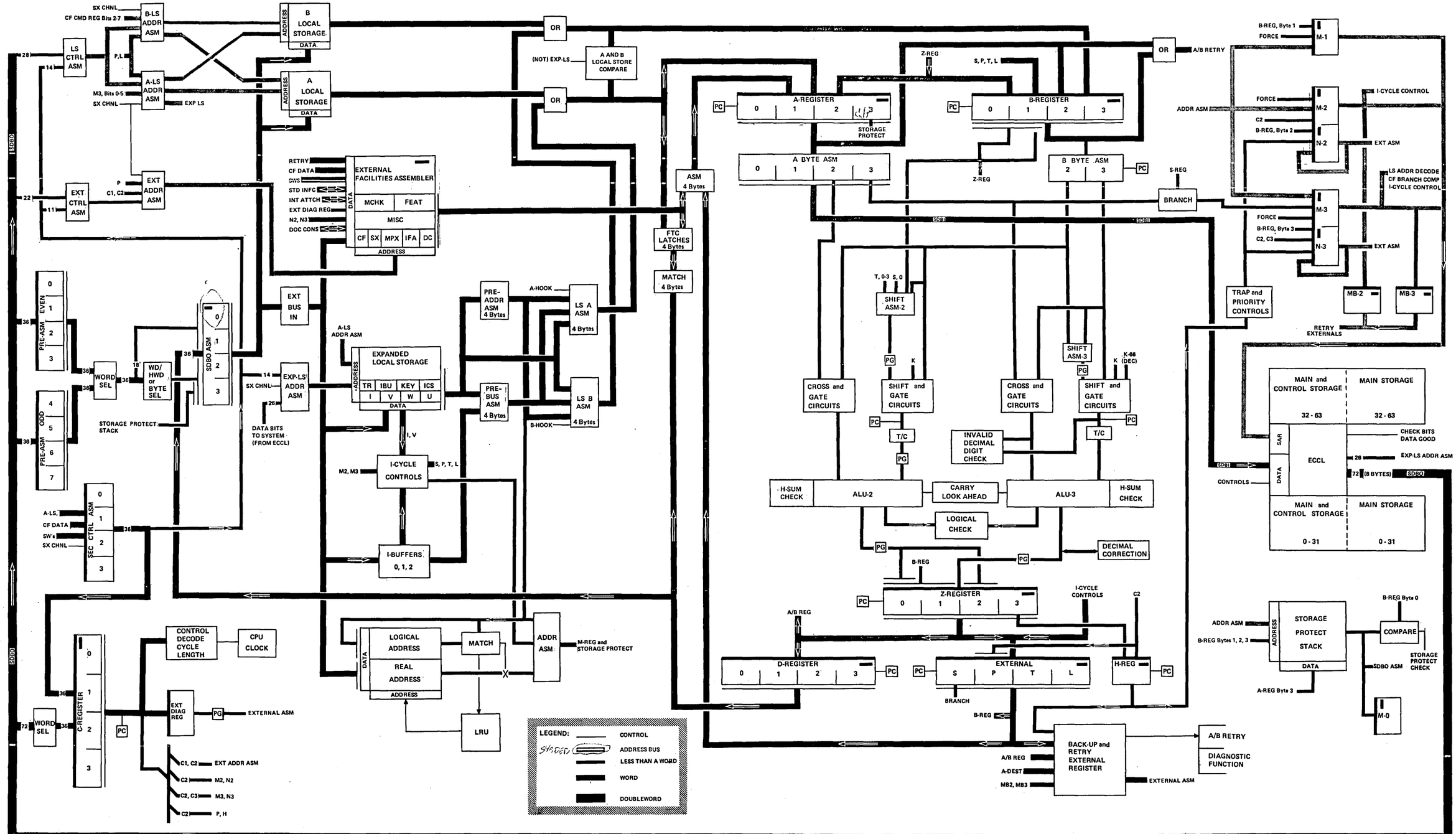
  Channels
  Console File
  Printer Keyboard
  CPU status conditions
  Checking circuits
  Retry circuits
  Integrated File Adapter Features

SPTL

(Chapter 2)

- Special external register.
- Composed of four byte registers: S, P, T, and L.
- S-register holds the status of arithmetic and logical results; controls some hardware functions.
- P-register is the base address register for local storage and external address formation.
- T-register is used with special branch functions, shifting, storing, and indirect byte addressing of local storage.
- L-register is used with P high bits to form indirect local-storage word addresses.
- S, T, and L can be used as general registers.

OVERALL DATA FLOW



STORAGE

Details concerning storage are in Chapter 3.

Storage for the Model 145 is implemented by monolithic technology. It is based on bipolar, semiconductor storage cells with nondestructive read capabilities. Unlike magnetic core storage, the content of storage is lost when power is turned off.

The main advantages of semiconductor storage are:

- price/performance
- reliability and serviceability (a storage card can easily be replaced).

STORAGE STRUCTURE

Monolithic: All 145 storage (local, control, and main) is implemented using monolithic technology.

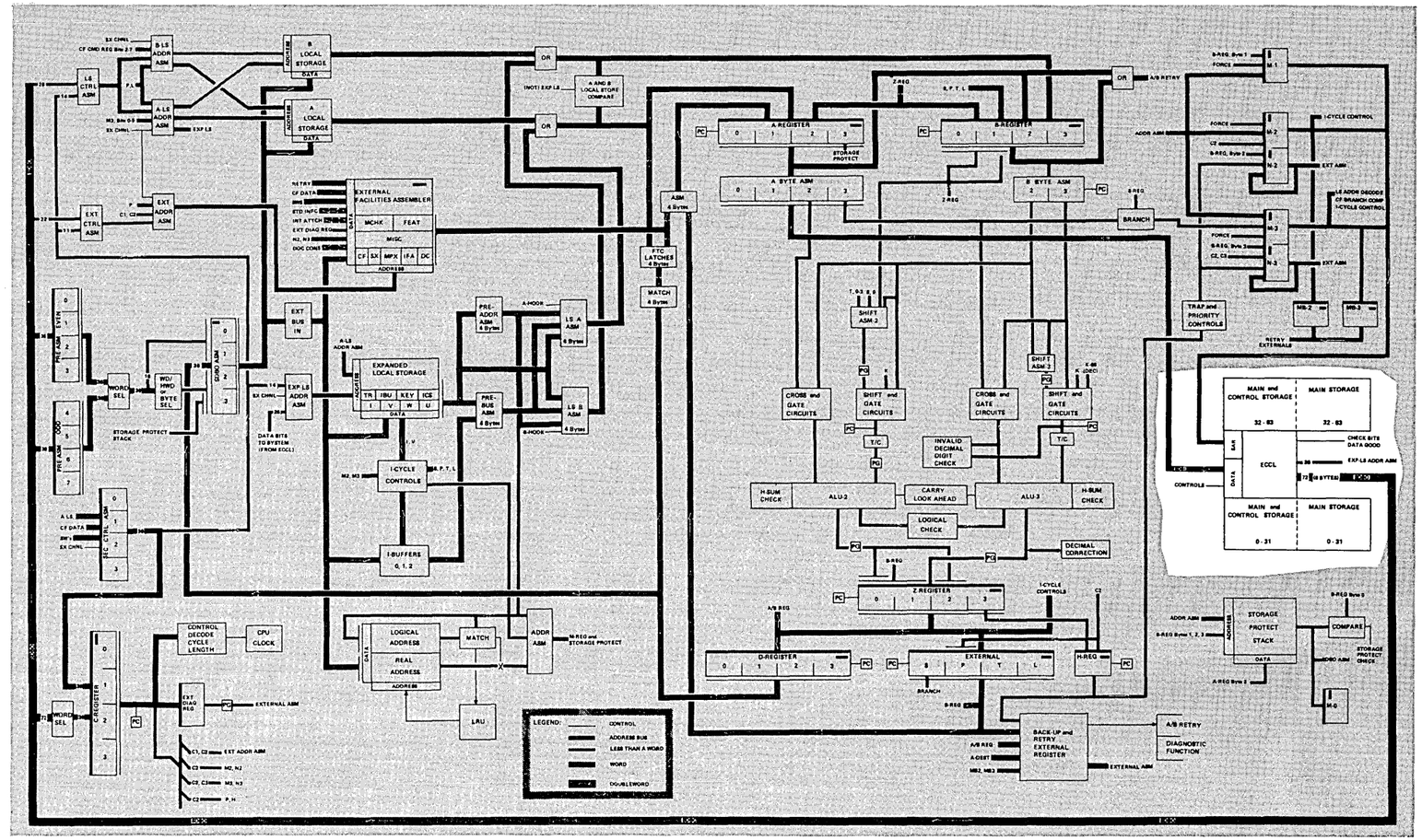
MAIN STORAGE

There are four models of the 145 processor with six storage-size options.

Model	Main Store
FED	112K
GE	160K
GFD	208K
H	256K
H with 3345 Model 1	384K
H with 3345 Model 2	512K

Data-Transfer Times

Storage data path width is eight bytes. The CPU will fetch a doubleword in 540 nanoseconds. It stores one word in 607.5 nanoseconds. (This involves fetching eight bytes, updating four of them, resetting the ECC, and storing back).



CONTROL STORAGE (CS)

The amount of control storage required is dependent upon features in the system.

Basic system microcode contains:

- Standard instructions
- Standard features
- Patch area and routine

and requires 26,000 bytes of CS.

Additional storage required for:

- Byte and block MPX UCWs
- Selector-channel support
- Block MPX feature
- Console support
- Integrated file adapter
- 14XX/7010/DOS emulators
- Floating-point arithmetic
- Direct control

Sample requirements

CS is assigned in the high-order range of available storage.

For example: 112K Model 145

	Amount	Address Range
Main Storage	112K	0-112K
CS	32K	112-144K

Instructions accessing control storage cause address checks; attempting to access a control word from main storage causes a machine check. This is checked by comparing the address to setting of the address check boundary register. The ACB register is part of the external facility.

Reloadable Control Storage (RCS)—Advantages

- Amount of control storage needed to minimized by recording a console-file cartridge for each customer according to the features he orders.
- Engineering changes can be easily effected.
- One storage system (single addressing design, circuits, data flow) allow greater serviceability.
- Functions implemented through RCS can be easily extended.

Speed

Fetch cycle: 540 ns, 8 bytes

RCS Size

32K bytes is the minimum. RCS may be expanded in increments of up to 64K at the expense of main storage.

The location of the RCS/main storage boundary is established by microcode at IMPL time.

M-REGISTER C

Details concerning the items on this page are in Chapter 2 except for Storage Protection, which is in Chapter 3.

- Addresses main and control storages.
- Composed of M1, M2, and M3, which provides a 20-bit (plus three parity bits) storage address.
- M1, M2, and M3 address both main and control storages. Storage is read out on a doubleword boundary and stored on a word boundary.

N-REGISTER C

- Composed of N2 and N3.
- Backup register for control-storage addressing
- N2 is set with the same information as M2 and is changed only when the control word being executed performs a module-switch functions.
- N3 is set with the same information as M3.
- N is not changed when a trap occurs.
- When a trap occurs, the M-register is set to the trap address. The trap routine stores the contents of N (the N-Reg contains the next address that would have been used had the trap not occurred). At the end of the trap routine, M ` and N are restored to their original value so that the control-word sequence may continue as if there had not been any trap.

MB-REGISTER C

- Composed of MB2, MB3.
- Set with the control-word address in M2, and M3 from M2 BFR and M3 BFR.
- When the CPU clock is stopped, MB contains the address of the last word executed.
- Its output is available to the retry and backup circuits as well as the external assembler.

Details concerning the preceding registers are covered in Chapter two.

Details concerning the items on this page are in Chapter 2 except for Storage Protection, which is in Chapter 3.

STORAGE PROTECTION D

The storage-protect unit has a 64 x 8 protection stack that applies to main storage locations (in sequential block of 2,048 bytes) zero through 131,072. Additional stacks are provided in the CPU when main-storage capacity exceeds 131,072 bytes.

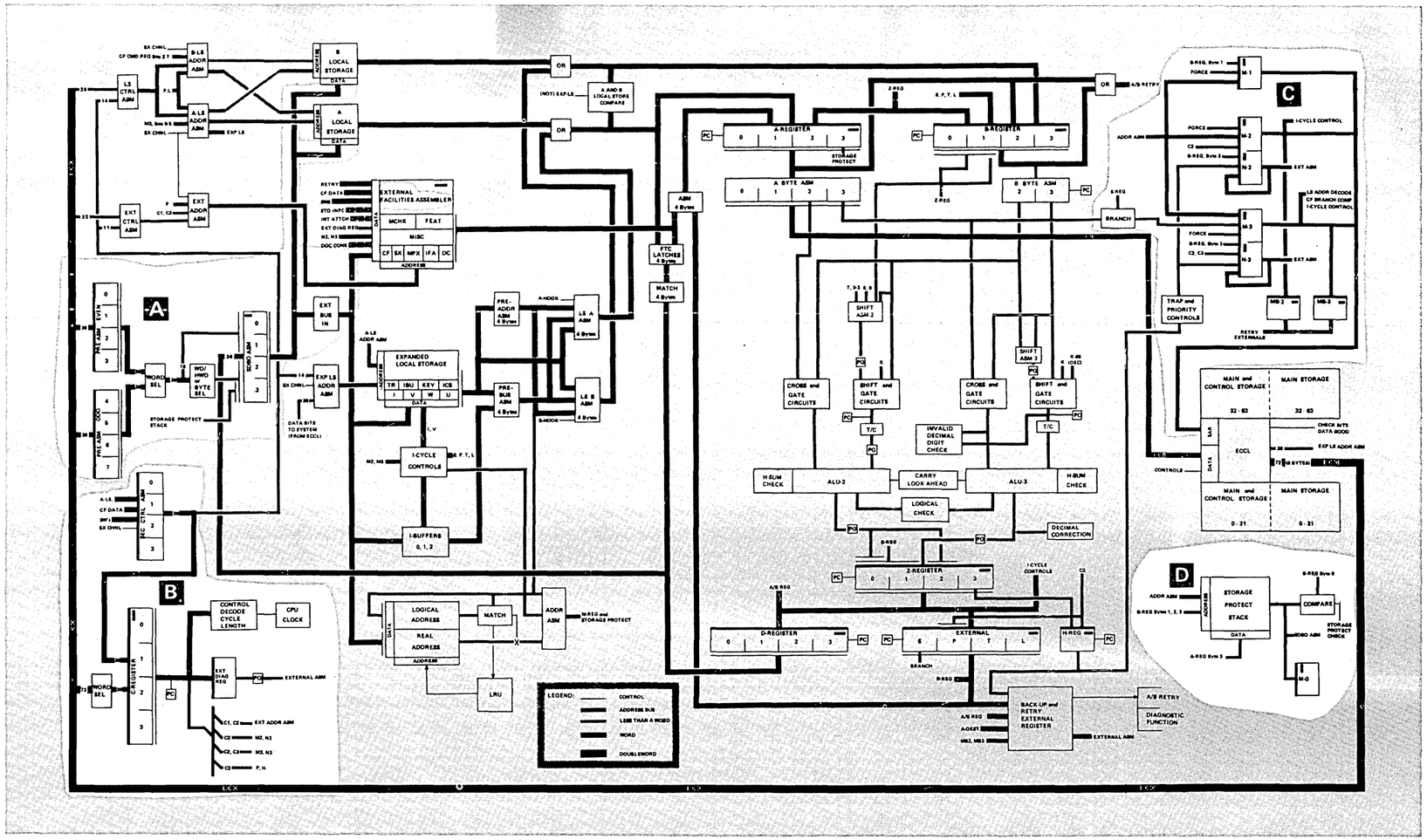
Storage-protect circuits prevent the accessing of protected areas during either store or fetch operations. To protect specific areas of storage, key bits are first stored in the array of the storage-protect circuit by a write-key operation. During a subsequent store or fetch operation, one of the prestored keys is accessed and compared with the key provided by the user. If the keys match, access to data storage is granted; if not, access is denied.

C-REGISTER B

The purpose of the C-register is to decode the control of CPU functions. Once the control word has been read out of control storage and gated to the C-register, it is decoded to determine:

- Word type
- CPU function
- CPU clock cycle and length

The C-register is set through the secondary control assembler during certain operations; for example, manually setting a control word from the switches on the console.



STORAGE DATA BUS-OUT ASSEMBLER A

- The Storage Data Bus-Out (SDBO) preassembler receives a doubleword of data from main storage.
- The output of the SDBO preassembler is gated to provide word, halfword, or byte selection.
- The SDBO assembler receives inputs from the SDBO pre-assembler, the storage-protect stack and the D-register. It provides an output that is used as data for external bus-in (EBI) and local storage.

A- AND B-REGISTERS A

Details concerning the items on this page are in Chapter 2.

The A- and B-registers, each with fullword capacity, provide the primary data inputs to the ALUs.

The B-register also feeds the M-register during address setup:

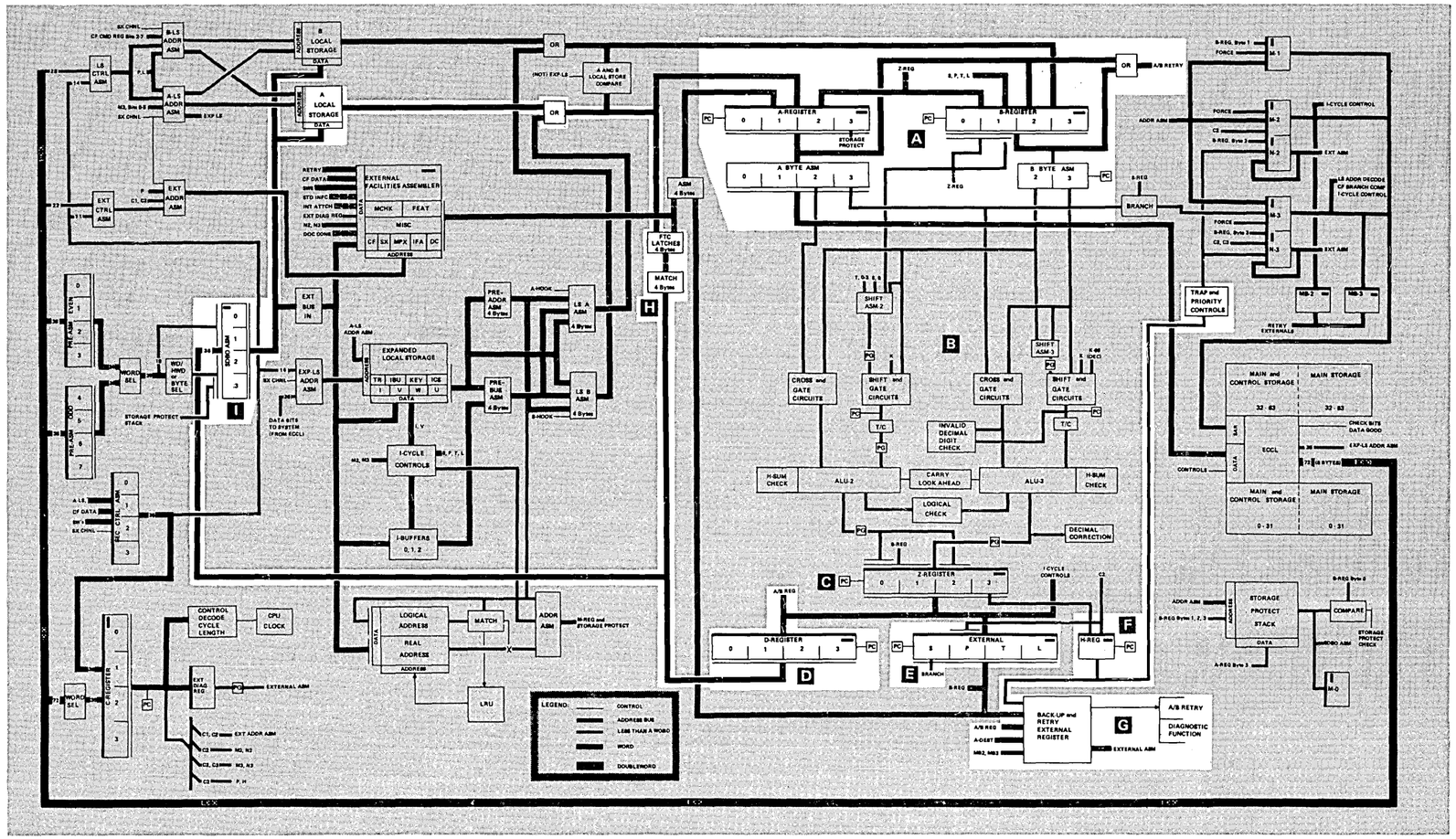
- In the first cycle of a storage word, or
- During a return function in which the return address is taken from local storage or an external facility.

ARITHMETIC AND LOGIC UNITS (ALUs) B

Two one-byte ALUs are provided: ALU2 and ALU3. The following operations can be performed by the ALUs in one CPU cycle.

- Binary addition, true or complement, of up to two fullword operands. Two halfword additions are performed to achieve the fullword add. Binary halfword addition is achieved by inputing the two low-order operand bytes of each halfword into ALU3, and two high-order operand bytes of each half-word into ALU2.
- Logical operation on two 1-byte operands. The operation can be AND, OR, or Exclusive OR.
- One-byte packed-decimal addition (true or complement).
- Operations and microprogram symbols are:

Symbol	*Operation*
,A,	AND
,OR,	OR
,OE,	Exclusive OR
+	True ADD
-	Complement ADD
,D+-,	Decimal ADD
+-	Binary ADD
,A-,	Complement AND



Z-REGISTER C

ALU results are set into the four-byte Z-register. The ALU result data can then be routed from Z to:

- The D register (normal gating).
- The S, P, T, or L registers.
- The A- or B-registers.

Also, Z-register data (that is, ALU result) is tested, if so specified in the control word being executed, to set/reset S-register bits.

D-REGISTER D

The D-register is used as an interim destination for data to be routed to external facilities or local storage. The data leaves the D-register on the following control-word cycle.

SPTL--SPECIAL EXTERNAL WORD E

- Addressed directly by control-word bits.
- Has special data patch to A- and B-register inputs.
- Only external that can be used as a B-source.
- Only facility that is destined in the same cycle.
- Composed of four one-byte registers: S, P, T, and L.

S-Register: holds the status of arithmetic and logical results; controls some arithmetic functions.

P-Register: base address register for local storage and external addressing.

T-Register: used in conjunction with special branch functions, shifting, storing, and indirect-byte addressing.

L-Register: used in conjunction with P-high bits to form indirect local-storage addresses.

H-REGISTER F

The setting of the latches in the H-register are used to determine the priority of traps. The bits are set during trap 2 cycle and prevent traps of lower priority from occurring. Detailed information is in Chapter 12.

BACKUP AND RETRY EXTERNAL REGISTER G

To allow recovery from certain kinds of errors, hardware registers (externals) are provided. The backup registers are set to the current cycle setting of the prime register; the retry registers are set to the cycle prior to the one currently being executed.

FLUSH-THROUGH CHECK H

Data routed to local storage, as the result of some control-word operation, other than a storage word read, is gated from the D-register through the SDBO assembler I to local storage. The data that is stored in local storage A is set into the Flush-Through Check (FTC) latches and is matched to the data routed from the D-register. If the match is unequal, an error condition is set. The same check is made on information routed to the external facilities from the D-register.

LOCAL STORAGE A

Detail concerning the items on this page are found in Chapter 2.

- Local storage A and B are identical monolithic stacks of 64 words each.
- Both stacks contain the same information at the corresponding address. This enables checking to ensure that data being operated on is correct.
- The microprogram uses the local-storage area as a buffer between main storage and the CPU hardware.
- Addresses are formed with combinations of bits from the control word, P-register, L-register, T-register, forced, and console file command register.
- Access time is 24 nanoseconds.

EXTERNAL FACILITIES B

- External facilities are composed of registers, buses, status lines, and other circuitry that form the communications line between the microprogram and:
  - Channels
  - Console file
  - Documentary console
  - Checking facilities
  - Retry circuits
  - Integrated file adapter
  - Features
- Addresses are formed from: control words, console-file data, selector-channel circuits, console switches, retry information, and local-storage address data.
- Data from the externals enters the data flow through the external assembler to the A-Reg.
- Data to the externals is gated through the SDBO assembler on the external bus in (EBI).

EXPANDED LOCAL STORAGE C

The expanded local storage (EXPLS) registers are hardware registers addressed as though they are local-storage (LS) registers. EXPLS operates similar to the external facilities; however, the output is routed to the A- and B-registers similar to LS.

I-CYCLE CONTROLS D

Hardware used to improve the CPU performance for System/360 and System/370 instructions by reducing the time during I-Phase of instruction processing.

I-BUFFERS E

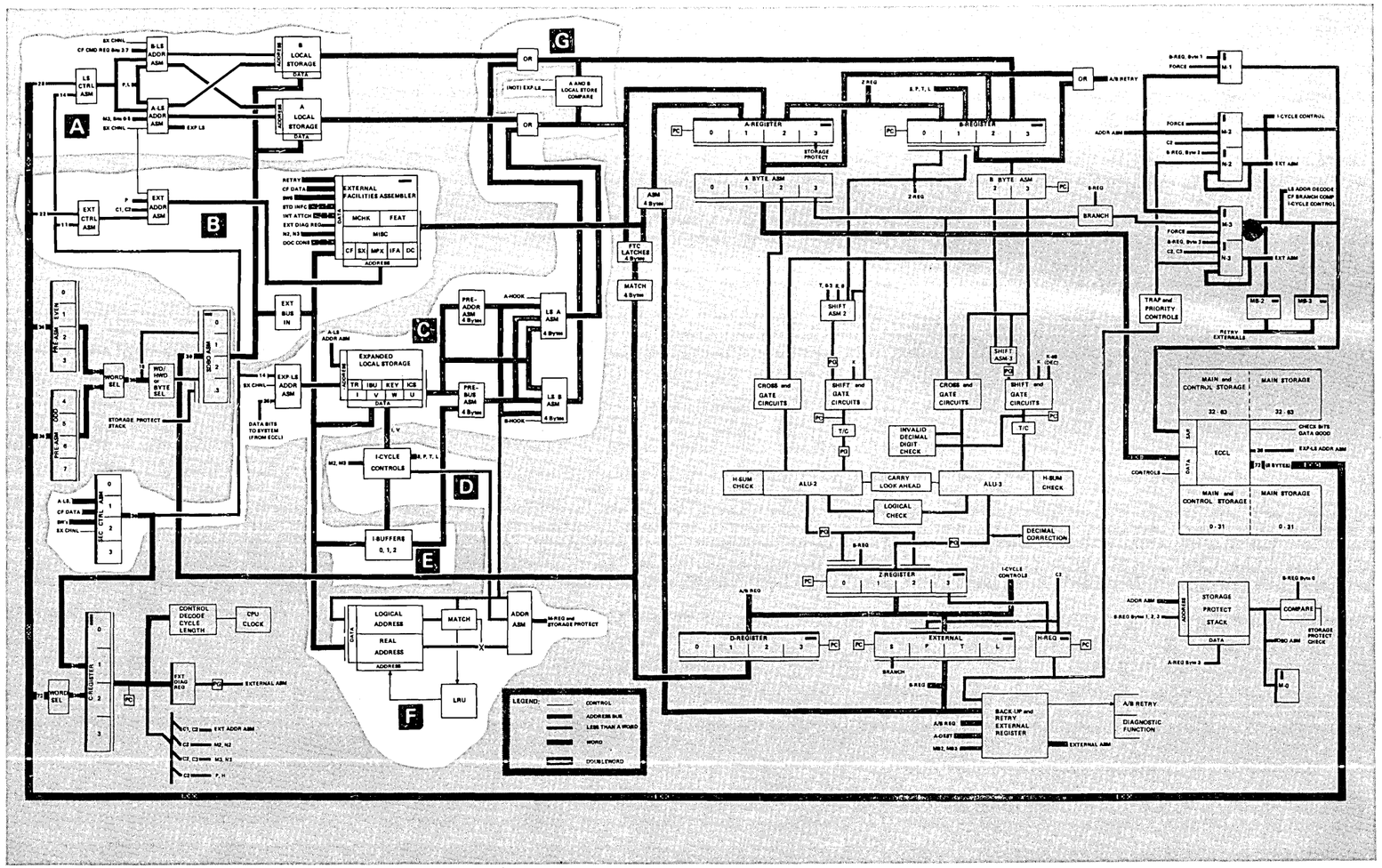
The I-buffers consist of three 1-word registers and are used to hold the present instruction plus the next doubleword of the instruction stream in most cases.

ADDRESS ADJUST F

The address-adjust logic is used by the OS/DOS compatibility feature. This feature provides the necessary logic for execution of a DOS supervisor and DOS programs under control of the OS supervisor in any main-storage location.

A- AND B-LOCAL STORE COMPARE G

Data stored in local storage is located at the corresponding address in both local-storage stacks. The data is read from both stacks and compared. If the data does not compare, an error condition is set. Note that the output of expanded local storage is routed along the same path but is not compared.



ERROR HANDLING

If application-program errors occur (such as illogical action requests), the operating system attempts to handle the exception and provide any necessary operator messages.

If a failure occurs within the CPU or an I/O unit, provisions are made to retry the failing operation. Error-logout facilities are incorporated into the system to record any such failures. (This is in addition to any provisions made by the operating system for error retry and error logging.)

Microprogram instruction retry, limited and extended channel logout, storage validation (Error Checking and Correction--ECC) for program and control storage, and other error-detection and error-handling provisions are standard.

MICROPROGRAM INSTRUCTION RETRY

The ability to recover from most intermittent failures is provided by retry techniques. CPU retry is done by microprogram routines that save the source data before it is altered by the operation.

When an error is detected, a microprogram routine returns the CPU to the beginning of the operation (or to a point during the operation that was executed correctly), and the operation is repeated. For a detailed description of microprogram instruction retry, refer to "Chapter 12, Error Handling".

ERROR CHECKING AND CORRECTION (ECC)

Error checking and correction circuitry for program and control storage automatically corrects single-bit errors. Automatic detection of double-bit errors is also provided. For a detailed description of ECC circuits, refer to "Chapter 3, Main and Control Storages". For a description of handling ECC errors, refer to "Chapter 12, Error Handling".

CHANNEL RETRY

This feature has been implemented to ensure that most failing channel operations can be retried by error-handling routines. Both a limited and an extended channel logout are implemented. When a channel error or a CPU error associated with a channel operation occurs, the channel status word (CSW) and an extended channel status word (ECSW) are stored in the fixed lower storage area during the I/O interrupt. The ECSW or limited channel logout data provides additional, more exacting status information about the channel failure. This data is formatted by the channel check handler routine and passed to a device-dependent error-recovery routine to be used in the retry of the failing I/O operation. The ECSW contains information as to:

- Which unit detected the error
- Which unit caused the error
- Successful retries
- Channel retries
- Validity flags
- Retry code--how far has the instruction progressed in execution

COMMAND RETRY

Command retry is a control-unit-initiated procedure between the channel and the control unit. (Not all control units have this capability.) No I/O interruption is required. The number of retries is device-dependent.

COMPATIBILITY BETWEEN THE 3145, OTHER SYSTEM/370 MODELS, AND SYSTEM/360

Within the storage capacity, internal and input/output channel processing rates, and type of input/output devices that can be attached, compatibility is maintained with other System/370 and System/360 models, with the following exceptions;

- Programs using machine-dependent data (for example, machine logouts).
- Programs using the ASCII bit (PSW bit 12).
- Programs that depend upon features or I/O devices that are not implemented on this system (such as special instructions for the System/360 Model 44).
- Programs that depend upon validity of data after the system power has been turned off and restored.
- Programs written for other System/370 or System/360 models that contain the following conditions or requirements should be evaluated on an individual basis to ensure proper operation.
- Time-dependent programs.
- Programs written to cause deliberate program checks.
- Programs that use storage locations between address 128 (decimal) and 704 (decimal) after a diagnostic logout into program storage. However, such programs may be executed:
  a. if the check-control switch is set to STOP AFTER LOG position. In this case, processing stops after the diagnostic logout into main storage takes place.
  b. if main-storage locations that are overlaid by the diagnostic logout are restored with the program requirements before an IPL and program restart.

Any attempt to continue processing after a diagnostic logout to main storage without restoring the program information to the logout area will have unpredictable results.

The 705-byte extent (the permanently assigned main-storage locations) can be reduced to 512 bytes by moving the 192 bytes (between locations 512 and 704) into another main-storage area.

CONTROL REGISTERS

The control registers may be thought of as extensions to the program status word (PSW). These control registers are part of some of the additional or expanded functions.

Masking of the timer interrupt and external interrupt in control register 0.

Machine-check subclasses masked through bits set in control register 14.

The pointer to an extended CPU logout area in control register 15.

Details of the control register used by the 3145 are in Chapter 12.

PROGRAM STATUS WORD CHANGES

Bit 7: External Mask bit is now a summary bit with control register one containing the individual mask bits.

Bit 12: is now reserved and must be zero. ASCII code is removed.

Bit 6: is the mask bit for channels 6 and over.

Bit 13 Hard Stop Bit: is now a summary bit. Control register 14 contains mask bits for subclasses of machine checks.

STANDARD INTERFACE

The standard interface for System/370 has all the lines used on the System/360 standard interface, plus several additional lines. The additional lines used by the 3145 are identified on this page. Detail explanations of how these lines are used are in Chapter 8.

DATA-IN

During read and sense operations, Data-In rises when data is available on Bus-In. During write and control operations, Data-In indicates that the control unit is ready to receive data.

Data-In indicates to the channel that data on Bus-Out was accepted by the control unit or that the control unit provided the requested data on Bus-In.

Data-In is effective with selector/block-multiplexer channels only.

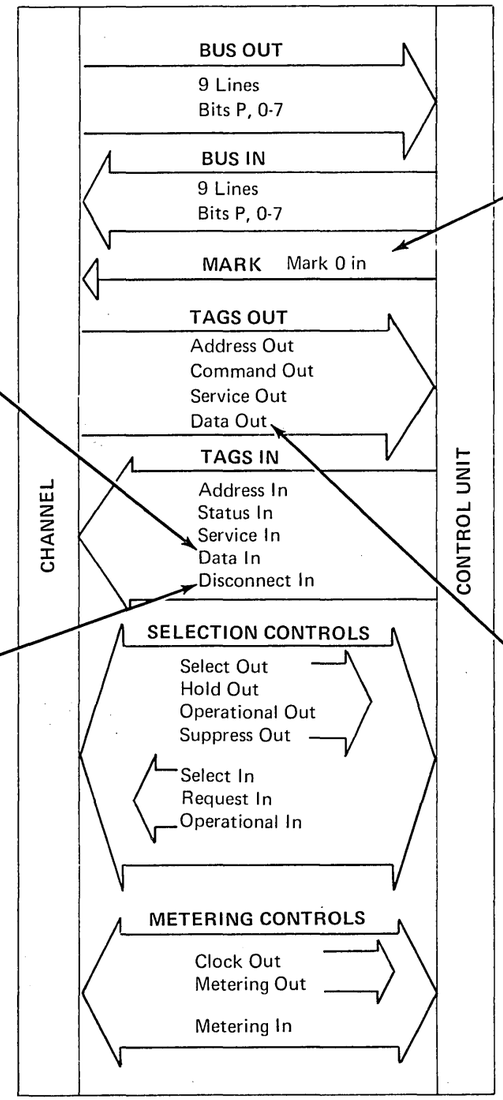
DISCONNECT-IN

Disconnect-In provides control units with the ability to alert the system of a malfunction that is preventing the control unit from signaling properly over the I/O interface. An example of this condition may be a microcoded control unit communicating with the channel at the time a read-only storage (ROS) error is detected. Such a control unit may be unable to complete an interface sequence properly.

'Disconnect in' can be raised by a control unit, only when it is connected to the channel (that is, it has 'operational in' up). When 'disconnect in' is used during a polling sequence, it has progressed at least to the point where 'address in' has been raised with the unit address on 'bus in' before raising 'disconnect in'.

The channel in response to 'disconnect in' performs a selective reset.

This line allows the I/O error alert feature to operate.



MARK 0-IN

When the command being executed encounters a condition requiring retry, the control unit indicates it by raising 'mark 0 in' and 'status in' while presenting 'unit check' and 'status modifier' ('retry status') together with 'channel end' (meaning that the control unit or the device is not yet ready to retry the command), or with 'channel end' and 'device end' (meaning that the control unit and device are prepared for immediate retry of the command). 'Device end', if not presented with 'channel end', is presented later, when the control unit is ready to retry the command.

The channel acknowledges the occurrence of command retry by accepting the status byte containing retry status and indicating chaining. If 'device end' accompanies 'channel end', 'mark 0 in', and retry status, the channel immediately initiates a normal, chained initial-selection sequence, reissuing the previous command. If only 'channel end' and 'mark 0 in' accompany the retry status, the retry is not immediately performed.

Rather, when the 'device end' or 'device end' with 'status modifier' is presented to the channel, it is accepted with chaining indicated and a normal reselection occurs to reissue the previous command; or, in the case of 'device end' with 'status modifier', the CCW following the previous command.

A channel indicates refusal to perform a command retry by accepting the status byte containing retry status without indicating chaining or by stacking the status byte. The stacked byte is treated as any stacked status.

DATA-OUT

Data-out is the response to data-in.

Data-out indicates to a control unit that data on bus-in was accepted by the channel or that the channel provided the requested data on bus-out.

Data-out is effective with selector/block-multiplexer channels only.

3145 CHANNELS--GENERAL DESCRIPTION

Detail information is found in Chapter 8.

The Model 145 has two types of channels available:

- Byte Multiplexer
- Selector
  The selector channel optionally may have the block-multiplexer feature attached.

Channels on the Model 145 are integrated in the CPU and share CPU cycles for I/O operations.

STANDARD FEATURES

- Byte multiplexer channel
- Selector channel 1 (2 if the IFA is present)
- Channel retry

OPTIONAL FEATURES

- Selector channels 2-4 (selector channel 3 if IFA is present).
- Block-multiplexer feature (no charge).
- Integrated File Adapter for 2319 DSF. (Displaces Channels 1 and 4.)
- Channel-to-Channel Adapter.

BYTE-MULTIPLEXER CHANNEL

Functionally equivalent to the System/360 multiplexer channel.

Data transfer is on a byte basis only.

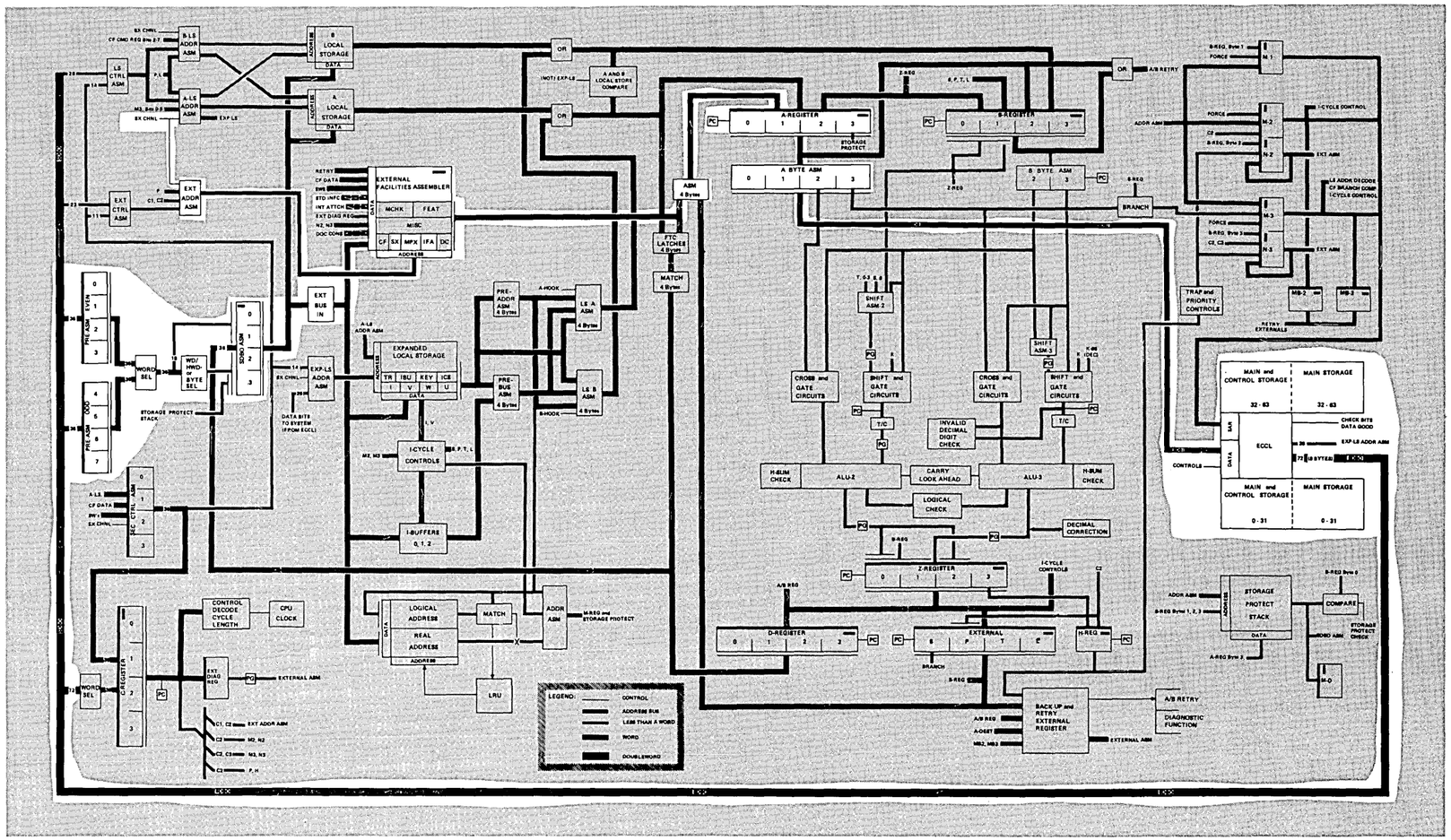
UCWs (Unit Control Words) are provided for subchannels in control storage. Each UCW is contained in four words (16 bytes).

- UCWs provide a place to store channel register data between data transfers, thus allowing multiplexing of data.
- A maximum of 256 subchannels is available on the Model 145.
- 16 UCWs are standard on the Model 145.
  A shared UCW can be shared by up to 16 devices, of which only one can operate at a time.

  A non-shared UCW can be used by one device only.

  Thus, with 16 UCWs, if 8 are shared and 8 non-shared, a total of 136 I/O devices can be attached.

  Up to eight control units can be attached per channel.
- Configurations of 32, 64, 128, or 256 subchannels are available. The number of subchannels must be specified so that the proper amount of control storage may be allocated and written on the console file.



Data Rates

- Aggregate data rate in byte mode is 50 kb. Note that the selector channels and IFA (Integrated File Adapter) can interfere with the byte multiplexer channel.
- Burst mode data rate is 180 kb.

SELECTOR CHANNELS

The 3145 has one selector channel as standard. Up to three more may be attached as an optional feature. DASD devices without command retry feature should not be attached to channel 4.

- Functionally equivalent to the System/360 selector channel.
- If the IFA (Integrated File Adapter) is installed, only channels 2 and 3 may be installed.

Data Transfer

Data is transferred one byte at a time unless the optional word buffer feature is installed on the channel.

- A four-byte buffer is provided for each channel if the feature is installed.
- The word buffer feature allows fewer accesses to main storage to be made while transferring data from the selector channels and increases channel data rates.
- A one-byte operation requires 585 nanoseconds; a one-byte fetch operation requires 517.5 nanoseconds. The word buffer feature allows four bytes to be transferred rather than one.
- The word buffer is required if the 2305 is to be attached. It is recommended if the 3330 is to be attached.

Data Rates

- Single Channel
  without word buffer .9 mb
  with word buffer 1.5 mb
- Aggregate data rate
  without word buffer 1.9 mb
  with word buffer 5.3 mb

BLOCK MULTIPLEXER FEATURE

The block multiplexer feature may be installed on the selector channels as an optional feature at no charge.

It is required if the 3330 and 2305 are to be attached.

The selector channel operates as a block-multiplexer channel when the mode bit in the control register is set ON.

A maximum of 512 UCWs is provided when the block-multiplexer feature is installed. These provide a pool that may be assigned to devices. Each UCW is contained in two words.

- UCWs may be shared or non-shared
- UCWs are contained in control storage
- Shared UCWs must be determined and assisgned device addresses.
- Non-shared UCWs are dynamically assigned to devices at start I/O time in blocks of eight. If no UCWs are available, a not-operational-condition code is returned.
- UCWs are provided in control storage in increments of 16 UCWs, each increment containing two groups of 8 UCWs.

Upon receipt of a signal from the previously disconnected device indicating that it is ready to use the channel data path again, the channel restarts the appropriate channel program.

The process is repeated for all active devices until each one is completely serviced.

If a channel is busy when a device reconnection is requested, the device must wait until the channel becomes available.

To facilitate channel scheduling, a new *channel available interrupt* has been defined for the block-multiplexer channel.

At disconnect time for a channel program, the channel is available for the resumption of an uncompleted channel program or the initiation of a new one. A channel available interrupt occurs at disconnect time if any I/O command was issued previously while the channel was busy.

Block Multiplexing

Block multiplexing allows the channel to disconnect a device at channel-end time. During the interval between channel end and device end, another device on the channel could be started or could complete data transfer for a previously started operation. Thus, a block-multiplexer channel can multiplex blocks of data from different devices giving a much greater effective data rate than a selector channel.

Block multiplexing occurs only if a control unit presents channel end and not device end during command chaining, and the channel is in block MPX mode.

The block-multiplexer channel can operate as regular selector channel so that existing System/360 channel programs can run unchanged.

Block-Multiplexer Operation

Because the channel is busy only during the time when data is actually being transferred, several channel programs can be executed concurrently by sharing the channel hardware. This could be called "Channel Multiprogramming."

The sequence of events in channel multiprogramming is:

A channel program controlling a device is started by the channel and remains active until the device signals that it has no need for the channel path at that stage of its operation.

The channel disconnects the channel program and stores all information needed to restart the program in UCWs that are in control storage.

The channel can start another channel program at this point if one is ready.

INTEGRATED FILE ADAPTER

- The Integrated File Adapter (IFA) feature connects from three to eight IBM 2314-type disk drives to the System/370 Model 145.
- The IFA feature is assigned exclusive use of the channel-1 address and functions as both channel and control unit for the files.
- Data transfer takes place one byte at a time on a share-cycle basis the same as with selector channels.
- The initiation of operations and the initiation of each step of the file sequence requires the use of the CPU controls and microprogramming.

The primary control for the IFA is contained in the CPU, where it can make use of the CPU hardware and microprogram for operation. The IFA adds hardware for control during periods of reading and writing the file records and for other periods of control when the microprogram is most required. The 2319-A1 contains the read clocking circuits and the storage module switching for up to eight files. The disk storage drives operate in the same manner as the 2314 system connected to a selector channel. The record format is identical and the operation requires the same programming systems.

The IFA control-unit operation is initiated as a channel operation using the I/O instructions and channel commands. Primary control information for the file operation is stored in the CPU. Operating commands are processed by micro-programs stored in the CPU. The microprogram starts the operation by developing the appropriate information for a portion of the sequence and issues a mini-op to the control-unit hardware. While the hardware is performing the mini-op, the microprogram stores a link address and returns to CPU operation. When the hardware finishes that portion of the sequence, it requests a trap to return to the microprogram link address to continue the operation. An operation may require several of these transfers between the microprogram and hardware to complete a command.

Data movement during the hardware control period is performed by requesting a selector-channel share cycle for each byte. The CPU or other channel operations have use of the CPU hardware and microprogram for other operations when time is not required by the IFA controls either for setup or data transfer. The file operation should never overrun during normal operation because of the assigned priorities.

Details on the IFA are in Chapter 10.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

CHAPTER 2. FUNCTIONAL UNITS

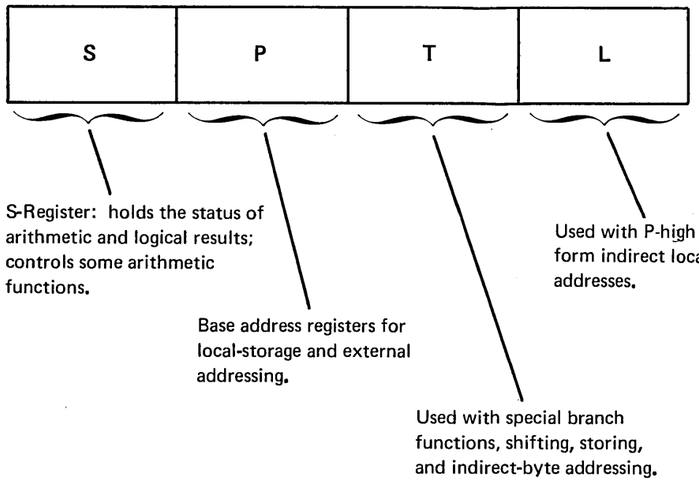
REMEMBER

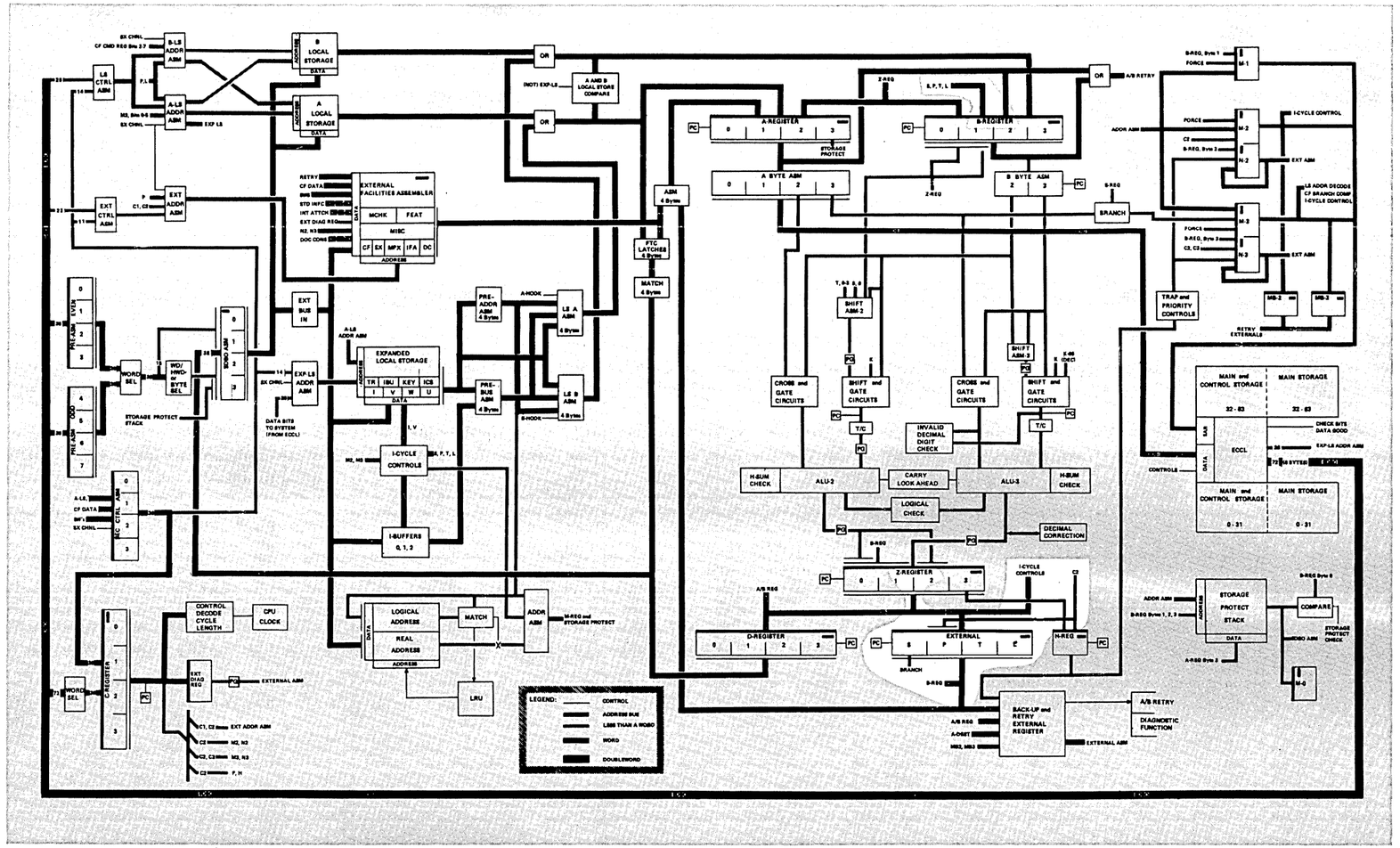
There is a Reader's Comment Form at the back of this publication.

SPTL

The SPTL word is a special external register that has a word address of 04.

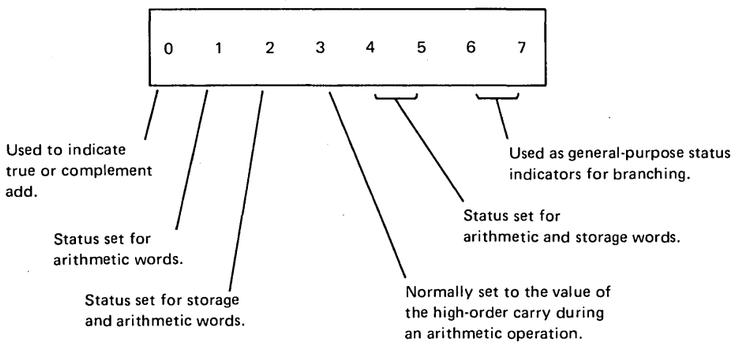
- Addressed directly by control-word bits.
- Has special data path to A- and B-register inputs.
- Only external that can be used as a B-source.
- Only facility (other than the H-Reg) that is destined in the same cycle.
- Composed of four byte registers: S, P, T, and L.





NOTE: For details concerning how SPTL affects control-word operations and addressing, see Chapter 4.

S-REGISTER



- May be used as a working register

S-register bits are used primarily to indicate the results of ALU operations or to specify how certain ALU operations are to be performed.

Branch fields in the control words can be set to specify branch testing of S-register bits. The results of the branch testing are used to determine a portion of the next control-word address. Therefore, a control-word sequence can be modified according to ALU results. For example, the following is frequently used in microprogram routines.

1. A control word specifying an ALU operation calls out set/reset of specific S-register bits, depending upon the ALU result.
2. A subsequent control word specifies branching on the same S-register bits.

Note: If the S-register is set with a control word and this same control word is branching on S-bits, the branch test is made on a previous S-register setting and not on the result of the current control-word operation.

3. The control word branched-to continues the microprogram sequence required by the ALU result.

The S-register can also be used as a general-purpose data register. For example, a control word can cause the S-register to be loaded with a byte for use in operations with subsequent control-word operations such that none of the following descriptions apply. Such use of the S-register is determined by the microprogrammer. The following listed S-register bit functions are not automatically performed. Any function must be explicitly specified in the control word for which the function is desired.

Note: There are duplicate S-registers designated S-register A- and B- respectively. The data from S-register B is displayable, and is gated to the A-register and B-register.

S0

The setting of S0 determines whether a true or complement add is to be performed in the ALU(s) when either a binary or decimal add is specified by the control word.

S0 Value	*Specifies*
0	True add
1	Complement add

The true/complement circuitry affects only the B-input (from the B-register or the K-assembler) to the ALU(s). The A-register input is always presented to the ALU(s) in true form, regardless of the value of S0.

In arithmetic word (type 10) shift operations, the value of S0 is shifted into the four high bits of the result word (shifted right) when the shift field of the arithmetic word specifies (SR4, S0). The bits (4-7) shifted out of the source-word byte 3 are set into T-register bits 0 through 3 respectively.

S1

1. In decimal operations, S1 is set to 1 if an invalid decimal digit is detected in the A or B inputs to the ALU. S1 is not changed if the decimal digits are all valid. An invalid decimal digit is greater than 1001 (binary). The test on the inputs is made before the original decimal data is binarily added in ALU3; no such test is made on decimal data when it is being sent to ALU3 on a ±6 correction cycle.

   In order for S1 to function in this manner, the arithmetic control word (type 10) must specify both decimal addition (C, D+-, C), and the S12 status set.
2. In binary operations, S1 is set to the value of the carry-out of:
   a. Bit 1 in single-type ALU operations,
   b. Bit 1 of byte 0 in fullword ALU operations.

The control word must specify the S12 status set, along with the appropriate binary ALU operation, in order for S1 to function in either of these two ways.

S2

1. In byte operations, S2 is set to 1 if the Z-bus (ALU result byte 3) is not zero. ($\overline{Z0}$) If the Z-bus is zero, S2 is not changed from its prior setting. The arithmetic control word calling

for the byte operation must specify the S12 status set in order for the S2 bit to function in this manner.

2. In full word binary operations, S2 is set to 1 if the entire 32-bit result is nonzero. S2 is not changed if the 32-bit result is zero. A status set of S12 must be specified in the arithmetic control word (calling for the fullword operation) in order for S2 to function in this manner.
3. In fullword binary operation, S2 is set to 1 if:
   a. A Z24 status set is specified, and
   b. the low-order 24 bits (ALU result bytes 1, 2, and 3) are nonzero. If the low-order 24 bits are all zero, S2 is not changed.
4. When an S2 status set is specified in a storage word, S2 is set to 1 if the count field is not zero after the count is decremented; S2 is set to 0 if the count field *is* zero after decrementing. The decrement-count function is specified in the storage word to effect the following actions:
   a. The 24-bit address, in bytes 1, 2, and 3 of the even word (of an even/odd pair of local-storage words), is updated.
   b. The count field (low-order 16 bits of the odd word of the pair) is updated.
   c. S2 is set according to the result of Step b.

Note that the count value is updated by circuitry and that S2 is set/reset if the S2 status set is specified *even if* decrement count is not specified. In this case, however, the updated count value is not stored back into the count location.

S3

S3 is set to the value of the carry-out of bit-0 of the ALU operation. This function is used in both byte and word operations. The arithmetic operation field must contain a bit configuration designated by a statement that contains a C at the left, in order for S3 to function in this manner. For example, in the statement:

```
S0 = 0
C + 0
```

the C specifies that S3 is to be set/reset.

In fullword arithmetic operations, S3 is set to the carry-out of:

a. Bit 0 of byte 0 if an S12 or no status set is specified.
b. Bit 0 of byte 1 if a Z24 status set is specified.

S3 is not set/reset in storage-word address and count updates.

S4 and S5

In arithmetic words (types 10 and 11) that specify a status set of S45, S4 and S5 are set/reset according to the bit values of the ALU result byte as follows:

Bit	Value	Indicates Result Byte Bits
S4	0	0-3 not equal to 0000
S4	1	0-3 = 0000
S5	0	4-7 not equal to 0000
S5	1	4-7 = 0000

If a Z6 status set is specified:

Bit	Value	Indicates Result Byte Bits
S4	0	0-5 not equal to 000000
S4	1	0-5 = 000000
S5	0	4-7 not equal to 0000
S5	1	4-7 = 0000

The S45 and Z6 status sets can be specified in an arithmetic word, only if a single-byte ALU operation is called for; S45 and Z6 do not pertain to fullword arithmetic operations.

In storage-word operations, S4 and S5 functions are the same as in the arithmetic words and are specified in the same manner (S45 and Z6). S4 and S5 are set according to the value of the low-order byte of the count field after the count has been decremented. The count is in bytes 2 and 3 of the odd word of an even-odd pair of local-storage words. The address is in bytes 1, 2, and 3 of the even-word location.

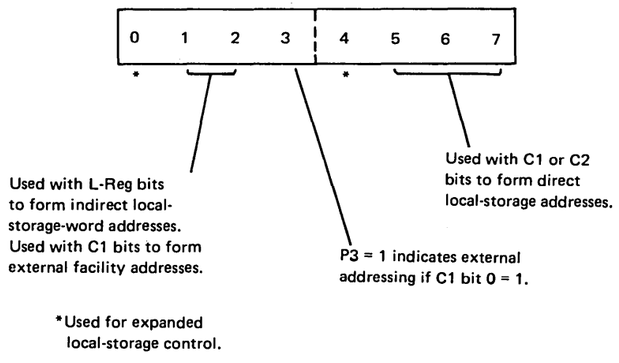
In an arithmetic word (type 10) that specifies an ABCK byte operation: S4 is set to 1 if a parity-check error is detected on the A input to the ALU(s).

Note: If an I24 status set is specified in a type 10 arithmetic word, no S-register bit setting occurs, regardless of any other specified status set. For example, if the *operation* specified is:

```
S0 = 0
C + 0
```

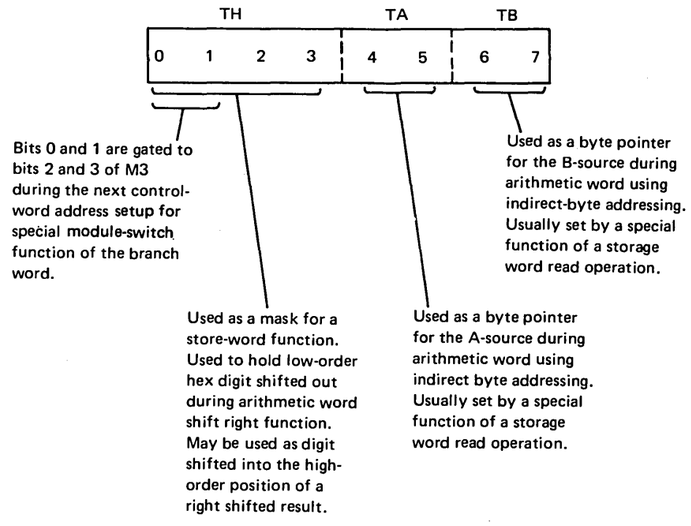
and I24 is also specified, then S3 and S0 are not changed even though the operation field calls for such set/reset functions.

P-REGISTER



The primary function of the P-register is to provide a base address during local-storage or external-register addressing. That is, the P-register is used to point to groups of external registers or areas of local storage; the remainder of the external/local-storage address is specified by the outputs of the C-register. In some cases, the L- and/or T-register contents are used to determine portions of the address.

T-REGISTER

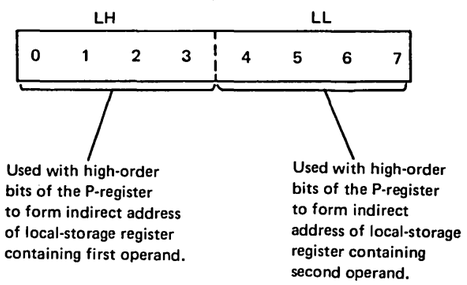


- May be used as a working register

The T-register is used in a variety of ways, which are summarized here.

1. T-register bits 4 and 5 and/or 6 and 7 are used in indirect-byte addressing and branching operations.
2. T-register bits 0 and 1 are used to form a portion of the next-control-word address when a module-switching operation is specified in the branch word.
3. T-register bits 0 through 3 are used in arithmetic fullword shift operations.
4. In certain storage-word read operations, T-register bits 4 and 5 or 6 and 7 are set to the value of the two low-order storage-address bits before the address is updated.
5. In certain storage-word store operations, T-register bits 0 through 3 are used to specify which bytes of a source are to be stored and what constant, if any, is to be used to update the storage address.

L-REGISTER



- May be used as a working register.

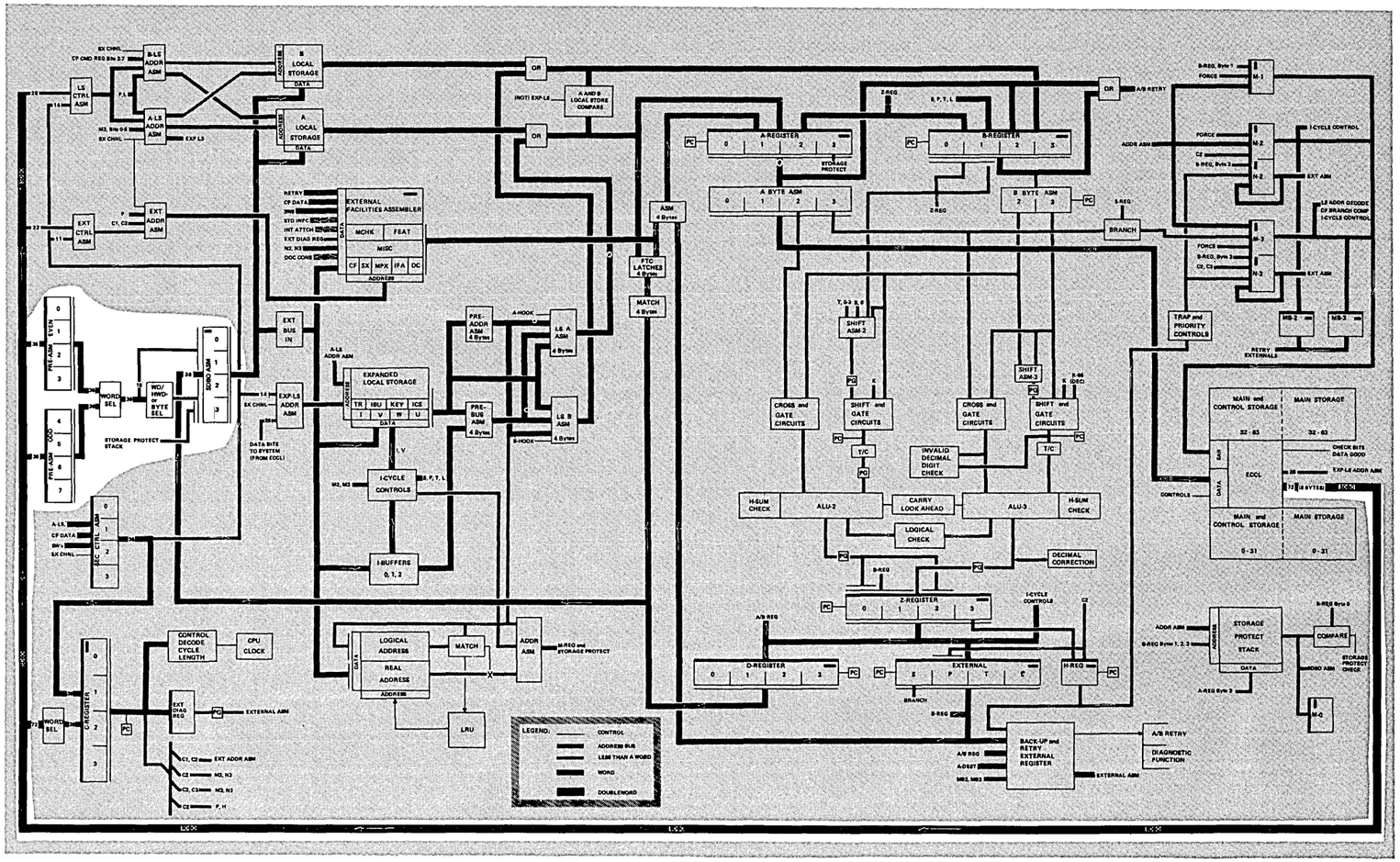
The primary purpose of the L-register is to hold the addresses of the general or floating-point registers, all of which are in local storage. The address of a general register can be specified by L0 through L3 or L4 through L7. For example, L0-3=0111 can specify general register 7. Note, however, that while this address corresponds to the hexadecimal address of general register 7, that register's address in local storage is determined by C, P, and L-register bits when the L-register is used in the addressing.

The address of a floating-point register is usually specified by L0-3 only (not L4-7).

SDBO PRE–ASM, ASM

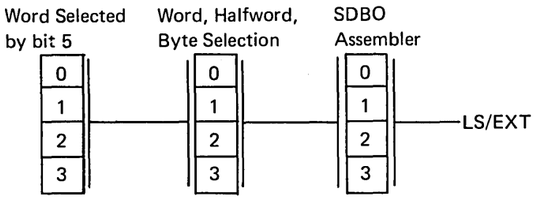
- The Storage Data Bus Out (SDBO) preassembler receives a doubleword of data from internal or external storage.
- The output of the SDBO preassembler is gated by M3 bits 5, 6, and 7 to provide word, halfword, or byte selection.
- The SDBO assembler receives inputs from the SDBO preassembler, the storage-protect stack, and the D-register. It provides an output that is used as data for External Bus In (EBI) and local storage.

The selection of data fed to the SDBO assembler is accomplished by decoding M3 bits 5, 6, and 7. The decode of the M3 bits causes corresponding gating lines to be activated, which cause data from the SDBO preassembler to be routed to the SDBO assembler.

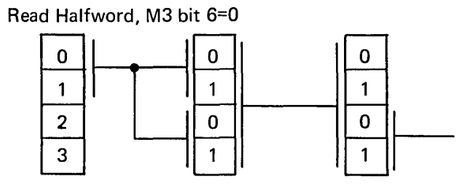


M3 Reg		Selects
bit 5	0	Even word bytes 0-3
	1	Odd word bytes 4-7
bit 6	0	Even halfword, bytes 0, 1
	1	Odd halfword, bytes 2, 3
bit 7	0	Even bytes 0, 2
	1	Odd bytes 1, 3

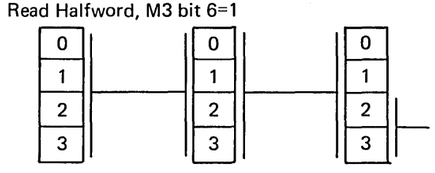
Read Word



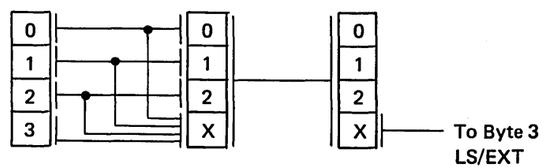
Read Halfword, M3 bit 6=0



Read Halfword, M3 bit 6=1

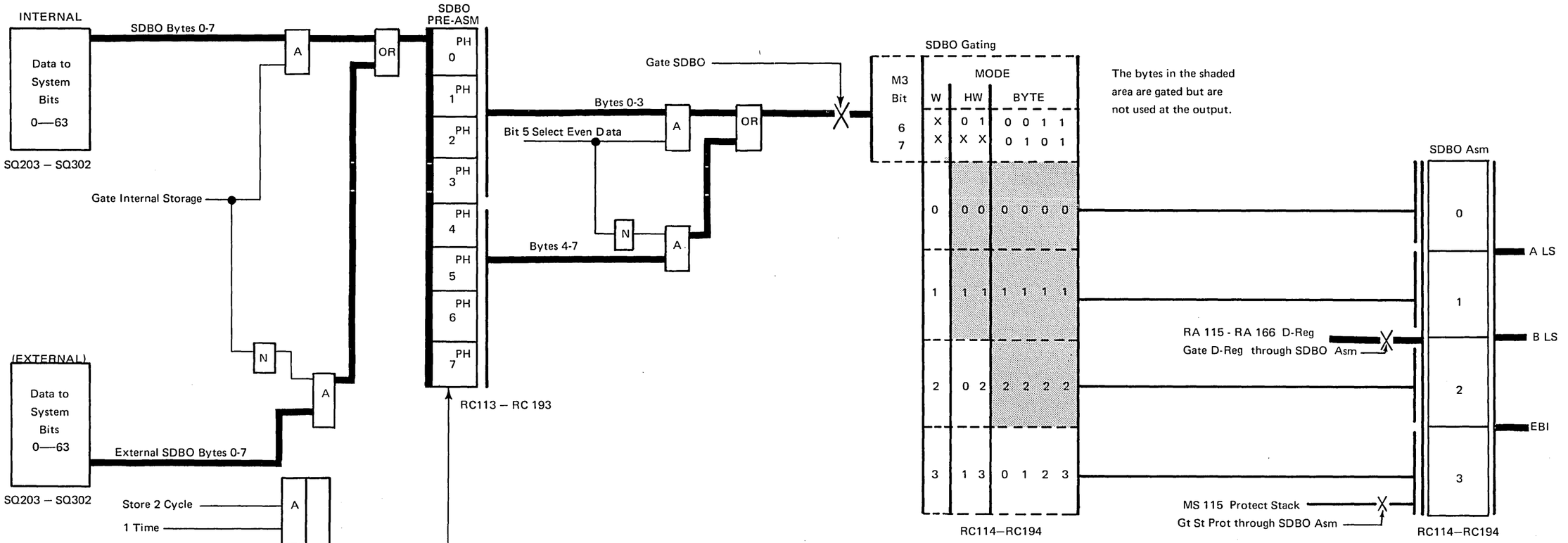


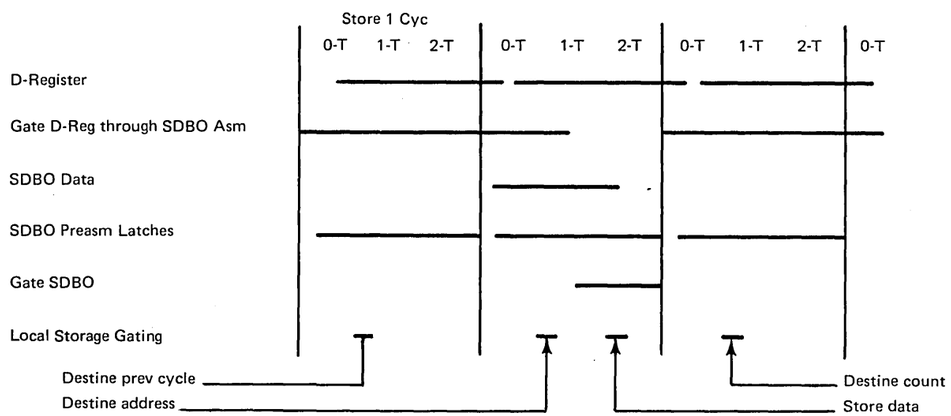
Read Byte



M3 bit 6,7=00, Byte 0 to X
M3 bit 6,7=01, Byte 1 to X
M3 bit 6,7=10, Byte 2 to X
M3 bit 6,7=11, Byte 3 to X

SDBO PRE-ASSEMBLER UNIT DATA FLOW





The two-word SDBO preassembler latches retain the storage entry for use during the cycle.

M3 bit 5 of the storage address defines which word of the storage access is to be gated to the SDBO assembler.

For a word-mode operation, the full word is gated to the assembler when the gate SDBO line is raised.

For a halfword mode operation, the upper or lower half-word, depending on M3 bit 6, is gated to bytes 2 and 3 while input bytes 0 and 1 continue to gate to output bytes 0 and 1.

For byte-mode operation, the byte defined by M3 bits 6 and 7 is gated to byte 3 while the remaining input bytes are gated to their respective output positions.

The protect stack read out is gated to the SDBO assembler byte 3 during the ISK instruction to allow transfer to a GP-register.

The four-byte output of the D-register is gated to the respective outputs of the SDBO assembler when neither the gate SDBO line nor the gate st prot line are active.

LOCAL STORAGE

- Local Storage (LS) consists of two monolithic stacks of 64 words each (A-LS and B-LS).
- Destined data is written into both A and B LS so that both stacks contain the same information at any corresponding address. This permits comparison checking of LS data.
- LS is used by the microprogram as a high-speed buffer. Access time is 24 nanoseconds.
- Readout is nondestructive.
- Address range within each stack is 00 to 3F (Hex).
- Addressing is accomplished with combinations of control-word bits, P-register bits, L-register bits, T-register bits, selector-channel share-cycle forced bits, and console-file command-register bits.

LS has assigned locations for specified functions. Refer to "Local Storage Map (370 Microprogram in Control Storage"). Locations included are:

16 general registers
4 floating-point registers
Selector-channel work area
CPU work area

These locations are valid when the 370 microprogram is located in control storage. When diagnostics are running, another set of LS assignments is in effect.

LS is external to main and control storage. Each 64-word stack is located on two MST cards:

	Bytes	Card Location
A-LS	0 and 1	A-C4B2
	2 and 3	A-C4C2
B-LS	0 and 1	A-B4P2
	2 and 3	A-B4M2

NOTE: Do not remove or replace LS array cards with power on.

LOCAL-STORAGE OPERATION

Read

- Either or both A-LS and B-LS can be accessed in one cycle.
- Data from A-LS is gated to the A-register.
- Data from B-LS is gated to the B-register.
- A-LS and B-LS sources can be different addresses.

Write/Read

- Data destined to LS always is written into both A and B LS.
- Data destined during any cycle is written during the next cycle.
- A write LS is always followed by a read LS. The read LS data is used for flush-through checking and A and B LS comparing.

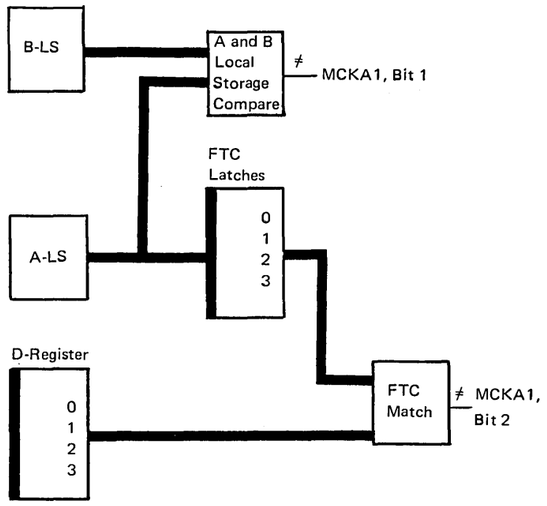
DATA CHECKING

Flush-Through Check (FTC)

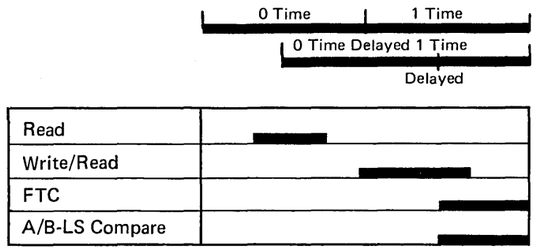
Data destined to local storage as a result of some control-word operation, other than a storage word read, is gated from the D-register through the SDBO assembler to local storage. The data in the D-register is compared with the data from the A-LS address that was the destination. If the compare is not equal, bit 2 of MCKA 1 sets to indicate an FTC error.

A and B Local-Storage Compare

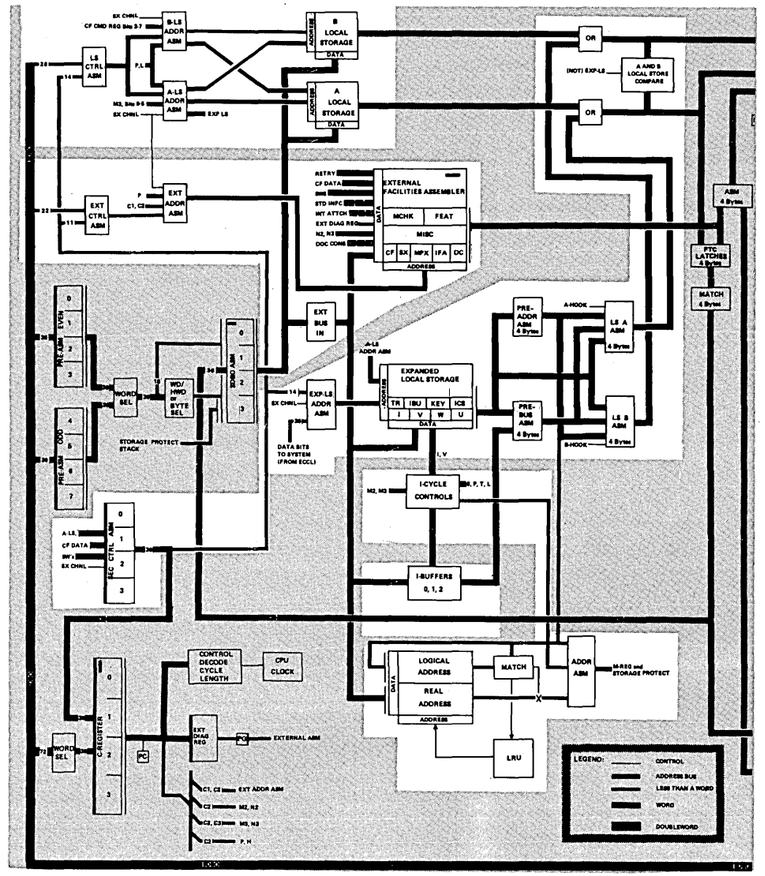
Data destined to local storage is stored at corresponding addresses in both local-storage stacks. The data is then read from these addresses and compared. If the compare is not equal, bit 1 of MCKA 1 sets to indicate the error.

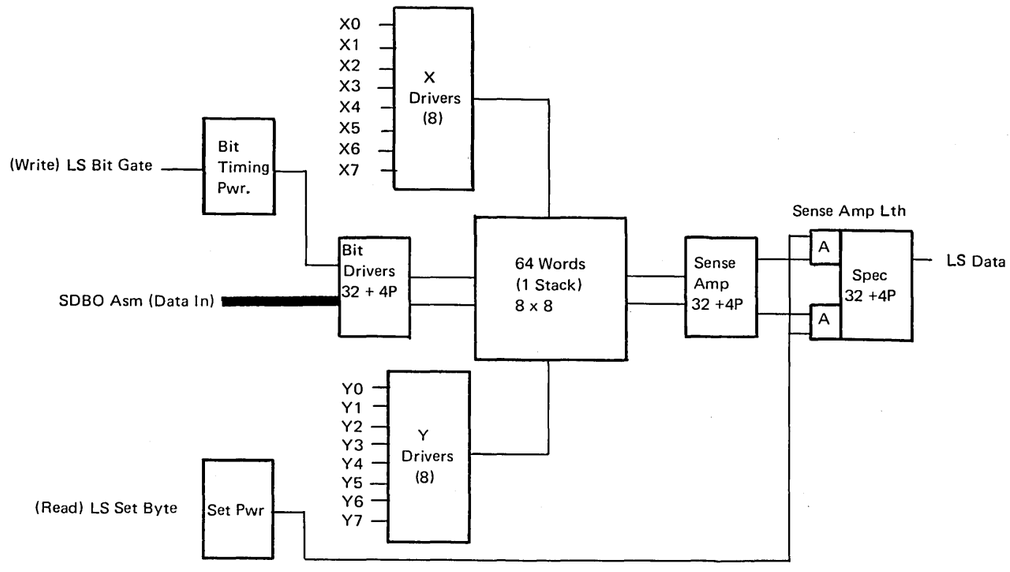


LOCAL-STORAGE TIMING



Write/Read, FTC, and A/B-LS compare occur during the cycle following the control-word cycle that the data was destined.

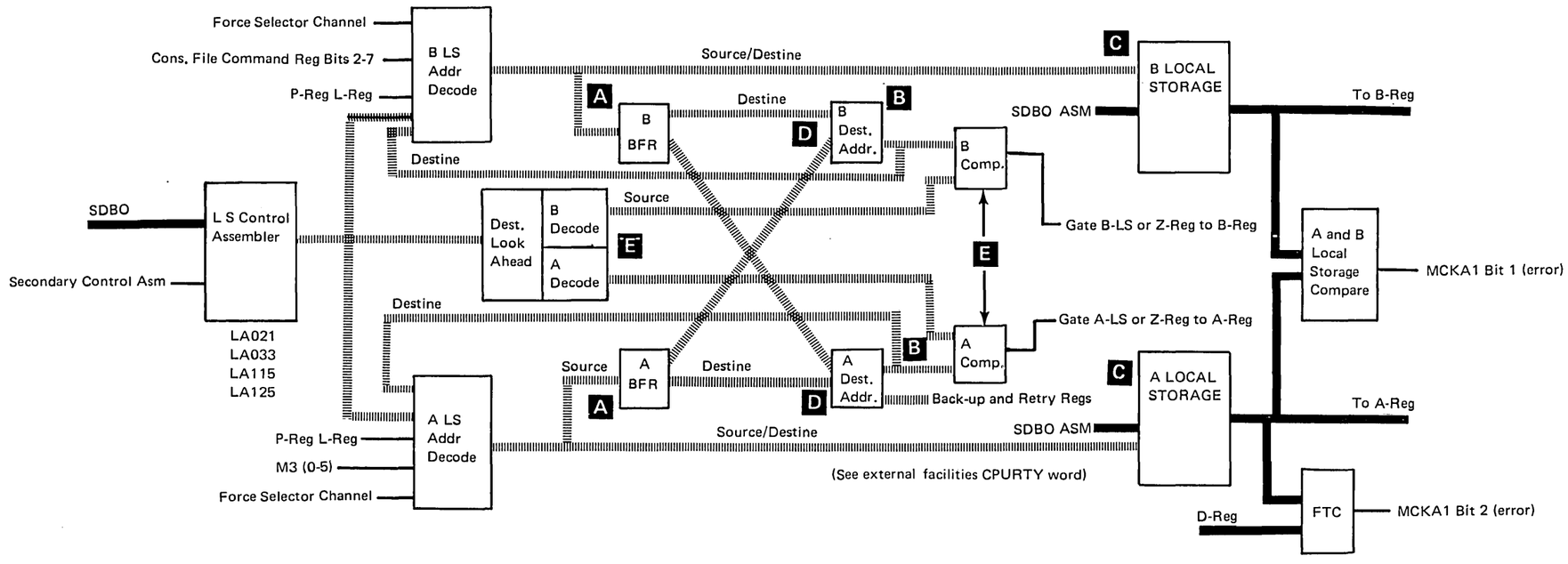




LOCAL-STORAGE DESTINATION ADDRESSING

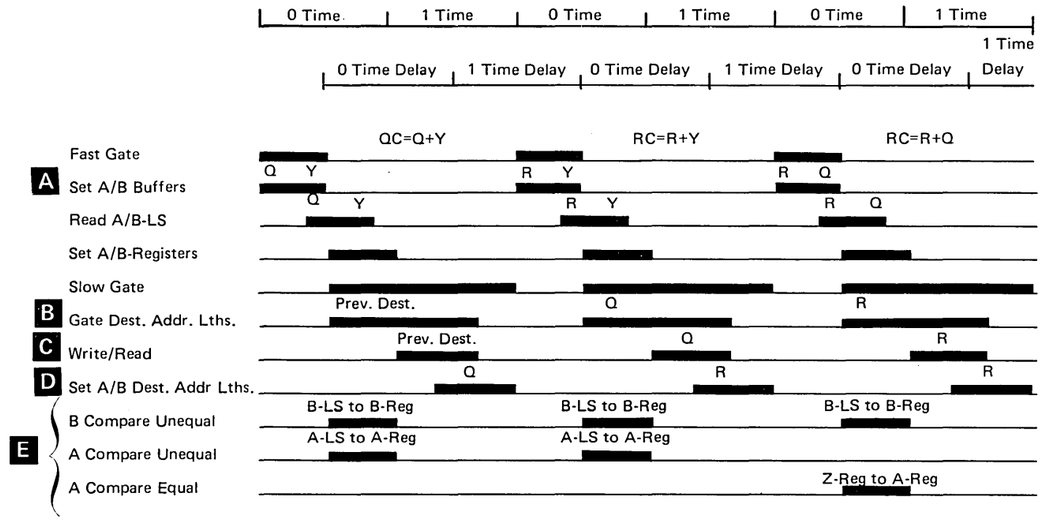
The type/form of the control word selects the source address (A or B) that is used for the destination address.

1. During fast gate, decoded source addresses are stored in the A and B buffers. **A**
2. At the beginning of slow gate, the previous control-word destination address is gated from the A and B destination address latches to the address decoders. **B**
3. The data destined during the previous cycle is written. **C**
4. The buffer (A or B) selected by the word type/form is gated to both the A and B destination address latches. **D**
5. At slow gate of the next cycle, this address is gated to the address decoders for destination write/read. **B**



Destination Look-Ahead **E**

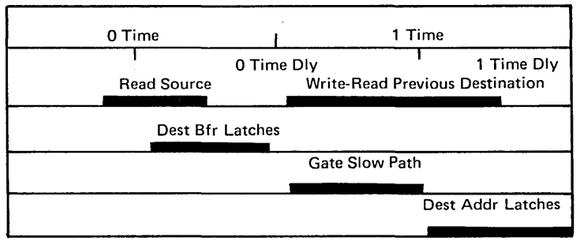
A new source address may be the same local-storage address as the previous destination address. When this condition occurs, the destined data in the Z-register is not stored into local storage in time to be accessed by the following control word as source data. Destination look ahead detects this condition by comparing A and B new source addresses with the previous destination address. An unequal compare (previous destination *not* new source) gates the new source data from local storage to the A- or B-register. An equal compare (previous destination *is* new source) gates the new source data directly from the Z-register to the A- or B-register.

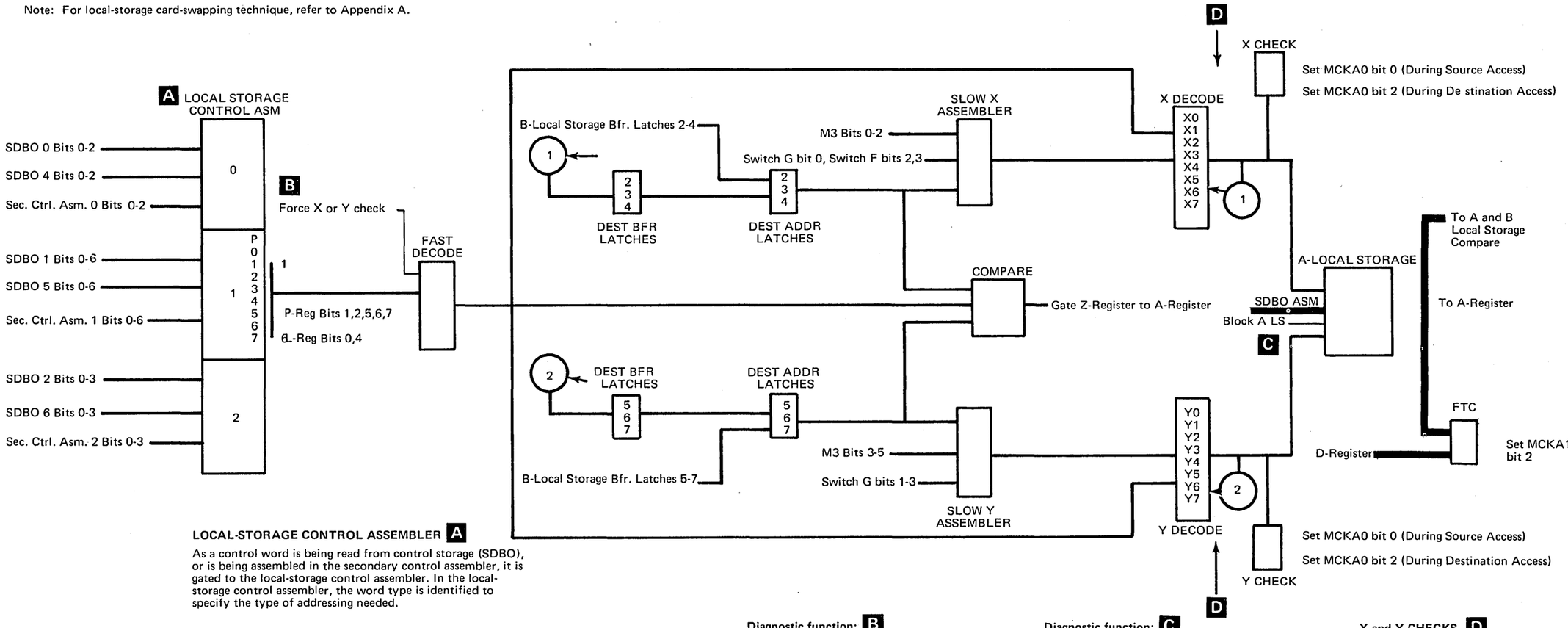


A-LOCAL STORAGE UNIT DATA FLOW

CIRCUIT	CARD	ALD
FAST DECODE	C4F2,C4N2,C4M2	LA011-LA018,LA021,LA111-LA117
DEST BFR LATCHES 2,3,4	C4G4	LA212
DEST BFR LATCHES 5,6,7	C4J4	LA222
DEST ADDR LATCHES 2,3,4	C4G4	LA212
DEST ADDR LATCHES 5,6,7	C4J4	LA222
SLOW X ASSEMBLER	C4G4	LA211
SLOW Y ASSEMBLER	C4J4	LA221
COMPARE	C4N2,C4L2	LA022-LA024, LA031-LA032
X DECODE	C4F2	LA015-LA016
Y DECODE	C4M2	LA114
X CHECK	C4G4	LA212
Y CHECK	C4J4	LA222
MONO BUFFER	B4P2,B4M2	LA311,LA327

Note: For local-storage card-swapping technique, refer to Appendix A.





LOCAL-STORAGE CONTROL ASSEMBLER A

As a control word is being read from control storage (SDBO), or is being assembled in the secondary control assembler, it is gated to the local-storage control assembler. In the local-storage control assembler, the word type is identified to specify the type of addressing needed.

Diagnostic function: B

With DIAG0 bit 5 on, and RTY backup Asm byte 0 bit 5 off, an extra X line is forced up to cause an X check.

With DIAG0 bit 5 on, and RTY backup Asm byte 0 bit 6 off, an extra Y line is forced up to cause a Y check.

Diagnostic function: C

With DIAG0 bit 4 or 5 on, and RTY backup Asm byte 3 bit 6 off, the storing of data into A local storage is blocked. This causes both a flush-through check and an A and B local store compare error.

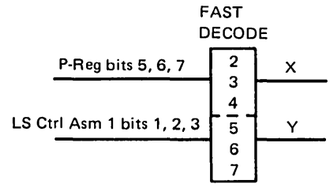
X and Y CHECKS D

The X- and Y-checks are tests to determine whether if at least one, but not more than one, X- and Y-line is active at a time.

A=LOCAL STORAGE ADDRESS ASSEMBLY

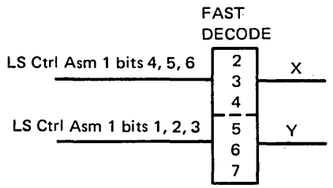
A-SOURCE DIRECT ADDRESS

When the A-source is called for by its symbolic name, or the actual address is used, bit 0 of byte 1 of the control word is 0 to flag direct addressing. Bits 1, 2, and 3 of byte 1 of the control word form the Y-line, and bits 5, 6, and 7 of the P-Reg form the X-line.



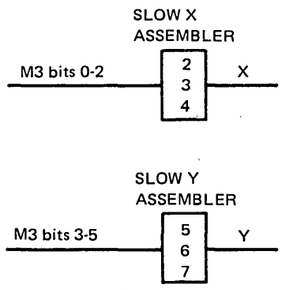
A-SOURCE DIRECT ADDRESS
(X- and Y- Lines carried in control word)

Two control words have the capability of carrying the X- and Y-lines in the bit structure of the control word. The Branch and Link or Return word carries the X- and Y-lines of the link register in bits 1-6 of byte 1. The word-move word carries the S- and Y-lines of the source or destination in byte 1.



LCSC MODE

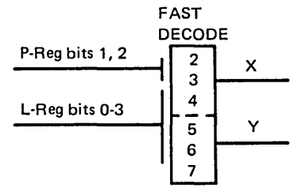
A special diagnostic function called LSCS (Local-Storage Control Storage) mode causes control words located in local storage to be read out and executed. The X- and Y-lines of the control word to be executed are formed from bits 0-5 of the M3-register, which is set up by the last word executed. Any of the other ways of addressing local storage may be used for data accesses during the execution of the control words read from local storage. For additional information about LSCS mode, refer to "Chapter 13. Diagnostic Hardware" under the heading "Basic Tests."



INDIRECT ADDRESSING

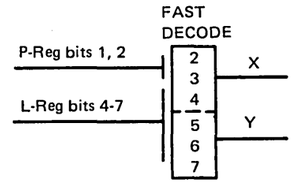
USING L HIGH (LH)

There are two symbols in the microprogram language that call for indirect word addressing. The symbol LH calls for local-storage addressing using the high bits of the L-register and bits 1 and 2 of the P-register to form the X- and Y-lines.



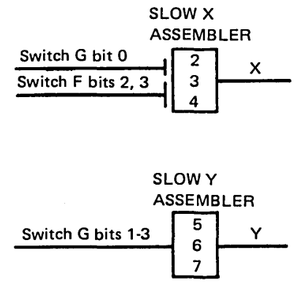
USING L LOW (LL)

The symbol LL calls for local-storage addressing, using the low bits of the L-register and bits 1 and 2 of the P-register to form the X- and Y-lines.



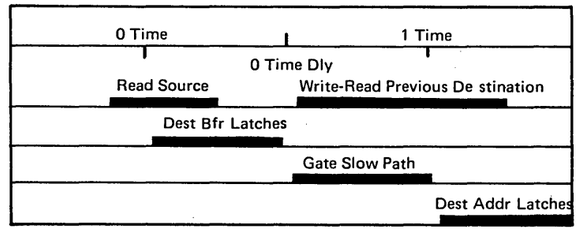
ADDRESSING FROM CONSOLE

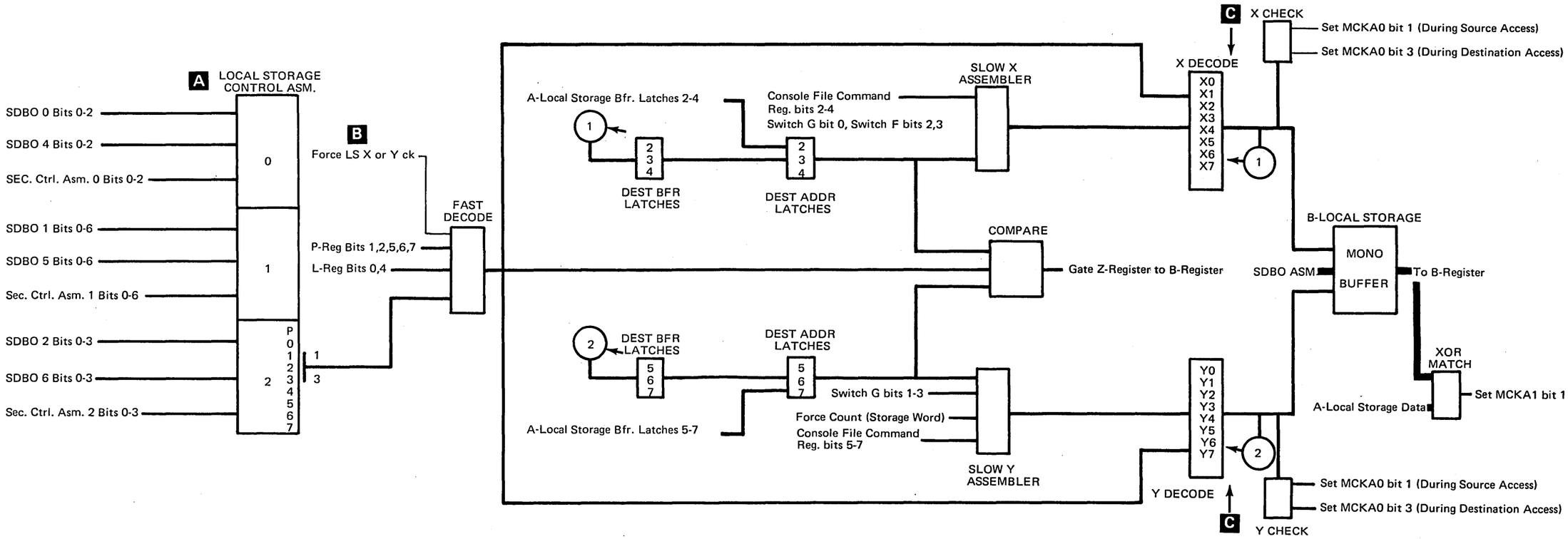
Local storage is also accessible from the operator's console. The X-line is formed from bits 2 and 3 of the F switch and bit 0 of switch G. The Y-line is formed from bits 1-3 of switch G.



B-LOCAL-STORAGE UNIT DATA FLOW

CIRCUIT	CARD	ALD
FAST DECODE	C4F2,C4N2,C4H2	LA011-LA018,LA021,LA121-LA127
DEST BFR LATCHES2,3,4	C4G2	LA232
DEST BFR LATCHES 5,6,7	C4J2	LA242
DEST ADDR LATCHES 2,3,4	C4G2	LA232
DEST ADDR LATCHES 5.6.7	C4J2	LA242
SLOW X ASSEMBLER	C4G2	LA231
SLOW Y ASSEMBLER	C4J2	LA241
COMPARE	C4N2,C4L2	LA022-LA024,LA031-LA032
X DECODE	C4F2	LA017-LA018
Y DECODE	C4H2	LA124
X CHECK	C4G2	LA232
Y CHECK	C4J2	LA242
MONO BUFFER	C4B2,C4C2	LA331-LA337, LA341-LA347





LOCAL-STORAGE CONTROL ASSEMBLER A

As a control word is being read from control storage (SDBO), or is being assembled in the secondary control assembler, it is gated to the local-storage control assembler. In the local-storage control assembler, the fields that indicate the source accessing are tested to determine the type of address formation needed to address local storage.

Diagnostic function: B

With DIAG0 bit 5 on, and RTY backup Asm byte 0 bit 5 off, an extra X-line is forced up to cause an X-check.
With DIAG0 bit 5 on, and RTY backup Asm byte 0 bit 6 off, an extra Y-line is forced up to cause a Y-check.

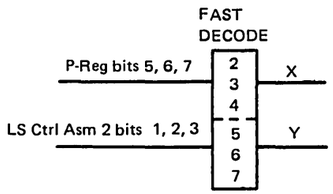
X- and Y CHECKS C

The X- and Y-checks are tests to determine whether at least one but not more than one X- and Y-line is active at a time.

B-LOCAL-STORAGE ADDRESS ASSEMBLY

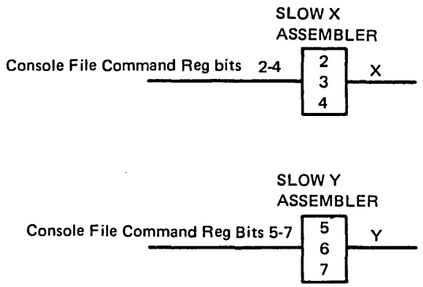
B–SOURCE DIRECT ADDRESS

When the B-source is called for by its symbolic name, or the actual address is used, bit 0 of byte 2 of the control word is 0 to flag direct addressing. Bits 1, 2, and 3 of byte 2 of the control word form the Y-line, and bits 5, 6, and 7 of the P-reg form the X-line.



ADDRESSING FROM CONSOLE FILE

There are console file commands that cause the control word currently in the C-register to be executed. The X- and Y-lines that would normally be formed by control word bits are formed from bits 2-7 of the console file command register.

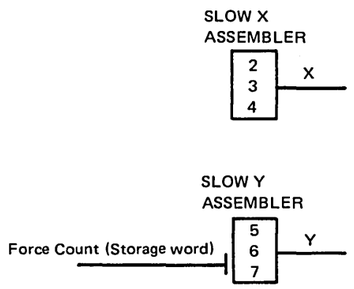


FORCE COUNT ADDRESS

When a storage word with the decrement count function is executed an odd local storage address must be forced. The Y-line in effect, for addressing source register, is assumed to be an even Y-line. The count to be accessed is in the next higher address; therefore the next higher Y-line is forced.

For example: A storage word is addressing the GD register for selector-channel 2 (see "Local Storage Map (370 Microprogram in Control Storage"). The X Y-lines for GD are X4, Y0. To access the count, Y1 is forced, and the GC register is addressed.

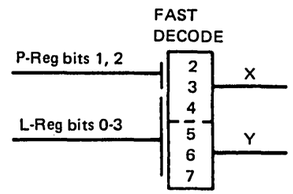
The X-line remains the same for the count access.



INDIRECT ADDRESSING

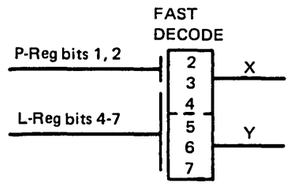
USING L HIGH (LH)

There are two symbols in the microprogram language that call for indirect word addressing. The symbol LH calls for local-storage addressing using the high bits of the L-register and bits 1 and 2 of the P-register to form the X- and Y-lines.



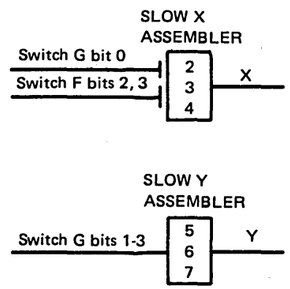
USING L LOW (LL)

The symbol LL calls for local-storage addressing using the low bits of the L-register and bits 1 and 2 of the P-register to form the X- and Y-lines.



ADDRESSING FROM CONSOLE (B - LS)

Local storage is also accessible from the operator's console. The X-line is formed from bits 2 and 3 of the F switch and bit 0 of the G switch. The Y-line is formed from bits 1-3 of switch G.



LOCAL-STORAGE MAP (370 MICROPROGRAM IN CONTROL STORAGE)

	Word Name	LS Location	Byte 0	Byte 1	Byte 2	Byte 3	X and Y Line	P_{low} Direct Access	P_{high} Indirect Access
		00		General Register 0			X0 Y0	0	0
		01		General Register 1			X0 Y1		
		02		General Register 2			X0 Y2		
		03		General Register 3			X0 Y3		
		04		General Register 4			X0 Y4		
		05		General Register 5			X0 Y5		
		06		General Register 6			X0 Y6		
		07		General Register 7			X0 Y7		
		08		General Register 8			X1 Y0	1	
		09		General Register 9			X1 Y1		
		0A		General Register A			X1 Y2		
		0B		General Register B			X1 Y3		
		0C		General Register C			X1 Y4		
		0D		General Register D			X1 Y5		
		0E		General Register E			X1 Y6		
		0F		General Register F			X1 Y7		
	AX	10		SRTN Temp Link			X2 Y0	2	2
	DI	11		Alter/Display Log Link			X2 Y1		
	RTX	12		Retry Link			X2 Y2		
	DTX	13		Translate Link			X2 Y3		
	X	14		Working			X2 Y4		
	R	15		Working			X2 Y5		
	Y	16		Working			X2 Y6		
	Q	17		Working			X2 Y7		
MPX CHANNEL	MA	18		Working			X3 Y0	3	
	MBS	19		Working			X3 Y1		
	MX	1A		Working			X3 Y2		
	MC	1B		Working			X3 Y3		
	MD	1C		Working			X3 Y4		
	MF	1D		Working			X3 Y5		
	MW	1E		Working			X3 Y6		
	CX	1F	CPU	Link	Register		X3 Y7		

	Word Name	IF A Name	LS Location	Byte 0	Byte 1	Byte 2	Byte 3	X and Y Line	P_{low} Direct Access	P_{high} Indirect Access
SX 2	GD		20					X4 Y0	4	4
	GC		21			Count		X4 Y1		
	GM		22		Protect CCW Address			X4 Y2		
	GW		23					X4 Y3		
SX 3	GD		24					X4 Y4		
	GC		25			Count		X4 Y5		
	GM		26		Protect CCW Address			X4 Y6		
	GW		27					X4 Y7		
SX 1	GD	FD	28					X5 Y0	5	
	GC	FC	29			Count		X5 Y1		
	GM	FM	2A		Protect CCW Address			X5 Y2		
	GW	FW	2B					X5 Y3		
SX 4	GD	FA	2C					X5 Y4		
	GC	FB	2D			Count		X5 Y5		
	GM	FS	2E		Protect CCW Address			X5 Y6		
	GW	FL	2F					X5 Y7		
			30		Floating-Point Register 0			X6 Y0	6	6
			31		Floating-Point Register 0			X6 Y1		
			32		Floating-Point Register 2			X6 Y2		
			33		Floating-Point Register 2			X6 Y3		
			34		Floating-Point Register 4			X6 Y4		
			35		Floating-Point Register 4			X6 Y5		
			36		Floating-Point Register 6			X6 Y6		
			37		Floating-Point Register 6			X6 Y7		
	SO		38					X7 Y0	7	7
	PM		39	P.E. Control	P.E. Code Group After Mask			X7 Y1		
	DM		3A		Adjustment Factor			X7 Y2		
	RW		3B		Address Adjustment Working			X7 Y3		
	DP		3C		IF A Low-Priority Link			X7 Y4		
	LNK		3D		I-Cycle Link			X7 Y5		
	P4X		3E		SX-4 Link Register			X7 Y6		
	P3X		3F		SX-1, 2, 3 Link Register			X7 Y7		

NOTE. Words 28 through 2F are shown with Selector Channel designations.

	System Mask	KEY	0WMP	Interrupt Code	ILC	CC	Program Mask	INSTRUCTION ADDRESS
	0 7	8 11	12 15	16 31	32 33	34 35	36 37 38 39	40 63
CURRENT PSW DISPLAYED IN:	External 10	EXP LS 50 Byte 0	EXP LS 53 Byte 1		EXP LS 53 Byte 0 Bit 0, 1	Byte 0 Bit 2, 3	EXP LS 53 Byte 0 Bits 4-7	EXP LS 50 Bytes 1, 2, 3

NOTE: The PSW can be manually displayed using the Console PR/KB. This procedure is contained in Chapter 7.

SCOPE PROCEDURE FOR LOCAL-STORAGE ADDRESSING

Use Tektronix Type 454 or equivalent
10X Probes
Set Time/Div: .05 us
Set Channel 1 Volt/Div: 50 MV
Set Channel 2 Volt/Div: 50 MV

Store 385E6FC8 (using system console rotary switches A through H) in an unused location of control storage. This is a wo word-move word — Y = LNK, SF, STOP

Set the P-register = 02

The word-move word is the version 1 type that carries the X- and Y-lines of the Source register in byte 1 of the control word. The mask of F specifies that all four bytes of the source are to be moved to the destination register. The STOP function is active (bit 4 of byte 3 = 1); therefore, the word-move word is continually executed.

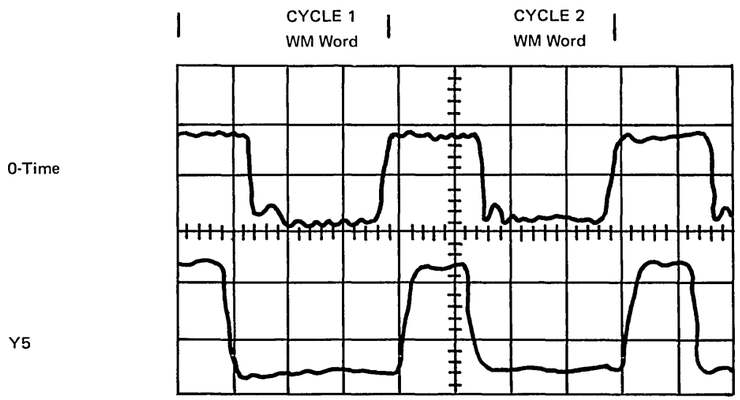
After storing the word and setting the P-register;

1. Dial the address of the word-move word into switches E-H.
2. Operate the control address set key.
3. Operate the start key.

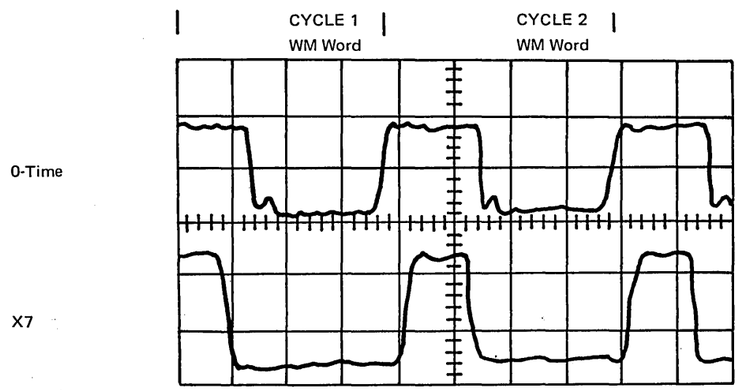
Note: The manual indicator is on because the soft-stop condition is set by the STOP function of the word-move word.

4. Sync with channel 1 on 0-time. Gate A C4 E2 G05

With channel 2, scope the + A LS ADDR Y EQUALS 5 line C4 M2 J12



1

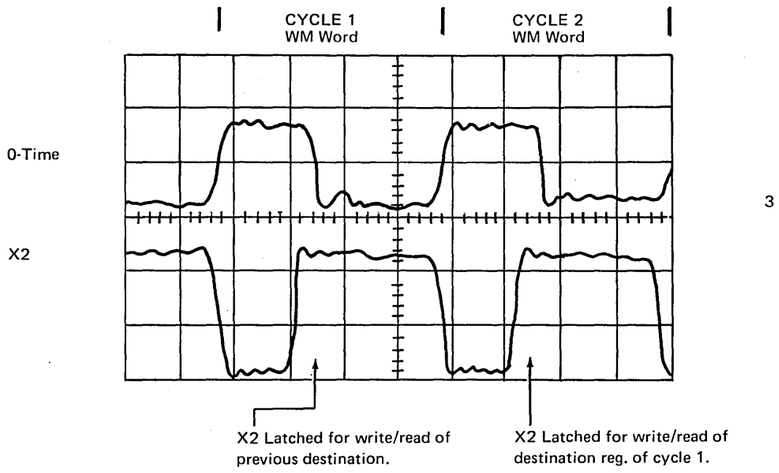


2

With channel 2, scope the + A LS ADDR X EQUALS 7 line C4 F2 G12

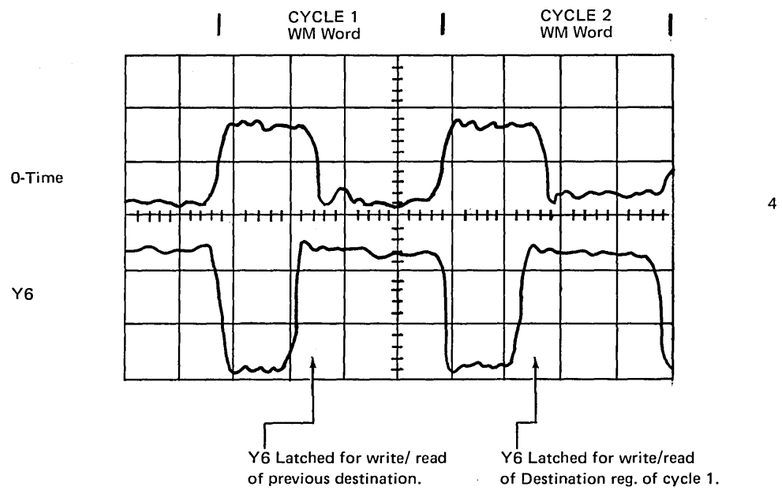
Scope pictures 1 and 2 show the X- and Y-lines for the source register LNK being activated early in the cycle of the word-move word.

With channel 2, scope the + SLOW AX PATH DECODE 2 C4 G4 B03



3

With channel 2, scope the + SLOW AY PATH DECODE 6 C4 J4 D02



4

Scope pictures 3 and 4 show the X- and Y-lines for the destination register Y being activated in the second half of the word-move word cycle. Destination addressing is always activated for the destining of the previous control word's results.

By altering the P-register setting, and the address bits in the word-move word, all the local-storage registers may be addressed.

For example;

Original word--385E6FC8 Statement– Y = LNK, SF, STOP
Change word to–385E7FC8 Statement– Q = LNK, SF, STOP

Change P-register setting to 03, leave the original control word in control storage. Statement–MN=LNK, SF, STOP

Refer to the local-storage map in Chapter 2 for the P-register settings necessary for addressing local storage.

Refer to the bit definition of the word-move word in Chapter 4 for variations of this word.

EXPANDED LOCAL STORAGE (EXPLS)

- Composed of hardware registers that are physically externals but are logically connected as local storage.
- Source addresses are formed through the expanded local-storage-address assembler.
- Destination addresses are formed through the local-storage-control assembler and the A-local-storage-address assembler.
- Expanded local-storage registers are not duplicated as are the local-storage registers.
- Used with I-cycles, selector channels, and address-adjustment circuits.

In order to access expanded local storage;

1. MODE register bit 1 must be on.
2. Direct local-storage addressing must be specified (C1 or C2 bit 0 = 0).

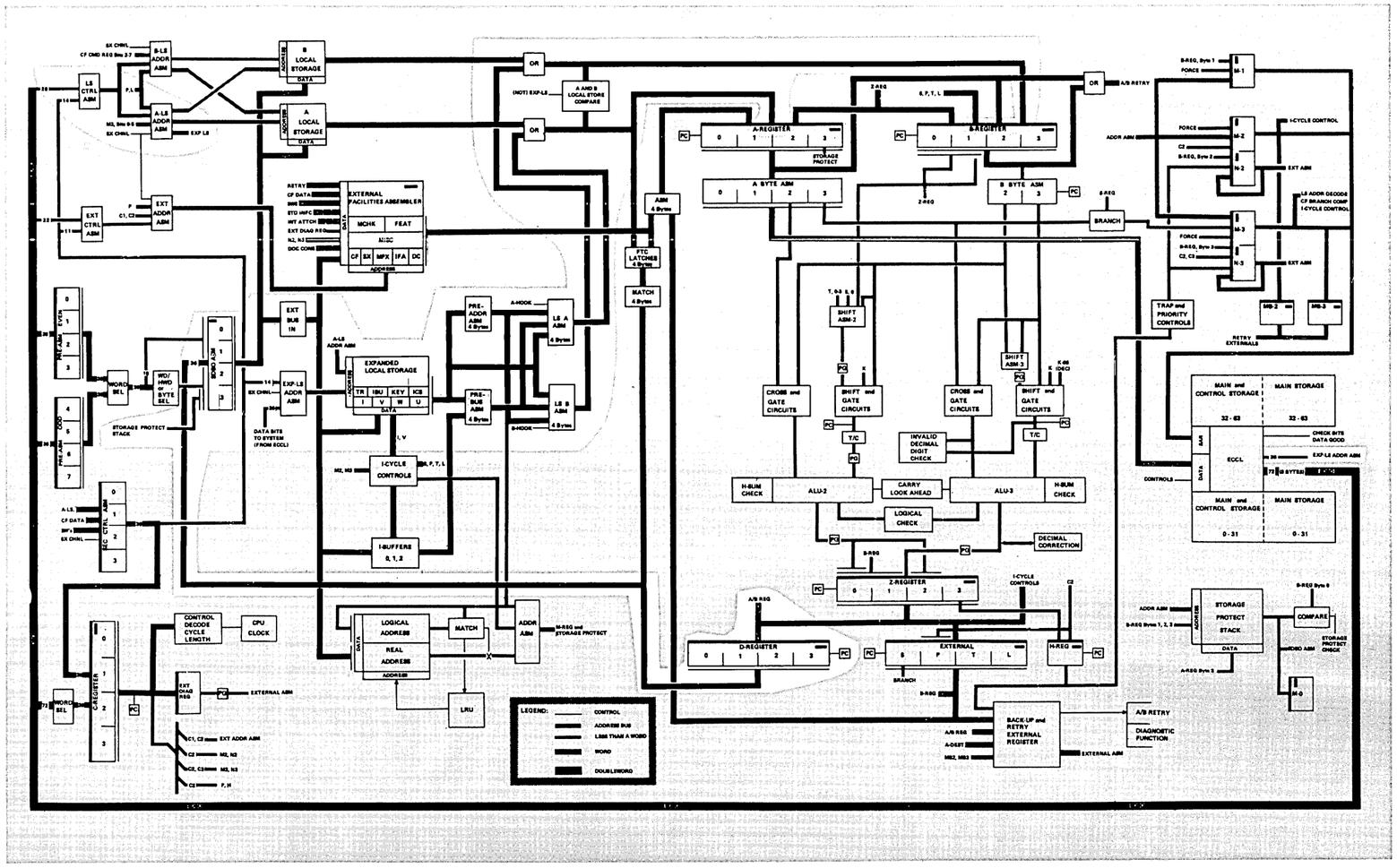
P-register bits 0, 3, and 4 control the expanded local-storage inputs to the A- and B-registers (shown on facing page).

Only one expanded local-storage register can be accessed as a source in any one control word.

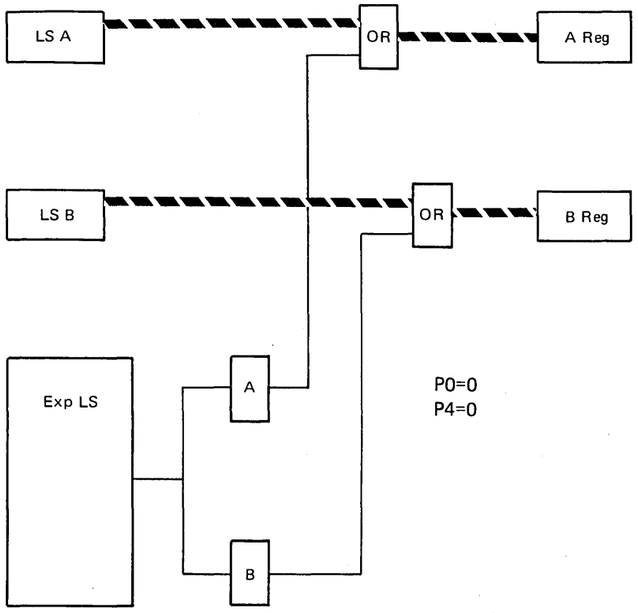
The branch and link/return word and the word-move word version 1 cannot access expanded local-storage registers.

All expanded local-storage registers may be displayed from the console, but altered only from the console printer keyboard.

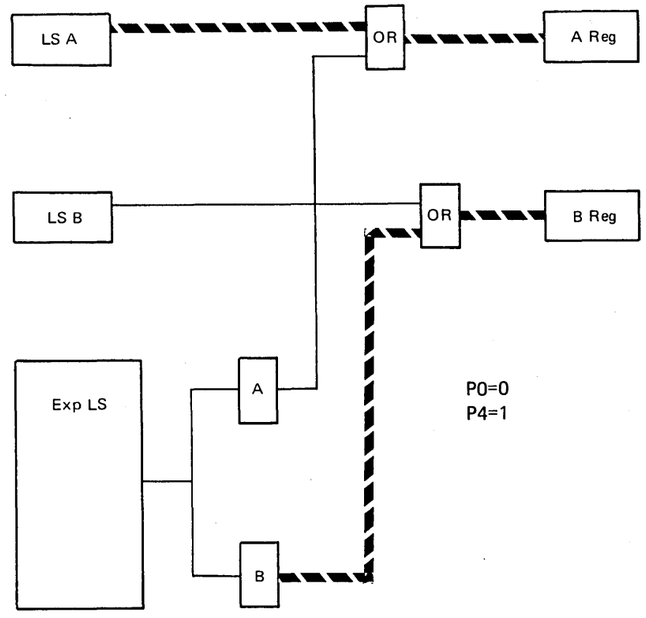
The I, V, U, and W-registers are not under control of bits 0 and 4 of the P-register, but do require that P low = 2 or A.



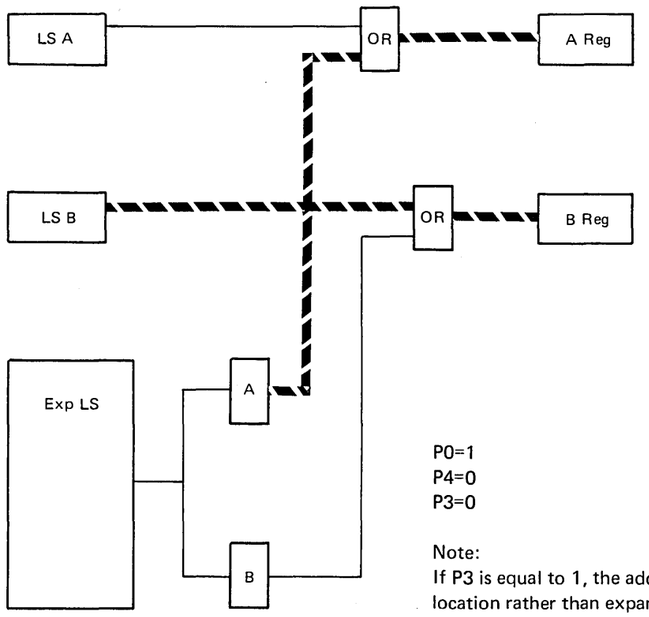
Local Storage Gating



Expanded Local Storage B Input



Expanded Local Storage A Input



P0=1
P4=0
P3=0

Note:
If P3 is equal to 1, the address would access an external location rather than expanded local storage.

I, V, U, and W are not dependent on the setting of P0 or P4.

EXPANDED LOCAL STORAGE MAP

Note: Expanded local storage may be altered from the printer-keyboard with the CE key on.

EXP LS	Word Name	Byte 0	Byte 1	Byte 2	Byte 3	X and Y Line
50	I	Key		I-Register		X2 Y0
51	V			V-Register		X2 Y1
52	W			W-Register		X2 Y2
53	U			U-Register		X2 Y3
54	IBU			IBU-Register		X2 Y4
55	TR			TR-Register		X2 Y5
56	ICS	1-Cycle Control Display				X2 Y6
		57 through 5F unassigned				
60	G2DRL	DATA ADDR (SX 2)				X4 Y0
61	G2DBRL	BACKUP DATA ADDR				X4 Y1
62						X4 Y2
63						X4 Y3
64	G3DRL	DATA ADDR (SX 3)				X4 Y4
65	G3DBRL	BACKUP DATA ADDR				X4 Y5
66						X4 Y6
67						X4 Y7
68	G1DRL	DATA ADDR (SX 1)				X5 Y0
69	G1DBRL	BACKUP DATA ADDR				X5 Y1
6A						X5 Y2
6B						X5 Y3
6C	G4DRL	DATA ADDR (SX 4)				X5 Y4
6D	G4DBRL	BACKUP DATA ADDR				X5 Y5
6E						X5 Y6
6F						X5 Y7
		70 through 77 unassigned				
78	SN				No. 1	X7 Y0
79	PN				No. 1	X7 Y1
7A	WK		Working Register			X7 Y2
7B	NP	PAA byte 1,2 Latched		Control	Control	X7 Y3
7C	DK	Local Addr. Reg.		Real Addr Reg		X7 Y4
7D	SS					X7 Y5
7E						X7 Y6
7F						X7 Y7

I-Register (EXPLS 50)

Instruction counter register. When used as a destination, byte 0, 24 bits, or fullword loading is possible.

Byte 0: If the I-Reg is destined, byte 0 is gated to the Key-Reg
Bytes 1,2, and 3 contain the instruction address

V Reg (EXPLS 51) *

Bytes 1, 2, and 3 usually contain the second operand address generated during I-cycles.

W Reg (EXPLS 52) *

Bytes 1, 2, and 3 usually contain the first operand address generated during I-cycles.

* If used in a storage word, the key register is gated as byte 0. The key register contains the storage protect key (bits 0-3, bits 4-7=0).
If V or W is used as a storage address in a storage word, the KEY reg is gated as byte 0. The KEY reg is always gated as byte 0 for I, TR, and IBU.

U Register (EXPLS 53)

When used as a destination, byte 2 may not be changed (loaded via hardware)

Byte 0 bits 0-1 used for ILC
bits 2-3 used for CC
bits 4-7 used for program mask
Byte 1 bits 0-1 reserved for FLP mult. and divide
bit 2 Indicates that GRs 0-3 need restoring.
bit 3 indicates LEX MODE
bit 4-7 used for OWMP
Byte 2 used for Op code
Byte 3 used for immediate byte information
Note:
Byte 0 bits 0 and 1 and Byte 2 bits 0 through 7 are set only by hardware.

IBU Register (EXPLS 54) *

Upon entering I-cycles, I REG bytes 1, 2, and 3 are set into IBU. If a retry condition is encountered during I-cycles, the instruction may be repeated (return to DFOC). In this event, IBU is moved to the I-Reg. This register may not be used as a destination (Loaded via hardware, only from the I-register).

TR Register (EXPLS 55) *

Address of next doubleword after the address in the I-register. This register may not be used as a destination, but will change if the I-Reg is destined.

ICS (I-Cycle Status)-Register (EXPLS 56)

This unique register is provided for manual display only, and is not accessible via microcode. While the specific signals occupy an expanded local-storage register address, no such register exists. Instead, various key signals from the I-cycle hardware (under hardware control) are gated via the register address to form a display. Interpretation of the display requires a fundamental knowledge of the functional operation of the I-cycle hardware.

Byte 0 forced to zeros
Byte 1 forced to zeros
Byte 2
bit 0 BR Read Latch
bit 1 OP LOAD Latch
bit 2 OP L2
bit 3 OP L1
bit 4 Prefetch required
bit 5 Prefetch inhibit
bit 6 FLP Long
bit 7 OP BR to DF
Byte 3
bit 0 I BFR 0 parity check latch
bit 1 I BFR 1 parity check latch
bit 2 Half adder check latch
bit 3 IMM Byte Modifier parity check
bit 4 X=0
bit 5 B=0
bit 6 Set Control Address
bit 7 Low Bit

ICS Bits: Functional Significance

Byte 2: Bit 0 (BR Read Latch) When on, indicates that a RTN to I-cycles will force an initial I-cycle address of DF0C.

Bit 1 (OP LOAD Latch) When on, indicates hardware is attempting to:
a) provide an initial I-cycle address of DF14 (if byte 2 bit 0 is off).
b) provide an address of DF14 for a further fetch of the instruction when within I-cycles and Set Control Address (Set CA) (byte 3 bit 6) is on.

Op Length		
1	2	Format
1	0	RR
0	1	RX SI
1	1	SS

Bit 2 (Op Length 2) from decode of two high-order bits of Op Reg, denotes that the data currently in Op Reg is not an RR format.

Bit 3 (Op Length 1) From decode of two high-order bits of Op Reg, denotes that the data currently in Op Reg is not of RX, RS, SI format.

Bit 4 (Prefetch req'd) conditions within I-cycle hardware, as a function of I Reg, I Bfrs, and current instruction indicate that the next instruction should be prefetched.

Bit 5 (Prefetch inhibit) When on, indicates that a Prefetch will not be allowed. This signal is also a function of I-Reg, I-Bfrs, and current instruction. Note that this signal does not take into account other functions, such as Real Instruction Address Compare mode.

Bit 6 (Floating Pt Long) When on, represents a partial decode of the data in the Op Reg. This signal is used with RR instruction format to determine a specific I-cycle path.

Bit 7 (Op Branch to DF) When on, indicates that the end of the hardware I-cycles will branch to the read and align phase of I-cycles. (the other half of the DF module instead of a CX module).

Byte 3: Bit 0 (I-Bfr 0 parity check latch) When on, indicates an incorrect parity condition for I-Bfr 0.

Bit 1 (I-Bfr 1 parity check latch) When on, indicates an incorrect parity condition for I-Bfr 1.

Bit 2 (Half Adder Check Latch) When on, indicates that a check condition occurred in the half adder during an I-register hardware update.

Bit 3 (Immediate Byte Modifier Parity Check) When on, indicates a parity check of the immediate byte modifier Reg. This signal is the check latch input. *Note:* The above four signals are combined to form the signal "I–Cycle Hardware Check".

Bit 4 (X=0) This signal has particular significance when the Op Reg data is for an RX format. When this bit is off, (and byte 3 bit 5 is off) double indexing is indicated for RX format instructions. Note that RS, SI, and SS format instructions force this signal to the ON state.

Bit 5 (B=0) When on, indicates that the data being gated through the base assembler of the I-Bfrs is zero. (refers to GPRO) This signal has no significance for RR format instructions.

Bit 6 (Set CA) When on, indicates either that the current control store address is not within the hardware I-cycle range; or that a branch point within that address range has been encountered.

Bit 7 (Low Bit) When on, indicates that hardware-detected conditions are currently satisfied for deviating from the normal branch to the execution. phase starting address of C (Op code) 0. Instead, the starting address will be C (Op code) 4. This signal has significance for certain branch instructions, floating-point, and shift-double instructions

Regarding the interpretation of the ICS signals byte 3, bit 0-3 provide a further definition of the cause of an I-cycle hardware check. Byte 2, bits 0-6 and byte 3, bits 4-6 may be used with discretion, to determine the starting address for I-cycle sequences. Byte 2 bit 7 indicates when the "read and align" phase will be used; and Byte 3 bit 7 indicates the status of hardware tests for branch on condition, floating-point reg, specification, etc.

G2DRL, G3DRL, G1DRL, and G4DRL (EXPLS 60,64,68, and 6C)

These registers function as a pointer to the next storage location used for a chare cycle.
BYTE 0 contains the protect key.

G2DBRL,G3DBRL,G1DBRL, and G4DBRL (EXPLS 61, 65,69, and 6D)

Only bytes 1 and 2 are in existence.

SN Register (EXPLS 78)

Byte 0	FF hex
Byte 1	00 hex

PN Register (EXPLS 79)

Byte 0	FF hex
Byte 1	00 hex

WK Register (EXPLS 7A)

Working Register
Byte 0 = FF

NP Register (EXPLS 7B)

Bytes 0 and 1	Logical Address
Byte 2	
Bit 0	
Bit 1	
Bit 2	
Bit 3	
Bit 4	
Bit 5	LEX MODE
Bit 6	
Bit 7	Reset Tables
Byte 3	
Bit 0	
Bit 1	
Bit 2	
Bit 3	
Bit 4	Execute Instruction
Bit 5	
Bit 6	
Bit 7	

DK Register (EXPLS 7C)

Byte 0	Bit 0		
	Bit 1		
	Bit 2		
	Bit 3		
	Bit 4		
	Bit 5		
	Bit 6		
	Bit 7		LOGICAL ADDRESS
Byte 1	Bit 0		
	Bit 1		
	Bit 2		
	Bit 3		
	Bit 4		
	Bit 5	0	
	Bit 6	0	
	Bit 7	0	
Byte 2	Bit 0		
	Bit 1		
	Bit 2		
	Bit 3		
	Bit 4		
	Bit 5		
	Bit 6		
	Bit 7		Real Address
Byte 3	Bit 0		
	Bit 1		
	Bit 2		
	Bit 3		
	Bit 4		
	Bit 5		
	Bit 6		
	Bit 7		

EXPANDED LOCAL-STORAGE: SOURCE GATING

When used as a source, expanded local storage is addressed by the expanded local-storage address assembler. The address bits from the control word are intercepted as the control word is being read from control storage, and gated to the expanded local-storage address assembler. Gating lines generated by the address assembler gate the proper expanded local-storage register to either the A- or B-register.

If expanded local storage is being gated to the A-register, the sense latches for A-local storage are blocked from being set. If expanded local storage is being gated to the B-register, the sense latches for B-local storage are blocked.

If the expanded local-storage register is also the destination, the destination latches in local storage are set to be used in the following control-word cycle.

Examples on the following pages show the various ways the expanded local-storage gates may be formed for source addressing.

Whether the expanded local-storage register is an A or a B source, it is gated to both the LS A and LS B assemblers. The source gating circuits then gate the A or B assembler to the A- or B-register.

There are control words in the microprogram listings that appear to be addressing two different expanded local-storage registers as sources in the same word. However, the B source address defaults to be the A source. For example:

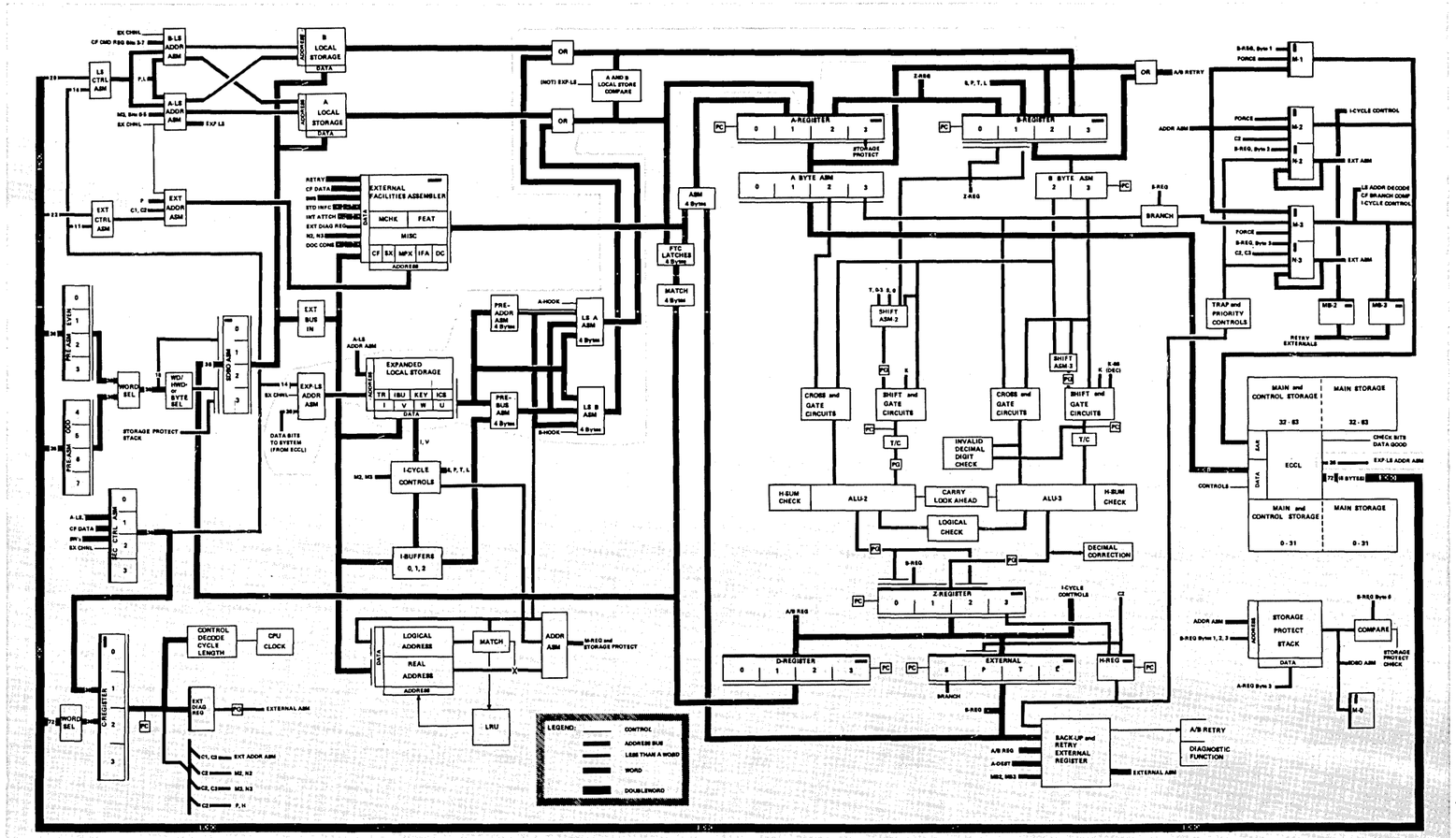
Control Word———————WK1 = NP2, OE, WK1

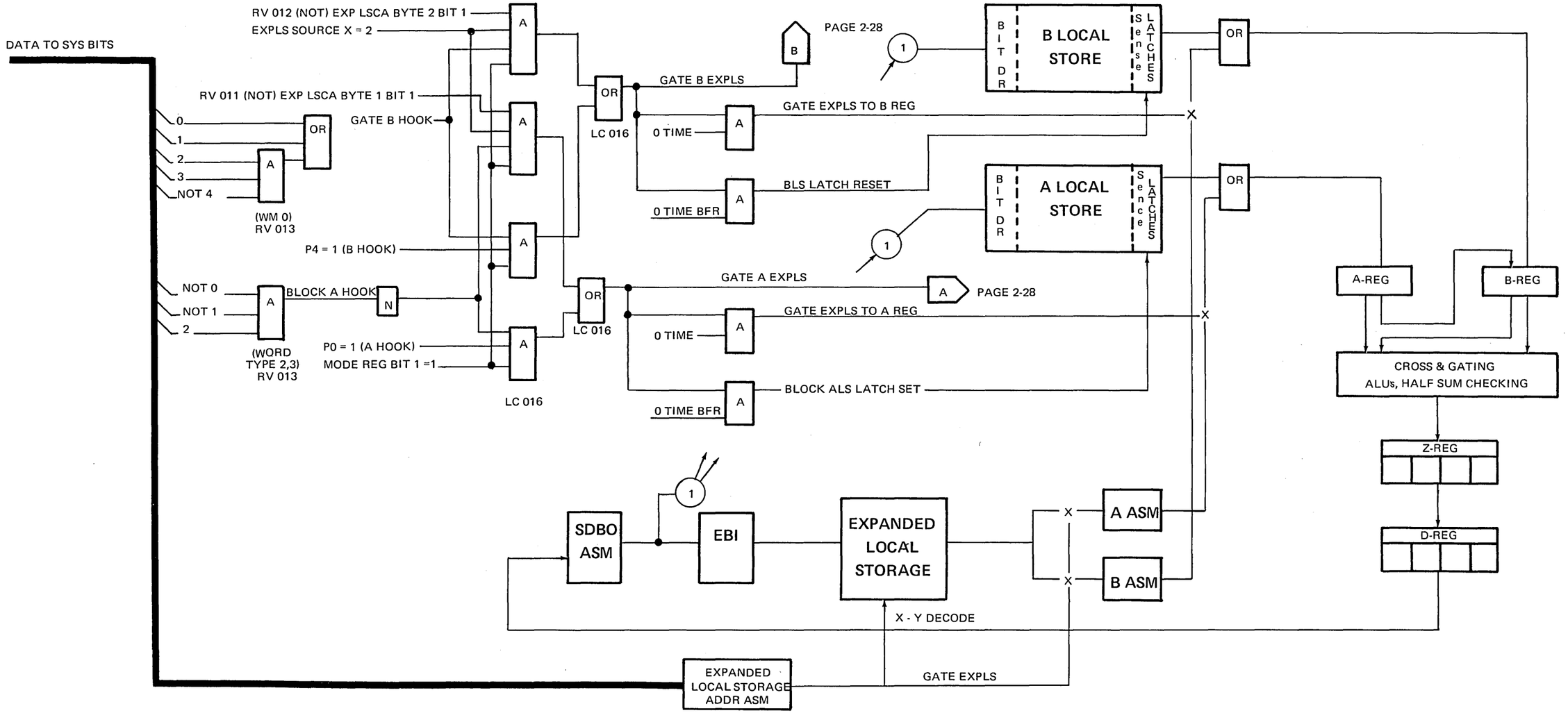
NP2 is the A source
WK1 is the B source

The decode of this word defaults to effectively read

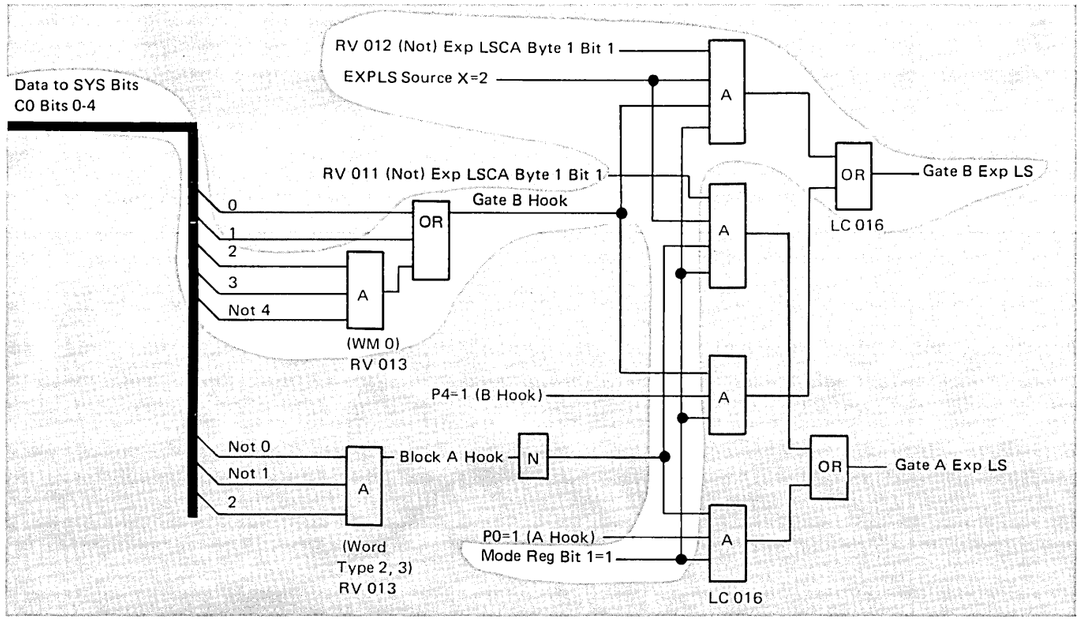
WK1 = NP2, OE, NP20

This type of control word is valid only if the B source is being blocked from ALU entry.




DATA TO SYS BITS
RV 012 (NOT) EXP LSCA BYTE 2 BIT 1
EXPLS SOURCE X = 2
RV 011 (NOT) EXP LSCA BYTE 1 BIT 1
GATE B HOOK
0
1
2
3
NOT 4
OR
A
(WM 0)
RV 013
P4 = 1 (B HOOK)
NOT 0
NOT 1
2
BLOCK A HOOK
N
(WORD
TYPE 2,3)
RV 013
P0 = 1 (A HOOK)
MODE REG BIT 1 =1
LC 016
OR
LC 016
GATE B EXPLS
B
PAGE 2-28
1
0 TIME
0 TIME BFR
GATE EXPLS TO B REG
BLS LATCH RESET
B LOCAL STORE
BIT DR
Sense
LATCHES
GATE A EXPLS
A
PAGE 2-28
GATE EXPLS TO A REG
BLOCK ALS LATCH SET
A LOCAL STORE
X
OR
A-REG
B-REG
CROSS & GATING
ALUs, HALF SUM CHECKING
Z-REG
D-REG
SDBO
ASM
EBI
EXPANDED
LOCAL
STORAGE
A ASM
B ASM
X - Y DECODE
EXPANDED
LOCAL STORAGE
ADDR ASM
GATE EXPLS

EXPANDED LOCAL STORAGE;SOURCE GATING EXAMPLES



Word-Move Version 0

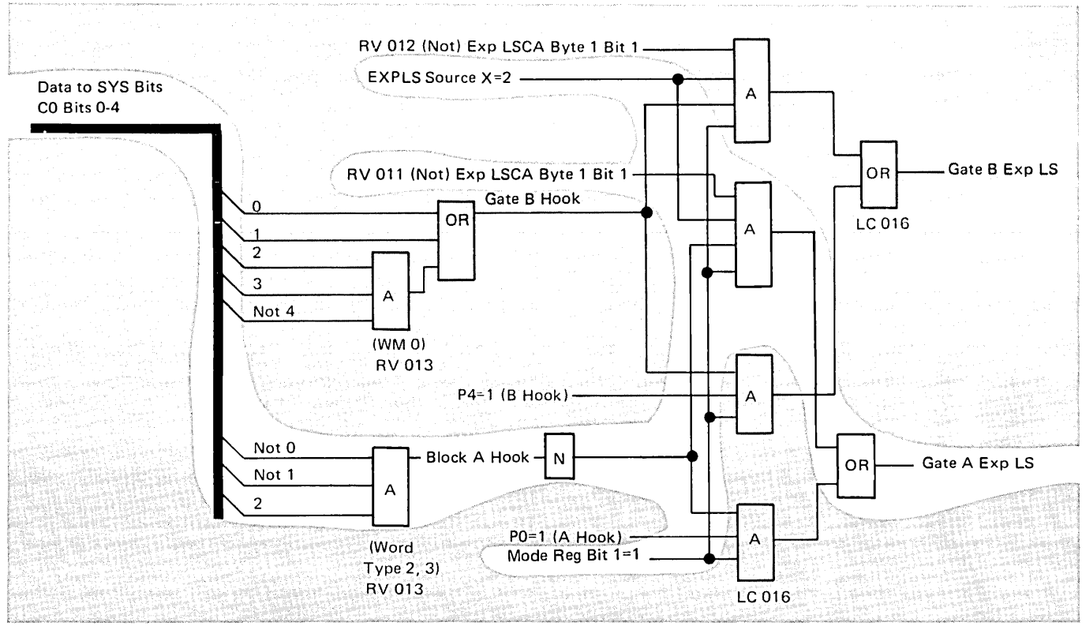
P Low=2 (X2 decode)

Statement C0 Bits 0-4=00110
C2 Bits 0-3=0010

Q=W,D7

In this word-move example, the expanded local-storage register W is the B-source. W is one of the four expanded LS-registers that do not rely on P0 or P4 to bring up a gating line.

The upper AND circuit is activated by this control word. The line 'Gate B Exp LS', gates the W-register (from the B-Asm) to the B local-storage bus-out. This gating line also blocks the B local-storage sense latch set.



Arithmetic Word (A destination)

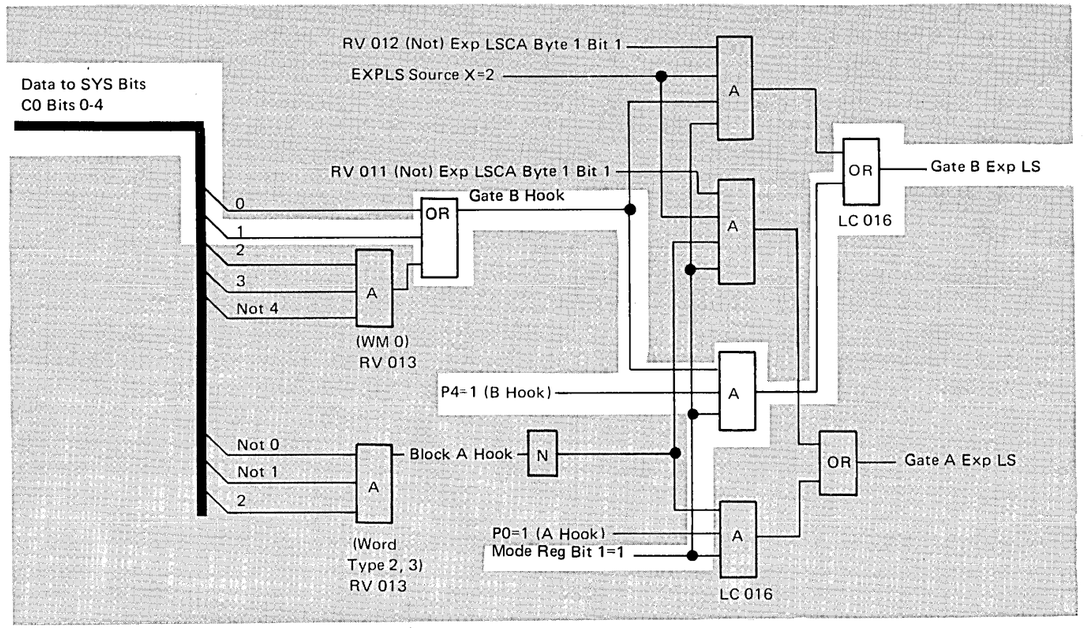
P Low=2 (X2 decode)

Statement C0 Bits 0-4=11000
C1 Bits 0-3=0001

V0=0

The decode of this arithmetic word specifies that the expanded local-storage register V is to be accessed as the A source and is also to be the destination of the arithmetic result.

The 'Gate A Exp LS' line is activated through the AND circuit highlighted in the diagram. The X and Y decode for the V-register is also set up in the A local storage destination latches for use in the following control word cycle.



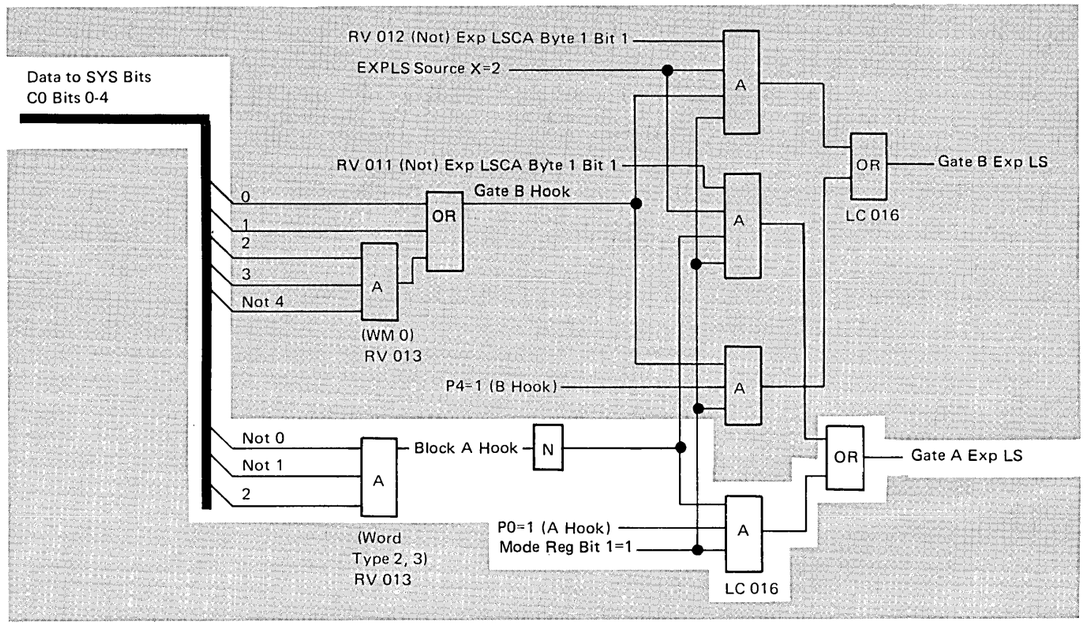
Storage Word (Read Word)

P Low=A (X2 decode)

Statement C0 Bits 0-4=01000
C2 Bits 0-3=0010

RDW RW WK, NOP

The expanded local-storage register WK is the address source for this Read Word. The highlighted AND circuit brings up the 'Gate B Expls' line. The set to the B local-storage sense latches is blocked to prevent any conflict on the B-register input.



Arithmetic Word (A destination)

P=87

Statement C0 Bits 0-4=10001
C1 Bits 0-3=0011

NP2=NP2, OR, K05

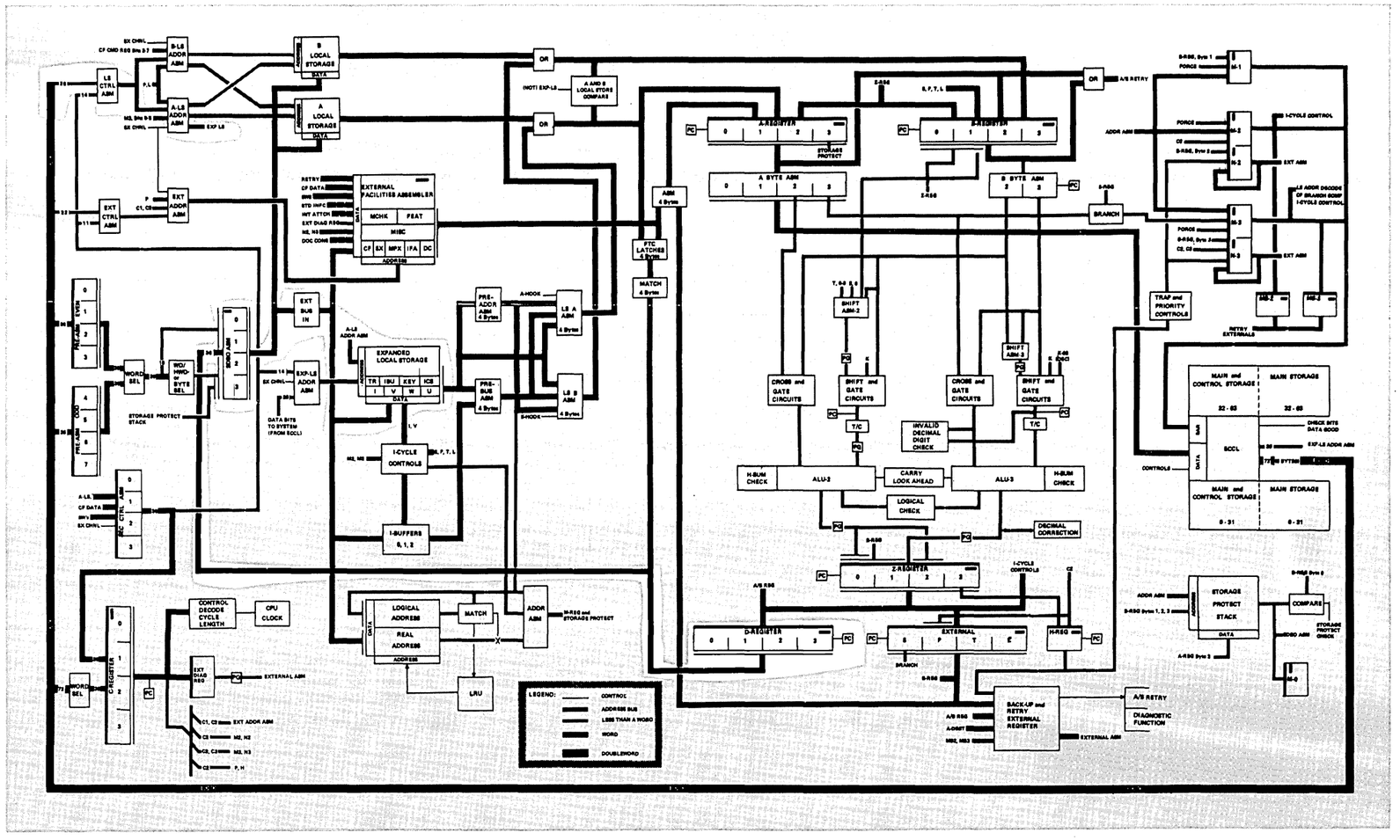
The expanded local storage register NP2 is the A source and destination in this arithmetic word. The 'Gate A Exp LS' line is brought up by the highlighted AND circuit.

The X- and Y-decode for NP is set up in the A local storage destination latches for use in the following control word cycle.

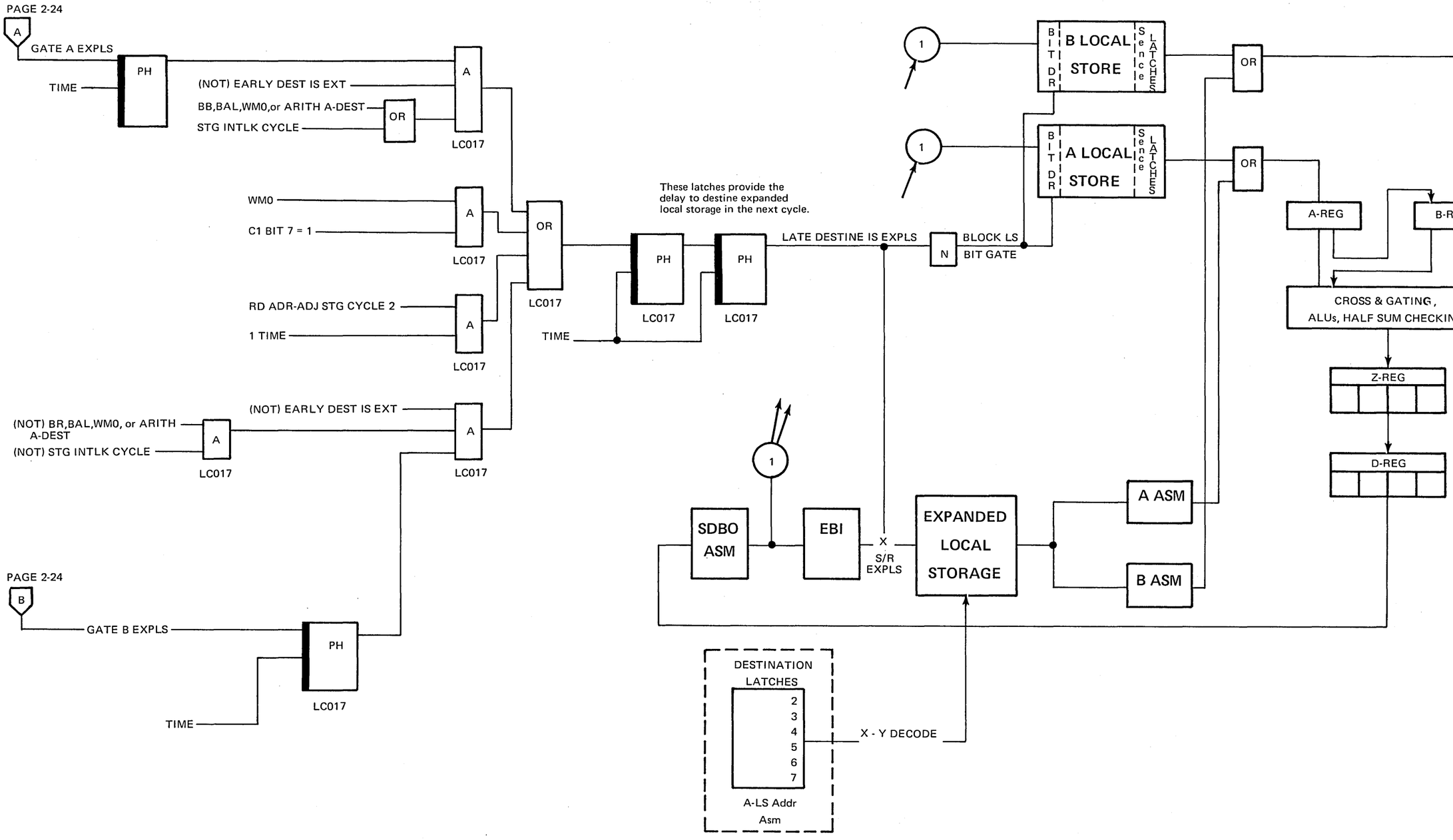
EXPANDED LOCAL STORAGE: DESTINING

When expanded local storage is addressed as a destination, the destination address latches of A local storage are used to retain the address until the following cycle.

At destination time, the bit gates for both A and B local storage are blocked to prevent local storage from being set. The data destined to expanded local storage is gated from the D-register through the SDBO assembler on the EBI to the expanded local-storage register addressed by the A local storage destination latches.



DESTINATION CONTROL



EXTERNAL FACILITIES

- External facilities are composed of registers, buses, status lines, and other circuitry that form the communications line between the microprogram and:
  - Channels
  - Console File
  - Documentary Console
  - Checking facilities
  - Retry circuits
  - Integrated File Adapter
  - Features

- Addresses are formed from: control words, console-file data, selector-channel circuits, console switches, retry information, and local-storage address data.
- Data from the externals enters the data flow through the external assembler to the A-Reg only. Externals cannot be gated to the B-Reg.
- Data to the externals is gated through the SDBO assembler on the External Bus In (EBI).

External facilities have restrictions associated with them because of the manner in which they are used. For example, certain externals cannot be addressed as destinations for data, others cannot be sources of data.

EXTERNAL CONTROL ASSEMBLER **A**

- Receives data from the SDBO early in the CPU cycle to form the source gates necessary to gate-in external data to the data flow in time to be used in source-control-word operation.
- Receives data from the secondary control assembler to form addresses in conjunction with source selector channel, console file, or display operation.

X-Y DECODES **B**

The X- and Y-lines are brought up only for destination addresses. The X- and Y-combinations are routed to the various external hardware locations to bring up set and gating lines.

The X- and Y-lines are checked to assure that one and only one X-, and one and only one Y-line is activated for a destination address. An X compare check sets MCKA3 bit 0. A Y compare check sets MCKA3 bit 1.

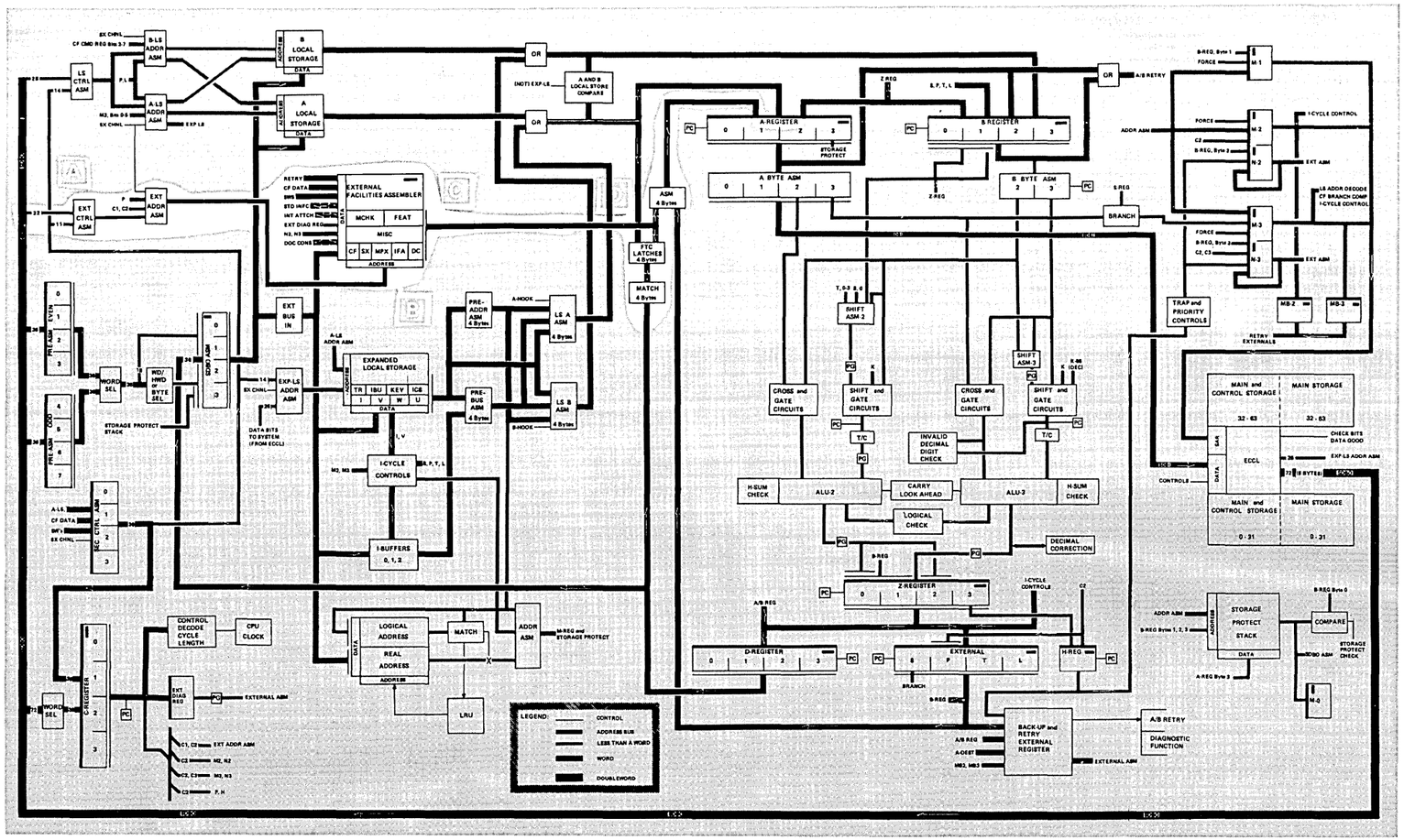
SOURCE ADDRESSING **C**

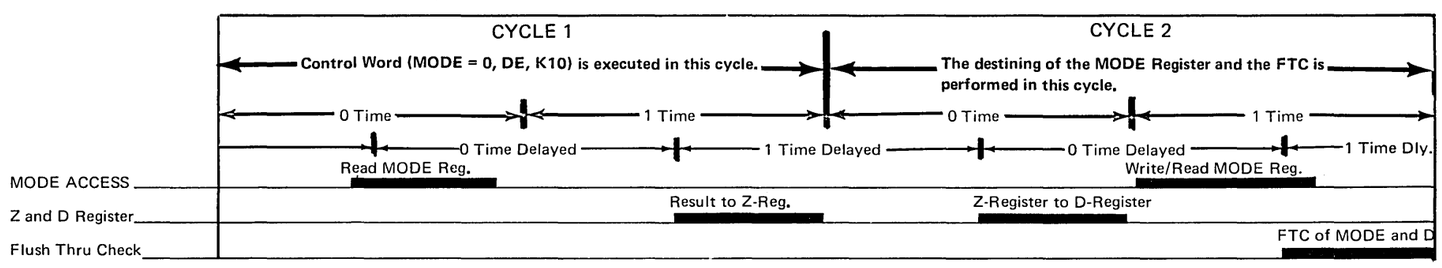
Source addressing is performed by generating gating lines that allow selected external buses to feed through the external or expanded external assembler into the CPU data flow.

FLUSH-THROUGH CHECK **D**

Data destined to some external facility is gated from the D-register through the SDBO asm. out on the External-Bus-In to the selected external facility. The data from the D-register is gated to the flush-through-check match circuit where it is compared with the data from the external that was the destination. If the data does not compare, bit 2 of MCKA1 is set to indicate the error.

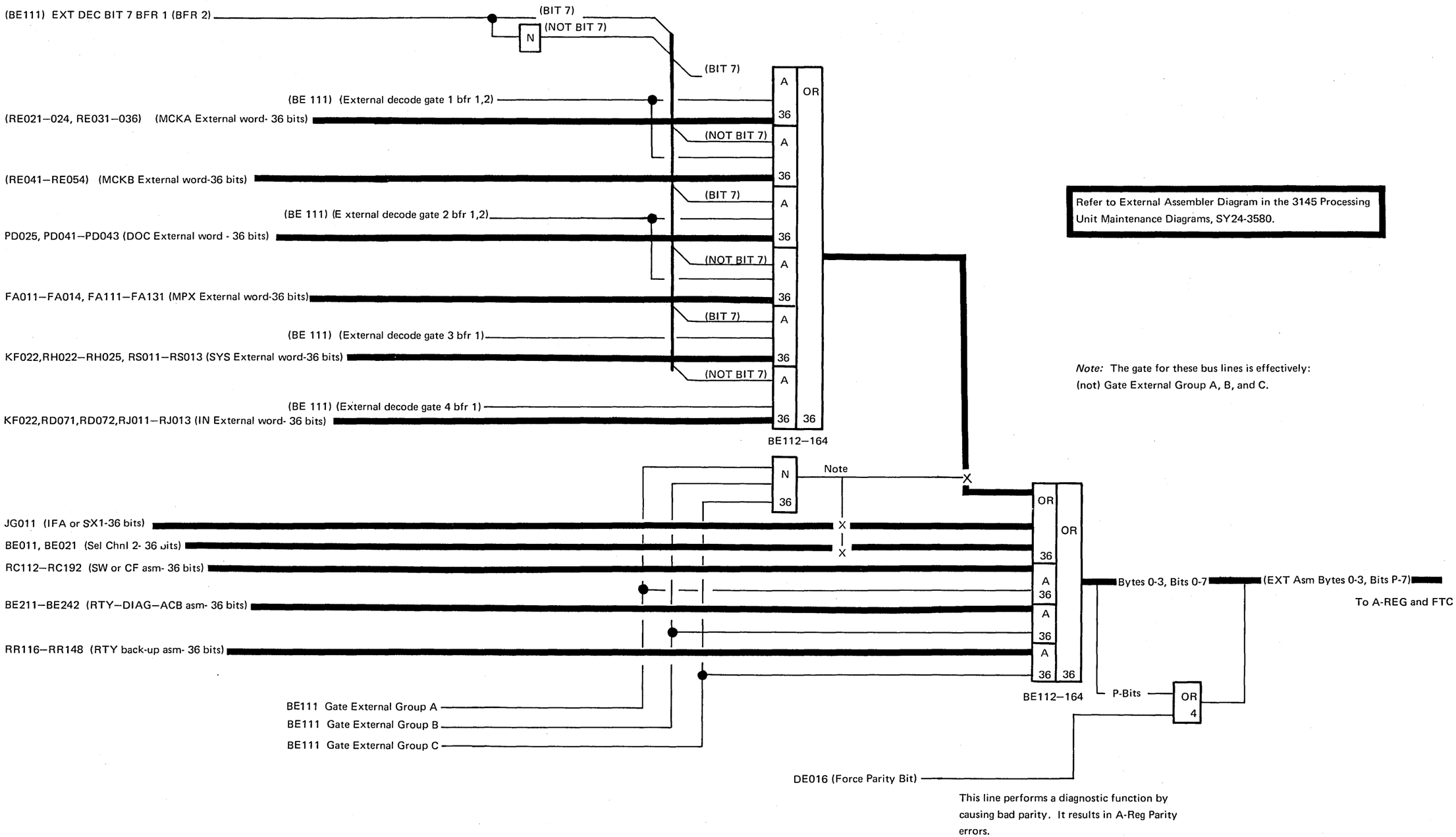
Data gated from the console file to the CFDR is not flush-through-checked.





Example of Source, Destination, and FTC Timing

EXTERNAL ASSEMBLER DATA FLOW

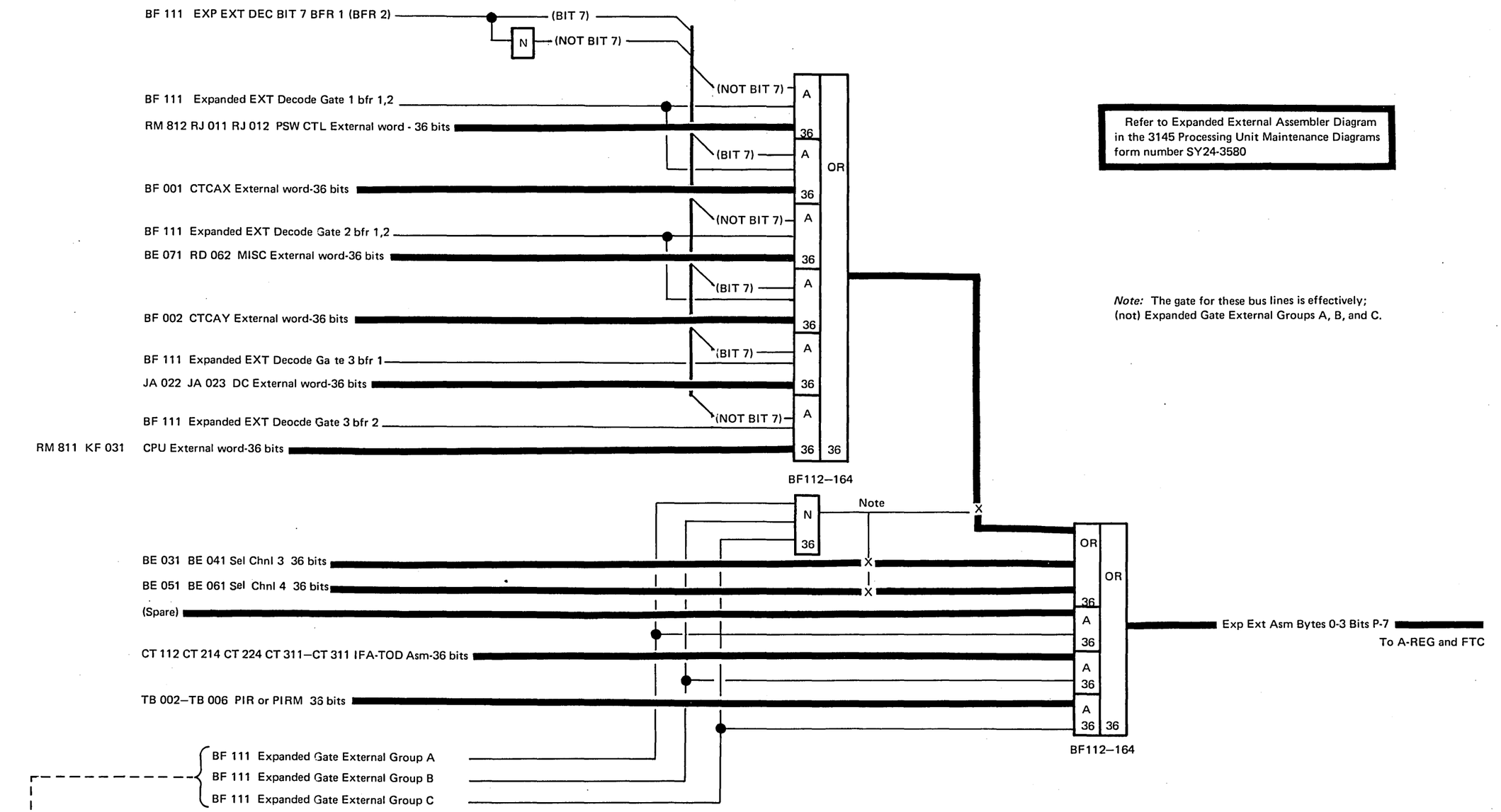


This line performs a diagnostic function by causing bad parity. It results in A-Reg Parity errors.

Refer to External Assembler Diagram in the 3145 Processing Unit Maintenance Diagrams, SY24-3580.

Note: The gate for these bus lines is effectively: (not) Gate External Group A, B, and C.

EXPANDED EXTERNAL ASSEMBLER DATA FLOW



Refer to Expanded External Assembler Diagram in the 3145 Processing Unit Maintenance Diagrams form number SY24-3580

Note: The gate for these bus lines is effectively; (not) Expanded Gate External Groups A, B, and C.

NOTE: These external group gates are not expanded local storage

EXTERNAL ASSIGNMENT AND INDEX MAP

Although there is only one SPTL word, and only one SYS word, they appear in every eight-word group. These two registers have direct-type addressing and are accessible with any P-register setting. SPTL is addressed when the hex digit C is specified in either the A-source or the B-source fields of a control word. SYS is addressed when the hex digit D is specified in the A-source field.

Bit MCKA and MCKB are set to zero when MCKA is used as a destination in a word-move word, with the NOREG as the source.

WORD ADDRESS	WORD NAME	BYTE 0	BYTE 1	BYTE 2	BYTE 3	X Y LINE
00	RTY	MB	MB 3	ECNT	RCNT	0 0
01	NOREG	NOREG0	NOREG1	NOREG2	NOREG3	0 1
02	DIAG	DIAG0	DIAG1	FEAT 2	FEAT 3	0 2
03	XXXXXXXX	XXXXXXXX	XXXXXXX	XXXXXXX	XXXXXXXX	0 3
04	SPTL *	S-REG	P-REG	T-REG	L-REG	0 4
05	SYS *	SYS0	SYS1	SYS2	H-REG	0 5
06	MCKB *	MCKB0	MCKB1	MCKB2	MCKB3	0 6
07	MCKA	MCKA0	MCKA1	MCKA2	MCKA3	0 7
08	CPU	MODE	CFDAR	LRUM	MATCH	1 0
09	CFDR	CFDR	CFDR	CFDR	CFDR	1 1
0A	ACB	ACB0	ACB1	XXXXXX	XXXXXXXX	1 2
0B	SW	SW0	SW1	SW2	SW3	1 3
0C	SPTL *	S-REG	P-REG	T-REG	L-REG	1 4
0D	SYS *	SYS0	SYS1	SYS2	H-REG	1 5
0E	MPX	MT0	MT1	MB1	MB0	1 6
0F	DOC	T1	TA	TT	TE	1 7
10	PSWCTL			MSKA	MSKB	2 0
11	CTCAX	CTCAX0	CTCAX1	CTCAX2	CTCAX3	2 1
12	MISC	EXTINT		EC LEVEL	SER 1	2 2
13	CTCAY	CTCAY0	CTCAY1	CTCAY2	CTCAY3	2 3
14	SPTL *	S-REG	P-REG	T-REG	L-REG	2 4
15	SYS *	SYS0	SYS1	SYS2	H-REG	2 5
16	IN	INTA	INTB	SER2	SER3	2 6
17	DC	DCB0	DCH1	TRBO	DCB1	2 7
18	ABRTY	ABRTY0	ABRTY1	ABRTY2	ABRTY3	3 0
19	SPTLB	SRTY	PRTY	TRTY	LRTY	3 1
1A	HMRTY		HRTY	MRTY2	MRTY3	3 2
1B	CPURTY	BYDST	RTYFLG	LSDST	EXTDST	3 3
1C	SPTL *	S-REG	P-REG	T-REG	L-REG	3 4
1D	SYS *	SYS0	SYS1	SYS2	H-REG	3 5
1E						3 6
1F						3 7

may not be used as a destination

*Not Flush Through-Checked

Both MCKA and MCKB are set to zero when MCKA is used as a destination in a word-move word, with the NOREG as the source.

Address	Described In
00	Chapter 12
01	Chapter 2
02	Chapter 13 (Bytes 0, 1) Chapter 2 (Bytes 2, 3)
03	Do not exist
04	Chapter 2
05	Chapter 12 (Bytes 0, 1, 2) Chapter 2 (Byte 3)
06	Chapter 12
07	Chapter 12
08	Chapter 2
09	Chapter 6
0A	Chapter 2
0B	Chapter 2
0C	Chapter 2
0D	Chapter 12 (Bytes 0,1,2) Chapter 2 (Byte 3)
0E	Chapter 8
0F	Chapter 7
10	Chapter 2
11	Chapter 9
12	Chapter 9 (Byte 0) Chapter (Bytes 2, 3)
13	Chapter 9
14	Chapter 2
15	Chapter 12 (Bytes 0,1,2) Chapter 2 (Byte 3)
16	Chapter 2
17	Chapter 9
18	Chapters 12 and 13
19	Chapter 12
1A	Chapter 12
1B	Chapter 12
1C	Chapter 2
1D	Chapter 12 (Bytes 0,1,2) Chapter 2 (Byte 3)
1E	---
1F	---

EXTERNAL ASSIGNMENT and INDEX MAP

WORD ADDRESS	WORD NAME	BYTE 0	BYTE 1	BYTE 2	BYTE 3	X Y LINE	
20	GBUF FBAK	GBO FWB	GB1 FCH	GB2 FCL	GB3 FOP	4 0	SX1/IFA
21	GBS FCND	GSP FDS	GBF FHC	GCT FED	GBD FMOD	4 1	
22	GSTAT FSTAT	GF FFL	GE FSC	GS FST	GL FGL	4 2	
23	GTAG FTAG	GTO FTO	GT1 FT1	GO FBO	GR FDR	4 3	
24	SPTL *	S-REG	P-REG	T-REG	L-REG	4 4	
25	SYS *	SYS0	SYS1	SYS2	H-REG	4 5	
26	FERR	FSB	FGT	FTS	FAT	4 6	IFA
27						4 7	
28	GBUF FRR	GBO FRRA	GB1 FRRC	GB2 FSC	GB3 FSR	5 0	SX4/IFA
29	GBS	GSP	GBF	GCT	GBD	5 1	
2A	GSTAT	GF	GE	GS	GL	5 2	
2B	GTAG	GTO	GT1	GO	GR	5 3	
2C	SPTL *	S-REG	P-REG	T-REG	L-REG	5 4	
2D	SYS *	SYS0	SYS1	SYS2	H-REG	5 5	
2E	ADJT	LOGICAL ADDR REAL ADDRESS				5 6	
2F						5 7	
30	GBUF	GB0	GB1	GB2	GB3	6 0	SX2
31	GBS	GSP	GBF	GCT	GBD	6 1	
32	GSTAT	GF	GE	GS	GL	6 2	
33	GTAG	GTO	GTI	GO	GR	6 3	
34	SPTL *	S-REG	P-REG	T-REG	L-REG	6 4	
35	SYS *	SYS0	SYS1	SYS2	H-REG	6 5	
36	TODH	TODH0	TODH1	TODH2	TODH3	6 6	
37						6 7	
38	GBUF	GB0	GB1	GB2	GB3	7 0	SX3
39	GBS	GSP	GBF	GCT	SX3	7 1	
3A	GSTAT	GF	GE	GS	GL	7 2	
3B	GTAG	GT0	GT1	GO	GR	7 3	
3C	SPTL *	S-REG	P-REG	T-REG	L-REG	7 4	
3D	SYS *	SYS0	SYS1	SYS2	H-REG	7 5	
3E	TODL	TODL0	TODL1	TODL2	TODL3	7 6	
3F						7 7	

may not be used as a destination

*Not-Flush-Through Checked

Address	Described In
20	Chapter 8 or Chapter 10
21	Chapter 8 or Chapter 10
22	Chapter 8 or Chapter 10
23	Chapter 8 or Chapter 10
24	Chapter 2
25	Chapter 12 (Bytes ϕ, 1, 2) Chapter 2 (Byte 3)
26	Chapter 10
27	---
28	Chapter 8 or Chapter 10
29	Chapter 8
2A	Chapter 8
2B	Chapter 8
2C	Chapter 2
2D	Chapter 12 (Bytes ϕ, 1, 2) Chapter 2 (Byte 3)
2E	---
2F	---
30	Chapter 8
31	Chapter 8
32	Chapter 8
33	Chapter 8
34	Chapter 2
35	Chapter 12 (Bytes ϕ, 1, 2) Chapter 2 (Byte 3)
36	Chapter 9
37	---
38	Chapter 8
39	Chapter 8
3A	Chapter 8
3B	Chapter 8
3C	Chapter 2
3D	Chapter 12 (Bytes ϕ, 1, 2) Chapter 2 (Byte 3)
3E	Chapter 9
3F	---

NOREG Word

This fullword facility is not really a register. It is used to zero-out other locations. For example, if a word-move control word specifies that bytes 1 and 3 of the NOREG are to be moved to a local-storage location, then bytes 1 and 3 of that location are set to all zeros with odd parity.

DIAG Word Bytes 2 and 3

(Feat 2) Byte 2, Bits 0-3	Main Storage Size	
	1 = 112K	4 = 256K
	2 = 160K	5 = 384K
	3 = 208K	6 = 512K
Bit 4	IFA	
Bit 5-6	Channels	(Note: IFA counts as one channel)
	00 = 1	10 = 3
	01 = 2	11 = 4
Bit 7	Word Buffer	
(Feat 3) Byte 3, Bit 0	Spare	
Bit 1	(3215)	
Bit 2	2nd selectric	
Bits 3-5	Spare	
Bit 6	Real Time Channel	
Bit 7	Direct Control	

CPU Word

MODE Register

Byte 0	
Bit 0	Hard-stop latch-control register 14 bit 0
Bit 1	Enable I-cycle and Adr/Adj Ctrl and expanded local storage.
Bit 2	Enable hardware retry
Bit 3	Full recording mode for single-bit failures in main storage
Bit 4	Full recording mode for single-bit failures in control storage
Bit 5	Threshold mode for single-bit failures in control storage
Bit 6	Reserved
Bit 7	Reserved
Byte 1	CFDAR (Console-file data-address register) Track and sector address used by console file.
Byte 2	LRUM
Bits 0-7	Indicate which adr/adj table register was least recently used.
Byte 3	MATCH
Bits 0-7	Indicate which adr/adj table register matches preaddress assembler. (useful only under diagnostic control)

SW Word (Console Switches)

SW0 through SW3 are the rotary console address/data switches:

SWByte	Console Switches
SW0	AB
SW1	CD
SW2	EF
SW3	GH

PSWCTL Word

Byte 0 EPSWA Bit	Name
0	
1	
2	
3	
4	
5	Translation mode
6	I/O master mask
7	External master mask

Byte 1 EPSWB Bit	Name
0	
1	
2	
3	
4	
5	Machine-check mask
6	Wait state
7	Problem state

Byte 2 MSKA Bit	Name
0	Timer mask
1	Interrupt mask
2	External signal mask
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

Byte 3 MSKB Bit	Name
0	MPX channel mask
1	Selector channel 1 mask
2	Selector channel 2 mask
3	Selector channel 3 mask
4	Selector channel 4 mask
5	Reserved
6	Reserved
7	Reserved

MISC Word Bytes 2 and 3

2. EC Level: External register 12 byte 2
   The last two digits in the 370 microprogram EC number are plugged. A test will be performed before the go-no-go test to determine whether the disk being loaded is at the proper level.
3. Serial Number
   External register 12 byte 3 contains the first two digits of the six-digit serial number. These digits are always plugged as 01.

IN Word (Interrupt Register)

An INTA or INTB (interruption) register bit is set on when the corresponding source has an interruption pending *and* the system mask is set to allow such an interruption. Bit names in the INT register are:

Byte 0 INTA Bit	Name
0	Spare
1	Spare
2	Timer
3	External signal
4	System control
5	CPU signal 0
6	CPU signal 1
7	Process stop

Byte 1 INTB Bit N	Name
0	Multiplex channel
1	Selector channel 1
2	Selector channel 2
3	Selector channel 3
4	Selector channel 4
5	I/O interrupt
6	Timer update
7*	External

* External is set on if all of the following conditions exist:

1. An external interruption signal is on (that is, from the EXTINT register of the CPU signal from the SCPU register).
2. The external mask bit = 1 (bit 7 of the EPSWA register).
3. For the timer MSKA bit 0 equals 1 or for externals 1 through 6, MSKA bit 2 equals 1.

Bytes 2,3
Contain the last four digits of the serial number.

ACB (Address Check Boundary) Register

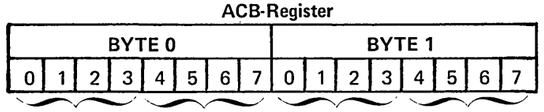
- The ACB register is a two-byte hardware register that contains boundary information used to check main- and control-storage accesses.
- The ACB-register is loaded at IMPL and is reloaded each time the System Reset routine (GRST) is executed.
- The ACB-register is addressed by the external address 0A and can be used as a source or a destination.

The ACB-register is set at IMPL with a specific value determined by the main-storage and control-storage configuration. Certain feature mixes may require additional control storage, above the 32K bytes that are standard. This expansion of control storage is made at the expense of main storage. The movement of the lower control-storage boundary into the main-storage area is done in 2K byte increments. The change in the boundary location between main and control storage results in a different setting for the ACB-register.

Once the feature mix and control-storage size is established, the 370 microprogram disk generated at the plant contains the proper ACB setting for that configuration.

For each access of main or control storage, a comparison is made between the ACB-register and the M-register. If a main-storage access attempts to address the control-storage area, an address check occurs. If a control-storage access is made to a main-storage location, a machine check occurs.

ACB-Register



ACB SETTINGS

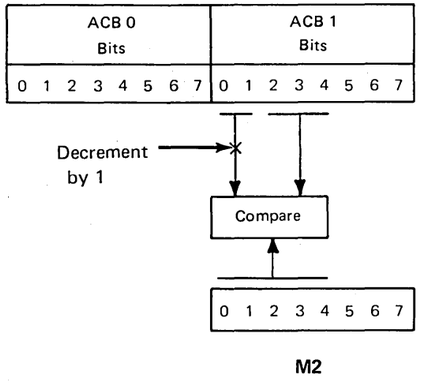
Main Storage	Control Storage	ACB	Main Storage	Control Storage	ACB
112K	32K	01C2	208K	32K	0340
110K	34K	01BA	206K	34K	0338
108K	36K	01B2	204K	36K	0330
106K	38K	01AA	202K	38K	0328
104K	40K	01A2	200K	40K	0320
102K	42K	019A	198K	42K	0318
100K	44K	0192	196K	44K	0310
98K	46K	018A	194K	46K	0308
96K	48K	0182	192K	48K	0300
94K	50K	017A	190K	50K	02F8
92K	52K	0172	188K	52K	02F0
90K	54K	016A	186K	54K	02E8
88K	56K	0162	184K	56K	02E0
86K	58K	015A	182K	58K	02D8
84K	60K	0152	180K	60K	02D0
82K	62K	014A	178K	62K	02C8
80K	64K	0142	176K	64K	02C0
160K	32K	0281	256K	32K	0403
158K	34K	0279	254K	34K	03FB
156K	36K	0271	252K	36K	03F3
154K	38K	0269	250K	38K	03EB
152K	40K	0261	248K	40K	03E3
150K	42K	0259	246K	42K	03DB
148K	44K	0251	244K	44K	03D3
146K	46K	0249	242K	46K	03CB
144K	48K	0241	240K	48K	03C3
142K	50K	0239	238K	50K	03BB
140K	52K	0231	236K	52K	03B3
138K	54K	0229	234K	54K	03AB
136K	56K	0221	232K	56K	03A3
134K	58K	0219	230K	58K	039B
132K	60K	0211	228K	60K	0393
130K	62K	0209	226K	62K	038B
128K	64K	0201	224K	64K	0383

If an external storage frame is attached, the ACB settings for the 256K internal storage configuration are changed to those listed below.

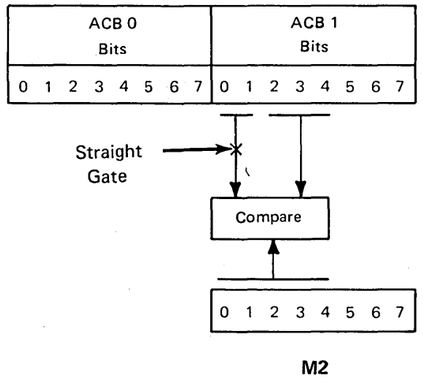
External 128K ACB	External 256K ACB
0404	0405
03FC	03FD
03F4	03F5
03EC	03ED
03E4	03E5
03DC	03DD
03D4	03D5
03CC	03CD
03C4	03C5
03BC	03BD
03B4	03B5
03AC	03AD
03A4	03A5
039C	039D
0394	0395
038C	038D
0384	0385

Note: The ACB setting for each 370 microprogram load may be found in the module chart in the back of the microlisting. Look up address FF08, the ACB setting is in Bytes 0 and 1.

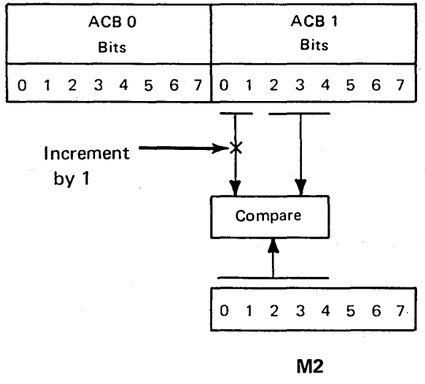
ACB Compare For Control-Storage Access



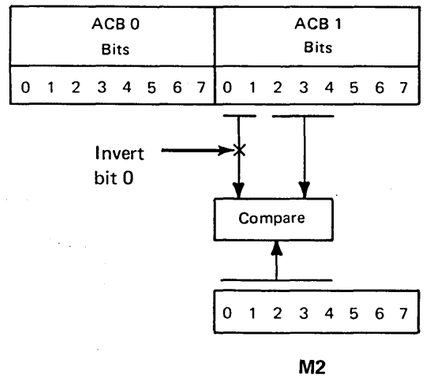
Main-Storage Size 112-K



Main-Storage Size 160K



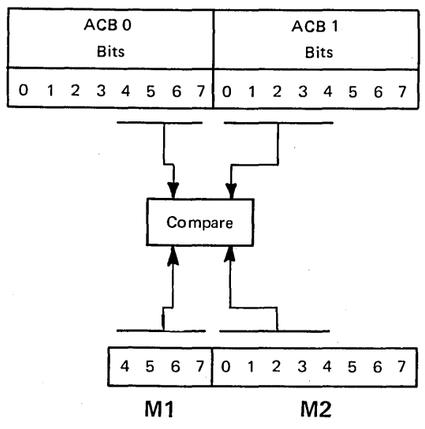
Main-Storage Size 208K



Main-Storage Size 256K and Above

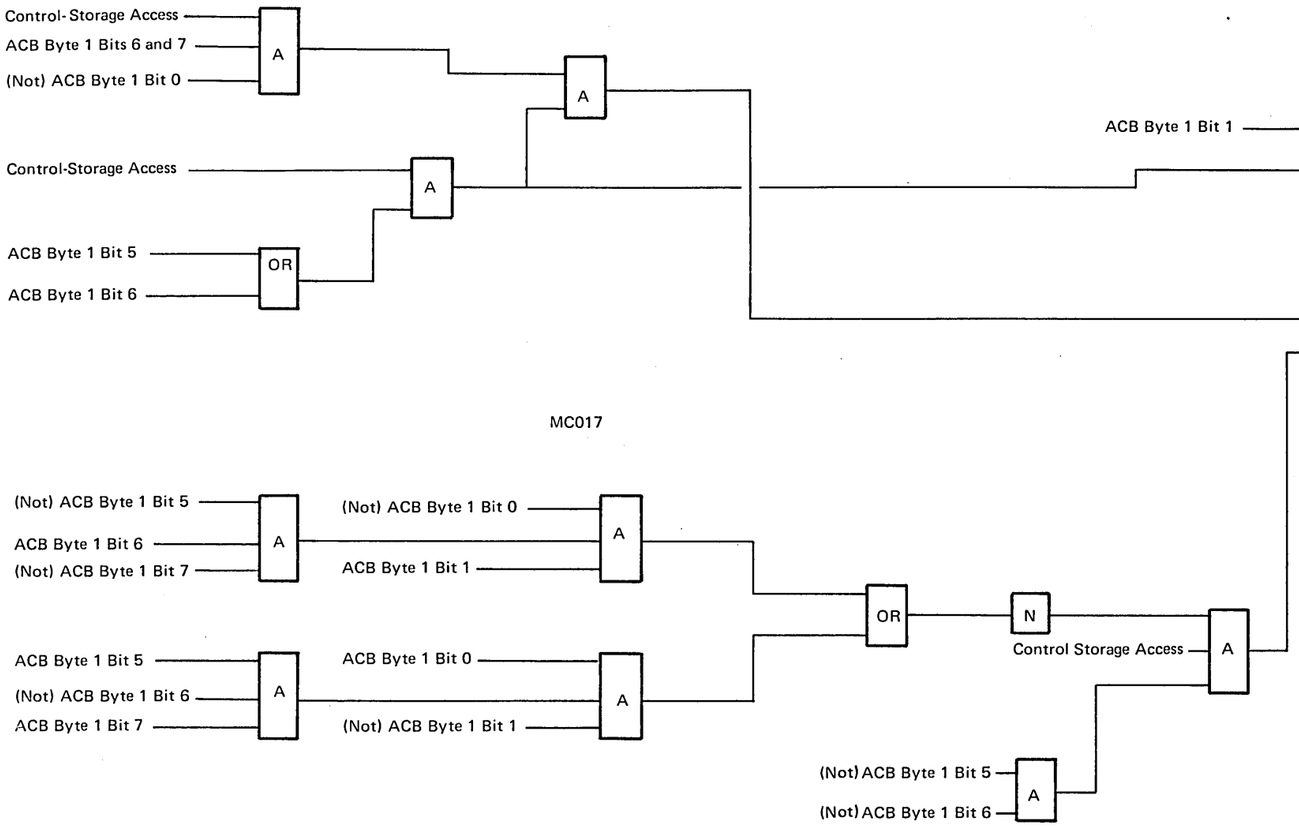
If the comparison indicates that the ACB value is more than the M2 value, a machine-check condition is specified.

ACB Compare for Main-Storage Access



If the comparison indicates that the ACB value is equal to, or less than, the M-register value, an address check occurs.

ACB Byte 1, Bit 0 and 1 Gating To Compare Circuits (Control Storage Only)



Refer to 'ACB and M1-Registers' in the 3145 Processing Unit Maintenance Diagrams Manual Form SY24 - 3580, for high-level diagram of ACB compare

Before comparison is made between the ACB-register and the M-register, for control-storage accesses, the output of bits 0 and 1 of ACB 1 may be altered on the input to the compare circuits. Circuits shown on this page test the status of bits 0, 1, 5, 6, and 7 of ACB 1.

Effectively, the output of bits 0 and 1 to the compare circuits is calculated in the following manner.

Main-Storage-Size	ACB 1 bits 0 and 1
112K	Decrement by 1
160K	Gate straight
208K	Increment by 1
256K (and above)	Invert bit 0.

SYS (System) Register

The system register gives the status or condition of the processor. It is an external hardware register located at word address 05.

System Register

Byte 0	
Bit 0	Machine-check interruption pending
1	Retry routine
2	Machine-check routine
3	Documentary console 2
4	Log present
5	Sub-block protection mode
6	Selector channel start I/O latch
7	Force module 0 to LSCS
Byte 1	
Bit 0	Address contents
1	CPU interrupt force
2	SAR interrupt force
3	PSW restart
4	
5	System control interrupt
6	Timer interrupt force
7	Reserved for priority interrupt
Byte 2	
Bit 0	Enable clear switch
1	IMPL
2	Load file wait bit
3	CE key in CE mode
4	} 00 System reset — 10 subsystem load (IPL)
5	} 01 Power-on reset — 11 system load (multiprocess)
6	Error in stop word
7	Instruction processing latch
Byte 3	H-Register
Bit 0	Machine-check trap
1	Retry trap
2	CPU high trap
3	Integrated file adapter (IFA) if installed Selector channels 1,2,or 3 if no IFA
4	Selector channels 1,2,or 3 if IFA installed Selector channel 4 if no IFA
5	Multiplex channel
6	IFA if installed
7	Store-display

PRIORITY OPERATIONS — H-REGISTER

Priority Operations

Priority operations, which may or may not be related to the current operation, can cause delay of the current microprogram routine. Most (not all) of the various priority operations are initiated by traps. A *trap* is basically a circuit-forced branch out of the current microprogram routine to a priority routine. After the priority routine is completed, a return can be made to the interrupted routine so that its execution will continue.

The interrupted routine is delayed further when several priority operations occur at the same time. Or in some cases, the interrupted routine may be ended by the occurrence of a priority operation. For example, if an instruction address that specifies an unavailable main-storage location is used, an address-check priority operation occurs. The microprogram routine, in which the invalid address is used, is discontinued.

A hierarchy of execution preference exists within the priority-operation structure. Execution preference is exhibited in two instances:

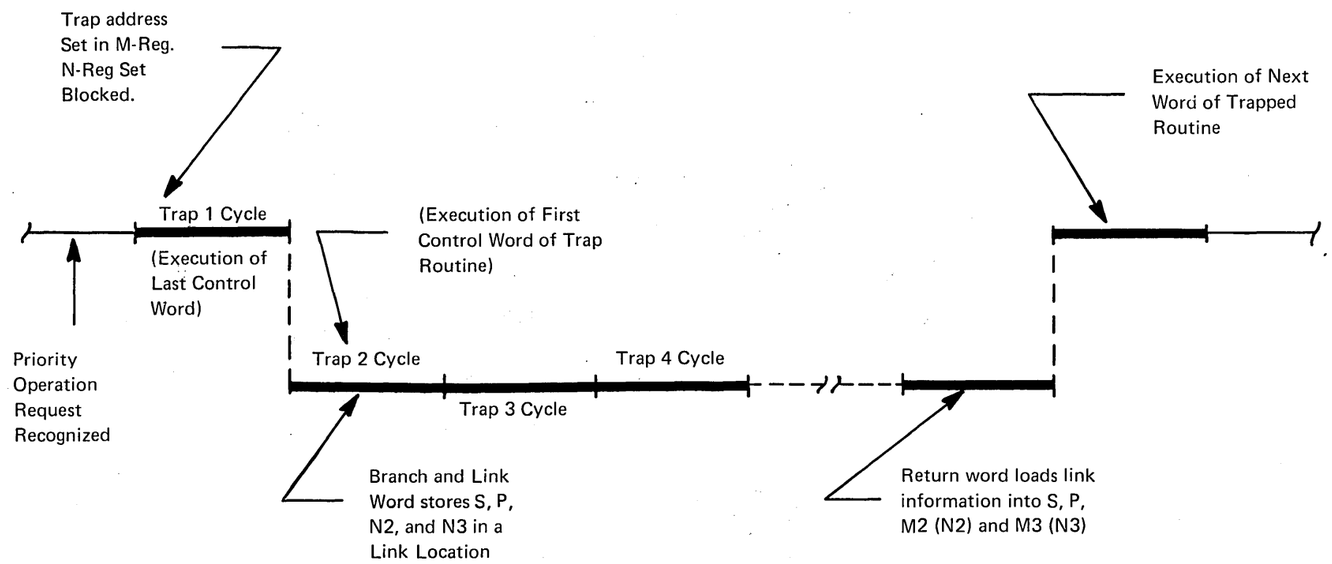
1. Circuit requests for two or more priority operations occur in the same CPU cycle. The highest-priority operation is executed first, the next highest second, etc.
2. During execution of a priority operation, a request for a *higher-* priority operation occurs. The higher-priority operation is executed; the lower-priority operation is delayed until completion of the higher-priority operation, subject to the rules of execution of priority operations specified later in this section.

priority but do not use the trapping mechanism. All of the other priority operations use the trapping mechanism, which functions in the following manner.

1. In a CPU cycle, a priority operation request is recognized.
2. The control-storage address of the first control word in the priority microprogram routine is set into the M-register in what is known as the *trap-1 cycle*, the cycle following the one in which the request is recognized. (The normal next-control-word address, generated by execution of the word in progress, is set into the N-register.) The address set into M is forced by circuitry and is dependent upon the priority operation for which the request is made.
3. The first word of the priority operation is read out of control storage and set into the C-register. Normally, this first word is a branch and link word.
4. The branch and link word stores the contents of S, P, N2, and N3 into a link location in local storage. (N2 and N3 contain the address of the word that would have been executed next if the trap had not occurred). The cycle in which this occurs is called the trap-2 cycle.
5. The priority routine is executed. Normally, the last word in the priority routine is a return word. This word loads the link information back into S, P, M2 (N2) and M3 (N3)
6. Execution of the interrupted routine is resumed at the control word specified by the link (return) address in M2 and M3.

Trap Operation

TRAP 1 CYCLE	TRAP 2 CYCLE	TRAP 3 CYCLE	TRAP 4 CYCLE
Set By: 1. Not Inhibit Traps. 2. Request 0-9 3. 0 Time	*Set By:* 1. Trap 1 INLK latch on. 2. 0-45 Time	*Set By:* 1. Trap 2 INLK latch on. 2. 0-45 Time	*Set By:* 1. Trap 3 INLK latch on. 2. 0-45 Time
Purpose: 1. Prevents any additional traps. 2. Sets TR1 INLK latch at 90-135 time. 3. Prevents normal set to M(N)-Regs. 4. Forces trap address to M2 (N2) & M3 (N3). 5. N2 and N3 set blocked. 6. Execute last word of interrupted routine.	*Purpose:* 1. Forces module address to M2 (N2). 2. Normal M3 (N3) addr update. *3. BAL operation stores S, P, N2, N3 in link Area. 4. TR2 INLK latch set at 90-135 time. *BAL is normally the first word of a Trap Routine.	*Purpose:* 1. Normal address update to M2 and M3. 2. TR3 INLK latch set at 90-135 time. *NOTE: The reason for Trap 3 and 4 cycles is to prevent continuous looping of Trap 1 and 2 cycles if an error is in the SPTL area.	*Purpose:* 1. Reset Block SPTL (Set at TR1 cycle).



H-Register

Many, but not all, priority operations cause an H-register bit to be set on in the trap-2 cycle. The priority operations and associated H-register bits are:

Operation	H-Register Bit	Trap Address
Selector share cycles	none	none
Machine check without I/O	H0	—
a. Normal		D000
b. H0 is already on		D004
c. One or more machine checks have already occurred (SYS0, Bit 2 = 1)		D008
d. H0 and SYS0 Bit 2 are already on		D00C
Machine check with I/O	H0	—
a. Normal		D010
b. H0 is already on		D014
c. One or more machine checks have already occurred (SYS0, Bit 2 = 1)		D018
d. H0 and SYS0 Bit 2 are already on		D01C
Retry	H1	—
a. Normal		D200
b. H1 is already on		D204
c. A retry trap operation is in progress (SYS0, Bit 1 = 1)		D208
d. H1 and SYS0 Bit 1 are already on		D20C
CPU High	H2	—
a. Set IC		D300
b. CA trap		D304
c. Address contents		D308
d. System reset		D30C
Integrated File Adapter	H3	—
a. Mini-Op end		D128
b. Error end		D12C
c. Index		D124
d. Gated Attn or D ADR		D120
Selector Channels 1,2,3 (without IFA)	H3	
a. Exceptional status trap		D120
b. Chaining (CC or CD)		D124
c. UCW handling		D128
d. D ADR trap		D12C

Operation	H-Register Bit	Trap Address
Selector Channels 2, 3/4 (with IFA-SX2,3) (without IFA-SX4)	H4	
a. Exceptional status trap		D100
b. Chaining (CC or CD)		D104
c. UCW handling		D108
d. D ADR trap		
Multiplexer channel	H5	D400
Integrated File Adapter	H6	
a. Return low		D480
b. Unused		D484
c. Unused		D488
d. Diagnostic		D48C
~~Store~~/display (KEY BOARD)	H7	—
a. ~~Store~~/display		D840
CPU low without I/O	None	—
a. Spare		—
b. Storage protect		D804
c. Address check		D808
d. ADR/ADJ exception		D80C
CPU low with I/O	None	—
a. Spare		—
b. Storage protect		D814
c. Address check		D818
d. Spare		—
Scan/Clear	None	—
a. Scan storage		D380
b. Clear storage		D384

The following rules apply to execution of priority operations:

A. A selector share cycle can break into any operation except the first cycle of a storage control word and during an ECC retry of a control word.

B. Any trap priority operation can not break into either:
   1. The first cycle of a storage word operation, or
   2. A trap-2 cycle operation. (That is, trap-2 cycle can not be a trap-1 cycle for another trap.)

C. If H1 is on, all other priority operations (except H0—machine check—and selector share cycles) are prevented. If, however, a diagnostic trap occurs, it will be executed, even though H1 is on. Also, if the system is in a single-cycle mode of operation, a store/display trap can be executed even if H1 is on.
D. If H3 is on, an H3, H4, H5, or H6 trap cannot be taken. If H4 is on, and H4, H5, or H6 trap cannot be taken. If H5 is on, H5 or H6 cannot be taken. If H6 is on, an H6 trap cannot be taken. In any of these cases, the H3, H4, H5, or H6 trap remains pending until after H3 (or H4 or H5) is turned off.
E. Selector share cycles can delay execution of other traps for an indefinite number of cycles, depending upon the rate at which share cycles occur.
F. Discounting the effects of the various non-H-Reg priorities (share cycles, CPU low, scan/clear), the following hierarchy applies.

H-Reg Bit	Blocks Trap Request for H-Reg Bit
H0	None
H1	H2,3,4,5,6,7
H2	H2
H3	H3,4,5,6
H4	H4,5,6
H5	H5,6
H6	H6
H7	H7

Share Cycle Priority Operation (Applies to: Selector Channel, Block Multiplexer, IFA).

Priority operation for selector-share cycles is as follows:

Without IFA

1. The priority sequence is selector channel 1, 2, 3, and 4.
2. A share-request for selector channel 4 will be taken if no other request is pending.

With IFA

1. The priority sequence is selector channel 2, IFA, and selector channel 3.

Machine-Check Priority Operation (H0)

A machine-check trap occurs (if allowed by the machine-check bit in the PSW) because a series of retry operations has been unsuccessful. The number of retry attempts is determined by a hardware counter. Basically, a machine-check trap occurs either because errors are occurring faster than can be handled or because a hard error cannot be successfully retried.

An attempt will be made to form logout information and initiate a machine-check interruption (depending upon the value of the machine-check bit in the PSW). The validity of such logout information may be unpredictable if the machine-check *trap* is called for.

Retry Priority Operation (H1)

The retry routine is entered through the retry priority operation (trap). The retry priority operation occurs when any machine check occurs if the retry counter is not full, retries are not masked off, and system register byte 2 bit 6 (indicates stop word error) is off. Depending upon the nature of the error and the word type, the error may be detected during execution of the failing micro-program word (Type 1), during execution of the following word (Type 2), or may be detectable but uncorrectable (Type 3).

CPU High-Priority Operation (H2)

System Reset Microprogram

The system-reset microprogram is executed after a circuit system reset has been performed. This action is initiated by operation of:
1. The system reset key,
2. The power-on key, when power is off, or
3. The load key.

The circuit system reset causes various CPU registers and controls to be reset.

Integrated File Adapter High-Priority Operations (H3)

Four trap addresses are provided: one for Mini-Op End, one for Error End, one for Index, and one for Gated Attention or D ADR.

Selector Channels or Block-Multiplex Channels 1, 2, and 3 (H3)

When IFA is not present, four trap addresses are provided for channels 1, 2, and 3: One for Exceptional Status Trap, one for Chaining (CC or CD), one for UCW Handling, and one to protect the next entry of the D ADR list.

Selector Channels and Block-Multiplex Channels 2, 3/4 (H4)

Four trap addresses are provided: one for Exceptional Status Trap, one for Chaining (CC or CD), one for UCW Handling, and one to protect the next entry of the D ADR list. When IFA is preset, this trap is shared by channels 2 and 3. For nonIFA, this trap is for the sole use of selector channel 4.

Multiplexer Channel (H5)

This trap is for the sole use of the multiplexer channel for handling data, status, and chaining functions.

Integrated File Adapter Low-Priority Operations (H6)

Four trap addresses are provided: one for Return Low, and one for Diagnostics. The use of the other two is not assigned yet.

Store/Display (H7)

Only one trap can be in operation at any one time. Store/Display pertains to printer-keyboard operations.

CPU Low (No H-Register Bit)

Address Check
This trap occurs when an access to an unavailable main-storage area is attempted.

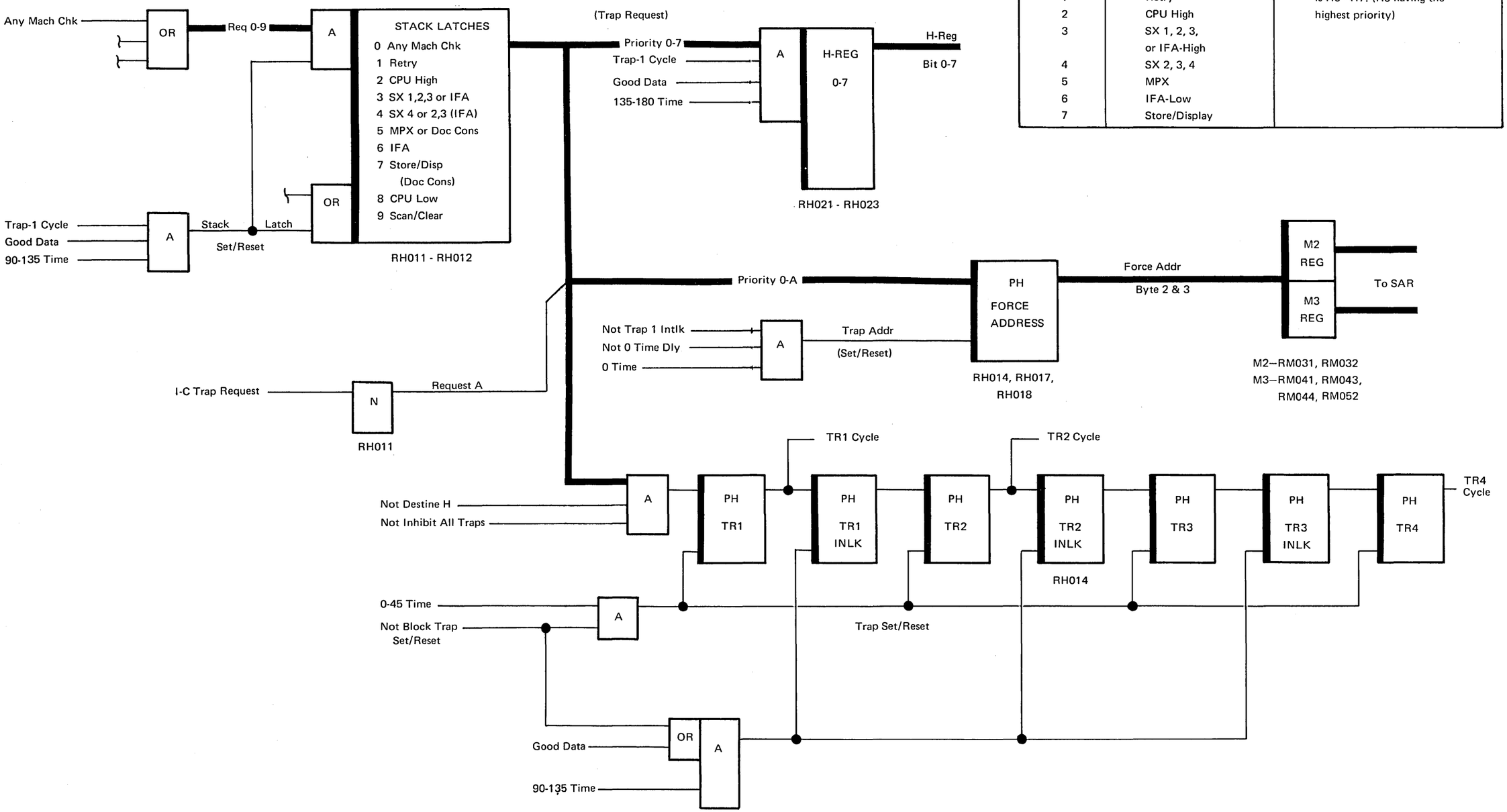
Storage Protection
This trap occurs because of a storage-protection violation.

Address Adjustment Exception
This trap is used with DOS emulator.

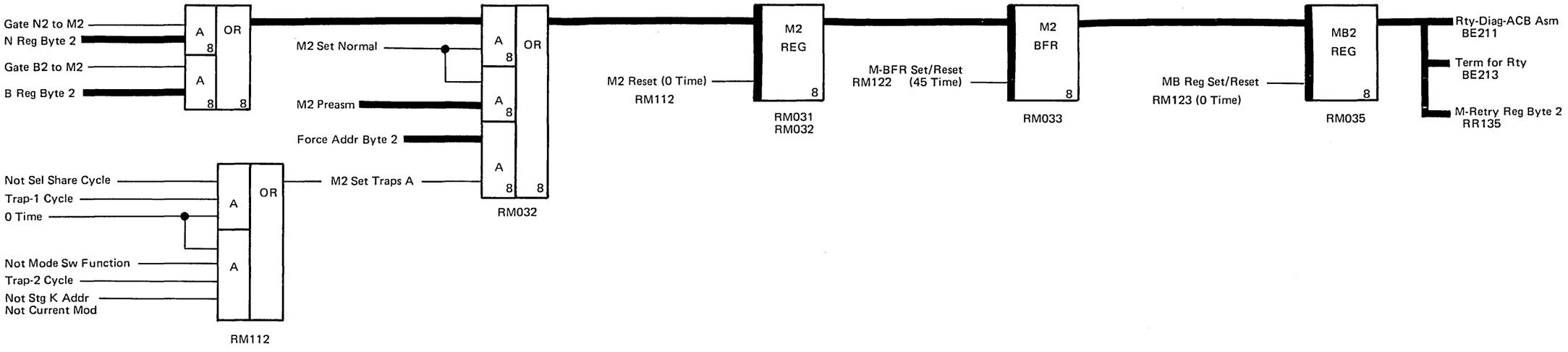
Scan/Clear (No H-Register Bit)
These traps are used for a clear-storage and a scan-storage operation.

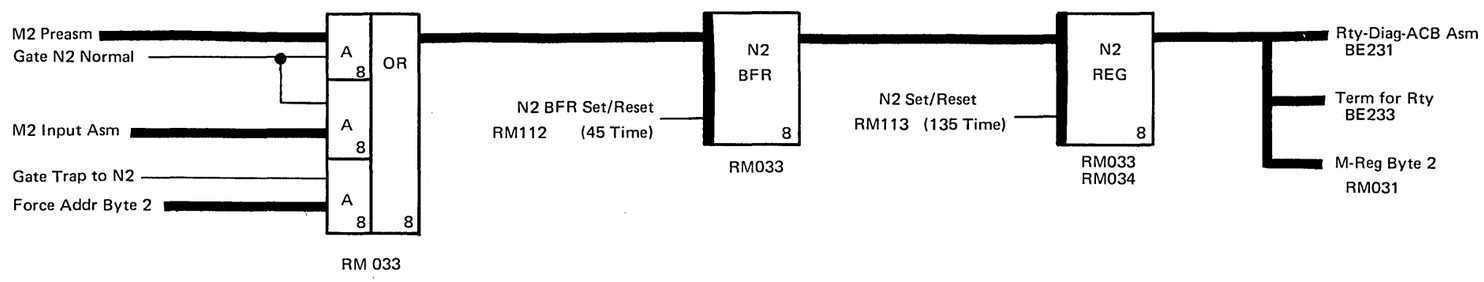
Priority and Trap Controls

H-REG	OPERATION	
Bit 0	Mach Check	*Note:* The priority sequence is H0 - H7. (H0 having the highest priority)
1	Retry	
2	CPU High	
3	SX 1, 2, 3, or IFA-High	
4	SX 2, 3, 4	
5	MPX	
6	IFA-Low	
7	Store/Display	

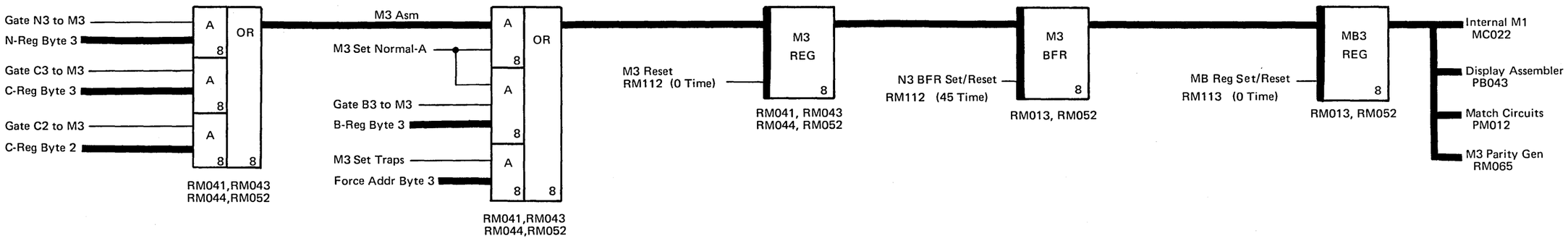


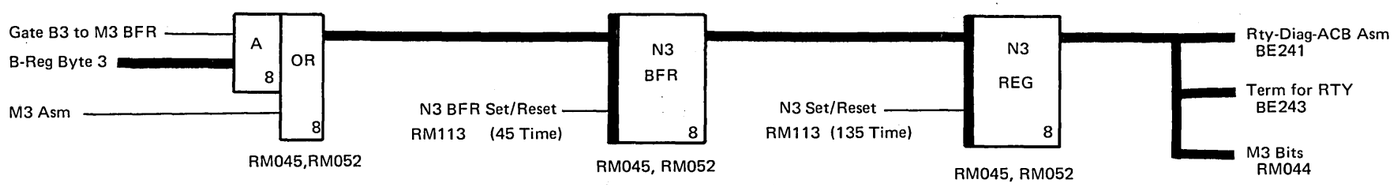
M2 Gating (Traps)





M3 Gating (Traps)





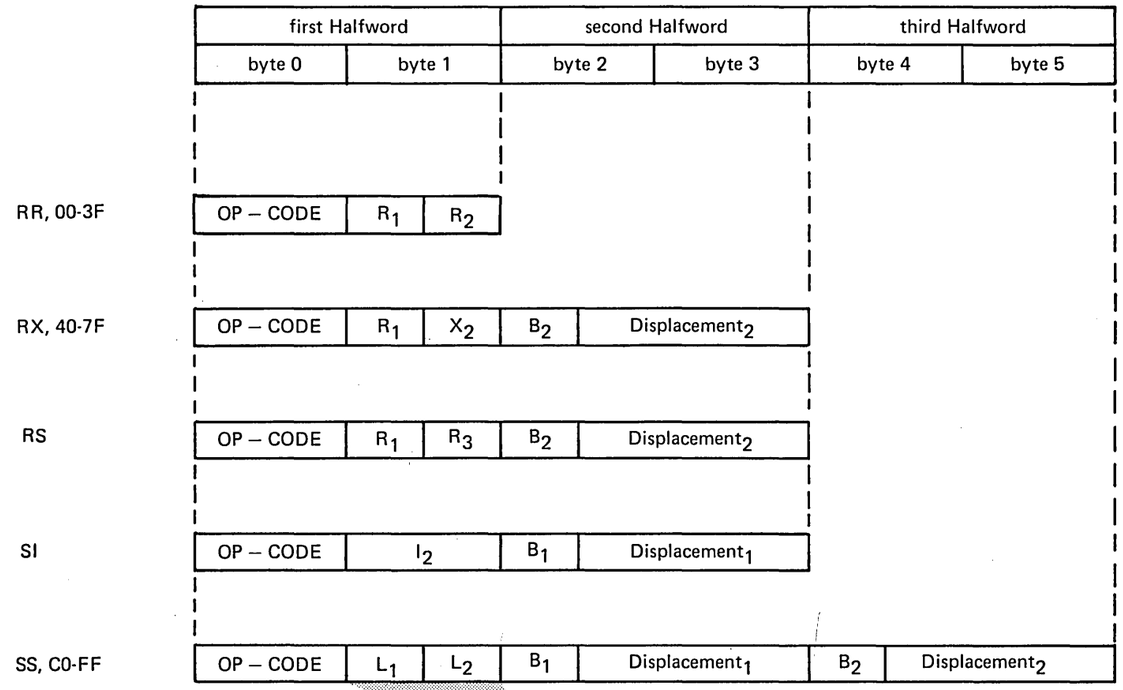
I–CYCLES

Processing a single software instruction may be divided into two parts: the I (instruction) phase and the E (execution) phase. Instructions are defined to be in different groups according to their format, length, and general form of execution.

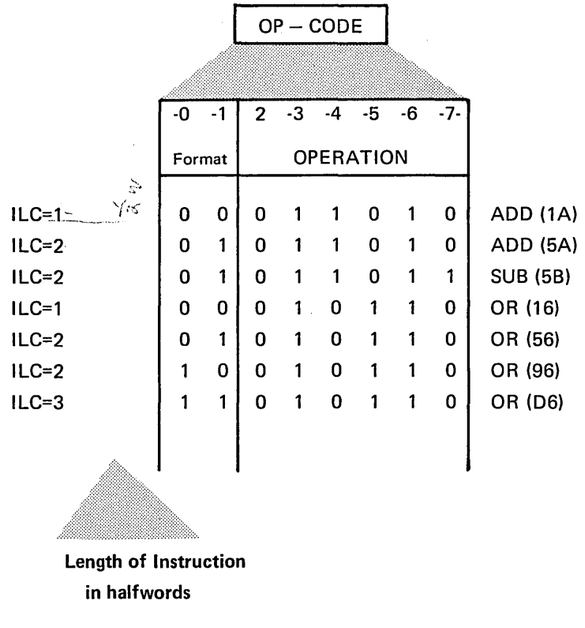
The I-phase of processing performs the following basic functions:

- Fetch instruction
- Initialize the CPU facilities for the completion of the processing.

During the E-phase, the CPU performs the unique functions specified by the instruction Op code.



The IMM BYTE is the byte following the OP CODE



Length of Instruction in halfwords

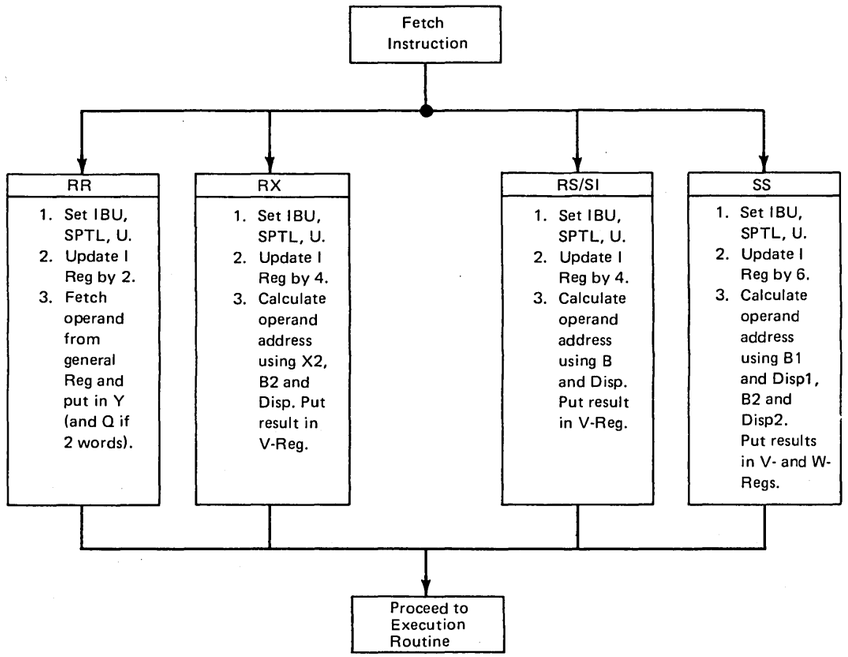
I-PHASE FUNCTIONS

The initialization of CPU facilities for the E-phase is partially dependent upon instruction type. All instructions require an updating of the instruction counter, the setting of the specified CPU Regs, and a branch to the start of the execution routine. In addition, some instructions require the fetching of the second operand from a general register, or the calculation of operand addresses.

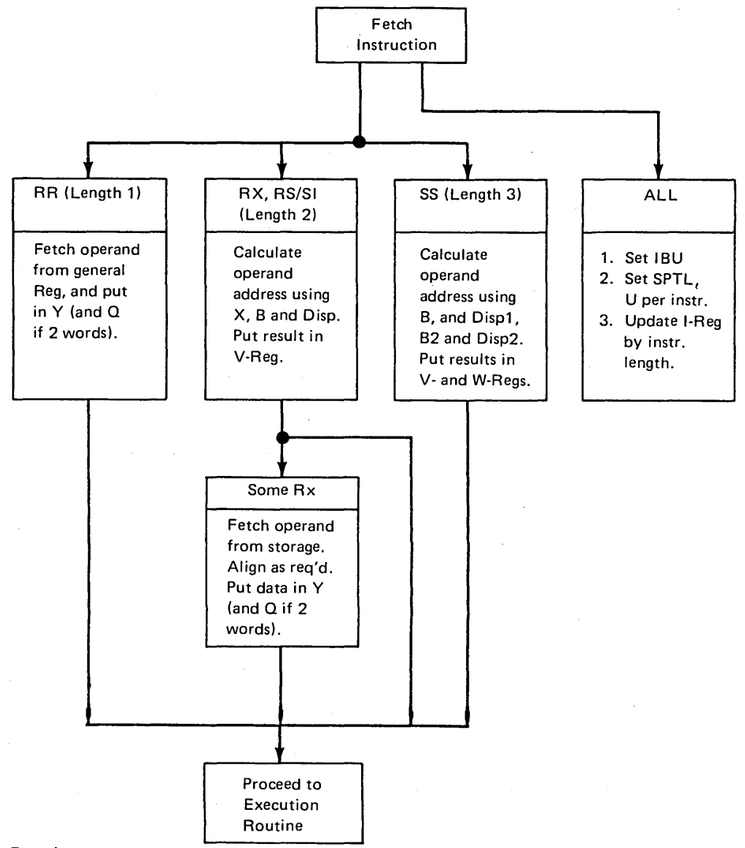
At this point, some observations may be made about the I-phase functions. For example, the RX and RS/SI functions are very similar. In fact, during the I-phase, an RS/SI instruction is handled exactly the same way as an RX instruction with the X2 field equal to zero. Also, some functions are identical, with only the data value being dependent upon the format and Op code (SPTL, U, and I update.)

It is also noted that the E-phase for some instructions is identical; such as, AR and A, NR and N. The difference between these RR and RX types of instructions occurs only in the source of the second operand (general register or storage). It is possible to save some control-storage words, and time, by including the operand fetch as an I-phase function for such RX format op codes.

The I-phase functions may now be illustrated as follows:



Basic I-Phase Functions



I-Phase Functions

Hardware Functions

Each software instruction processed requires performing the previously mentioned I-phase functions. It is obviously desirable to minimize this time and thereby reduce the time required to process a given instruction. To minimize the number of machine cycles required during the I-phase (that is, the I-cycles,) some functions are performed by hardware. Additionally, some other characteristics of the machine are more fully exploited by hardware means.

First consider the previously defined I-phase functions which apply to "ALL" instructions. Hardware is used to perform the setting of IBU, SPTL, U, and the I-Reg update. These functions do not require microwords to be performed; hence, they do not require any additional time during I-cycles.

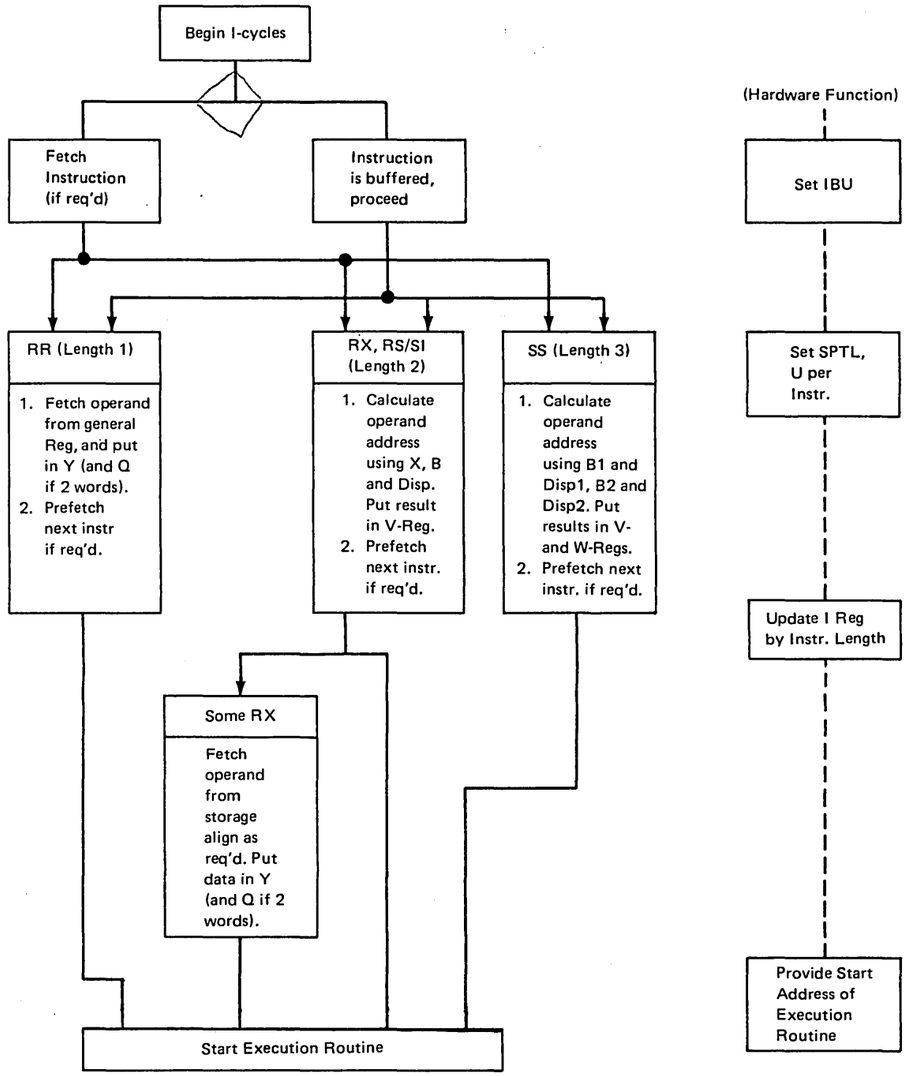
Now consider the function Proceed to Execution Routine. By strictly defining the starting control-storage address of each execution routine as a function of instruction Op code, it is possible to perform a hardware-forced branch. The hardware branch on the Op code does not require any additional time, because there is no microword used to perform the branch and module switch.

The interface between storage and the CPU provides a doubleword transfer of eight bytes of data. During a read type of microword, the SDBO assembler provides the selection of the addressed word (halfword, or byte) from the doubleword actually read. The I-cycle hardware provides for buffering the entire doubleword from storage, via a time-slotting of data from SDBO to EBI. When an instruction is fetched from storage, the addressed word is routed from SDBO to the I-buffer, via EBI, during the normal destination time in storage 2 cycle. During the next cycle time, the odd word is gated to EBI, and placed in the I-buffer. This time-slot action occurs when there is no decrement count function specified by the storage microword. It is, therefore, possible to buffer up to two words of data from the instruction stream when fetching one instruction from storage. Upon completion of the instruction being processed, the next instruction may be available in the buffer and, therefore, need not be fetched from storage.

A savings in processing time becomes obvious, especially if the double word being buffered represents four RR instructions. The concept of buffering a portion of the instruction stream will now be extended to include pre-fetching. Although the buffer does speed subsequent instructions, the fetch of the first (current) instruction does require some time. It is always more desirable to have the current instruction resident in the buffer. To get to this I-buffer condition, it is necessary to have obtained the instruction at some point during the previous instruction. This function of reading the next instruction from storage to the I-buffer is termed prefetching and is performed during I-cycles.

As described in the Expanded Local Storage section, the TR Reg always contains a value representing the next doubleword address beyond the current I Reg value. The TR Reg is always used as the storage address during a prefetch, and the SDBO time-slotting is forced to provide the even word; then odd word. This guarantees the I-buffers to be loaded with sequential data.

The I-phase functions at this point are:



Microcode-Hardware Functions

The complexity of the I-cycle functions have now been increased. Where, on a previous diagram, it was necessary to select one of three paths after fetching the instruction, it is now also required to:

- determine source of instruction (storage vs buffer).
- select path if instruction is in the I-buffer.
- determine whether a prefetch is required.

Additionally, it is desired to minimize the time required to perform both the above functions, and the basic I-phase functions.

Microcode branch operations requires CPU time. I-cycle hardware can force a control-storage address to the M-Reg. (for example branch on the op code). This facility is expanded to include all addressing within I-cycles. The I-cycle hardware provides the starting address of the I-cycle routine to the M-Reg, as a function of I-buffer status, instruction format and prefetch requirement. When the CPU encounters a RTN word (to I-cycles) this address is set into the M-Reg and the I-phase of the instruction begins. Thereafter, except for some parts of the RX-align routine, the I-cycle hardware provides the next control-storage address and a gating signal to the M-Reg, until the execution routine has begun. The I-cycle hardware than initializes for the start of the next I-phase.

The minimization of time spent performing the basic I-phase functions requires more hardware control. The I-cycle hardware controls the data inputs and setting of the SPTL Regs, and uses this facility to select general registers from local store. By setting the P-Reg to a value of 02 (or 62) and gating a portion of the instruction to the L-Reg (R2, X, or B), the microcode can indirectly address any general register, including floating-point registers. This gating to SPTL is done by hardware, and is, therefore, transparent to the microcode. Note that it is necessary to set SPTL to the desired value one machine cycle before use.

Furthermore, the I-cycle hardware can force and/or block gating of data through a portion of the Expanded Local Store. This capability is utilized as follows: a microword is executed performing the arithmetic operation of V = LL+V. During the previous machine cycle, a value is set into the P- and L-Regs of: P = 02, and L = R1B2 (for an RX format instruction). The underlined data value gives the general register specified by the B2 field of the instruction as the data source of the A-Reg. Although the microword is attempting to source the V-register, the I-cycle hardware blocks this Expanded Local Store source, and forces the Disp2 field through the gating to the B-register. The microword function of adding the A- and B-Regs is then completed, with the result destined to the V-register. Thus, the microcode and I-cycle hardware are combined to perform the function of:

V = Base + Displacement

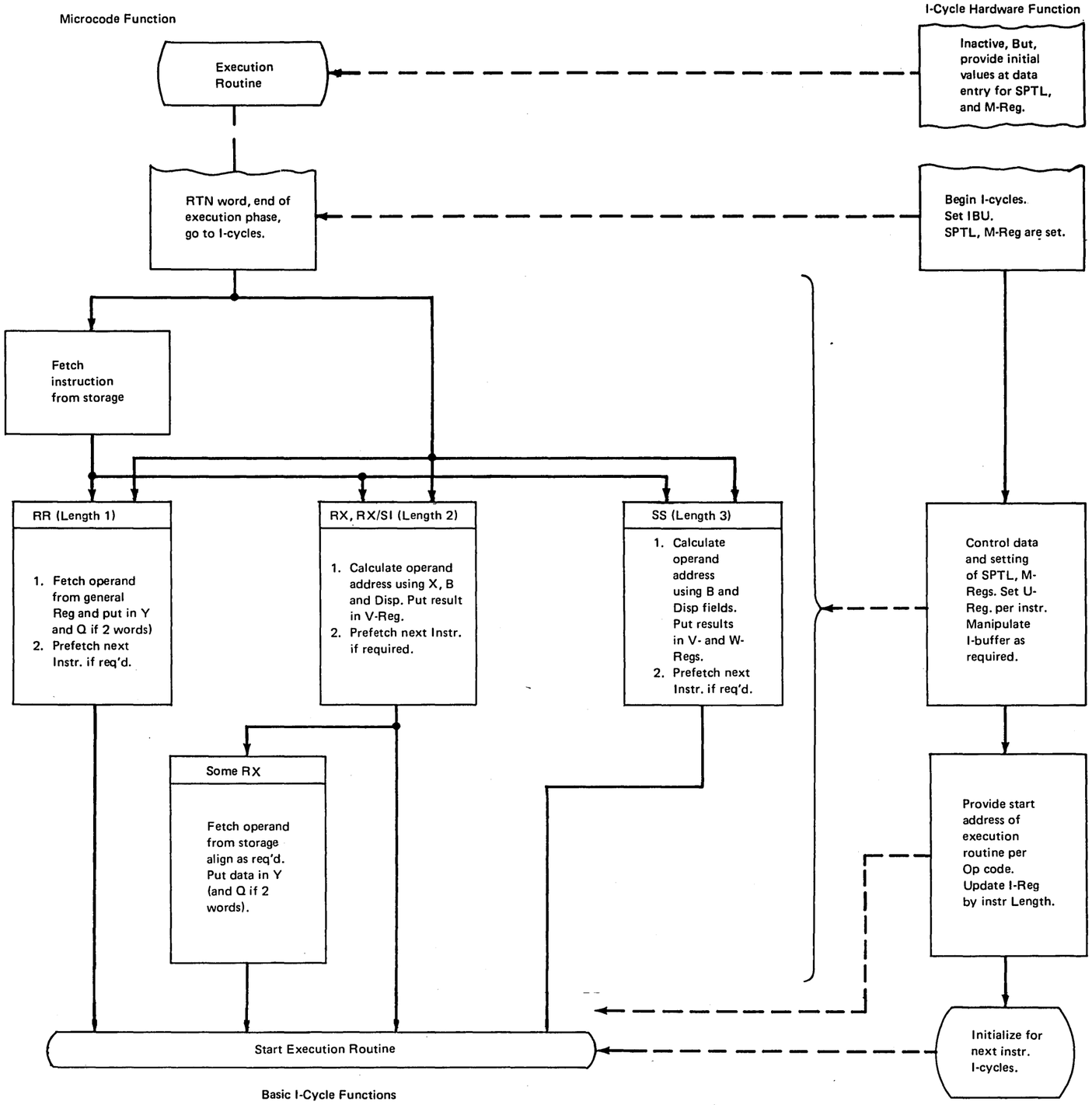
Most of the I-cycle microcode/hardware functions are performed this way.

Another significant interaction of microcode and hardware occurs when prefetching and calculation of an operand address are performed simultaneously. The microword executed during a prefetch is of the form: RDW LL ADJ, V + 4. First, the V-Reg is blocked as an address source. Then, the TR-Reg is substituted as the address source for the M-Reg, with gating performed via the PAA, and I-cycle/ADR ADJ path to the M-Reg. At the same time, the Disp field is gated from the I-buffer through the Expanded Local Store to the B-Reg. The A-Reg data source is an indirectly addressed general register, as previously described. This form of microword normally performs an update of the B-register value; however, this function is blocked and changed to an A + B operation. The function; V = Base + Displacement, is, therefore, performed during the storage 1 cycle. During storage 2 cycle, the destination of data to local store is blocked, and SDBO is time-slotted to the I-buffer, as previously described.

Because the I-cycle hardware and microcode are expected to operate simultaneously, the microcode must consist of specific microwords at fixed addresses. This is obvious because the hardware is providing the control-storage address for the microwords, and then performing hardware functions coincident with the microword execution. What has not been obvious is how the hardware remains in sync with the microcode. This function is performed by routing the M-Reg output back to the I-cycle hardware. Thus, when the M-Reg contains a value corresponding to the control-storage location of an I-cycle microword, the I-cycle hardware can determine what functions are to be performed at the next 0 time (that is coincident with the microword execution).

This introduction has provided the basic functional concepts of the hardware I-cycles. Significant omissions include trapping, share cycles, correction cycles, error conditions, etc. It is expected that the hardware description in the rest of this section will have sufficient explanation for these conditions. The basic I-cycle functions are described by the following flow diagram.

Microcode-Hardware Relationship



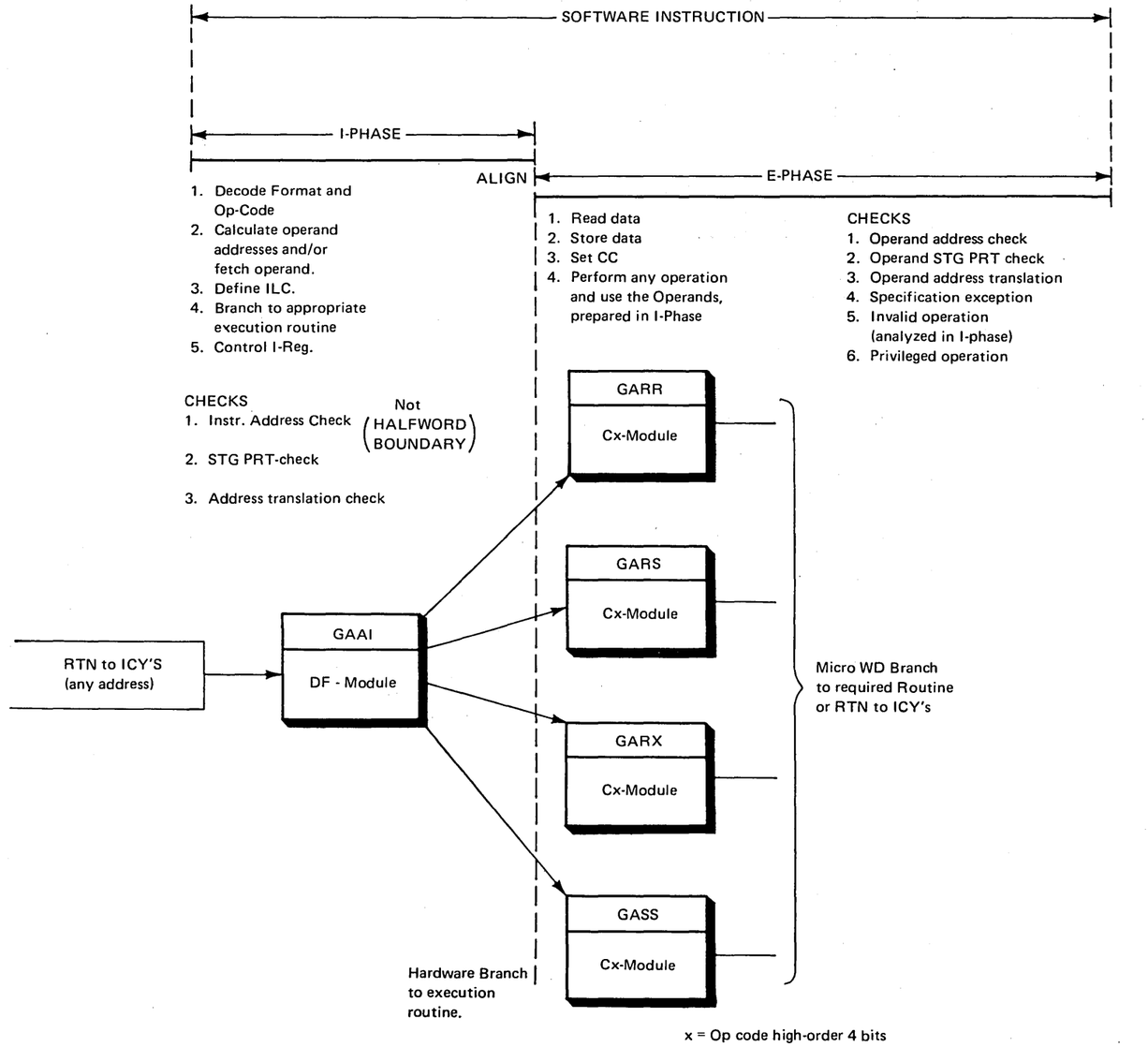
Basic I-Cycle Functions

I-Cycles Microcode Module Assignment

The I-cycle microcode routine, (GAAI) resides in the DF module of control storage. There is a direct relationship between the control-storage address, hardware functions generated, and microcode. Specifically, the hardware generates control signals (and a next control-storage address) from the contents of the M-register, to be active (and coincident) with the control word executed from that address.

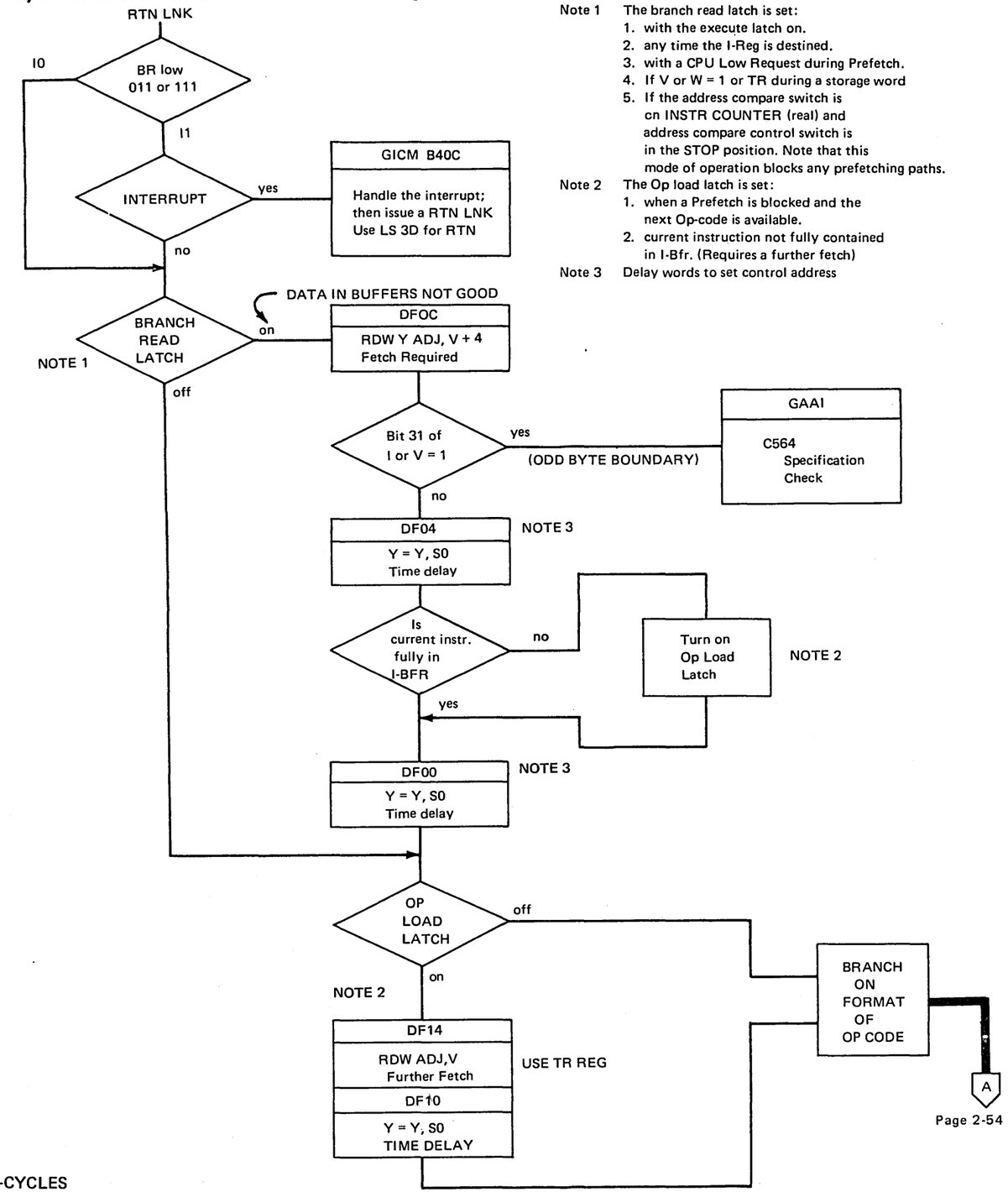
The DF module is active if bit 1 of the Mode Reg (external address 08) is on.

Microprogram Module Assignment



x = Op code high-order 4 bits

I-Cycles Microcode and Control Hardware-Loading of I-Buffers



Page 2-54

Note 1 The branch read latch is set:
1. with the execute latch on.
2. any time the I-Reg is destined.
3. with a CPU Low Request during Prefetch.
4. If V or W = 1 or TR during a storage word
5. If the address compare switch is on INSTR COUNTER (real) and address compare control switch is in the STOP position. Note that this mode of operation blocks any prefetching paths.

Note 2 The Op load latch is set:
1. when a Prefetch is blocked and the next Op-code is available.
2. current instruction not fully contained in I-Bfr. (Requires a further fetch)

Note 3 Delay words to set control address

I-Cycle Entry

I-cycles may be entered (and the I-cycle controls enabled) by either of 2 RTN words:

- Conditional: testing for interrupt. C3 bits 5-7 will be (111). If there is an interrupt pending, the RTN word will be executed normally, and will not go to the I-cycle routine. If there is no interrupt, the return is to I-cycles.
- Unconditional: goes directly to I-cycles when C3 bits 5-7 are (011).

The M-register controls the decode of the return and accesses control storage using the address inputs to the M-register from I-cycles. The I-cycles inputs to the P- and L-registers are also gated. Data is maintained on these inputs by the I-cycle hardware when not in I-cycles (except when performing a storage address adjustment access).

Initial I-Cycle Address

All I-cycle addresses are in the DF module. The initial address for a given instruction is determined by:

- I latches, which represent how much of the instruction is in the I-buffers (I-Bfr).
- The instruction itself and,
- The requirement for prefetching the next instruction.

Current Instruction Not Fully Contained in I-Bfrs

First consider the two cases where the instruction is not completely contained in I-Bfr. Before proceeding further, the instruction must be completely within the I-Bfr.

1. The first case (and highest address priority) occurs when the branch read latch is on. This occurs not only from the most obvious case of a macroprogram branch, (detected by the I-register being loaded from EBI), but also from:

- program modification (detected by storing within the present, or next, storage doubleword address as compared to the I-register).
- a prefetch condition that was not filled during the last I-cycle phase.
- blocking a trap during a prefetch in the last I-cycle phase.
- being in real instruction address compare mode.
- performing an execute macroinstruction.

These examples are summarized as "whenever the instruction must be read from storage." When the branch read latch is on, all other initial addresses are blocked and an address of DF0C is sent to the M-register input.

2. The second case (and next highest address priority) occurs when the Op load latch is on. The condition for setting this latch occurs when part of the instruction is in I-Bfr, but the remainder is in storage. The latch is set during the previous I-cycle phase if it is determined that a prefetch is required, but blocked, and only part of the instruction is in I-Bfr. When this latch is on, all other initial addresses are blocked and an address of DF14 is sent to the M-register input.

Current Instruction Fully Contained In I-Bfrs

With the instruction fully contained in I-Bfr, the next condition considered for an initial address is prefetching. The rule for prefetching is that a prefetch will be performed if the:

- Present instruction ends at a doubleword boundary (The Next op code is not available) or,
- Next instruction crosses a doubleword boundary. This is determined by hardware as a function of:
  - A. The halfword address of the present instruction within the doubleword.
  - B. The length of the present instruction and,
  - C. The length of the next instruction.
- A prefetch is blocked if the present instruction is decoded to be a branch type, or a special addressing case of an SS instruction, at an address of 6 or C.

The initial address is gated to the M-register according to the following table:

Instruction	Without Prefetch	With Prefetch
RR (but not flt. pt. long)	DF20	DF34
RR (flt. pt. long only)	DF24	DF3C
RX (double index only)	DF4C	DF5C
RS, SI, RX (not double index)	DF48	DF58
SS	DF6C	DF7C

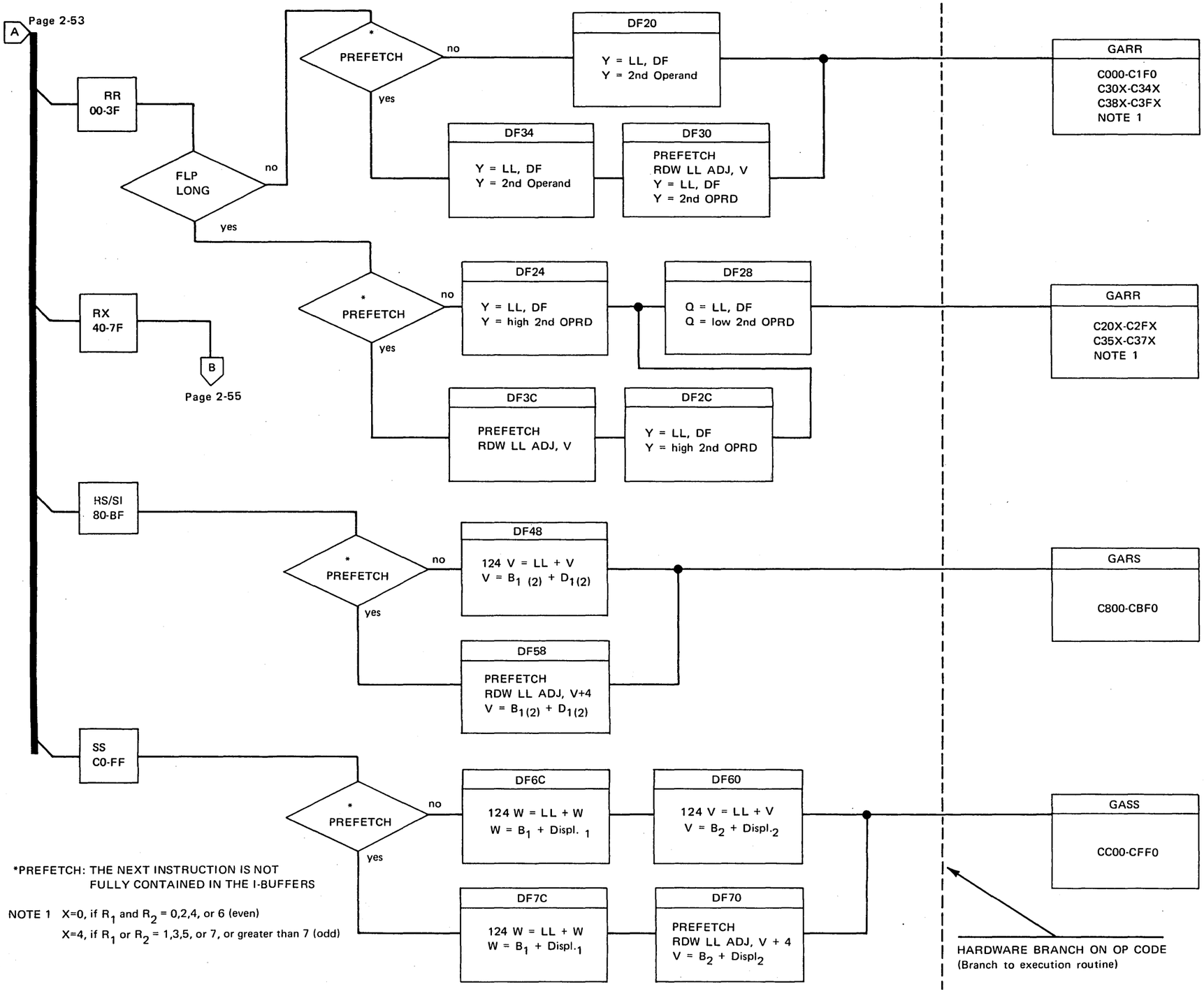
I-BFR SET CONDITIONS FOR MOVE I BFRs

1. RR Op and even halfword	=do nothing	
2. SS Op and odd halfword	=S/R 0,1	S/R 1
3. Neither of the above conditions	=S/R 0	S/R 1
4. Prefetch and the next instruction is not fully contained in the I BFRs	=S/R 0, 1	S/R 1

I-BUFFER SET CONDITIONS FOR STORAGE WORDS (begin in Stg 2 cycle)

1. Prefetch and the next Op is not available	=S/R 0, 1, 2	S/R 1.2
2. Prefetch and the next Op is available	=S/R 1,2	S/R 2
3. Branch Load Latch	=S/R 0, 1, 2	S/R 1, 2
4. Op Load Latch	=S/R 1, 2	S/R 2

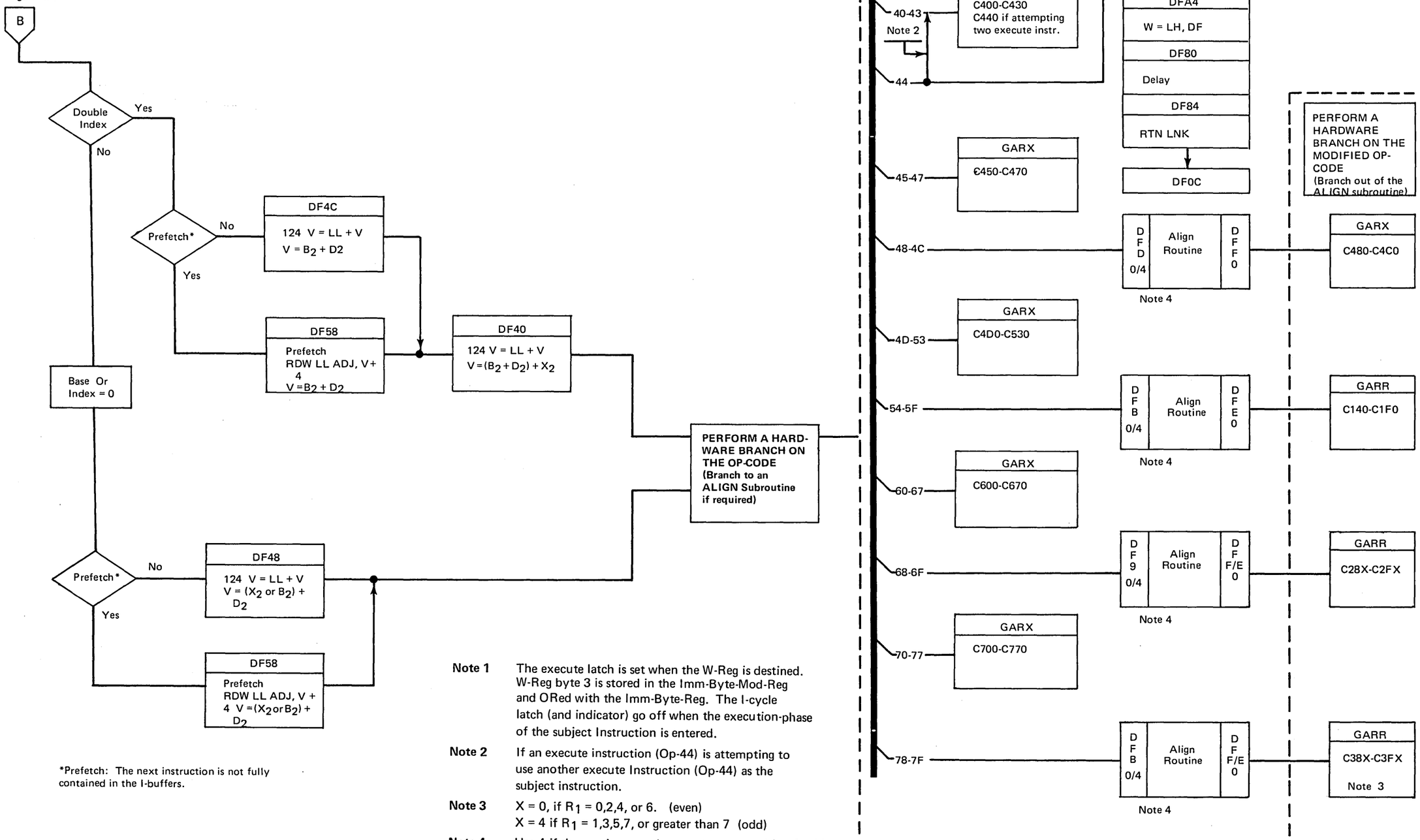
I-Cycles Microcode and Control Hardware-Calculate Operand Address and Perform Prefetches



*PREFETCH: THE NEXT INSTRUCTION IS NOT FULLY CONTAINED IN THE I-BUFFERS

NOTE 1 X=0, if R_1 and R_2 = 0,2,4, or 6 (even)
X=4, if R_1 or R_2 = 1,3,5, or 7, or greater than 7 (odd)

Page 2-54



*Prefetch: The next instruction is not fully contained in the I-buffers.

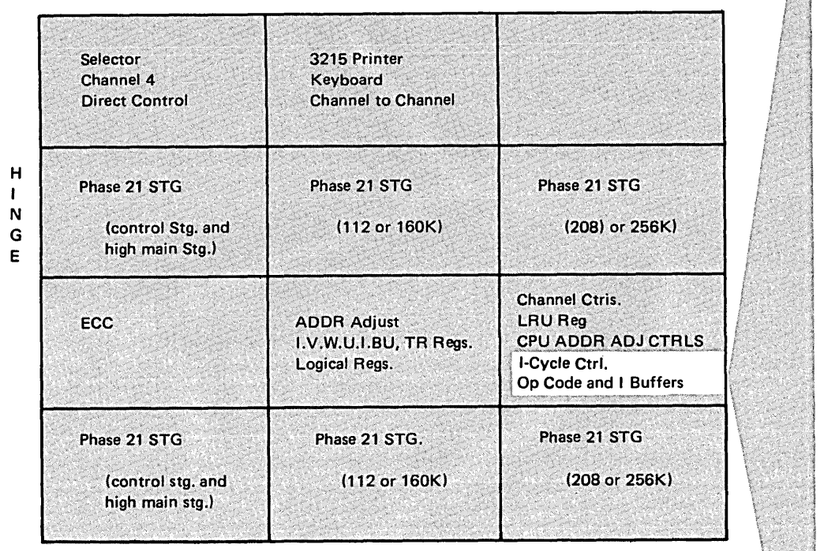
Note 1 The execute latch is set when the W-Reg is destined. W-Reg byte 3 is stored in the Imm-Byte-Mod-Reg and ORed with the Imm-Byte-Reg. The I-cycle latch (and indicator) go off when the execution-phase of the subject Instruction is entered.

Note 2 If an execute instruction (Op-44) is attempting to use another execute Instruction (Op-44) as the subject instruction.

Note 3 X = 0, if R_1 = 0,2,4, or 6. (even)
X = 4 if R_1 = 1,3,5,7, or greater than 7 (odd)

Note 4 Use 4 if the previous word was a storage word (prefetch)
Use 0 if the previous word was not a storage word.

I-Cycle Hardware Locations



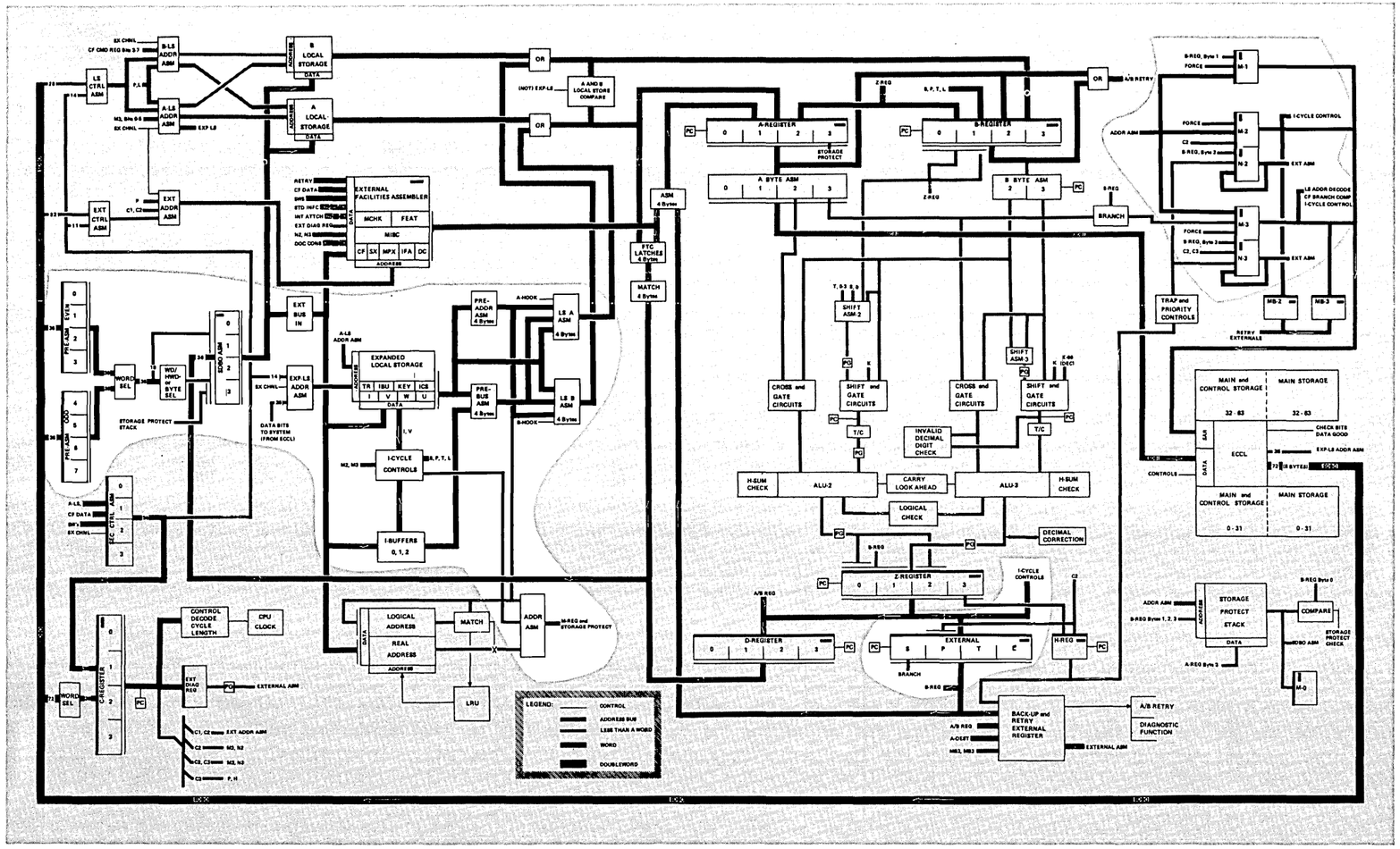
B GATE (CARD SIDE)

CARD LOCATION & TYPE		ALD PAGE	
B-C3D2	Type 8551	RU011	Op-U2 Reg-Op Decode
		RU012	Op-U2 Reg-Op Decode
		RU013	Op-U2 Reg-Op Decode
		RU014	Op-U2 Reg-Op Decode
		RU015	Op-U2 Reg-Op Decode
		RU016	Op-U2 Reg-Op Decode
B-C3E2	Type 8552	RU021	Imm Byte-U3 Regs
		RU022	Imm Byte-U3 Regs
		RU023	Imm Byte-U3 Regs
		RU024	Imm Byte-U3 Regs
		RU025	Imm Byte-U3 Regs
		RU026	Imm Byte-U3 Regs
B-C3F2	Type 8553	RU031	I-Cycles Generation
		RU032	I-Cycles Generation
		RU033	I-Cycles Generation
		RU034	I-Cycles Generation
		RU035	I-Cycles Generation
B-C3G2	Type 8554	RU041	I-Buff Ctrls and Gates
		RU042	I-Buff Ctrls and Gates
		RU043	I-Buff Ctrls and Gates
		RU044	I-Buff Ctrls and Gates
		RU045	I-Buff Ctrls and Gates
B-C3H2	Type 8558	RU051	PAA Latches
		RU052	PAA Latches
		RU053	I-Cycles Controls
		RU054	I-Cycles Error Latches
B-C3B2	Type 7771	RU111	I-Buffer
		RU112	I-Buffer
		RU113	I-Buffer
		RU114	I-Buffer
		RU115	I-Buffer
		RU116	I-Buffer
		RU117	I-Buffer
		RU118	I-Buffer
B-C3C2	Type 7771	RU121	I-Buffer
		RU122	I-Buffer
		RU123	I-Buffer
		RU124	I-Buffer
		RU125	I-Buffer
		RU126	I-Buffer
		RU127	I-Buffer
		RU128	I-Buffer

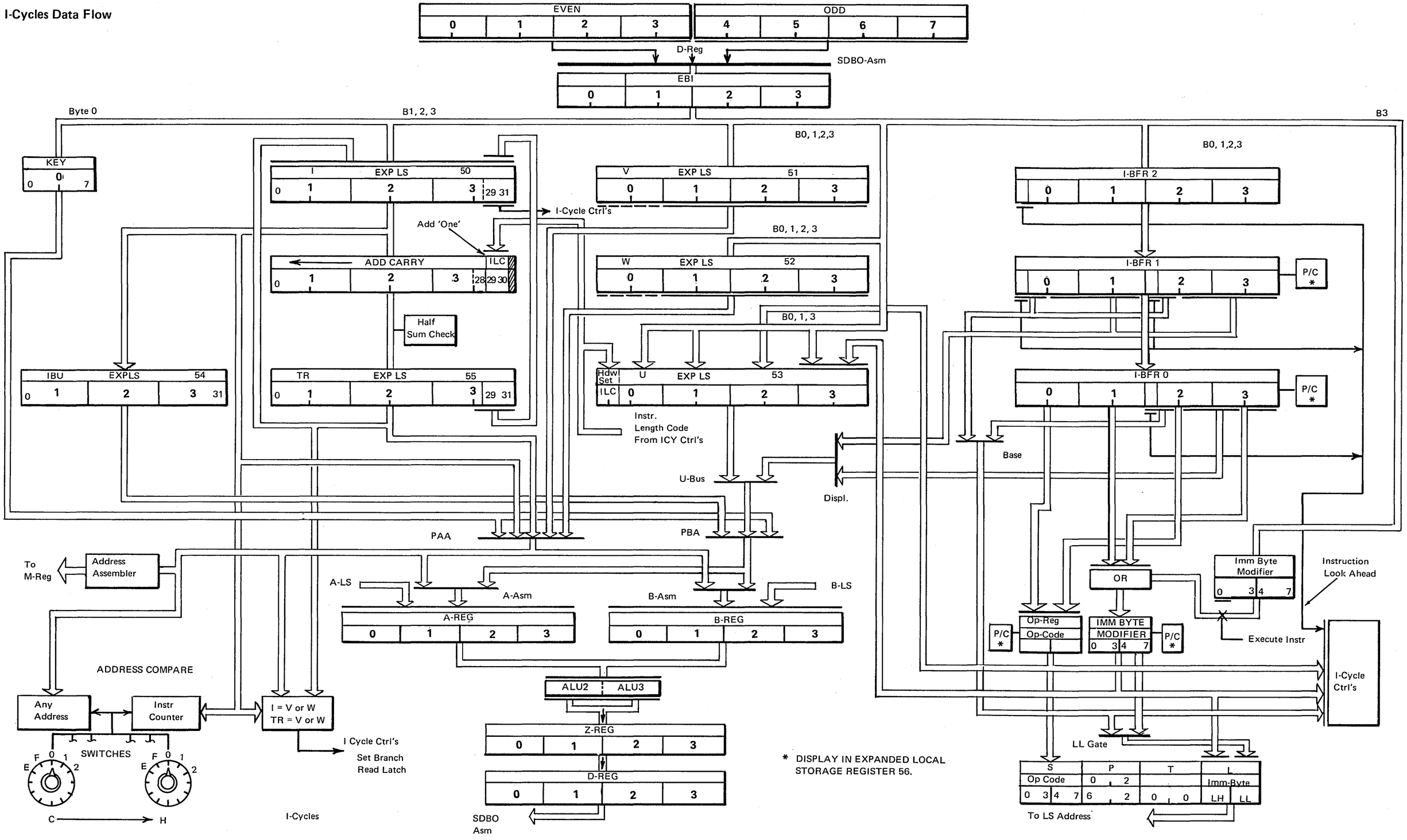
I-Cycle Hardware Description

The hardware I-cycle concept increases CPU performance by:

- Buffering instructions and prefetching (using a hardware generated address for the next doubleword storage location) instructions while calculating an operand address.
- Performing hardware controls concurrent with micro-program execution.
- Instruction decoding via a hardware forced branch on the eight bit Op code.



I-Cycles Data Flow



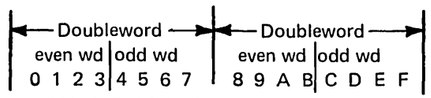
* DISPLAY IN EXPANDED LOCAL STORAGE REGISTER 56.

A I-BFR 2, I-BFR 1, and I-BFR 0

Instruction Buffer (I-BFR): three one-word registers are used to hold the present instruction and next doubleword (where possible). Loading of the registers is from EBI.

Instructions are assembled on a halfword basis to obtain the OP-Code and immediate byte. The base and displacement fields are gated from the I-Bfr through separate assemblers as required

When the TR-Reg is used, the even/odd time-slot of data is forced. When the I-Register is used, (DFOC) M3 bit 5 controls the gating.
Bit 5 off = even/odd
Bit on = odd/odd



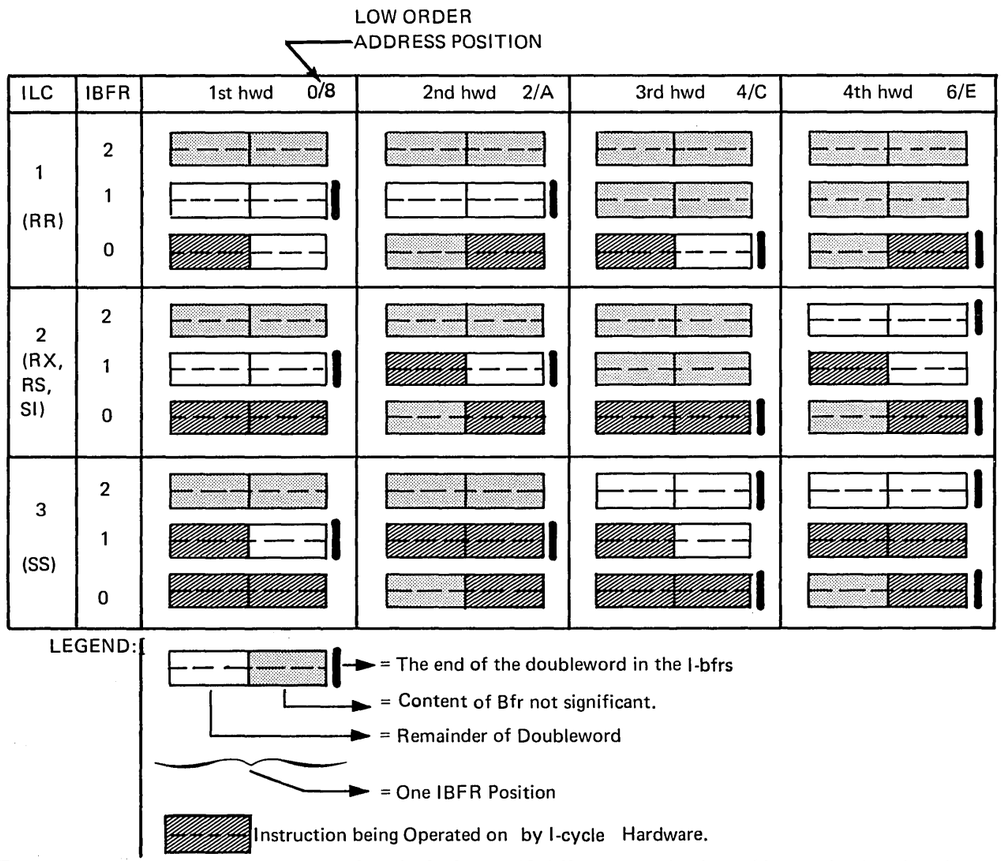
If M3 bit 5=1 it indicates that the odd word, information will be loaded into all 3 I-buffers.

I BFR SET CONDITIONS for MOVE I BFRs

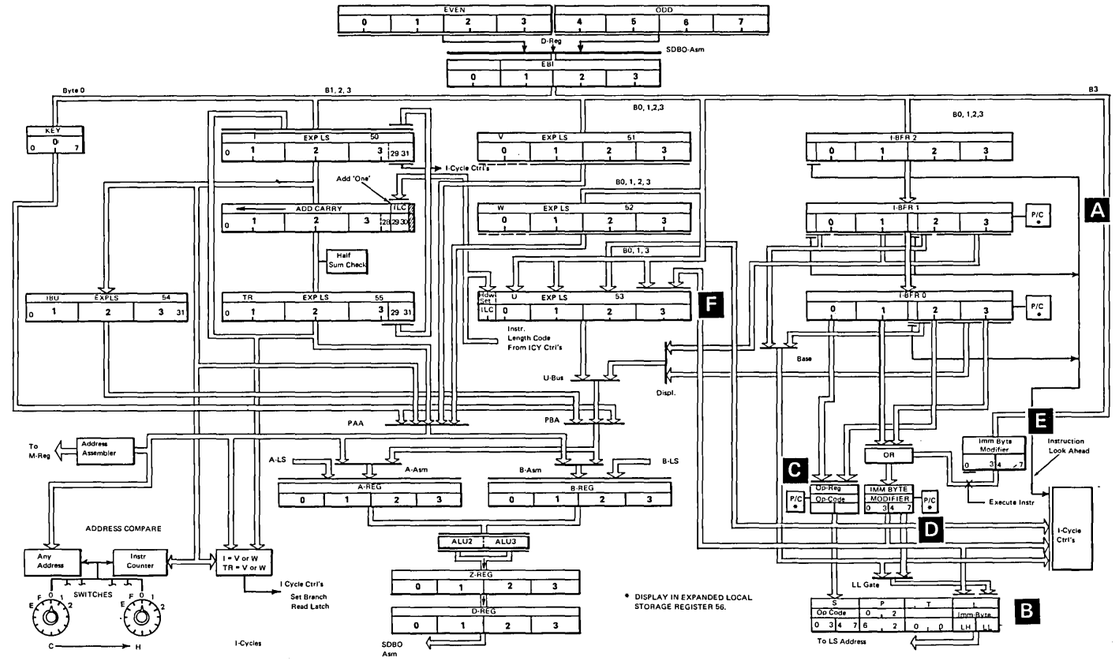
1. RR Op and even halfword	=do nothing	
2. SS Op and odd halfword	=S/R 0, 1	S/R 1
3. Neither of the above conditions	=S/R 0	S/R 1
4. Prefetch and the next instruction is not fully contained in the I-Bfr's	=S/R 0, 1	S/R 1

I BUFFER SET CONDITIONS for STORAGE WORDS (begin in Stg 2 cycle)

1. Prefetch and the next Op is not available	=S/R 0, 1, 2	S/R 1, 2
2. Prefetch and the next Op is available	=S/R 1, 2	S/R 2
3. Branch load latch	=S/R 0, 1, 2	S/R 1, 2
4. Op load latch	=S/R 1, 2	S/R 2



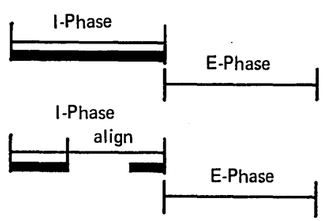
The above chart represents the contents of the I-Bfrs for the 12 different combinations of instruction length and instruction location within a doubleword.



B SPTL Registers

S-Reg Set from the Op reg for further use by the execution routine.

Set time for SPTL.



P-Reg Set to a value of 02 (or 62 for floating point) to allow addressing of local-storage areas containing general registers. CPU working area, and floating-point registers.

T-Reg Reset to zero for use in an align routine (if required).

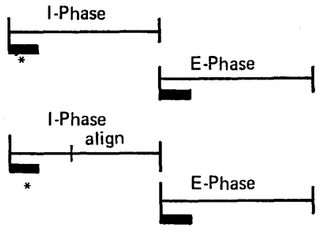
L-Reg Set from the immediate byte register or base field assembler to allow indirect addressing of the general registers and floating-point registers.

C

Op Register

The Op register is used to hold the present instruction Op code during I-cycles. The output is gated to the U- (2) and S-registers, and is also used for I-cycle decoding and branching to the instruction execution routine.

The decode of the Op-register is used to build the ILC, which is gated to the U-register byte 0, bits 0 and 1.

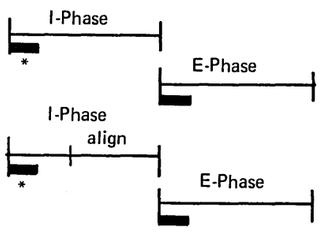


D

Immediate Byte Register

The Imm byte reg is used to hold the second byte of the present instruction during I-cycles. The output will:

- Load the U3-register.
- Be assembled with the base field to be gated to the L-register.
- Be gated to the I-cycle controls.



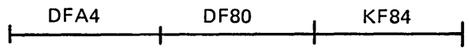
*Only with branch-read (DFOC)

E

Immediate Byte Modifier Register

- Used only by the Execute software instruction to modify the second byte of the subject instruction (if the R_1 field of the Execute instruction is not zero).
- The register is set when the W-reg is destined, with data from byte 3 of the GR specified by the R_1 field. (Refer to the explanation of the Execute software instruction.)

W=LH, DF

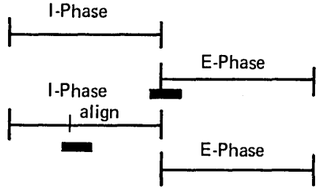


F

U-Register (Exp LS 53)

Part of this register is set only by hardware. The two-bit Instruction Length Code (ILC) is set to a value determined by the Op-reg decode. The condition code (two bits) is used by the I-cycle controls to determine whether a branch-on-condition code instruction will branch. Byte 2 is set to the Op-code, by hardware only. Byte 3 is initialized to the immediate byte by hardware. Bytes 0 (except bits 0, and 1), 1, and 3 may be loaded from EBI by microcode.

Byte 0	Bits 0-1	Instr. length code (hardware set only)
	Bits 2-3	Condition code
	Bits 4-7	Program mask
Byte 1	Bits 0-3	Special CPU use
	Bits 4-7	0MWP bits
Byte 2	Bits 0-7	Op code (Hardware set only)
Byte 3	Bits 0-7	Immediate byte



G

Special address-matching function--on a doubleword basis (bits 8-28).

- I-reg is compared to PAA. (V or W) TR-reg is compared to PAA (V or W). (These matches are required to determine whether program modification is taking place in that part of the instruction stream that may have been loaded in the I-BFRs.)
- If the comparison is equal during a store operation, the BR read latch is set. (This forces a new loading of the I-buffer Op-reg and IMM byte reg).

H

V-Register (EXP LS 51)*

Bytes 1, 2, and 3 usually contain the second operand address.

*If used in a storage word as a storage address, the Key register is gated as byte zero.

I

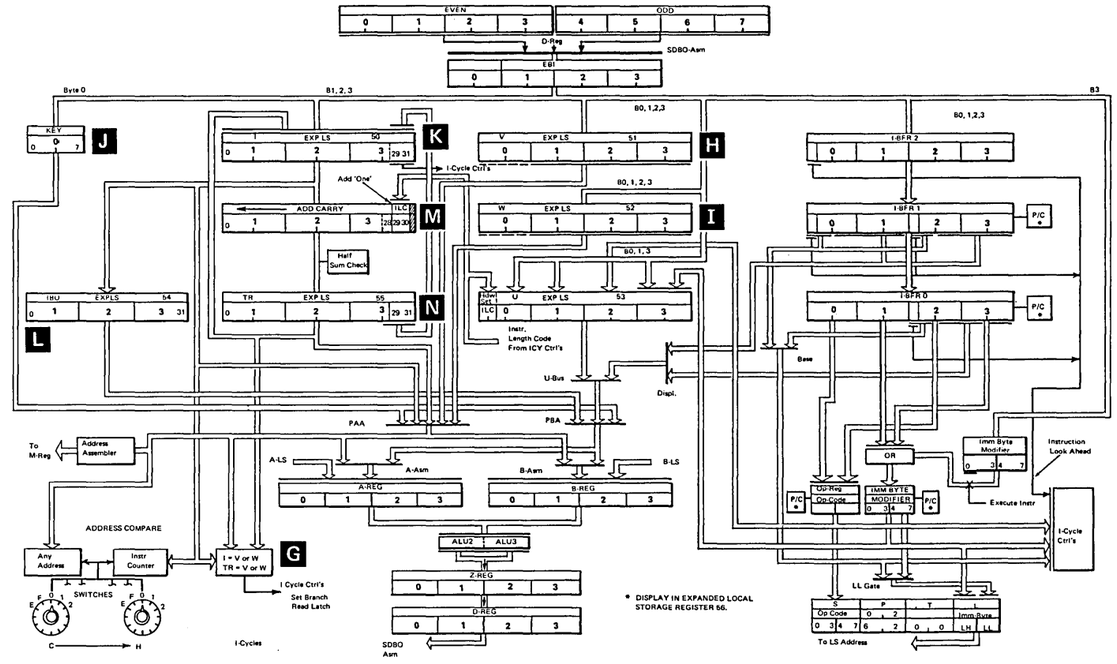
W-Register (EXP LS 52)*

Bytes 1, 2, and 3 usually contain the first operand address.

J

Key Register

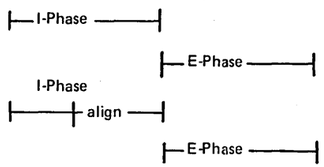
The KEY REG is byte 0 of the I-reg when it is destined. It contains the storage-protect key, bits 0-3, bits 4-7=0).
If V or W is used in a storage word as a source, the key reg is gated as byte 0. If I, IBU, or TR is used, the key reg is always gated as byte 0. The key-reg is set when the I-reg is destined.



K I-Register (EXPLS 50)

Instruction counter register. Byte 0: If the I-Reg is destined, byte 0 is gated to the key register. Bytes 1, 2, and 3 contain the instruction address.

The I-Reg is updated as follows:



L IBU-Register (EXPLS 54)*

Upon entering I-Cycles, I-Reg bytes 1, 2, and 3 are set into IBU. If a retry condition is encountered during I-cycles, the instruction cycles may be repeated (Return to DF0C). In this event, IBU is moved to the I-Reg by the retry microprogram.

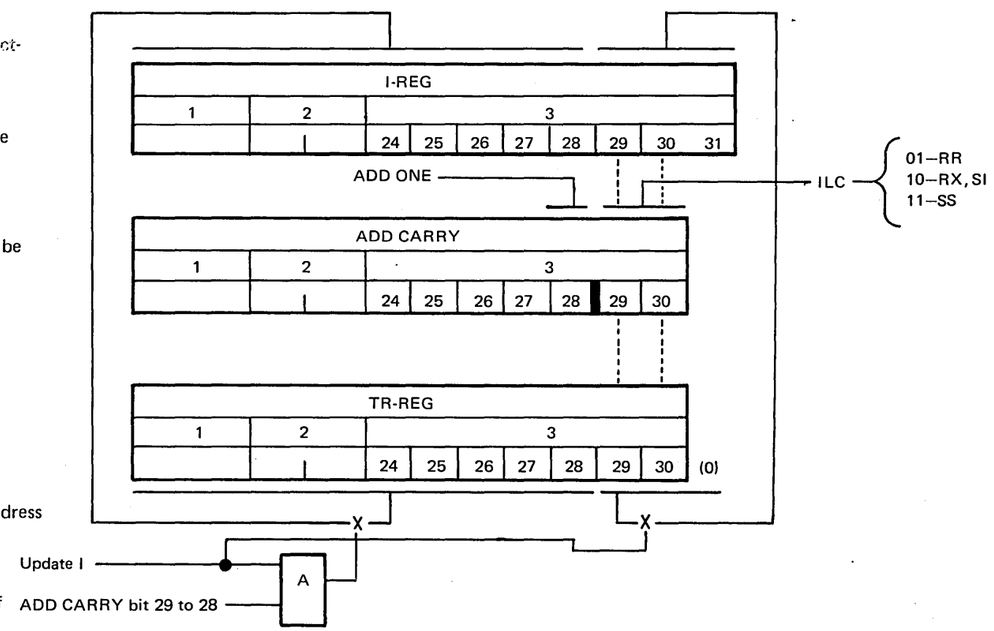
M Add Carry (Adder)

The two low-order adder positions (24-bit adder) add the instruction length to the I-register bits 29-30.

- Adder (except the low two bits) adds a "1" to I-Reg bit position 28. This sum represents the next doubleword storage location.
- A carry-out of position 28 is added to bits 27-8.
- A carry-out of position 29 indicates that TR bits 8-18 are to be loaded into I for a hardware update.
- An adder check turns the I-cycle hardware indicator on.

N TR–Register (EXPLS 55)

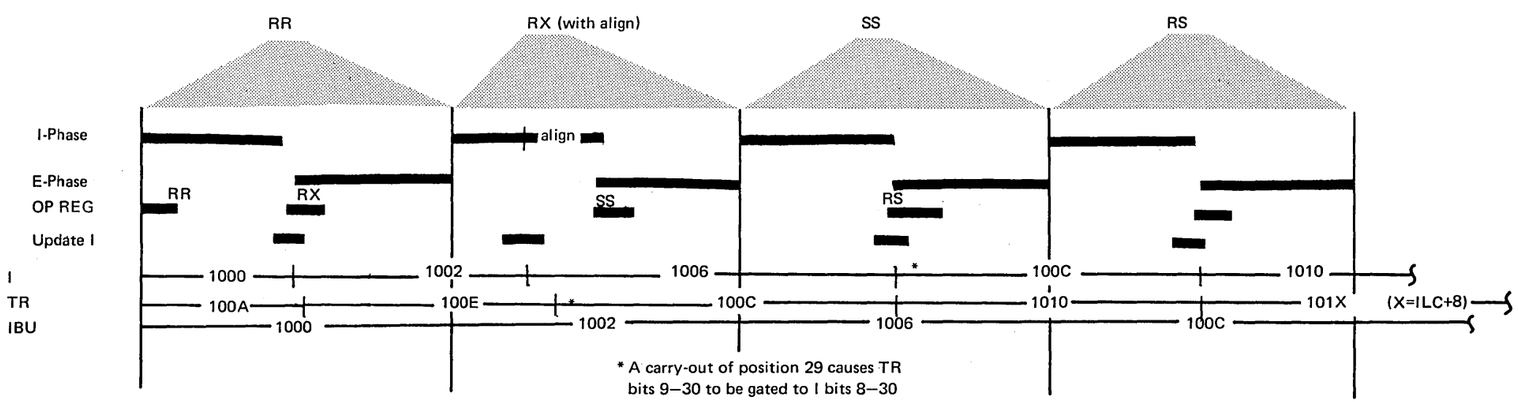
- The TR-Reg consists of bits 8-30, with bit 31 always forced to a zero. (There is no bit 31 latch).
- Contains an address within the next doubleword after the address in the I-register, to be used when prefetch or further fetch (DF14) is required.
- Used to buffer the Addr output during hardware updating of the I-register.
- I hardware update, consists of: Loading I bits 29-31 from TR 29-31. Bit 31 of the I-register is not gated through the addr. I bits 8-28 from TR 8-28 if the Adder bit 29 had a carry-out. (TR = I + ILC + 8)



Carrys from add carry bit 29 to bit 28 are not allowed, but remembered. If at I update time, a carry was remembered, it causes all of TR to be gated back to I. Otherwise, only bits 29 and 30, are used to update I.

I=1000

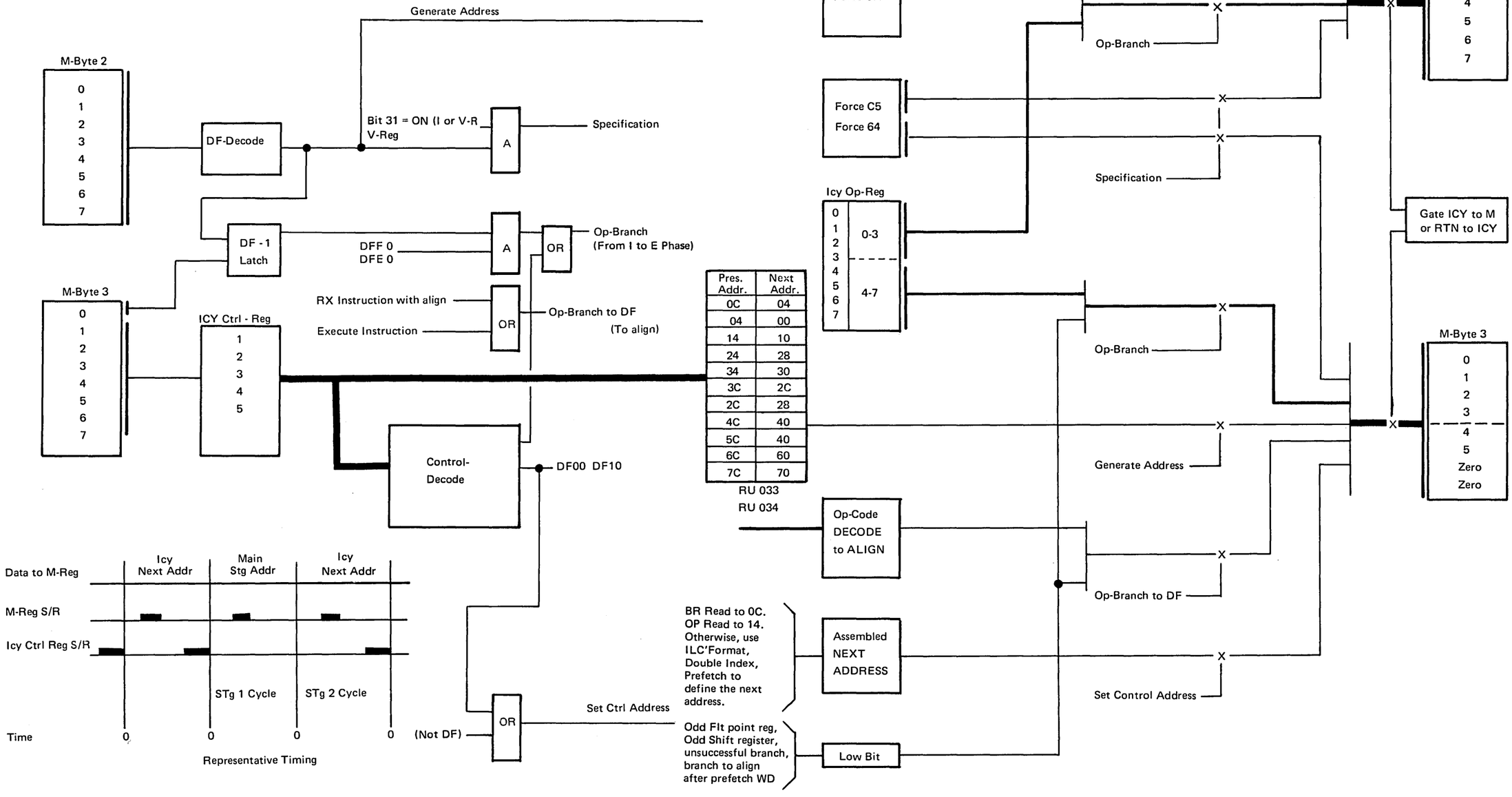
1000 RR
1002 RX
1006 SS
100C RS



* A carry-out of position 29 causes TR bits 9–30 to be gated to I bits 8–30

Instruction Cycles-Address Generation

High-Level Data Flow



I-Cycles Control Line Generation

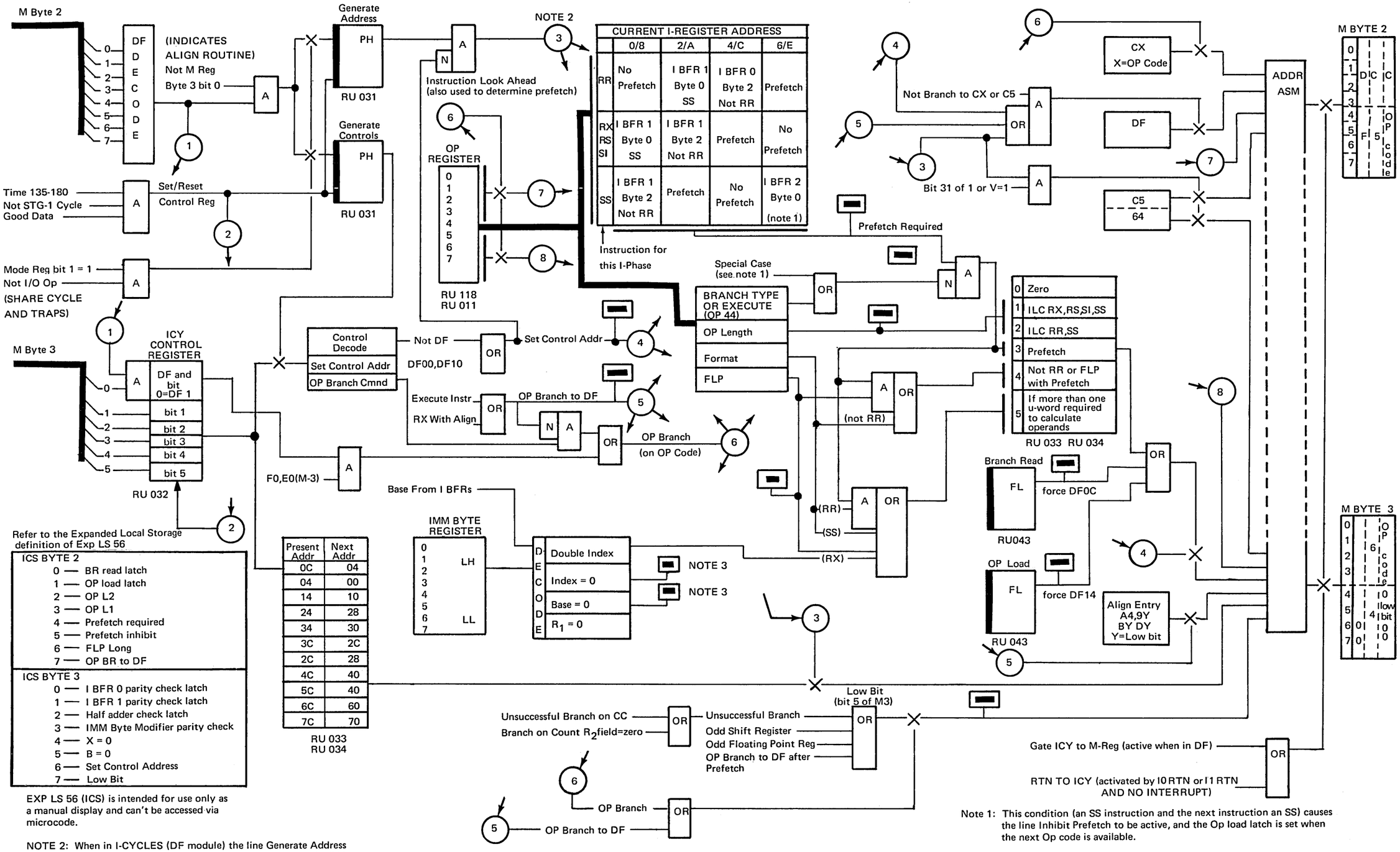
ADDRESS	Command Branch Load	Force I to B	Command Op Load	Force TR	Command Prefetch	OP Branch Command	L plus one	Load Op-Reg Imm Byte	Command Move I-Bfr's	Gate D_1 and B_2	Gate D_2	Set Control Address	Set SPL Reset T	Gate Imm Byte to L
DF00								2 5				2	1 6	
DF04								2 5					1 6	
DF0C	2	2											1 6	
DF10												2	1 6	
DF14			2	2									1 6	
DF20						2			2				1 6	
DF24							2		2				1 6	
DF28						2							1 6	
DF2C							2						1 6	
DF30				1 3	2	2			4				1 6	
DF34													1 6	
DF3C				1 3	2				4				1 6	
DF40						2							1 6	1
DF48						2			2	2			1 6	1
DF4C									2	2			1 6	
DF58				1 3	2	2			4	2			1 6	1
DF5C				1 3	2					2			1 6	
DF60						2			2		2		1 6	1
DF6C										2			1 6	
DF70				1 3	2	2			4		2		1 6	1
DF7C										2			1 6	
Not DF												1		
RTN to I-Cycles													1 7	

1. These controls are not the result of the Control Register decode
2. The Control line is activated by the corresponding address.
3. This line is activated by Command prefetch.
4. The set/reset of the I-Bfr's is also controlled by Command Prefetch.
5. This control line is activated, but not used.
6. From DF00 Through DF7C SPTL is controlled by I-Cycles. DFE0 or DFF0 will activate this control line again to restore SPTL after an align.
7. Set P, set LL.

CONTROL REGISTER DECODE

ICY CTRL REG BITS 1	2	3	4	5	CONTROL LINE	FUNCTION
0	0	–	1	–	Command Branch Load	Load I-bfr 0 (I-Bfr 1 in next cycle)
0	0	–	1	–	Force I	Gate I-Reg to B (and Address Adjust)
0	0	1	0	1	Command OP Load	Load I-Bfr 1 (I-Bfr 2 in the next cycle)
0	0	1	0	1	Force TR	Gate TR-Reg to B (and Address Adjust)
–	1	1	–	0	Command Prefetch	Activate prefetch
1	0	1	1	–		Force TR to ADR ADJ Asm
0	1	1	1	–		(and B if DF 30 or DF 3C)
–	1	–	–	0	OP Branch Command	Use Op-Reg to define the next Cxxx address; or go
1	–	–	–	0		to the Align Routine
0	1	0	–	1	L plus one	Force bits 7 and P of data being gated to L-Reg to be inverted for FLP long
0	0	0	–	–	Load OP, IMM Byte	Initial load of Op and IMM Byte
–	1	0	0	–	Command Move I-Bfr	Used with I-Reg to activate the set
1	0	0	1	–		or reset of the I–Bfr's
1	–	–	1	–	Gate D_1 and B_2	Used to gate the correct base or displacement
1	1	–	0	–	Gate D_2	field from the I-Bfr's 0 and 1 to L - low or B-Reg
0	0	–	–	0	Set Control Address	Set next address for ICY sequence

ICY Address Generation and Control Decode

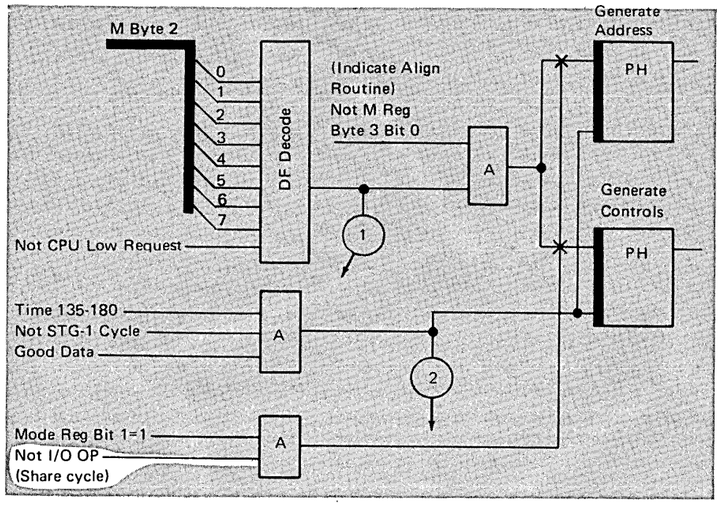


Unique Conditions During ICY

Share Cycle

If a share cycle is attempted during ICY, the ICY hardware is deconditioned. This occurs when the line 'Not I/O Op' becomes inactive. (See ICY address generationand control decode page.)

With 'Not I/O Op' inactive, the generate address, and generate controls latches are deconditioned.



Trap (Not Machine Check)

In the event of a trap during I-Cycles, the M-Reg will be set to the appropriate trap address, and the trap will be taken. When the trap address enters the M-Reg, the DF Decode will turn off, suspending I-cycle operation.

Trap (Machine Check)

If a machine-check trap occurs during ICY, the failing instruction is retried eight times before entering the hard machine-check routine.

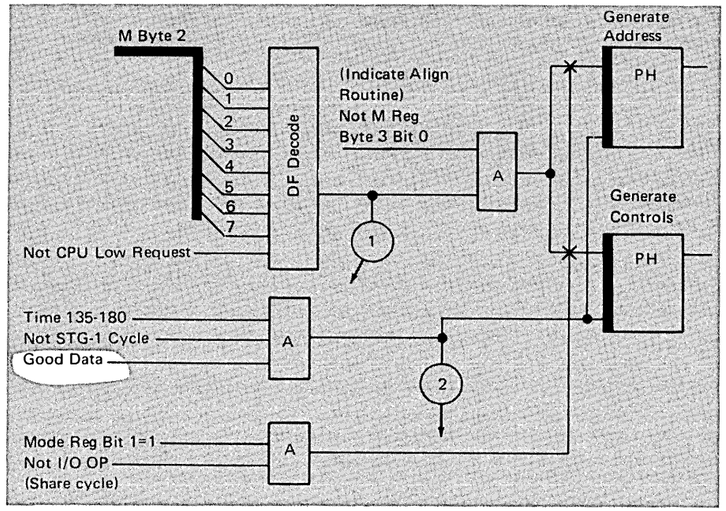
The Retry Microrutine takes the contents of IBU (I-Reg Backup; that is, the current instruction address) and place it in the I-Reg. The Retry procedure is to then return to I-cycles to begin processing the instruction again.

I-Cycle Error Conditions

Parity-check errors for I Bfr 0, I Bfr 1 Op and IMM Byte regs as well as half sum check errors are indicated in expanded local-storage register 56 (ICS) I-Cycle control display.

Storage Correction Cycle

If a storage correction cycle occurs during I-cycles, the latches Generate Address and Generate Controls are not S/R. The line 'good data' blocks the clock pulse. (See I-cycles Address Generation and Control Decode Page).



CPU Low Request (Not Prefetch)

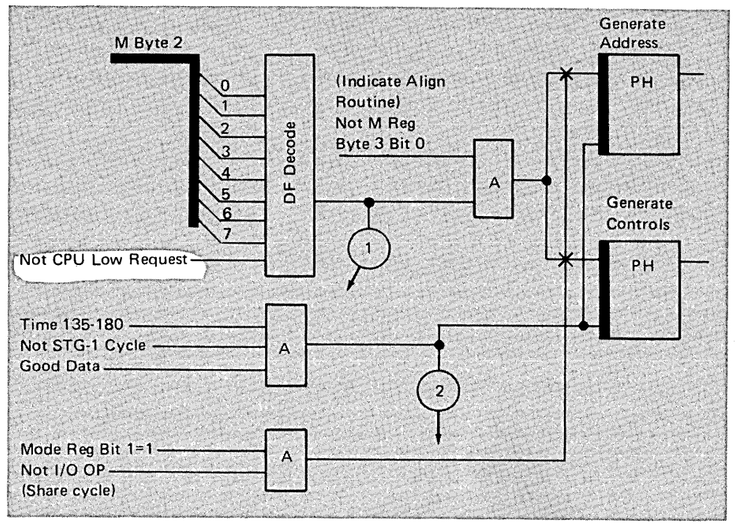
The line 'CPU Low Request' may be activated by:

A. Storage-protect check (caused by a mismatch of the storage keys).

B. Address check (TR pointing to the doubleword above the top of storage.)

C. Address translation check.

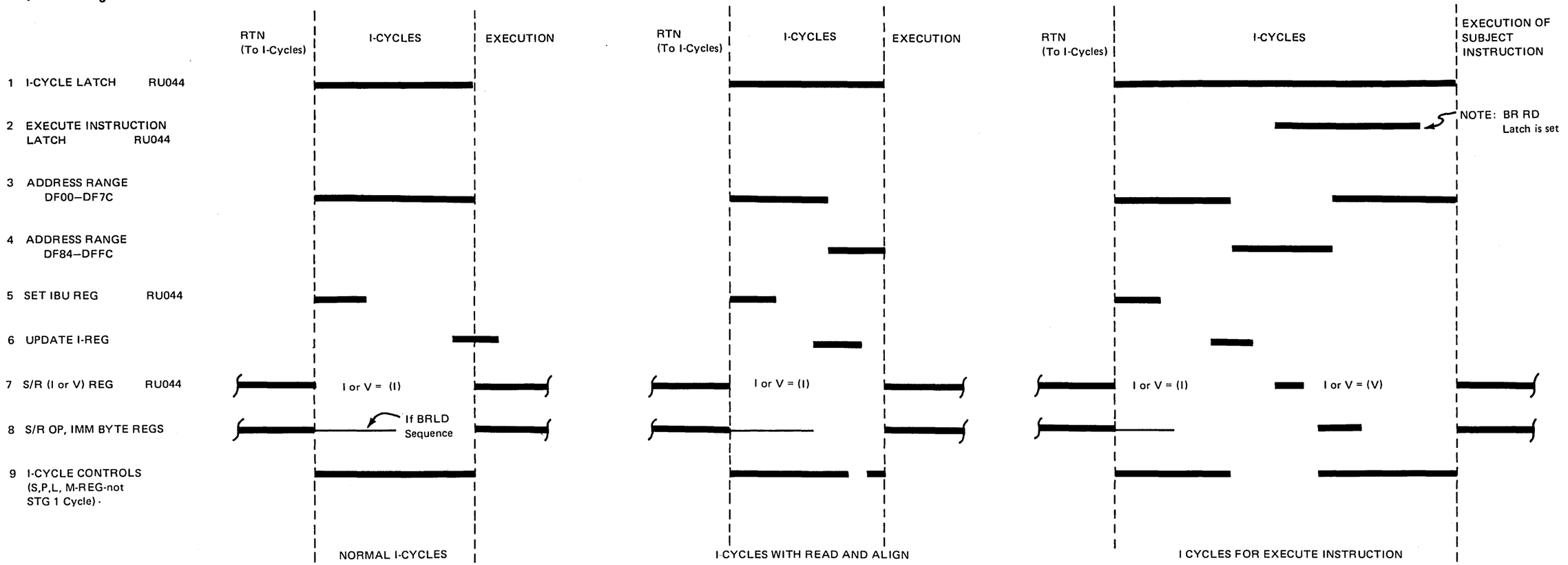
These three conditions may be considered as program errors that cause a program check and go to the GICM routine.



CPU Low Request (Prefetch)

CPU low request is blocked during Prefetch. If a storage check occurs during a Prefetch, it sets the branch read latch. Upon return to I-cycles, the I-Reg is used (instead of the TR-Reg) to refill the I-Buffer. If a second CPU low request occurs, it indicates a program check.

I-Cycle Timings



I-Cycle Operational Description

Software instruction decoding on the 3145 is accomplished by a unique interaction of microprogram and hardware.

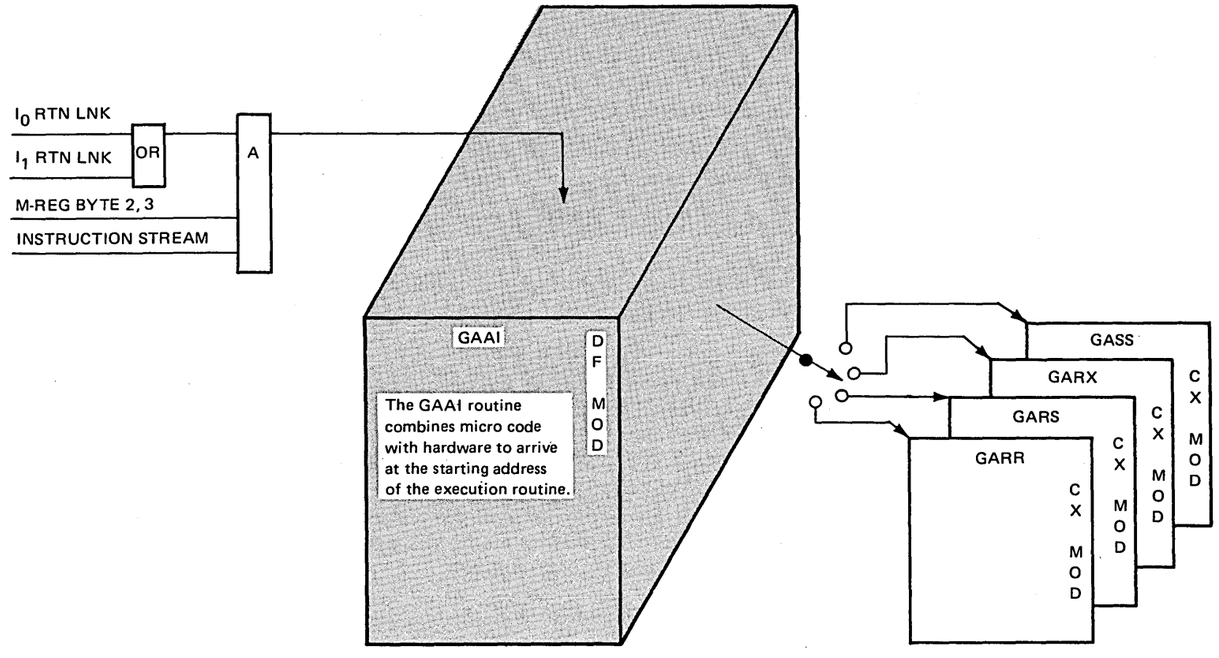
The microprogram used for instruction decoding is the GAAI routine, which resides in the DF module of control storage.

I-Phase

During I-phase, the instruction is read out of storage and placed in I-buffers. Certain determinations may then be made concerning format, Op code, and instruction length code. The purpose of the I-phase is to ensure that the correct data will be available for use during the E (execute) phase. Upon exit from the I-phase, the Op-code is used to point to the next control-word address of the microroutine for that format. (for a 1A add instruction, address C1A0 is used as the entry to the GARR routine).

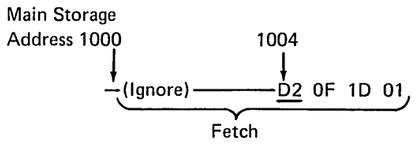
E-Phase

In the E (execute) phase, data is read, stored, and the correct condition code (CC) is set. The operations indicated in I-phase are performed, using the operands fetched during I-phase. The address sent to the M-Reg is that of the entry to the execution routine.



X = Op code high-order four bits

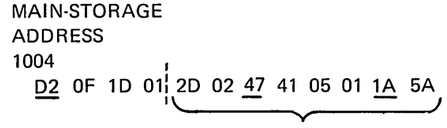
Fetch Operations



FETCH

- I-Bfrs are initialized
- This case may have occurred by a program branch to the instruction at address 1004

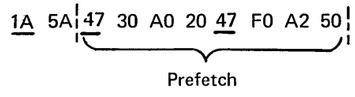
Note: See Fetch Sequence (page 2-70) and Loading I-Buffers (page 2-53) for additional information.



Further Fetch

- Required when the current instruction is not fully contained in the I-Bfrs.
- Recognized by:
  1. The decoding of the current Op-code, which determines the length of the current instruction.
  2. The I-register points to an address within the doubleword that the first byte of the present instruction is located.
- Accomplished by: (storage word at DF14)
  1. Setting the Op load latch during DF04 and forcing DF00 to branch to DF14 with the gating line Set Control Address.
  2. DF14 is a storage word that forces the TR-Reg to the B-Reg, thus fetching the next doubleword from storage, using S/R I buffers 1 and 2 with the even word late in the storage 2 cycle of DF14. During DF10, (early in the cycle) I-buffer 2 will be S/R with the odd word.

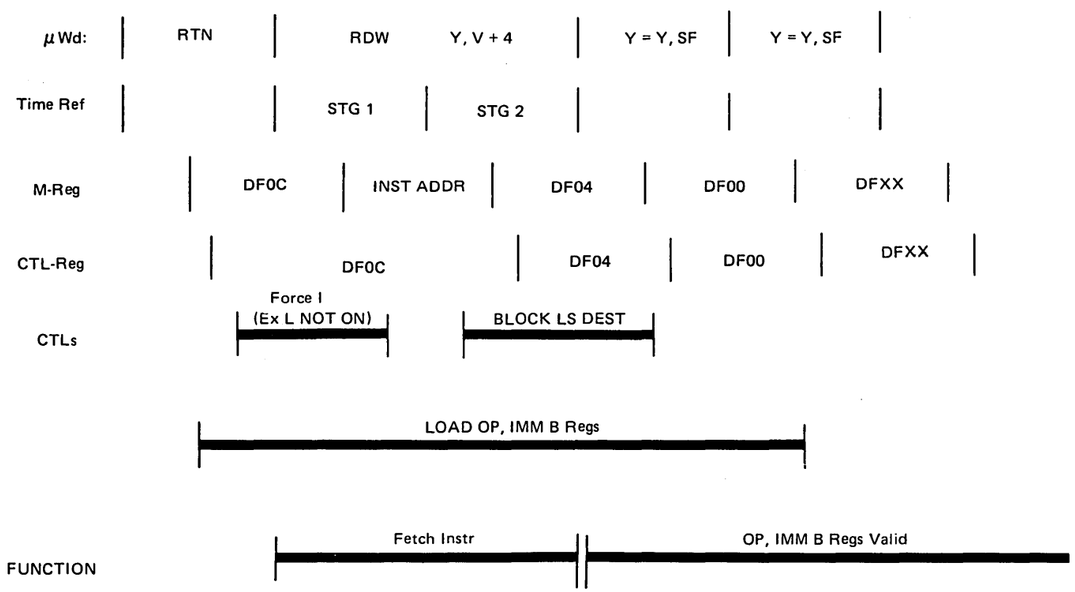
Note: see Fetch Sequence (page 2-70) and Loading I-Buffers (page 2-53) for additional information.

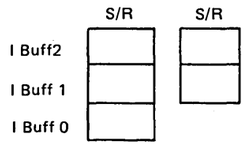


Prefetching

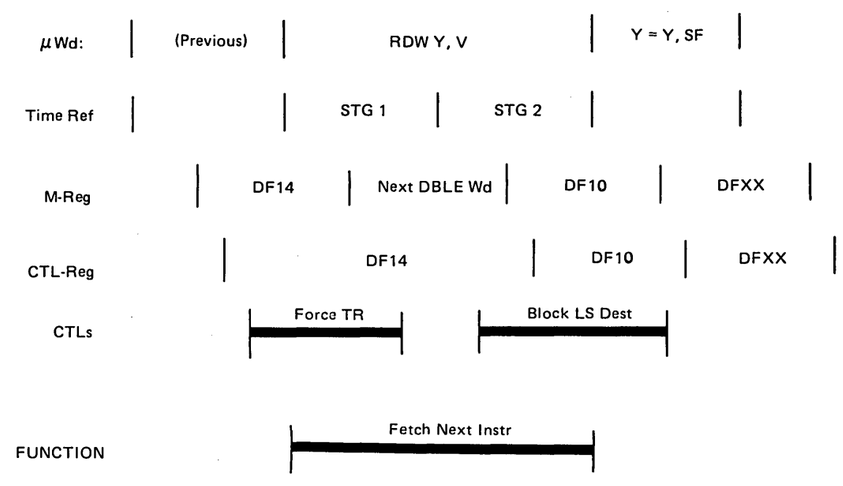
- Required when the next instruction is not fully contained in the I-Bfrs.
- Recognized by a combination of decoding:
  1. The I-Reg points to where the instructions come from within the doubleword.
  2. The Op code indicates the length of the present instruction.
  3. Instruction look-ahead knows the format of the next instruction and therefore, knows the length of the next instruction.
- Prefetch is blocked under the following conditions:
  1. All branch-type instructions.
  2. Two SS instructions in succession (TR is pointing to the doubleword just fetched).

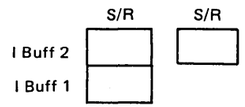
Fetch Sequence
(BRANCH LOAD SEQUENCE)





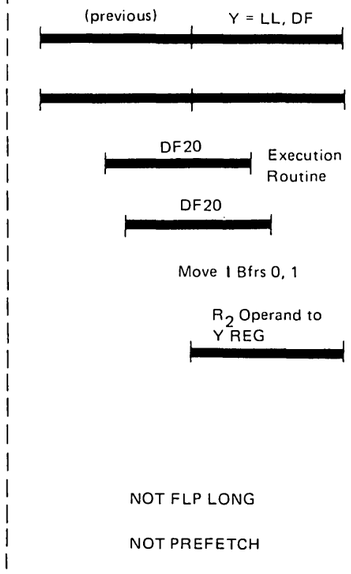
Further Fetch Sequence
(OP LOAD SEQUENCE)





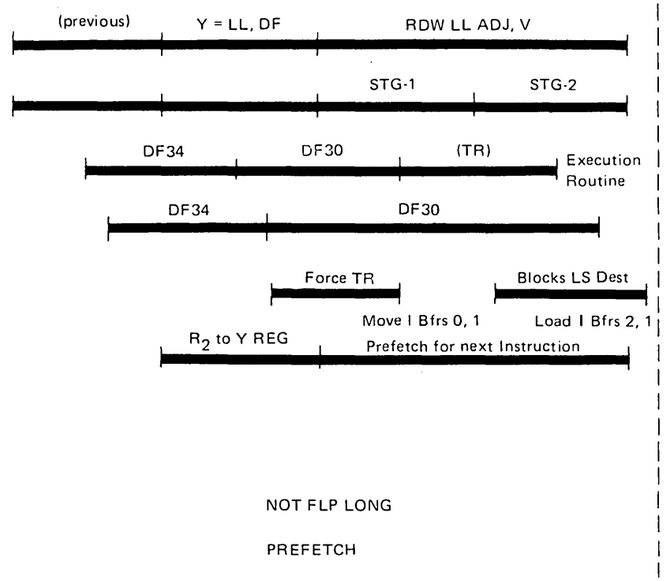
RR Sequences

1. CONTROL WORD
2. TIME REFERENCE
3. M-REG
4. ICY CTRL REG
5. CONTROLS
6. FUNCTION



NOT FLP LONG

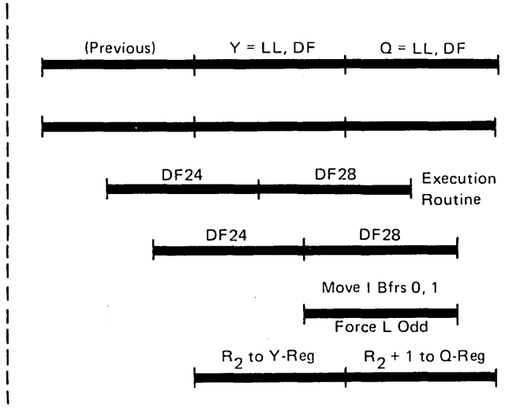
NOT PREFETCH



NOT FLP LONG

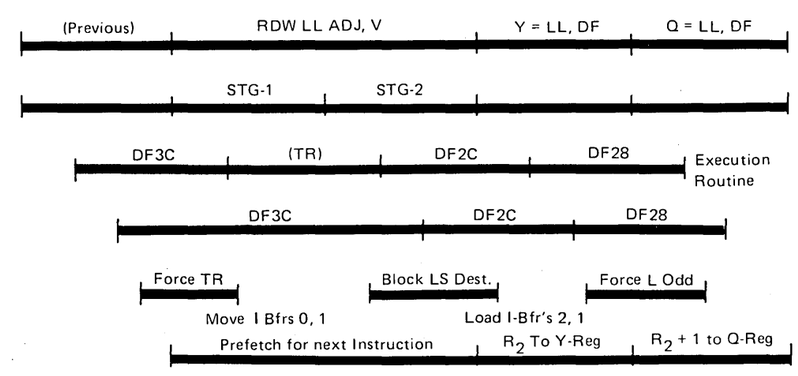
PREFETCH

1. CONTROL WORD
2. TIME REFERENCE
3. M REG
4. ICY CTRL REG
5. CONTROLS
6. FUNCTION



FLP LONG

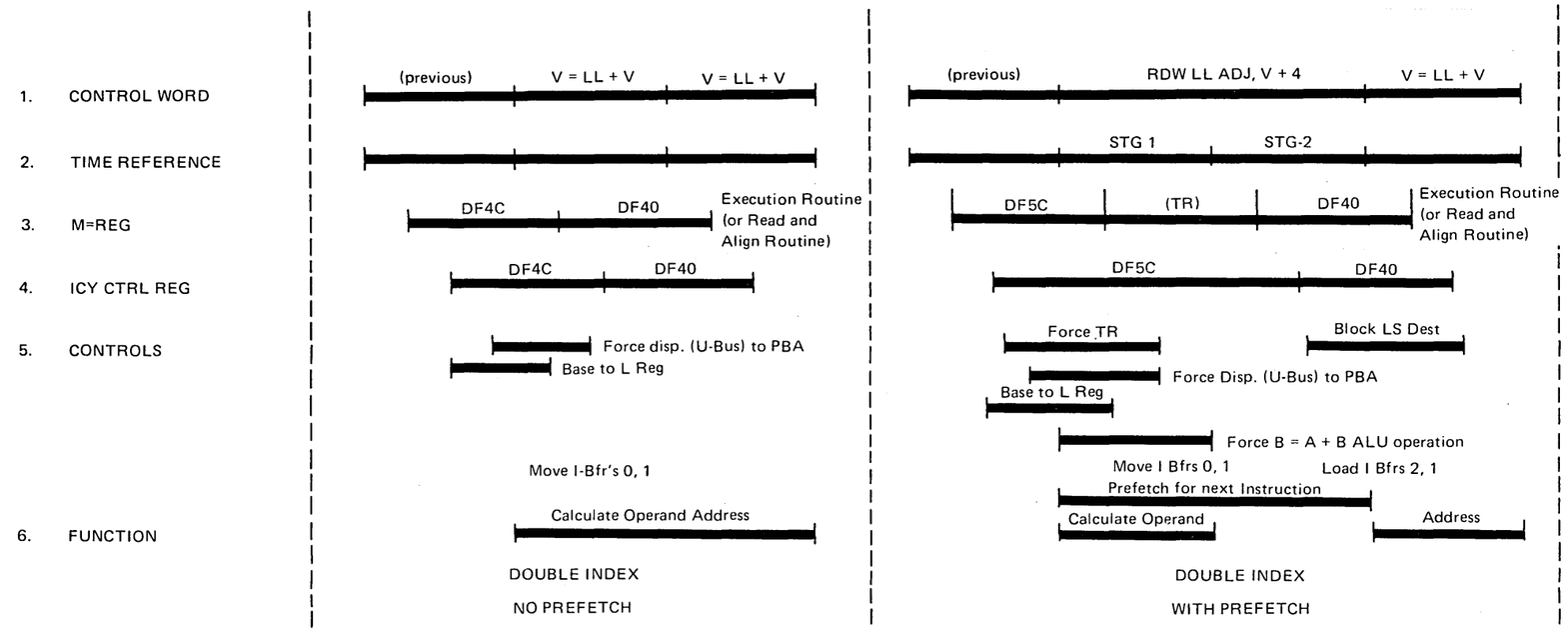
NOT PREFETCH



FLP LONG

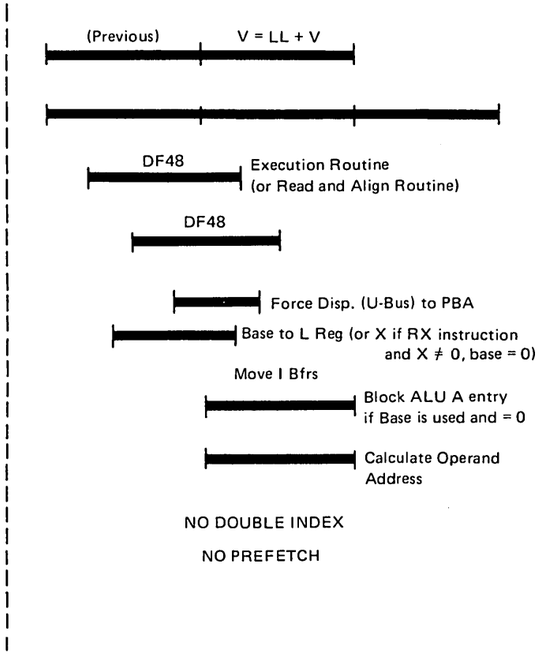
PREFETCH

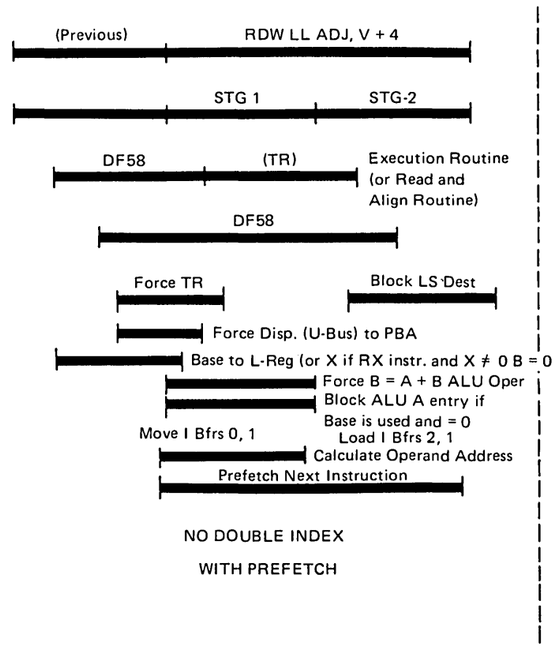
RX Sequences



RX, RS, SI Sequences

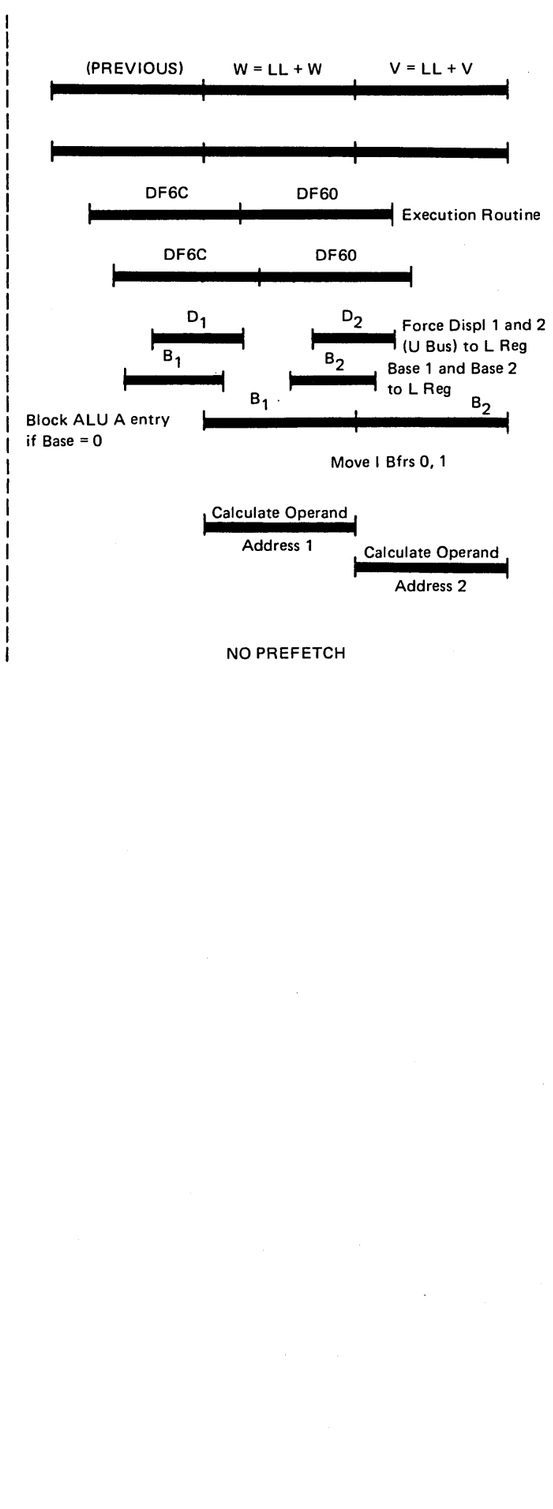
1. CONTROL WORD
2. TIME REFERENCE
3. M REG
4. ICY CTRL REG
5. CONTROLS
6. FUNCTION

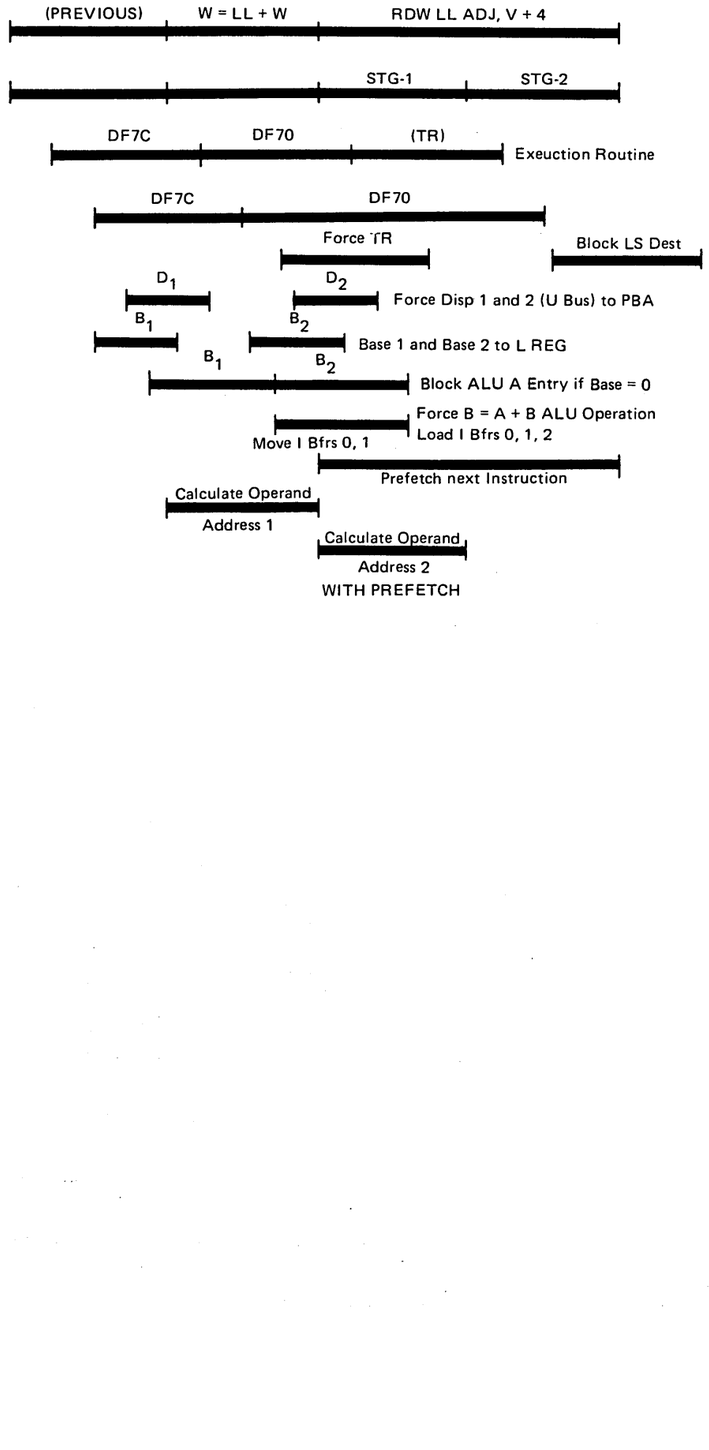




SS Sequences

1. CONTROL WORD
2. TIME REFERENCE
3. M REG
4. ICY CTRL REG
5. CONTROLS
6. FUNCTION

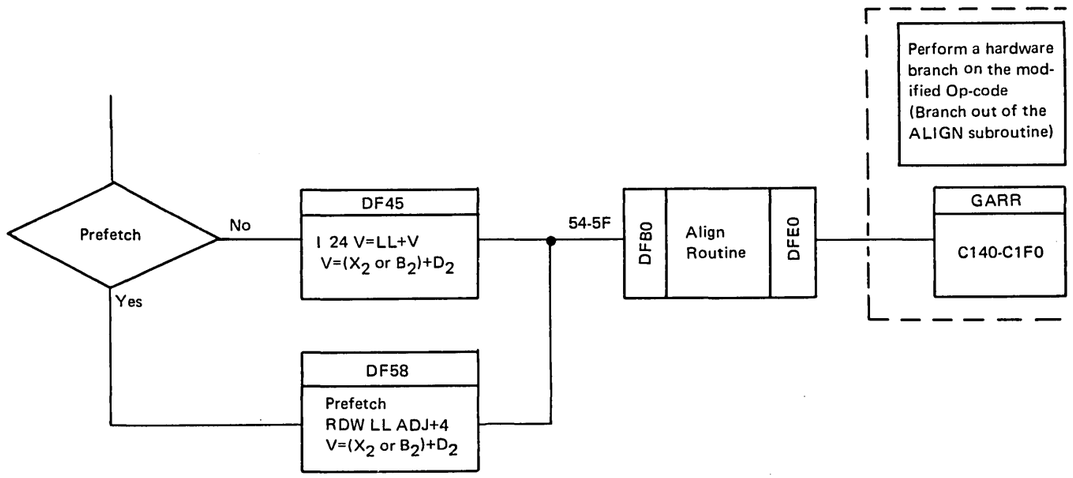


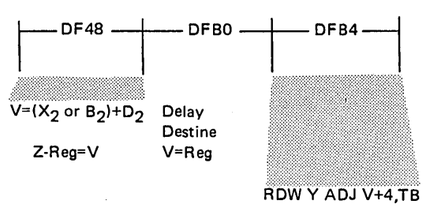


I-Cycles Alignment Routine

I-Cycle Entry with No Prefetch

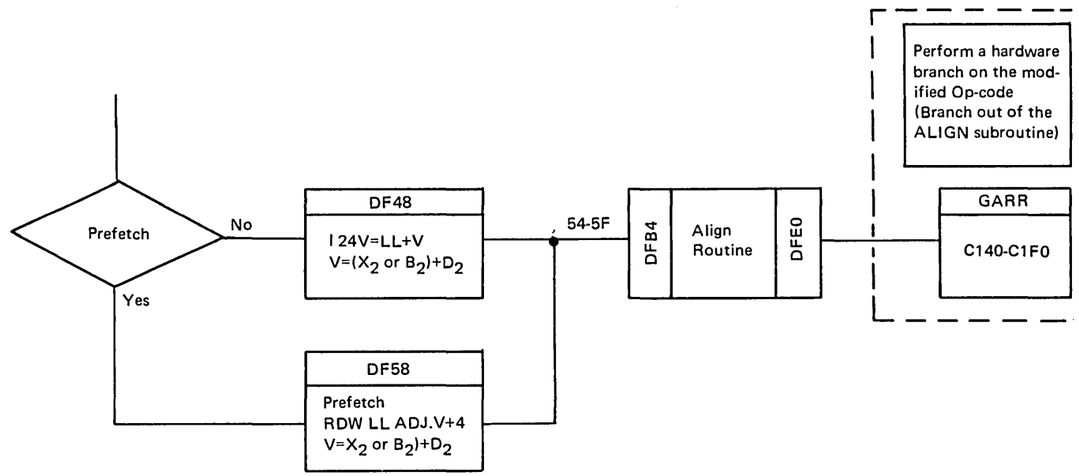
(Previous word was not a storage word.)
The align routine is entered at address DFB0, which is a delay word. The delay word allows the destine of the V-Reg.

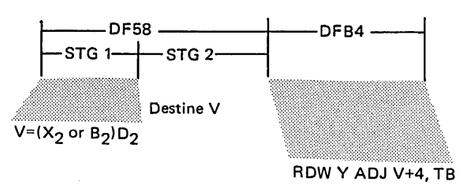




I-Cycle Entry with Prefetch

(Last word was a storage word, requiring a storage 1 and storage 2 cycle.)
The align routine is entered at address DFB4, thereby eliminating the delay word. The delay word is not necessary in this instance because the storage 2 cycle allows sufficient time for the V-Reg to be destined.



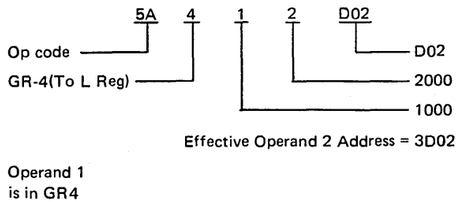


I-Cycle Alignment

The align routine enables data in main storage not on a word boundary to be aligned to appear on a word boundary.

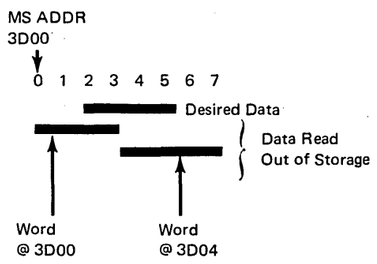
Consider the following example:

GR 1 = 00 00 10 00
GR 2 = 00 00 20 00

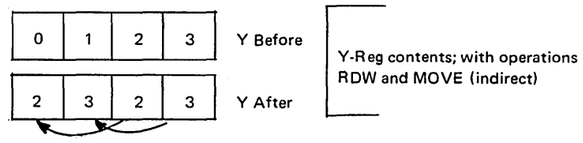


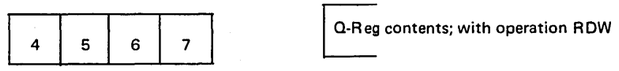
The data at main-storage address 3D02 is to be added to the contents of GR-4 and the result placed in GR-4.

In order to get the data at address 3D02, the problem arises that the desired data is not on a word boundary.

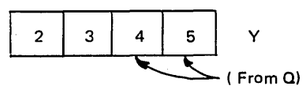


During the storage access, the bytes 0, 1, 2, and 3 are placed in the Y-Reg. Through the use of the TA-TB control-word function, bytes 2 and 3 are moved to bytes 0 and 1.





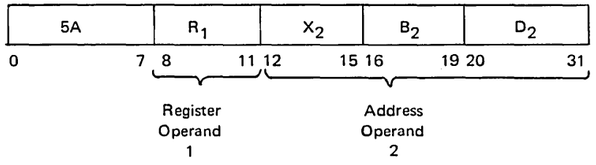
The information is then place in the Y-Reg (via an indirect MOVE operation) in the form;



I-Cycle Exit from the Alignment Routine

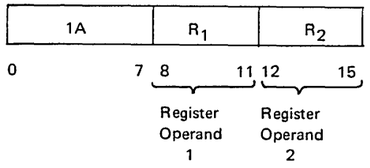
In some instances, it is possible to take an RX instruction and place the operand 2 data into a local-storage register. Once the operand 2 information is located in the register, it may use the RR format-execution routine. Consider the following example:

A $R_1, D_2(X_2, B_2)$ [RX]

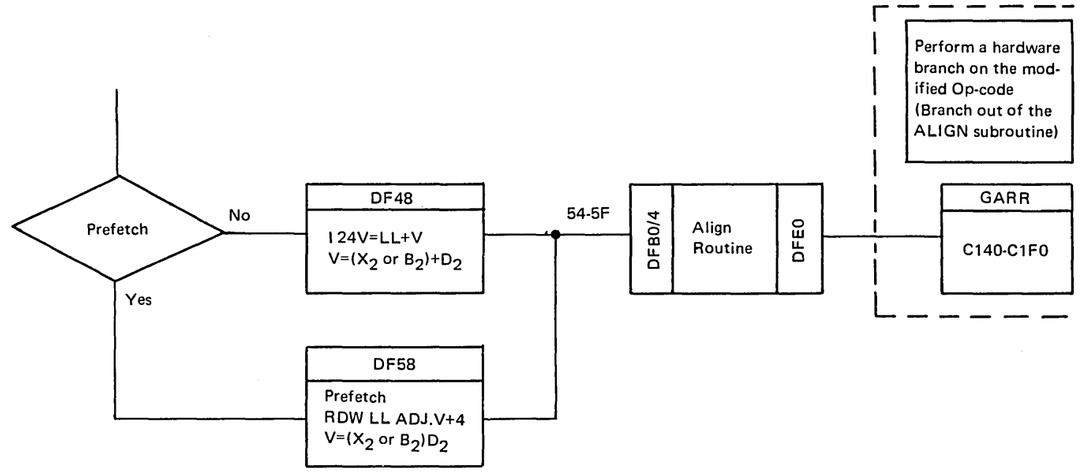


An RX '5A' add instruction has two operands, consisting of R_1 and $X_2 + B_2 + D_2$. If the operand two fields are combined, the instruction would then appear as a '1A' RR instruction. The operand 2 data for both formats is placed into the Y-register before entering the execution phase.

AR R_1, R_2 [RR]



During the align routine, the operand 2 data is placed in a local-storage register. Upon exiting the align routine, the RR execution routine may be used.

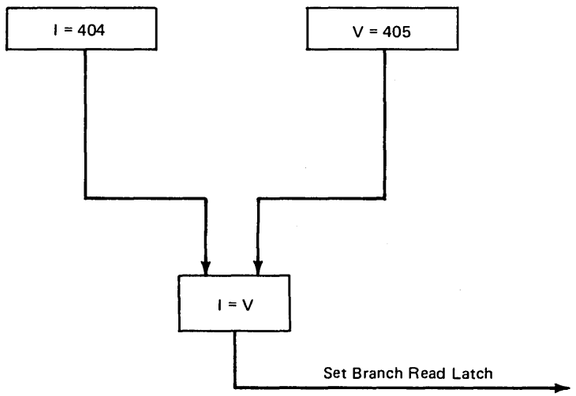


I-Cycles Program Modification

Program modification is detected by a store operation using V or W, as a storage address within the present, or next storage double-word address, as compared to the I- or TR-Reg. Only bits 8-28 are compared.

When program modification is detected, the branch read latch is set, and the modified instruction is loaded into the I-Bfrs.

In this example, the Insert Character Op code causes the modification of a portion of storage that is already loaded in the I-Bfrs. To continue operating on the old information in the I-Bfrs would cause the wrong result. To replace the old information, the branch read latch is set; and the modified information is loaded into the I-Bfrs early in the next I-phase.



I - BFR 2

D2	00	F1	24

I - BFR 1

D2	00	F1	24

I - BFR 0

43	20	04	05

```
ADDR.   INSTR.          LABEL
400     43200405                IC 2, INOUT + 1
404     D200F124F700    INOUT   MVC OUTPUT (00), INPUT
```

REMEMBER

There is a Reader's Comment Form at the back of this publication.

Control Word DF0C

ADDRESS	WORD	STATEMENT
DF0C	40 68 18 00	RDW Y ADJ, V + 4

C-Reg

C0	0 0	Storage Word	
	1 1		
	2 0		
	3 0	Read Word	
	4 0		
	5 0		
	6 0	Branch High (0) (To bit 4 of M-3)	
	7 0		
C1	0 0	Data Register (Y-Reg)	
	1 1		
	2 1		
	3 0		
	4 1	Increment +	
	5 0	Stat Set	
	6 0		
	7 0		
C2	0 0	Address Source (V-Reg)	
	1 0		
	2 0		
	3 1	Mode (Addr Adj)	
	4 1		
	5 0		
	6 0	Special Stat Set —	
	7 0		
C3	0 0	Next Address	
	1 0		
	2 0		
	3 0		
	4 0		
	5 0		
	6 0	Branch Low (0)	
	7 0		

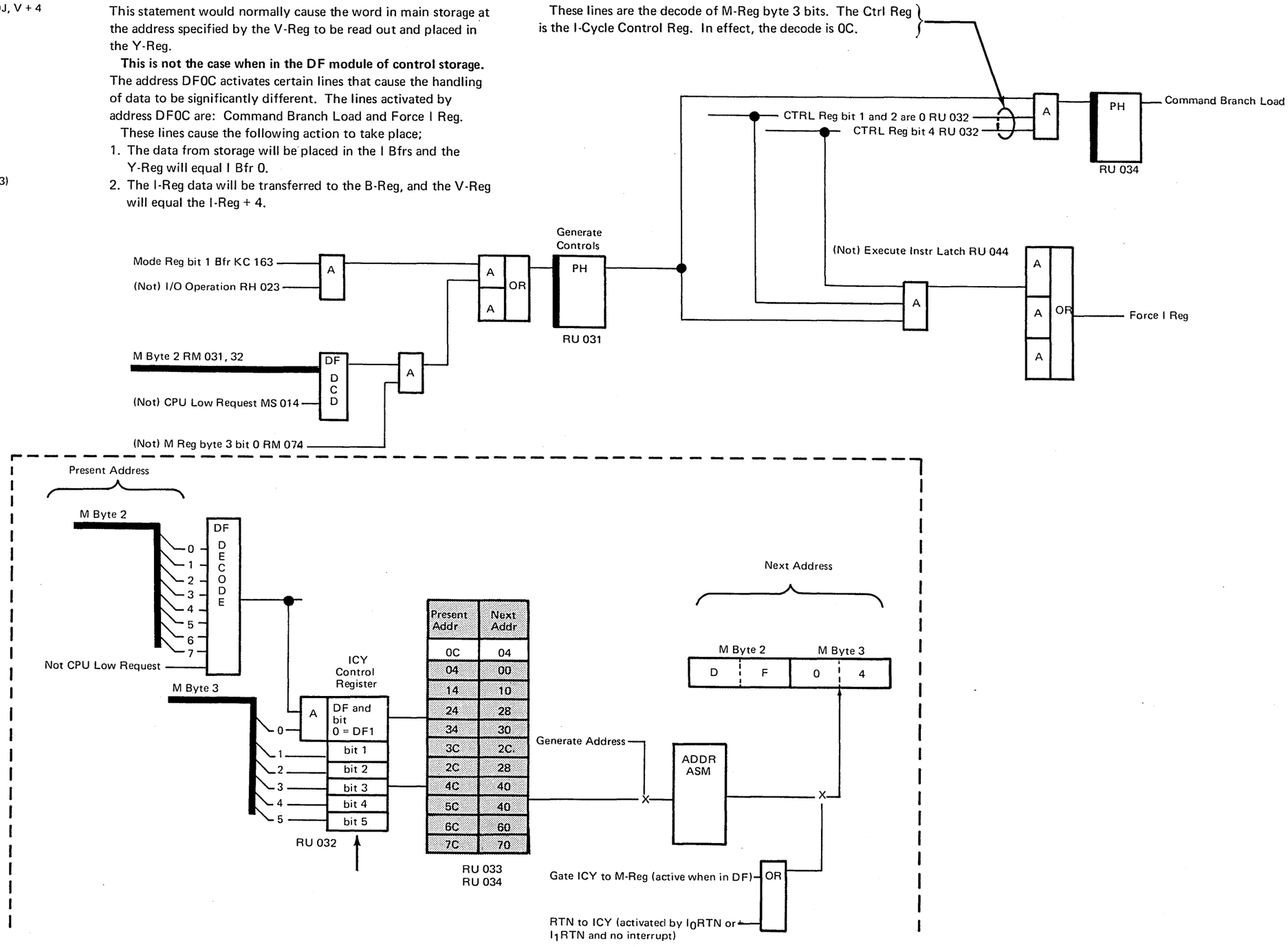
RC 111-191

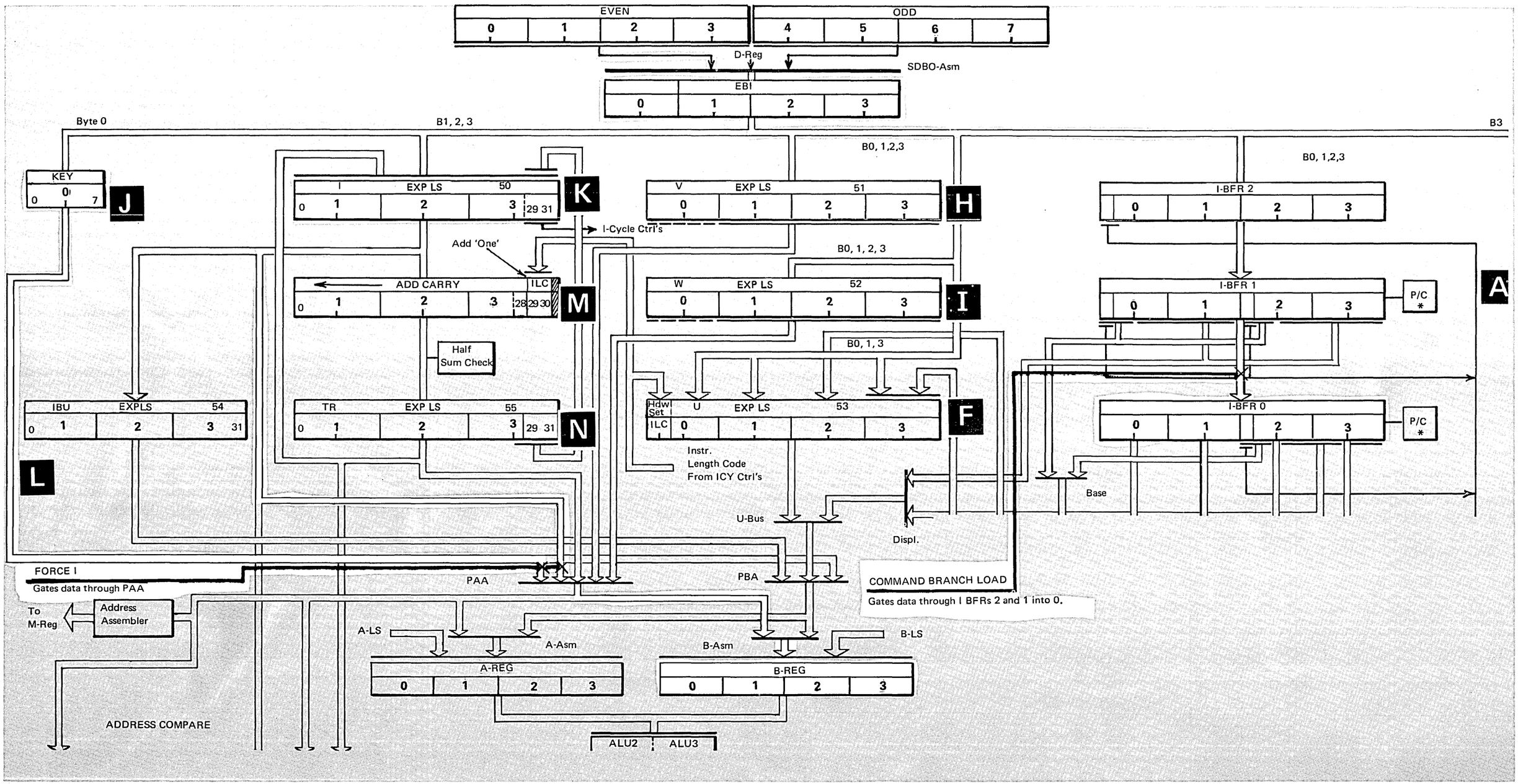
This statement would normally cause the word in main storage at the address specified by the V-Reg to be read out and placed in the Y-Reg.

This is not the case when in the DF module of control storage. The address DF0C activates certain lines that cause the handling of data to be significantly different. The lines activated by address DF0C are: Command Branch Load and Force I Reg.

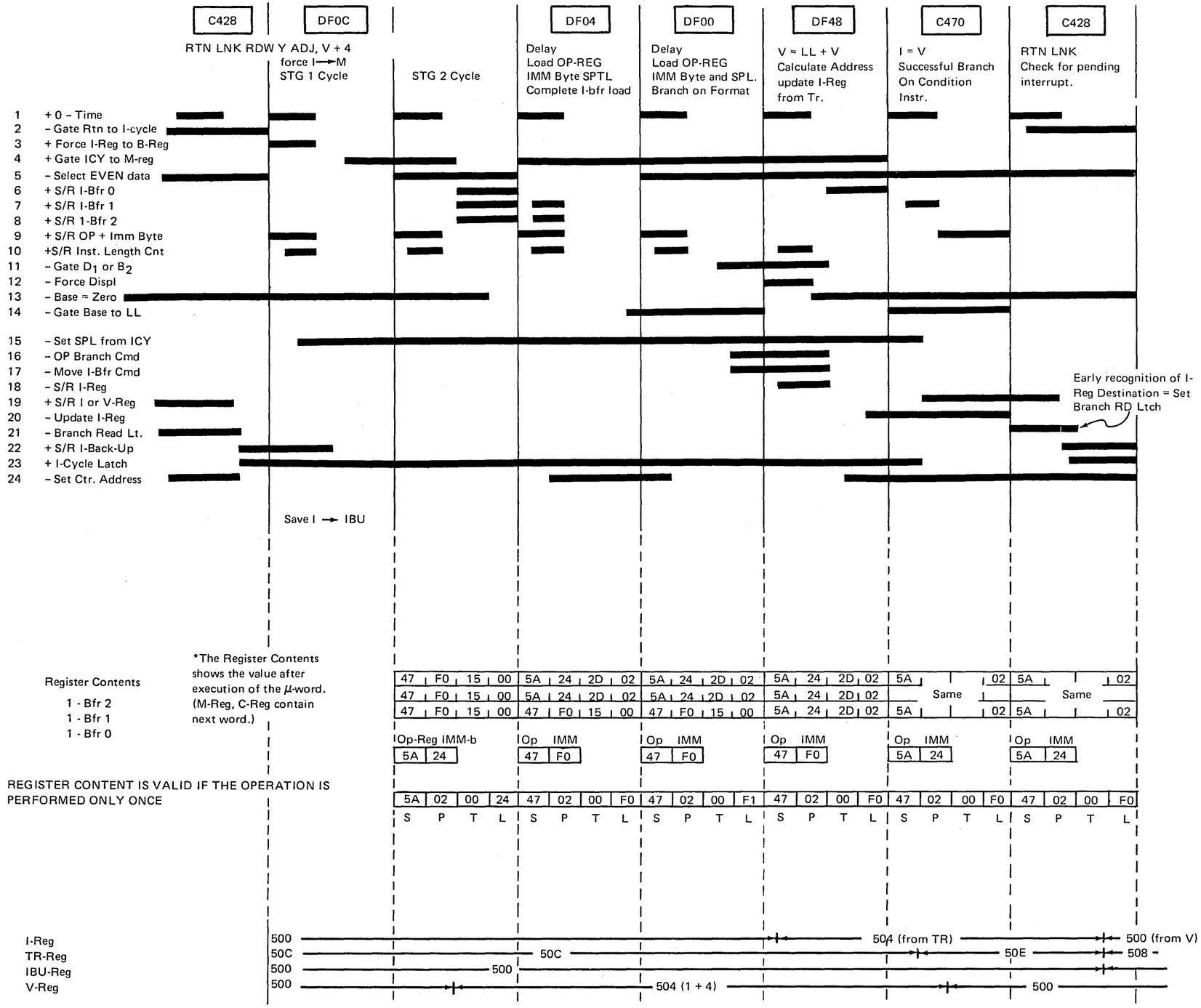
These lines cause the following action to take place;

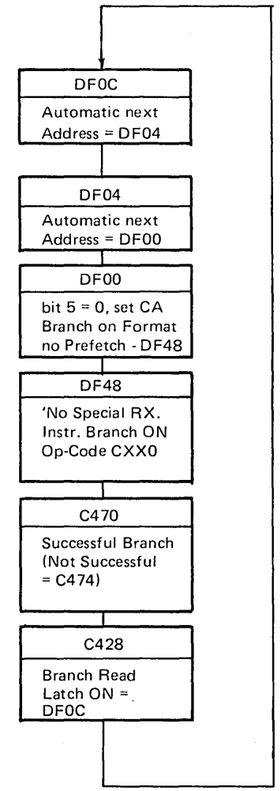
1. The data from storage will be placed in the I Bfrs and the Y-Reg will equal I Bfr 0.
2. The I-Reg data will be transferred to the B-Reg, and the V-Reg will equal the I-Reg + 4.

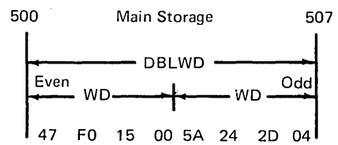



EVEN
0
1
2
3
ODD
4
5
6
7
D-Reg
SDBO-Asm
EBI
0
1
2
3
Byte 0
B1, 2, 3
B3
B0, 1,2,3
B0, 1,2,3
KEY
0
7
J
I
EXP LS
50
K
V
EXP LS
51
H
I-BFR 2
I-Cycle Ctrl's
Add 'One'
B0, 1, 2, 3
ADD CARRY
ILC
M
W
EXP LS
52
I
I-BFR 1
P/C
A
Half
Sum Check
B0, 1, 3
IBU
EXPLS
54
TR
EXP LS
55
N
Hdw
Set
ILC
U
EXP LS
53
F
I-BFR 0
P/C
L
Instr.
Length Code
From ICY Ctrl's
Base
U-Bus
Displ.
FORCE I
Gates data through PAA
PAA
PBA
COMMAND BRANCH LOAD
Gates data through I BFRs 2 and 1 into 0.
To
M-Reg
Address
Assembler
A-LS
A-Asm
B-Asm
B-LS
A-REG
B-REG
ADDRESS COMPARE
ALU2
ALU3

Branch Loop Example







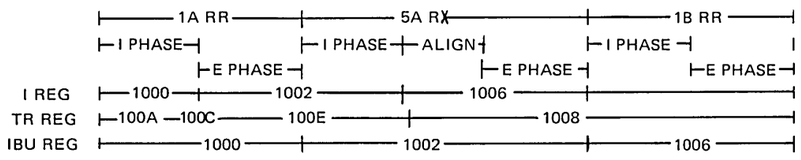
GR1 = 00000000

The base register is assumed to be zero for this example, which causes an unconditional branch to the first instruction (47). The ADD instruction (5A) is used only to provide data for the registers.

Partial Instruction Stream I-Cycles Example

The following instructions are contained in main storage.

1A 32 *5A* 41 2D 02 *1B* 46 (IC set to address 1000)



Add (1A) Instruction

The 1A instruction starts at address 1000 in main storage. To start processing at that address, assume that the set IC key is pressed. During the set IC microroutine, the I-Reg is destined. (This sets the instruction counter to address 1000). The IBU Reg has address 1000 and the TR Reg contains address 100A, the beginning address of the next doubleword. Destining the I-Reg as a result of the set IC microroutine causes the branch read latch (RU 043) to set. Upon completion of the set IC routine, the start key must be pressed, which initiates the start microroutine. At the end of the start microroutine is a I0 RTN LNK microword. The combination of the I0 RTN LNK and the fact that the M-Reg is not at address DFXX along with branch read latch (previously set on) forces the M-Reg to address DF0C. The M-Reg addresses control storage and reads out the control word at address DF0C.

CONTROL WORD AT DF0C- (RDW Y ADJ, V + 4)

1. M-reg is at DF0C.
2. This is a RDW, force I to B and read the doubleword at MS 1000 and put it on SDB0.
   a. This double word (1A32 5A41 2D02 1B46) will be in the SDBO Pre-Asm early in stored bycle 2 (storage cycle 1 was addressing MS).
   b. Late in Stg Cycle 2, the even word from the SDBO Pre-Asm will be gated to the SDBO Asm through EBI to the I-bfrs. I-Bfrs 0, 1, and 2 will be "set/reset" (S/R); and at the end of storage 2 cycle, all three Bfrs will contain the same information (the even word).
   c. During the control word DFOC, the M-reg was set to address DFOC. Through hardware, the M-Reg is forced to DF04 by using the gating line 'Generate Address' (*RU 031).*
   d. The M-Reg reads out the control word from control-storage address DF04.

Control Word at DF04 – (Y = Y, S0)

1. Early in the cycle, the odd word of the doubleword read out during DFOC is gated from the Pre-Asm to the SDBO Asm through EBI to the I-Bfrs. This time *only* I-Bfrs 1 and 2 will be 'S/R'. I-Buffers 1 and 2 contain the same information, the odd word I-Bfr 0 contains the even word.

   I BUFFERS

2	2D02	1B46
1	2D02	1B46
0	1A32	5A41

2. The Op code (RU 118) and the Imm byte (RU 128) are set into the Op Reg and Imm byte Reg respectively. (I-Reg indicates which byte to gate to the Op-Reg).
3. The ICY controls decode the Op code and set the ILC in U0 Bits 0 and 1. The ILC is also made available to the add-carry (it is available to the add-carry at the time the I-Reg is updated).
4. The Op Reg decode and the I-Reg are used to determine whether the present instruction is fully contained in the buffers. The Op code indicates the length of the current instruction, and the I-Reg indicates what part of the doubleword the current instruction came from. Therefore, through this combination of decoding, it can be determined whether further fetching is needed. (*Further fetch* is defined as when the present instruction is not fully contained in the I-buffers). In this case, no further fetching is required. The present instruction is fully contained in the Bfrs. The Op-load latch (RU 043) is not turned on.

Control Word at DF04 (Cont'd)

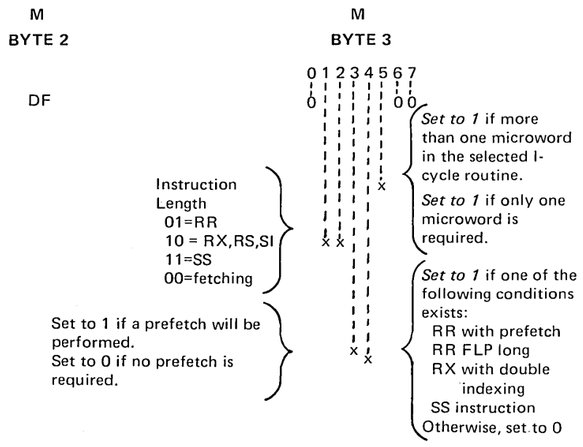
5. The present instruction is fully contained in the I-Bfrs. The I-Bfrs must be decoded to see whether the next instruction is fully contained in the I-Bfr's. Instruction Look Ahead is used to decode the format of the Op code that is following the current instruction. The decoding of this format indicates the length of the instruction that follows the current instruction. The I-Reg keeps track of the address that the data came from within the doubleword. Through this combination of decoding, it can be determined whether the next instruction is fully contained in the I-Bfr's. If not, a prefetch is performed (a *prefetch* is defined as: when the *next* instruction is not fully contained in the I-Bfr's) the next doubleword is loaded using the TR-register contents as an address. If there is not enough room in the I-Bfr's for the next doubleword, the prefetch is blocked. Prefetch is also blocked on all branch type O-codes. (In this case a prefetch is not required).
   The S- and L-Re gs are set from the Op-Reg and Imm-Byte-Reg. This occurs during ICY and when a cycle is taken. LL = operand 2. LH = Operand 1
6. During the control word DF04, the M-Reg was at address DF04. The M-Reg is forced via hardware to address DF00, by the gating line 'Generate Address' (RU 031).
   M-Reg at DF00, the M-Reg reads out the control word from control storage address DF00.

Control Word at DF00 (Y = Y, S0)

1. This is a delay word to allow the hardware to develop the next address.
2. The gating line 'Set Control Address' (RU 032) is needed because a new sequence of Control words is being started. Note that DF20 does not always follow DF00. In this case, the Op code has been decoded to 1A and will cause a branch to DF20.
3. During the control word DF00, the M-reg was forced to DF20.

Next ICY address formation

Once the instruction is contained within the I-buffers, the next address is formed as follows:



M-Reg at DF20, the M-Reg reads out the control word from control-storage address DF20.

Control Word at DF20 (Y = LL, DF) (the last cycle in I-Phase)

1. The word Y = LL, DF (LL points to operand 2).
2. The contents of GR2 are placed in the Y Reg.
3. At the end of this cycle, the value to be destined to the Y-Reg will be in the Z-Reg.
4. U2 and U3 are set from the Op Reg and Imm-Byte-Reg respectively, so that the information will be available to the microprogram during E Phase.
5. Update I-Reg (I = 1002) (*Note:* TR will be updated after the S/R to the I-Reg).
6. At the conclusion of I-phase, the Op code has been decoded (in this case 1A) and the next address is known (C1A0).
7. The M-Reg is forced to C1A0.

M-Reg at C1A0, the M-Reg reads out the control word from control-storage address C1A0.

EXECUTE PHASE (I-CYCLES)

The control word at address C1A0 is read out (The first cycle of E-Phase of the 1A instruction).

Control word at C1A0 - (LHC = LH + Y)

A. The following hardware functions are completed for I-cycles.
   1. The Bfr's are still being move.
   2. S/R the Op-Reg and the Imm Byte Reg from the next Op Code and Imm Byte. The I-Reg was updated just before leaving I-phase; therefore, the I-Reg is pointing to the next instruction.
   3. In E-Phase, the Op Reg and Imm byte Regs cannot be set into the S- and L-Regs. The system must be in I-cycles to gate the Op-Reg and Imm-byte-Reg to the S- and L-Regs.
4. 4. The TR-Reg will be updated to point to the next double-doubleword.

B. The control word read from address C1A0 performs the f following.
   1. LH is pointing to GR3
   2. Y has not been destined (Z = Y).
   3. At source time, the Z-Reg is gated back to the B-Reg controlled by Destination Look Ahead, and the contents of GR3 are gated to the A-Reg. At the end of this cycle, the sum of GR3 and 2 will be in the Z-Reg and will be destined to GR3 in the next cycle.
   4. The 1A add instruction is now complete.
      a. The S-Reg will be changed by Stat sets.
      b. The U2- and U3-Regs still contain the Op code and Imm byte of the current instruction.
   c. The L-Reg still contains the address of operations 1 and 2.
   d. The microcode sets the condition code, test overflow, etc.
   e. While in E-phase of the 1A instruction, the ICY controls are decoding the next Op code to determine the controls required for the next I-phase operation and the first address that will be used upon returning to ICY.

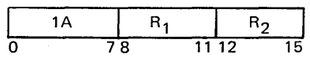
Example 2B Add (5A) Instruction (Double Indexing with Alignment)

1A 32 *5A* 41 2D 02 *1B* 46

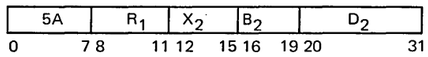
Operand 1 = GR4
Operand 2 = main storage addr 3D02

Add

AR R_1, R_2 [RR]



A $R_1, D_2(X_2, B_2)$ [RX]



The second operand is added to the first operand, and the sum is placed in the first operand location.

RX With Double Indexing and Alignment

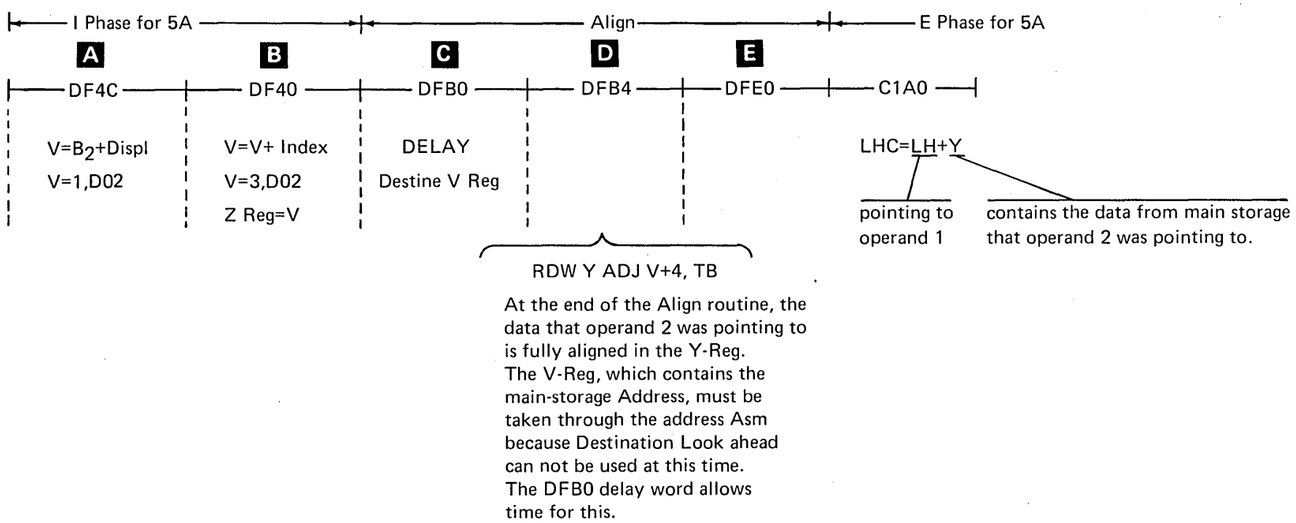
- The definition of *double indexing* is: Neither base nor index fields are using GR0 (zero). Another control word is needed to calculate the operand.
- Alignment: (operand 2 is pointing to a storage address; the data at that address will be added to the contents of the GR that operand 1 is pointing to).

  Fetch data from storage (using operand 2).

  Align data in the GR.

  Both operands are now in local storage.

  The same routine may be used as in the RR Add to add the two operands because both operands are in local-storage registers.



A *Developing address DF4C after E-Phase of the 1A Op-code.*
During E-phase of the 1A Op code, the Op-Reg contains the next Op code 5A. During this time, ICY controls are setting up for the next address in the SF module by decoding the next Op code 5A. The next address 4C is developed by the lines ILC, RX, RS, SI, SS, and IF MORE THAN ONE M-WORD. (The line 'if more than one M-word', is developed from the decode of double index). After the execution phase of the Add (1A) operation is complete, the RTN LNK microword develops the gate line 'set control address' (not DF'). 'Set control address' gates the developed address 4C to the M3 assembler, and gate DF, to the M2 assembler. Rtn to ICY gates the assemblers to the M-Reg, setting the M-Reg to the next address DF4C.

At the end of DF4C, the V-Reg = Base + Displacement (2D02)

B The transition from address DF4C to DF40 is accomplished by using the decode of the ICY Ctrl Reg 4C and gating line 'Generate Address'.

At the end of DF40, the V-Reg will contain X + B + D (3D02).

C Address DF40 to address DFB0; (going to the Alignment Routine)

1. 'RX with alignment' develops the gating line'Op-Branch to DF'.
2. 'Op Branch to DF' gates 'Align Entry' (in this example, the last control word in ICY is not a prefetch; therefore, the 'Low Bit Y' will be zero and the Op code 5A will develop a B) to the M3 assembler, and DF to M2 Assembler.

'Gate ICY to M-Reg' gates the assemblers to the M-Reg. The M-Reg contains DFB0, which is the first address in the align routine.

Being in the align routine deactivates the ICY controls. This is done by M3 bit 0 being on, which deactivates 'generate address' latch and 'Generate Controls'. With the ICY controls deactivated, the control words in the align routine develop the next address. (The exception to this is the first and last addresses of the align routine, which are developed by ICY controls).

Note the setting of U2-U3, updating I-Reg, etc., when going from DF40 to DFB0 (Op Reg and Imm byte will be set in the first control of E-phase of the 5A Op-code).

The first control word in the align routine is a delay word to allow the destine of the V-Reg. This is necessary because the next

D control word 'DFB4' in the alignment routine is RDWY ADJ J + 4, TB. The V-Reg is addressing storage, and 'ADJ' (hardware adjustment) specifies that the V-Reg must be taken through the Addr Asm to address storage. Destination Look Ahead may not be used to get data to the Addr Asm; therefore, DFB0 must be used to allow destine of the V-Reg so that it will be available to the address Asm during 'DFB4'.

At the end of the alignment routine, the data that operand 2 was pointing to is fully aligned and is in local storage (in the Y-Reg).

E The last address in the align routine is DFE0, M3 bit 0 being on and the line *'F0, E0 (M3)'* activates the gating line 'Op Branch' (on Op code). Upon exiting the align routine, the Op code 5A is still in the Op-code register. The gating line 'Op Branch' *(on Op-code)* causes a hardware branch. *LH is pointing to operand 1 (GR4), and the data that operand 2 was pointing to is aligned and in the Y-Reg. It is now possible to branch to the RR add routine and add the two locations in local storage. The first control word found in the RR add (at address C1A0) is LHC = LH + Y. Hardware takes the present Op-code 5A and 'minus 4' from the left hex digit of the present Op-code. Therefore, the present Op-code (5A) is changed to Op-code of 1A. This is how the ICY controls set up the next address when leaving the alignment routine of some RX instructions and Op-branch (on Op-code) to the execute routine for a RR instruction.

MVC (D2) Instruction Example

MAIN-STORAGE
ADDRESS
1004

D2 0F 1D 01 ¦ 2D 02 *47* F1 05 01 *1A* 5A

Further fetch required because the current instruction is not fully contained in the I-Bfr's.

I-Phase for MVC (D2) Requiring Further Fetch

- Set I/C to 1004.
- DF0C (M-3 bit 5 is on, forcing the odd/odd time slot). Gate in storage cycle 2 I-Bfr's 0, 1, and 2 will be S/R with the odd word.
- DF04: early in the cycle, I-Bfr's 1 and 2 will again be S/R with the odd word. The Op-Reg, Imm byte, ILC and S and L will be set. If the current instruction is not fully contained in the Bfr's the 'Op latch' turns on.
- DF00: delay 'set control address'.
- DF14: 'Op load' latch on and 'set control address' caused generation of DF14. RDW: force TR to B-Reg (fetch next doubleword). Storage 2 cycle S/R Bfr's 1 and 2 from even word. Next, the 'generate address' gating line is needed.
- DF10: (delay) S/R Bfr 2 from odd word. Format the branch. A new sequence of words is being started in ICY. Therefore, the gating line 'set control address' is needed. The Op code in the next address '6C' is developed by:

  ILC, and SS develop the '6'.
  If there is more than one control word and the control line 'not RR or FLP with prefetch' will develop the "C".

- DF6C W = LL + W (add base 1 + displ1 to W-Reg) *Block 'W' as a source and gate Displ. 1 from Bfr 0 to the B-Reg. LL points to base register. At the end of this cycle, the Z-Reg will contain operand 1, B2 is gated from Bfr 1 to LL. (B2 must be in LL for the next address to calculate operand 2). To develop the next address, the gating line 'generate address' is activated.
- DF60 V = LL + V, Block V as a source, gate displ. 2 from Bfr 1 to the B-Reg. LL is pointing to Base Reg 2. During this cycle, the W-Reg was destined. At the end of this cycle, the Z-Reg will contain operand 1. 'Op Branch' to CD20. Leaving I-phase to E-phase, set U2 U3. Update the I-Reg (must update I before the Bfrs are moved and set Op-Reg and Imm byte). MWve Bfrs (this overlaps into first cycle of E-phase).
- CD20: Set Op-Reg and Imm byte with Op-code and Imm byte of the next instruction. (The I-Reg indicates where the Op-code and Imm byte are in the Bfr's.
- CD20: Set ILC (from next Op code). Update TR. The S and L Regs *will not* be set from the Op and Imm byte because the addressing range is out of ICY (DFXX Module).
- During E-phase of the MVC, the microprogrammer has the Op-code and Imm byte available in U2 and U3. The length will be decremented in U3. The condition code will be set in U0 bits 2 and 3. The S-Reg will be changed by Stat sets. Move characters. During E-phase, the ICY controls are decoding the next Op-code 47 to develop the first address that will be used when returning to ICY after E-phase of the MVC Op-code.
- XXXX - RTN LNK I1 - The ICY controls provide an I-cycle starting address of: DF48. If there is no interrupt pending, this is the data value set into the M-register.
- DF48 - V = LL + V - Block V as a source, gate displ2 from Bfr 1 to the B-Reg. LL is pointing to base Reg 1. 'Op Branch' to C470. Leaving the I-phase, set U2U3. Update the I-Reg and move the I-buffers.
- C470: Beginning of E-phase for branch instruction. V-Reg is destined with sum Base + Displacement.
- Note that a software branch has occurred. Refer to example 1 for any further explanation.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

Execute 44 Instruction Example

The execute instruction causes one instruction in main storage to be executed out of sequence without actually branching to the object instruction. For example, assume that a move (SI) instruction is located at address 3820, with format as follows:

Machine Format

OP Code	I_2	B1	D1
92	66	C	003

0 7 8 15 16 19 20 31

Assembler Format

OP CODE	D_1	B_1	I_2
MVI	3 (12),		X'66'

where register 12 contains 00 00 89 16.

Further assume that at storage address 5000, the following execute instruction is located:

Machine Format

OP CODE	R_1	X_2	B_2	D_2
44	1	0	A	000

0 7 8 11 12 15 16 19 20 31

Assembler Format

OP CODE	R_1	D_2	X_2	B_2
EX	1,	0	(0,	10)

where register 10 contains 00 00 38 20, and register 1 contains 00 0F F0 99.

When the instruction at 5000 is executed, bits 24-31 of register 1 are ORed inside the CPU with bits 8-15 of the instruction at 3820:

Bits 8-15: $0110\ 0110_2 = 66$

Bits 24-31: $1001\ 1001_2 = 99$

Result: $1111\ 1111_2 = FF$

causing the instruction at 3820 to be executed as if it originally were:

Machine Format

OP CODE	I_2	B_1	D_1
92	FF	C	003

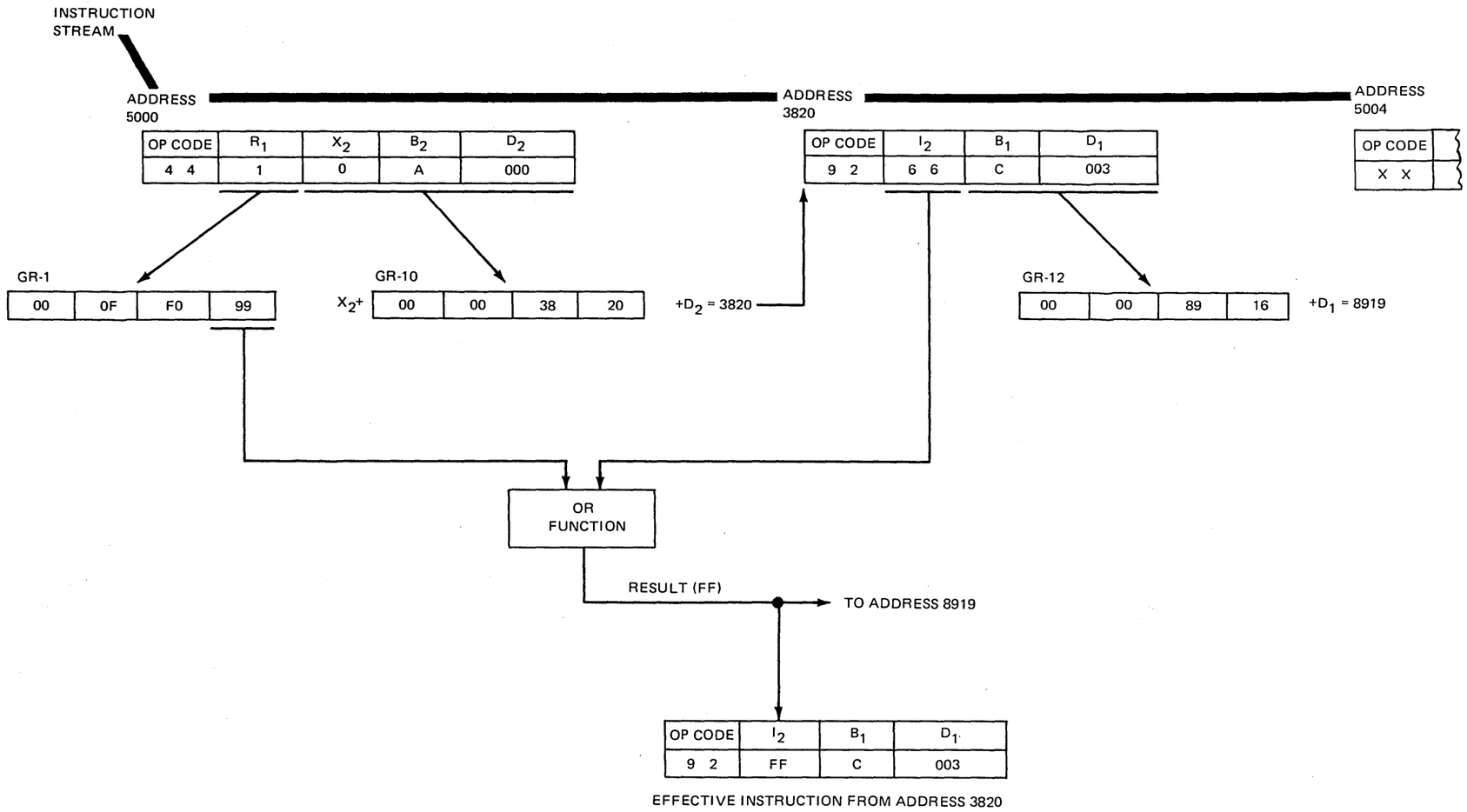
0 7 8 15 16 19 20 31

Assembler Format

OP CODE	D_1	B_1	I_2
MVI	3 (12),		X'FF'

However, after execution:

Register 1 is unchanged.
The instruction at 3820 is unchanged.
Storage location 8919 contains FF.
The CPU next executes the instruction at address 5004
(PSW bits 40-63 contain 00 50 04)



EFFECTIVE INSTRUCTION FROM ADDRESS 3820

1. a. Assume that the instruction being processed had prefetched storage location 5000.
   b. During the E-phase, the I-cycle starting address of DF48 is provided as data input to M-Reg.
2. *I0 for Instruction O-Code* '44'

- Base 1 was gated to LL during the Rtn to ICY.
- I-Reg = 5000
- During E-phase of the previous instruction, ICY controls recognized that the current and the next instructions are fully contained in the I-Bfr's. Therefore, the first address of ICY for Op-code 44 is DF48.
- DF48 'set control address' and the lines 'ILC, RX, RS, SI, SS, and not RR or FLP with prefetch' cause address DF48 to be gated to the M-Reg.

  NOTE: Op Branch to DF (The three control words used for the 'Execute Instr' are referred to as "align". That is, they are in the DF module; and when entering these three words, the ICY controls are inactive. The I-Reg will be updated to 5004 (next sequential instr).
- DFA4: 'Op Branch to DF gates out align entry A4.
- The word at DFA4 moves the GR specified by the R1 field to the W-Reg, and in doing so:
  a. Sets the execute latch on (I- or V-Reg will be V)
  b. Loads the immediate byte modifier Reg-and ORs it with Imm byte Reg because R1 is not GR0.
- Rtn Lnk (DF84): The first address in ICY will be DF0C because the 'branch read' Latch was found ON during DF84.

I-Phase for the Subject Instruction

- DF0C: The 'execute' latch is on, indicating that the V-Reg should be used in place of the I-Reg to address storage. The V-Reg contains the address of the subject instruction that was calculated as operand 2 during I-phase of the execute instruction.
- V + 4, used if further fetch is required to load the subject instruction.
- After E-phase of the subject instruction, return to DF0C and reload the Bfr's.

The I-Reg was not updated during the subject instruction; therefore, the next sequential instr. will be loaded from the instr. stream (from address 5004).

A-REGISTER, B-REGISTER, AND ALUs

Arithmetic and logic operations in the CPU are processed by two one-byte arithmetic and logic units (ALUs).

ALU operations and program symbols:

Symbol	Operation
,A,	AND
,OR,	OR
,OE,	Exclusive OR
+	True ADD
–	Complement ADD
,D+–,	Decimal ADD
+–	Binary ADD
,A–,	Complement AND

The two input operands for the ALUs are entered into the A-register and the B-register and gated through their output assemblers.

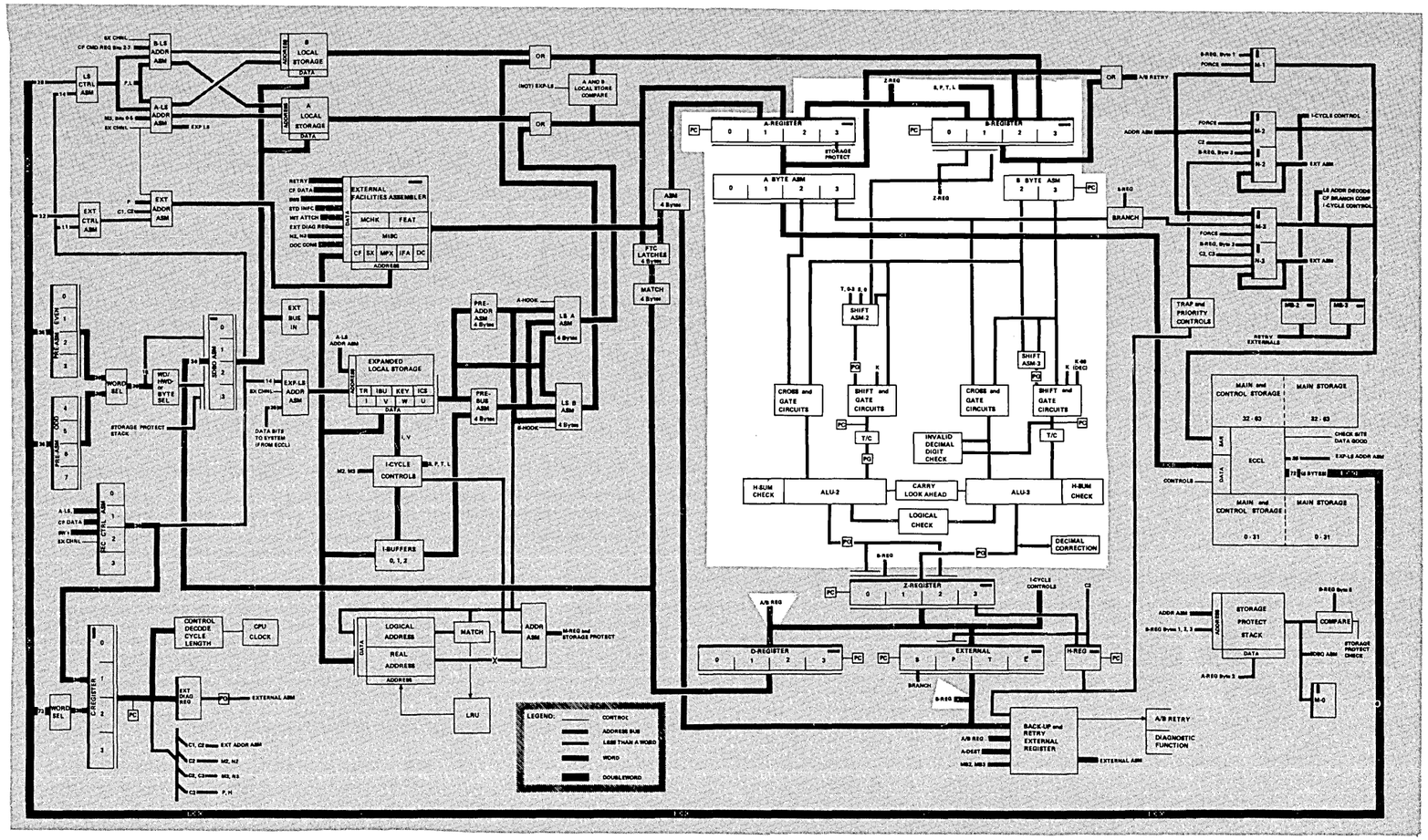
Two ALUs are provided to allow simultaneous operation in halfword operations and to provide checking during logic-byte operations.

By using two ALU cycles during a CPU cycle, the ALU system can binary-add two fullword operands.

Normal operations are for either byte operands or fullword operands as defined by the control word.

Special gating controls are provided in the entry of each ALU to allow manipulation of byte operands for logic and arithmetic operations.

The description of ALU operation in control words is given in Chapter 4 of this manual.



A-REGISTER AND A-BYTE ASSEMBLER

The A-register serves as the A source entry for the ALU system and to enter data into main storage through the SDBI.

Input data for the A-register comes from the A local storage, the external registers, or a previous ALU output from the Z-register.

Parity is checked on all four bytes of the A-register during entry. An error condition is reported in MCKA byte 2 bit 4.

The A-register is a fullword (four-byte) register that normally enters the fullword.

The A byte-assembler provides the means of presenting the A-register bytes to the ALU system and to the EDBI.

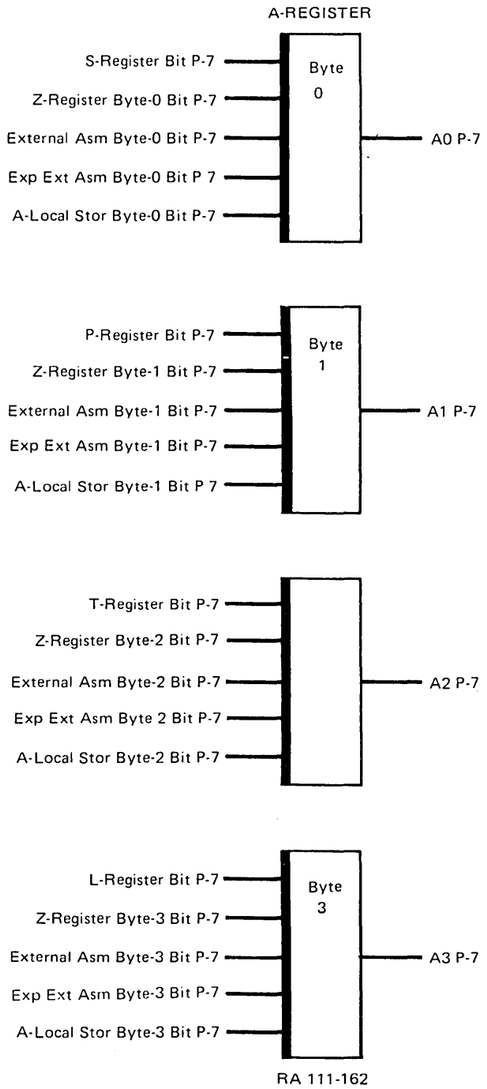
The A byte-assembler has a four-byte output to allow assembling a fullword to feed the EDBI.

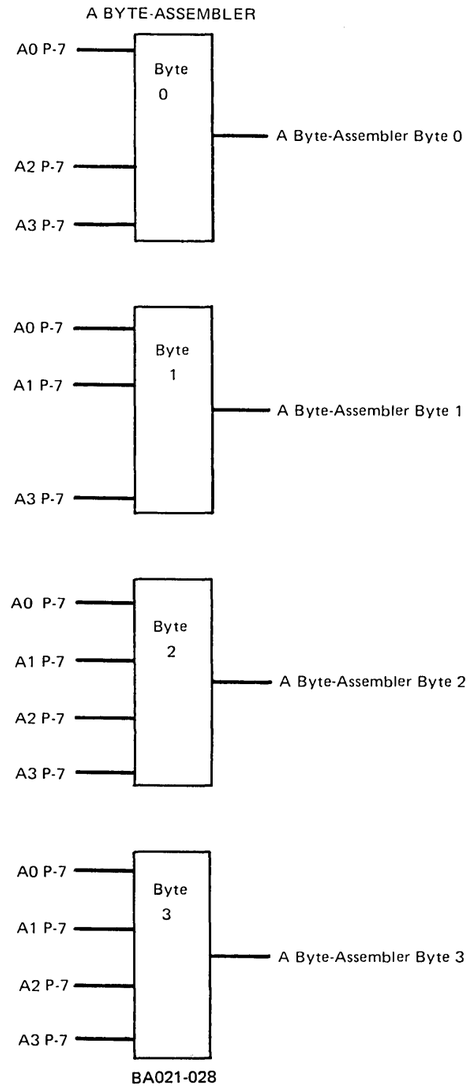
Only the byte 2 and byte 3 assemblers feed the ALU system and have gating to enter any of the four A-register bytes into either output.

Byte 0 assembler cannot gate the A-register byte 1, and byte 1 assembler cannot gate the A-register byte 0 into their outputs, because they are not required in defined operations.

The high-order three bytes of the A byte-assembler do not have a parity check but depend on the receiving area checks.

The byte 3 assembler is parity-checked because of its use in decimal adding, and the error condition is reported as an A-register error.





B-REGISTER AND B-BYTE ASSEMBLER

The A-register serves as the B source entry for the ALU system and the data address entry for the M and N registers.

Input data for the B-register comes from the B local storage, the A-register, the SPTL external, or a previous ALU output from the Z-register.

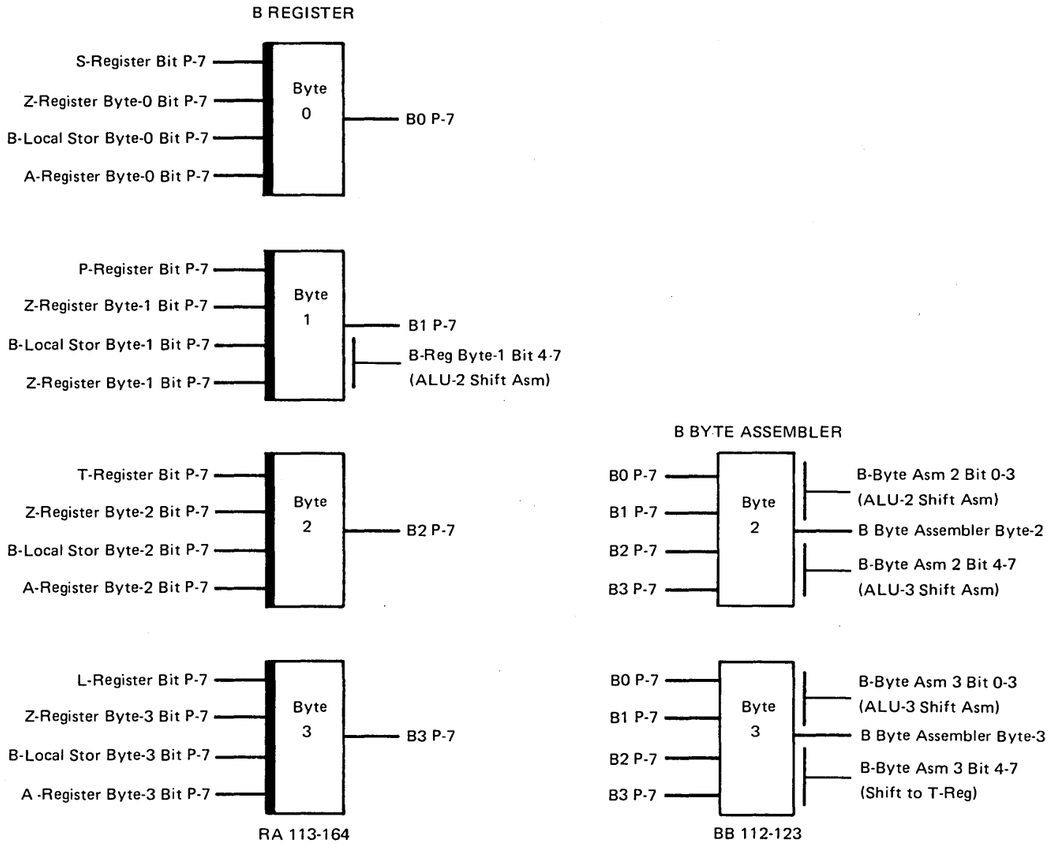
Parity is checked in all four bytes of the B-register during entry. An error condition is reported in MCKA byte 2 bit 5.

The B-register is a fullword (four-byte) register that normally enters a fullword.

The B byte-assembler provides the means of presenting the B-register bytes to the ALU system.

The B byte-assembler can gate any byte of the B-register into both byte 2 and byte 3 outputs to feed the respective ALUs.

The B byte-assembler has a parity check on both outputs, and an error condition is reported as a B-register error.



ALU A ENTRY GATING

The logic requirements of the system make it necessary to be able to block or transpose (cross) a part or all of the A entry byte operand.

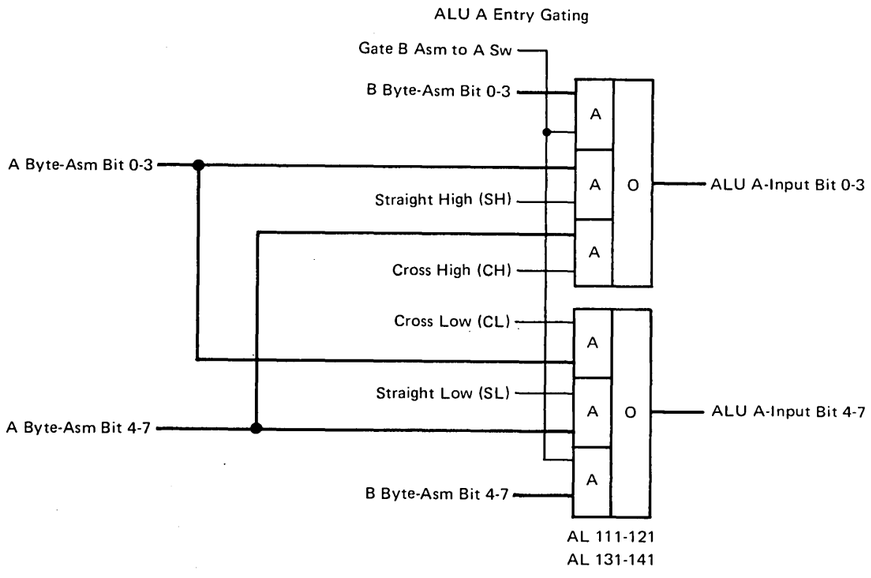
The high (0-3) and low (4-7) portions of the operand are separately gated to allow moving a portion straight to the ALU or to cross high and low in entry to the ALU.

When none of the ALU entry gates are activated, the entry is blocked and the operand is presented to the ALU as zeros.

For a normal entry, both straight gates are raised and the full byte is entered.

For a crossed or transposed entry, both cross gates are raised.

For operations requiring only one portion of the operand, the appropriate gate is raised to enter the desired portion into the specified position of the ALU with zeros in the ungated portion.



Op Sym	Operation	Entry HHHH	Byte LLLL	Gate SH	CH	CL	SL
BS	Straight	HHHH	LLLL	x			x
BS0	Block High and Low	0000	0000				
BSH	Gate High; Block Low	HHHH	0000	x			
BSL	Gate Low; Block High	0000	LLLL				x
BSX	Cross High and Low	LLLL	HHHH		x	x	
BSXH	Cross; Gate High; Block Low	LLLL	0000		x		
BSXL	Cross; Gate Low; Block High	0000	HHHH			x	

ALU B ENTRY GATING

The logic requirements of the system make it necessary to be able to block all or part of the B entry byte operand and to enter the shift and K factors.

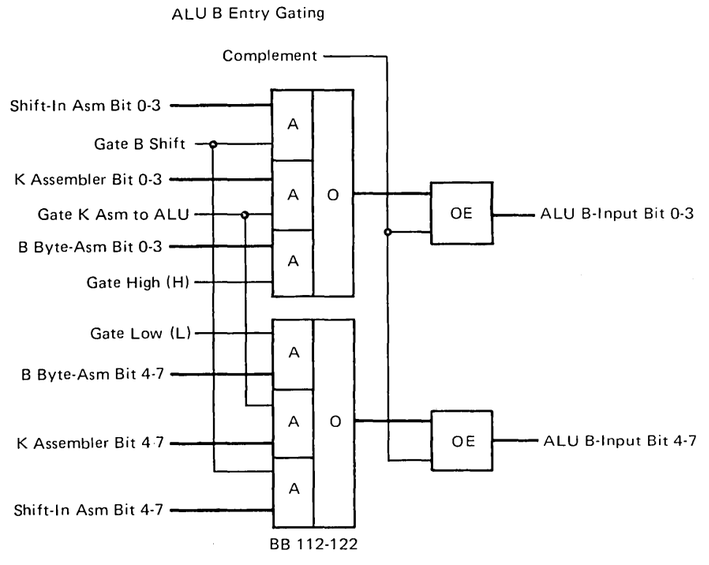
The high (0-3) and low (4-7) portions of the operand are separately gated to the ALU entry in order that one or both portions may be blocked and entered as zeros for the operand.

For a normal entry, both of the gates are raised and the data is transferred straight to the ALU.

When the normal entry is blocked, the byte developed in either the shift assembler or the K assembler can be gated.

The ALU B entry has a true/complement gating that reverses the binary bit levels of the operand byte when the complement line is raised.

The complement line is under control of the minus operation sign or the presence of the S0 bit when the operation sign is ± in the control word.



Op Sym	Operation	Entry HHHH	Byte LLLL	Gate H	L
BS	Straight	HHHH	LLLL	x	x
BS0	Block High and Low	0000	0000		
BSH	Gate High; Block Low	HHHH	0000	x	
BSL	Gate Low; Block High	0000	LLLL		x

SHIFT GATING

Special ALU entry gating is provided to allow the two-cycle right shift (four bits) operation.

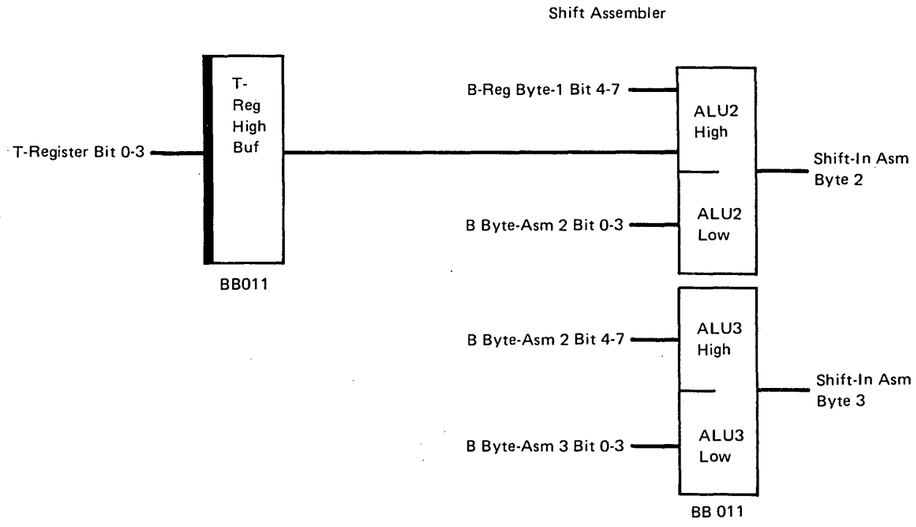
During both cycles, the low-order four bits of byte 2 and the high-order four bits of byte 3 are assembled as a byte for the B entry of ALU-3.

During the first cycle, the high-order four bits of the T-register are set into buffer latches in the shift assembler, and the low-order four bits of byte 3 enter the T-register.

During the first cycle, the low-order four bits (4-7) of byte 1 and the high-order four bits (0-3) of byte 2 are assembled as a byte for the B entry of ALU-2.

During the second cycle, the buffered T-register bits and the high-order four bits (0-3) of byte 2 are assembled as a byte for the B entry of ALU-2.

The A entries for both ALUs are blocked and enter zeros for both cycles.



ALU K ASSEMBLER

The K assembler for the ALU gates fixed constant and constants defined by the control word to the B entries of the ALUs.

Byte 2 of the control word in the C-register can be gated directly as a full byte to the ALUs.

The low-order four bits (4-7) of byte 2 in the C-register can be gated to the low-order (4-7) of the K byte with zeros in the high-order (0-3).

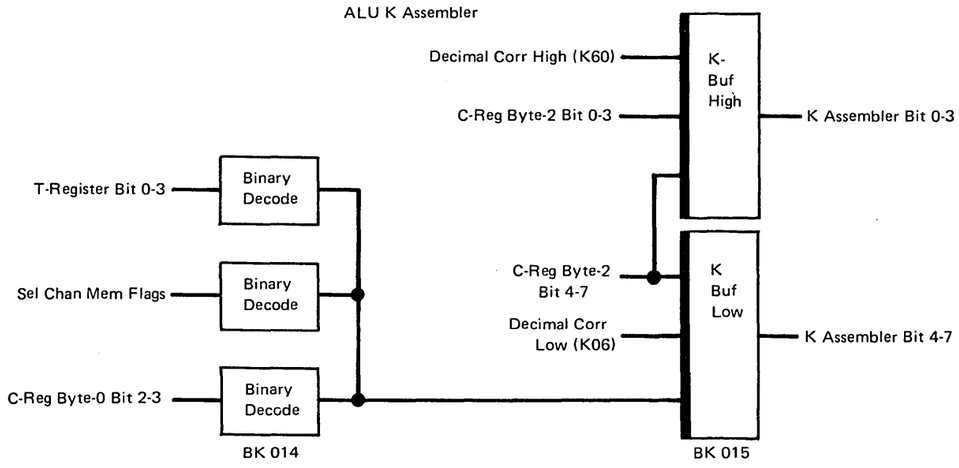
The low-order four bits (4-7) of byte 2 in the C-register can be gated to the high-order (0-3) of the K byte with zeros in the low-order (4-7).

The high-order four bits (0-3) of the T-register that define the bytes to be read or stored are converted to a binary count byte for adjusting the address and the count.

The four memory flag bits from the selector-channel buffers are converted to a binary count byte for adjusting the address and count during share cycles.

When C-register byte-0 bits 2 and 3 are used to define the size of the data transfer in a storage word, these bits are converted to a binary count.

For a decimal operation requiring binary to decimal adjustment, a K66, K60, or K06 value is developed during the second pass.



ARITHMETIC AND LOGIC UNIT (ALU)

The arithmetic and logic unit (ALU) performs the logic manipulation and adding operations in the CPU.

Two ALU units are provided to allow halfword binary and word-move operations in one pass.

Two ALU passes can be made during a CPU cycle to complete a fullword binary or word-move operation.

Each ALU consists of the A and B entry gates, the logic and arithmetic circuits, and output gating to position the output byte in the Z-register.

ALU-3 entry lines are checked for invalid decimal digit to ensure a correct decimal output with errors reported in S1 bit.

A carry look-ahead circuit is shared by the ALUs to allow simultaneous arithmetic processing of two successive digits or bytes.

During logical operations, both ALUs are fed with the same data, and a logical check circuit compares the results and reports errors through MCKA byte 2 bit 2.

The ALU logic circuits develop four outputs for any input presentation, and gates defined by the control-word operation select the appropriate output to set the Z-register.

Three of the ALU circuits provide the logic AND, OR, and OE outputs; the fourth combines these with the carry inputs to develop a full sum output used for decimal arithmetic.

The A-logic function is performed by raising the complement line in the B entry and gating the AND output.

A parity prediction (generator) circuit on each ALU develops an output parity based on the inputs and the operation and is proven by the parity check of the Z register.

In word operations the ALU-2 output is set into Z0 and Z2 and ALU-3 output is set into Z1 and Z3 on the first pass with an update of Z0 and Z1 on the second pass.

For byte operations ALU-3 output is set into all Z-register bytes to allow the SDBO byte assembler to enter the byte into any position.

Half-Sum Checking

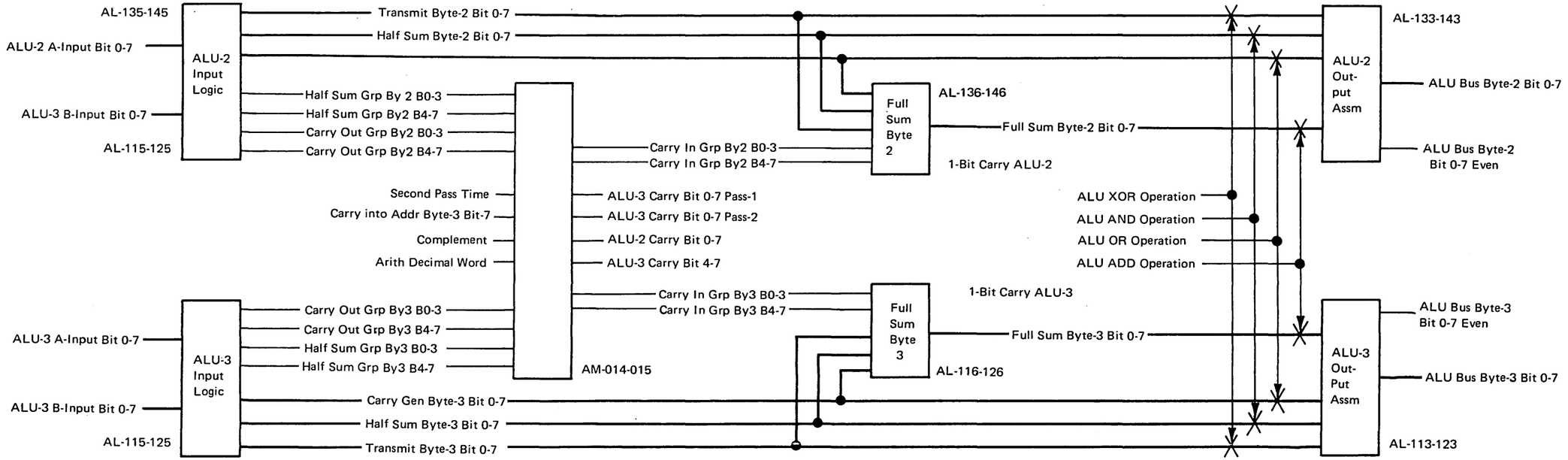
The half-sum lines (OE) developed from the two ALU operands are tested against the parity bits of the two operands to check the parity of the entry data.

Each half-sum line indicates the odd/even relation of a bit position within the operands.

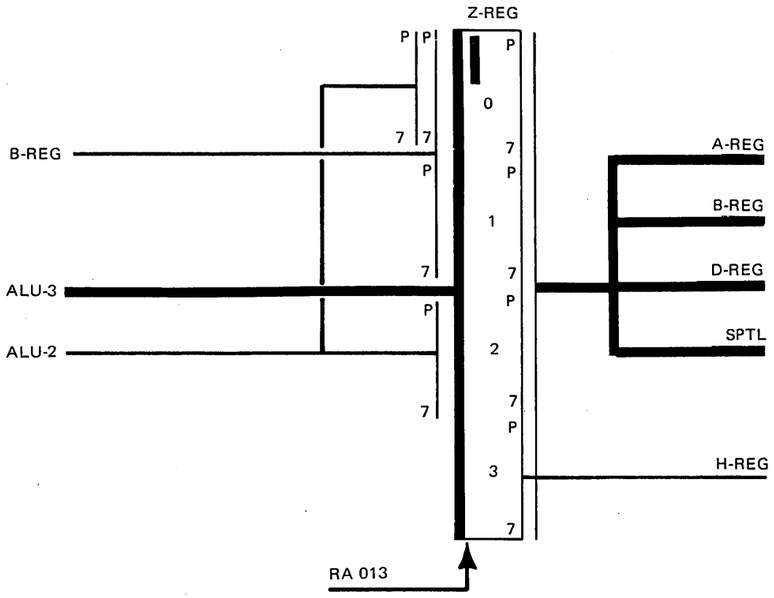
Developing an odd/even count from the eight half-sum lines indicates the level of the entry data bits (16) that should check with the level of the entry parity bits.

A separate latch is set for each pass and each ALU, but the ALU-2 indications are blocked for decimal operations.

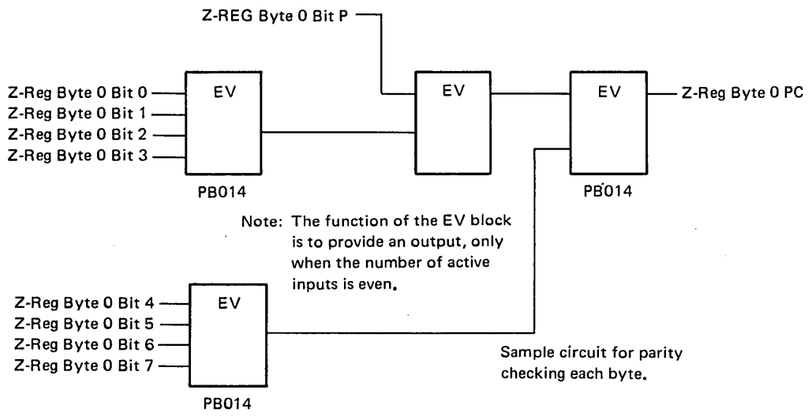
A detected error sets an ALU check indicator in MCKA byte 2 (bit 0 for ALU-2; bit 1 for ALU-3).



Z-REGISTER and D-REGISTER



Z-REGISTER PARITY CHECKING

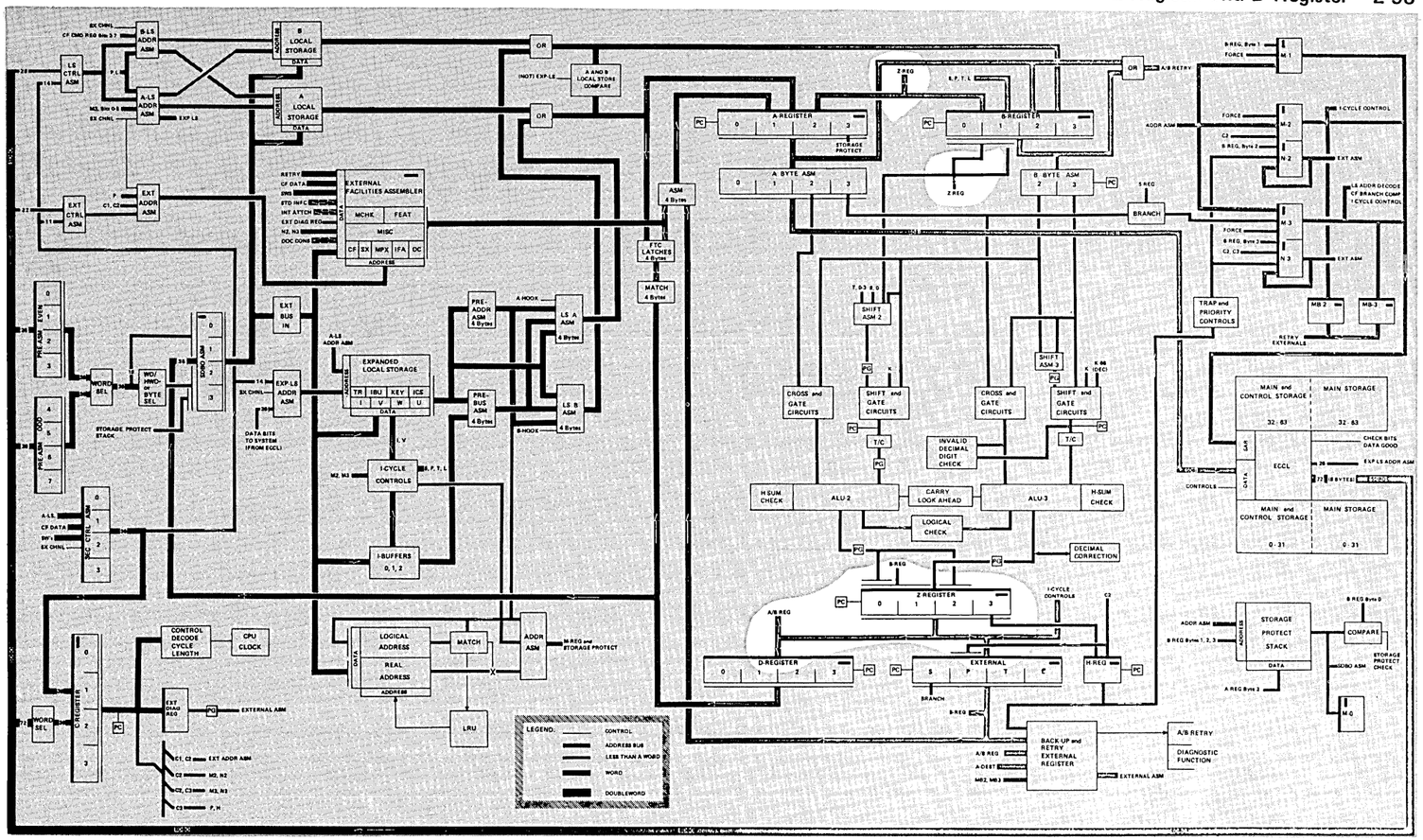


Z-REGISTER

The ALU results are set into the four-byte Z-register. The ALU result data can then be routed to:

- The S, P, T, or L-register
- The A- and B-registers
- The D-register

Z-register data (ALU result) is tested, if so specified in the control word being executed, to set or reset S-register bits.



The direct paths from the Z-register to the S, P, T, and L-registers permits the setting of these externals earlier than other external facilities. This capability is necessary because these registers are frequently used by the next control word for status information and for local-storage and external-register addresses. Data destined to all other facilities is set into the D-register and destined during the next control-word cycle.

ALU data set into an external register (other than S, P, T, or L) in one control-word cycle is not available as source data for the next control-word operation.

One-byte Ops feed all four bytes of the Z-Reg with the same data.

A path is provided from the Z-register to the A- and B-registers. This path is to allow ALU data destined to local storage (not externals) by one control word to be used as source data by the next control word.

Example:

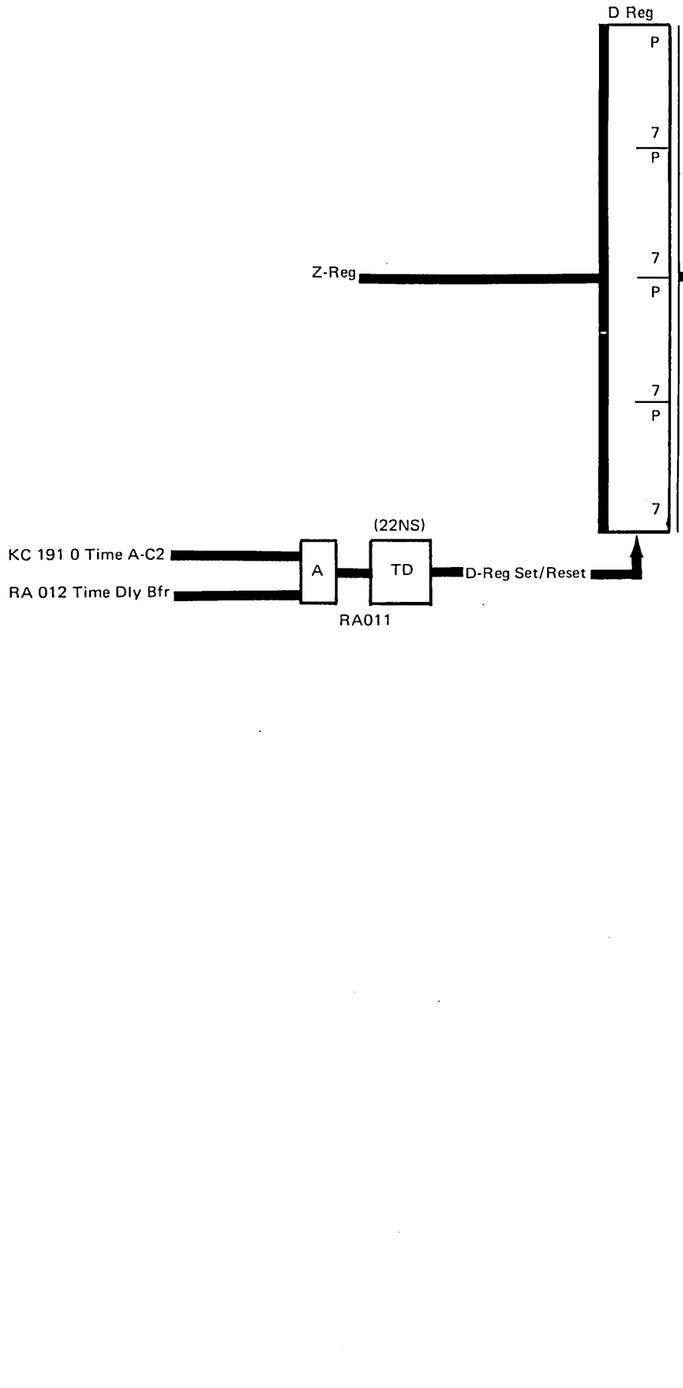
A control word is executed, and the result is gated to the R-register.

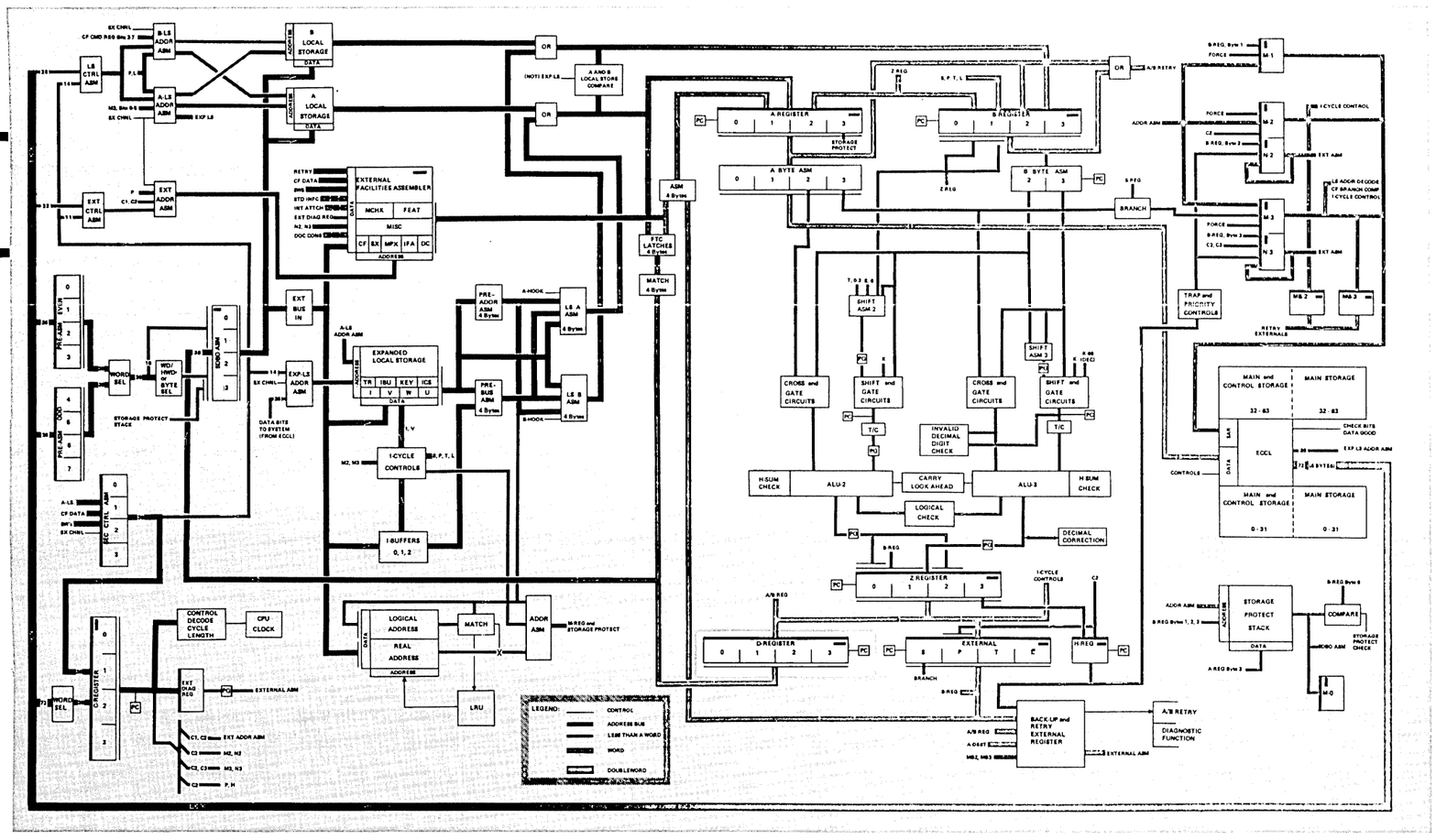
The next control word requires the R-register as the B-source.

The Z-register to B-register gate is activated because the result of the previous control word has not been set into the R-register. The Z-register is also gated to the D-register in normal fashion.

Refer to Local Storage Destination-Look-Ahead in Chapter 2.

Any combination of byte selection is possible. For example: if only byte 2 of a local-storage location is altered, then only that byte is routed to the A- and B-registers from the Z-register. Bytes 0, 1, and 3 come from the local-storage location addressed by the control word being executed. Consequently, no matter which bytes of a local-storage location are altered and destined by a control-word operation, the altered data is available as source data for the next control-word operation. Address information derived from the control word being executed determines whether Z-register data is routed to the A-register, the B-register, or both.

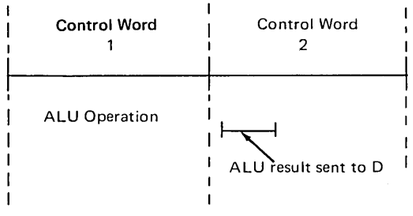




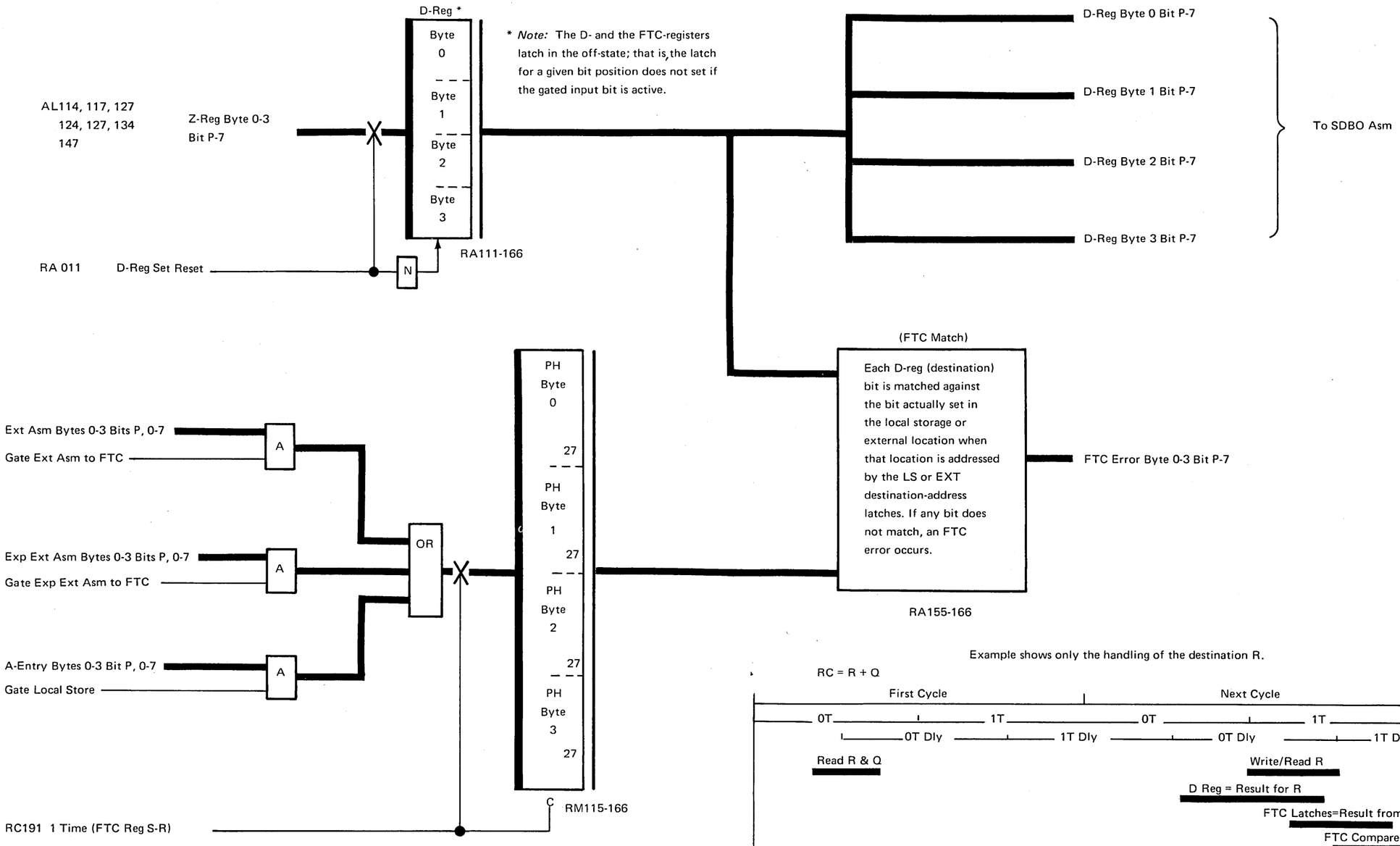
D-REGISTER

The four-byte D (destination) register is sent to the ALU result data (from Z) early in the next control-word cycle.

The D-register data is sent to the SDBO assembler, where it can be routed to externals (via external bus in) or to local storage.



D-REGISTER and FLUSH-THROUGH-CHECK (FTC) REGISTER



Example shows only the handling of the destination R.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

CONTROL (C) REGISTER and CPU CLOCK

C-REGISTER

The C-register contains 32 bits plus four parity bits. The purpose of the C-register is to decode the control word and provide control and gating of CPU functions. Once the control word has been read out of control storage and gated to the C-register, it is decoded to determine:

- Word type
- CPU function
- CPU clock cycle and length

The functions of the bytes of the C-register are:

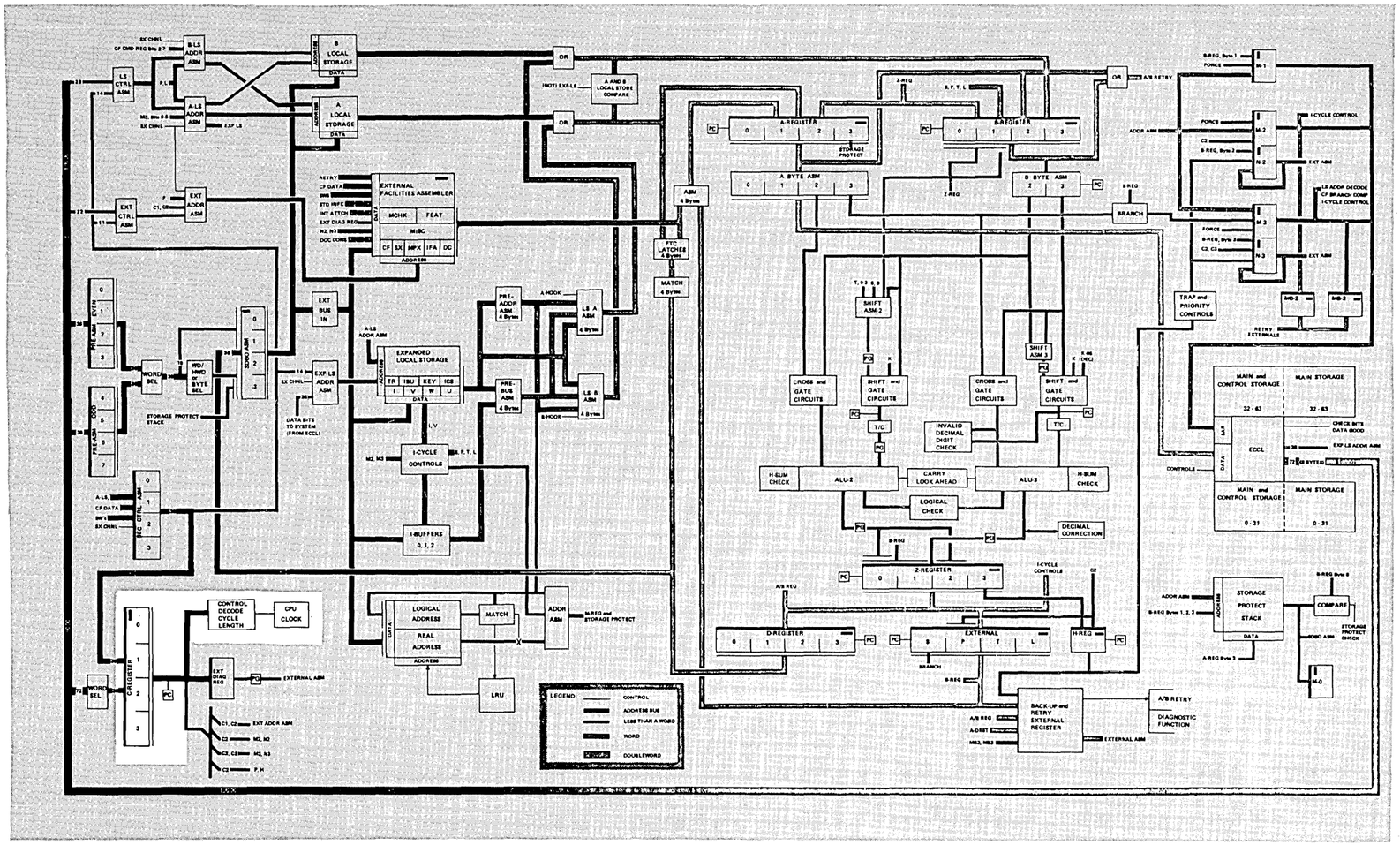
C0 Define word type and format Branch High Address (M3 B4)

C1 Specify A source or destination (an external register or an A-local-storage address). Stat sets and/or special functions.

C2 Specify a B-local-storage source or destination. Contains Mask or K values special Stat sets.

C3 Specify next control-word address Branch Low Address (M3 B5).

The C-register is set at 0 time of each control cycle.



CPU CLOCK AND TIMING

A. Oscillator

Timing for the 3145 is developed from a 22.222 MHz crystal-controlled oscillator. This oscillator is fed to an oscillator control card, where it goes through one frequency divider stage and is controlled for distribution to each board clock. From this, each board clock then develops six basic timing signals to time the CPU circuitry.

B. CPU Clock

This clock is a variable-cycle clock that is designed to operate 180-, 225-, and 270-nanosecond cycles, with each cycle having the further capability to extend by 22.5-nanosecond increments (referred to as pauses). This then allows (with one pause) 202.5 ns from a 180 ns cycle; 247.5 ns from a 225 ns cycle; and 292.5 ns from a 270 ns cycle. With two pauses, a cycle may be extended 45 ns as in the case of the Storage 1 Cycle Write. (270 becomes 315).

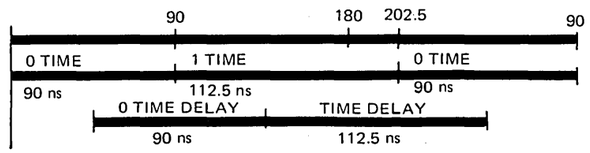
The CPU control word decodes determine cycle length. They provide 180-, 225-, and 270-cycle control signals that determine the cycle with which the clock should operate and the number of pauses the cycle should contain.

The CPU clock runs for one cycle under control of the clock start latch. The clock start latch has many input controls (start switch, set IC, CF clock start, etc.).

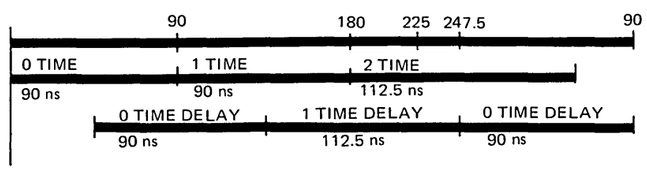
The clock consists of six latches operated in an overlapped configuration to produce six timing pulses.

180-Cycle (202.5)	0 Time 0 Time Delay 1 Time 1 Time Delay
225-Cycle (247.5)	0 Time 0 Time Delay 1 Time 1 Time Delay 2 Time
270-Cycle (292.5)	0 Time 0 Time Delay 1 Time 1 Time Delay 2 Time 2 Time Delay

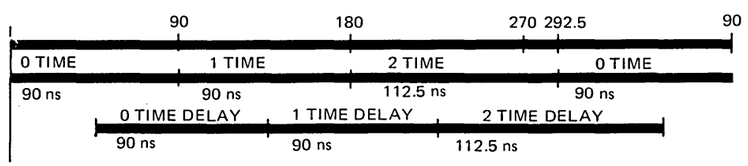
180-CYCLE 202.5 ns



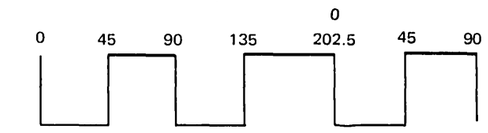
225-CYCLE 247.5 ns



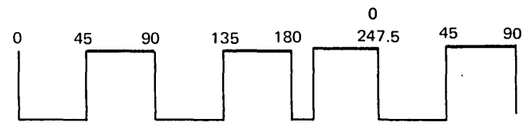
270-CYCLE 292.5 ns



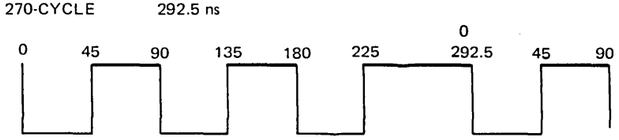
180-CYCLE 202.5 ns



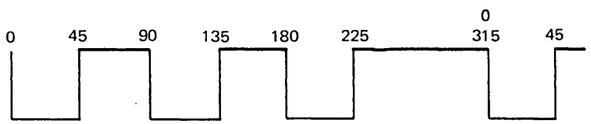
225-CYCLE 247.5 ns



270-CYCLE 292.5 ns



STORAGE 1 CYCLE WRITE 315.0 ns



CPU Clock Checks and Adjustments

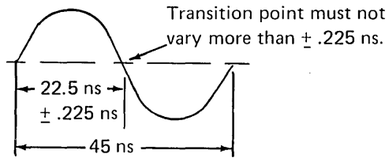
Equipment Required

Tektronix* type 454 oscilloscope or equivalent. Oscilloscope probes of equal length and equal attenuation.

NOTE: All measurements should be made with the displays centered on the scope face.

Oscillator

Oscillator frequency is 22.222 MHz ± 0.05%. The symmetry of the output sine wave must be within a 1% tolerance. Symmetry is checked at 01AB2L2S05. If the transition point varies more than shown, replace the oscillator card.



Clock

The CPU clocks are initially synchronized at the factory and must be readjusted, only when additional boards are installed.

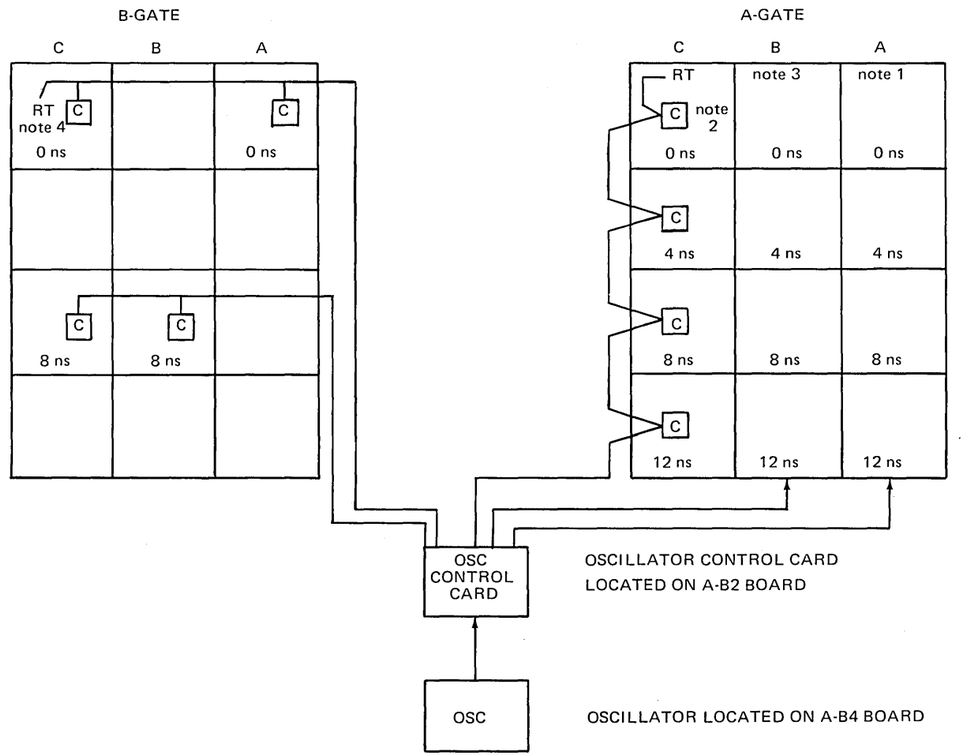
Each clock has a programmable delay-line adjustment. Pin G10 of each clock card (6735) is the oscillator test point. To ensure oscillator synchronization during the following adjustment, sync negative on the G10 test point.

1. To determine 'late clock' sync:
   a. Set rate switch to SINGLE CYCLE HARD STOP (CLOCK STOP indicator on).
   b. Ensure that zero delay is plugged in the clock cards in boards A-A1, A-B1, and A-C1.
   c. Sync channel 1 (minus) on clock card at A-C1G2G10 and display the signal.
   d. Using channel 2 to display the G10 pins of the clock cards in A-A1 and A-B1 boards, determine the latest (in time) of the three clocks.
   e. Place the channel 1 probe on the latest of the three clocks. Channel 1 is now synchronized on 'late clock'.

   Note: By swapping the input signals at the oscilloscope, verify that the oscilloscope is calibrated. The relationship between the two signals must not change. If signal relationship does change, use another oscilloscope before continuing with this adjustment.
2. With channel 1 sync and display on 'late clock', display all other clocks. If any clock is more than 1.0 ns earlier or later than 'late clock', change the programmable delay line for that clock to being it to within ± 1 ns of 'late clock' (negative pulse at G10).

The oscillator signal should now be synchronized for the system. If the oscillator test point is checked and found to be greater than ± 2.0 ns out of sync, resynchronize the clock.

*Trademark of Tektronix, Inc.



CPU Clock Locations

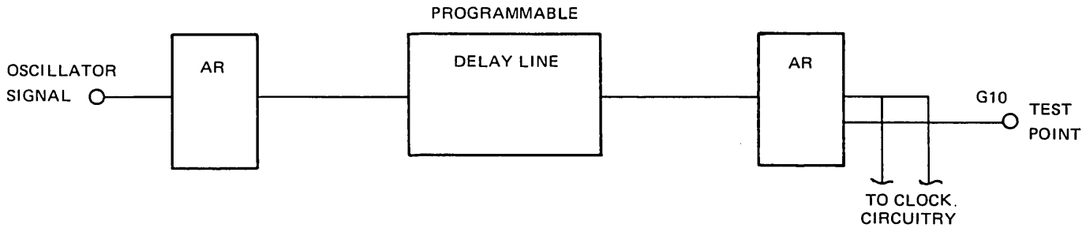
Gate A-	C1G2	C3G4	B1C4 (note 3)	B3H4	A1K2	A3C4	
	C2J2	C4E2	B2M2	B4K2	A2C4	A4Q4	
Gate B-	A1C4	B1	C1M2 (note 4)	B3V3	C3J2		

Oscillator Location:

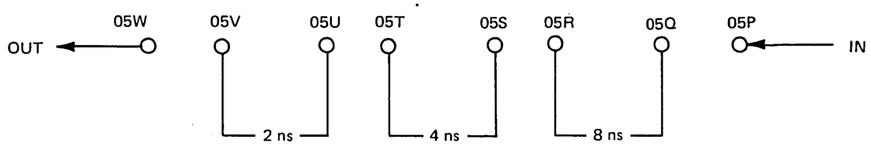
Oscillator and oscillator control card location: A-B4A3
A-B2C2

Notes:
1. These numbers are initial programmable delay settings.
2. Separate oscillator signal for select.
3. For IFA version 003 machine, clock is located in B1B4 socket on the A gate.
4. Clock-card position is feature-sensitive. Refer to the KC ALD pages.

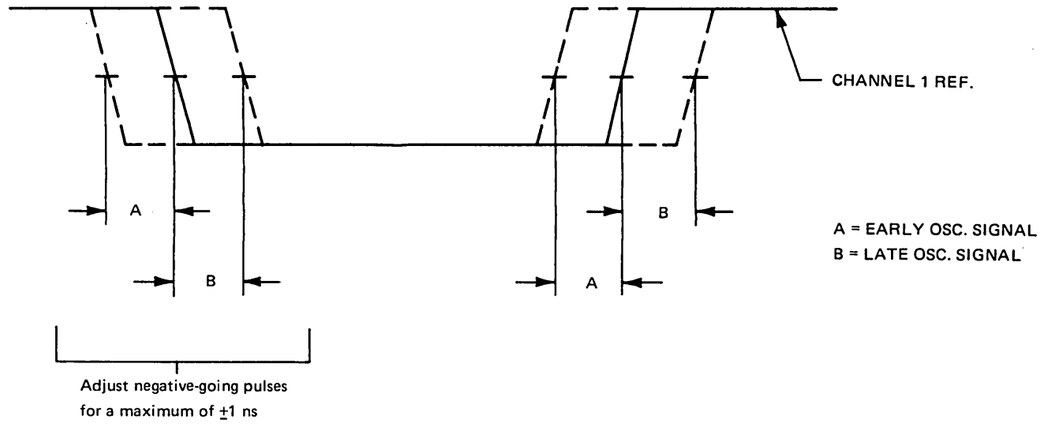
Circuitry Located on Each Clock Card and Clock-Start Control Card



PROGRAMMABLE-DELAY LINE LAYOUT



CLOCK CARD (6735)

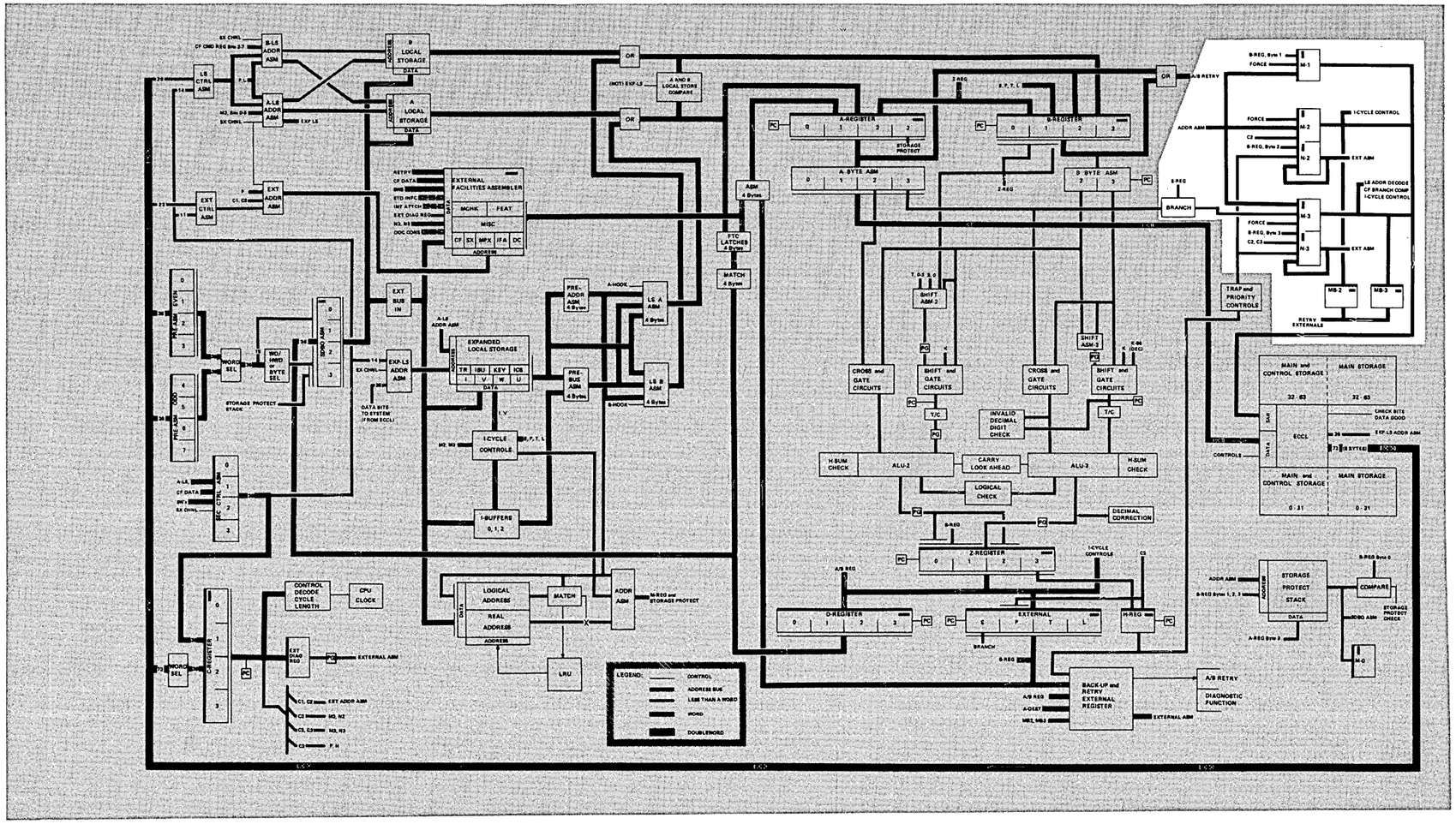


M, N, and MB REGISTERS

M-REGISTER

- Addresses main and control storage.
- Feeds storage address buses (SAB).
- Made up of M1, M2, and M3, which provide a 20-bit (plus 3 parity bits) storage address.
- M1, M2, and M3 address both main and control storage. Storage is read out on a doubleword boundary and stored on a word boundary.
- M3 bits 5, 6, and 7 and the storage word being executed provide the following selections for storage-word operations.

Read	Store
Odd/Even word, Halfword, or byte	Word, Halfword, or odd/even byte. Note: Information may be stored under mask (any byte selected).



Setting M-Register for Main-Storage Addressing

- M1, M2, and M3 are set from the ADR/ADJ circuits or B-Reg bytes 1 (bits 4-7), 2, and 3.
- M1, M2 may be forced to zero for direct main-storage addressing.

Setting M-register for Control-Storage Addressing

- M1 is set to zero for display purposes. A line address CTRL store is sent to the ECC board.
- M2 selects the control-storage module and is set from C2, ADR/ADJ circuits, trap circuits, or N2 (no module-switch function). M2 may be forced to FF for direct control-storage addressing.
- M3 selects one of 64 words within the module selected by M2. M3 is set from C2 (K-adr STW), C3, trap circuits, or ADR/ADJ.

N-REGISTER

- Made up of N2 and N3.
- Backup register for control-storage addressing.
- N2 is set with the same information as M2 and is changed only when the control word being executed performs a module-switch function.
- N3 is set with the same information as M3.
- N is not changed when a trap occurs.
- When a trap occurs, the M-register is set to the trap address. The trap routine stores the contents of N (the N-Reg contains the next address that would have been used had the trap not occurred). At the end of the trap routine, M and N are restored to their original value so that the control-word sequence may continue as if there had not been any trap.
- N2 sets M2 for every control-storage word access except when a module switch occurs.

MB-REGISTER

Made up of MB2, MB3.

Set with the control-word address in M2, and M3 from M2 BFR and M3 BFR.

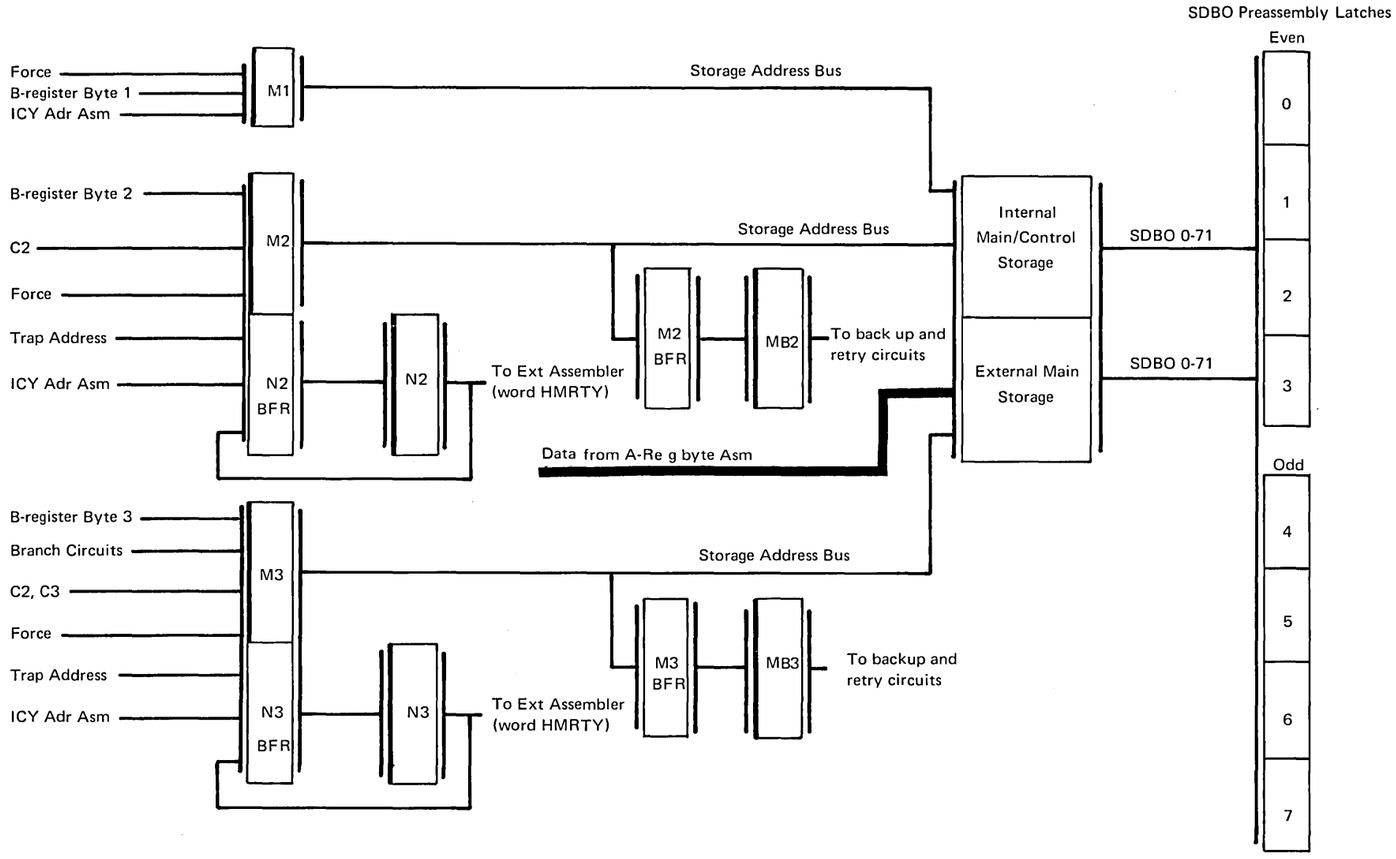
When the CPU clock is stopped, MB contains the address of the last word executed.

Its output is available to the retry and backup circuits as well as the external assembler (word RTY).

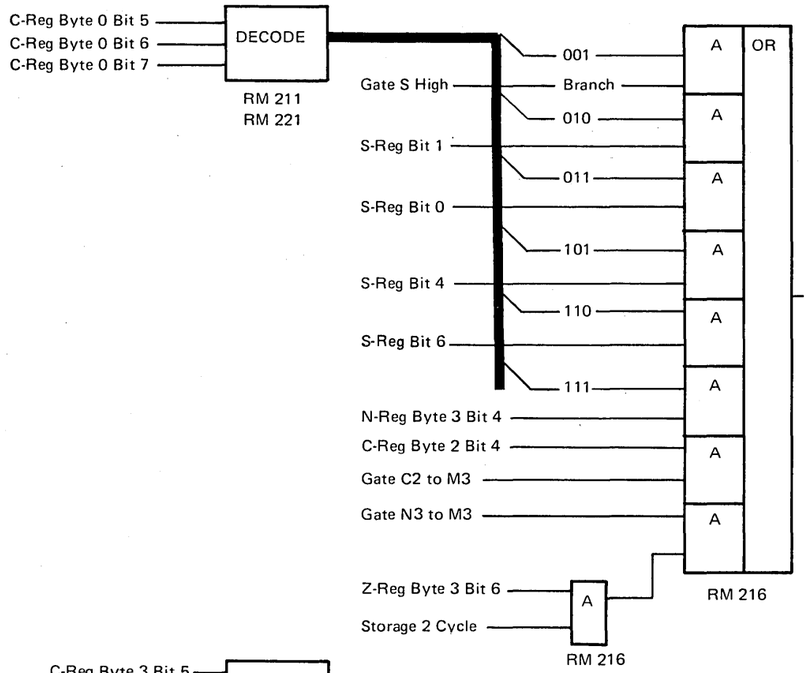
Buffer Registers

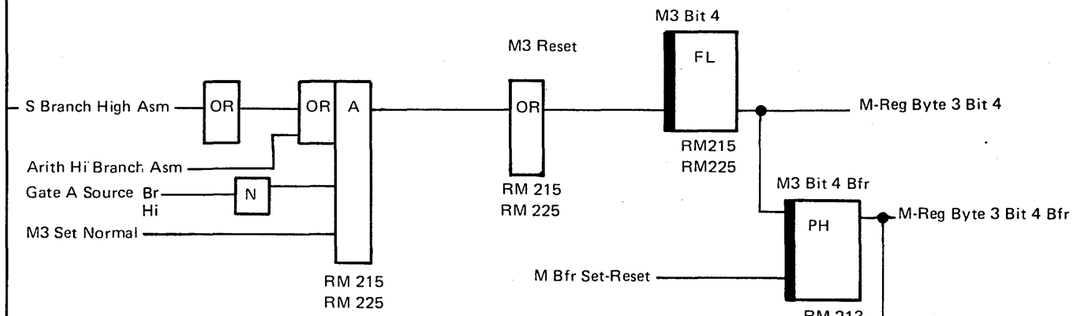
The M-buffer registers are an interim set of latches between the M-register and the MB-register. This allows cycle-to-cycle communications.

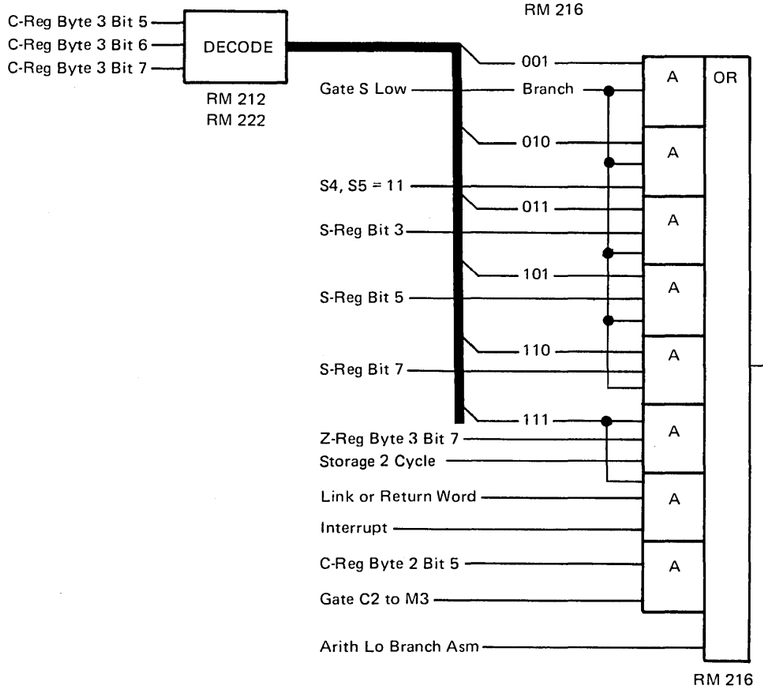
The N-buffer registers perform a similar function.



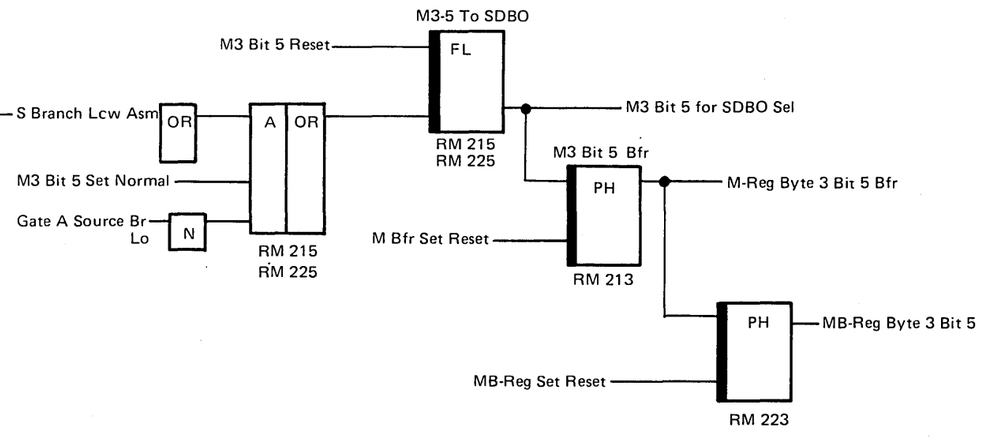
M-, N-, and MB-REGISTERS BRANCH CIRCUITS







M3 bits 4 and 5 are used for high and low branches respectively. M3 bits 6 and 7 are used to maintain parity.

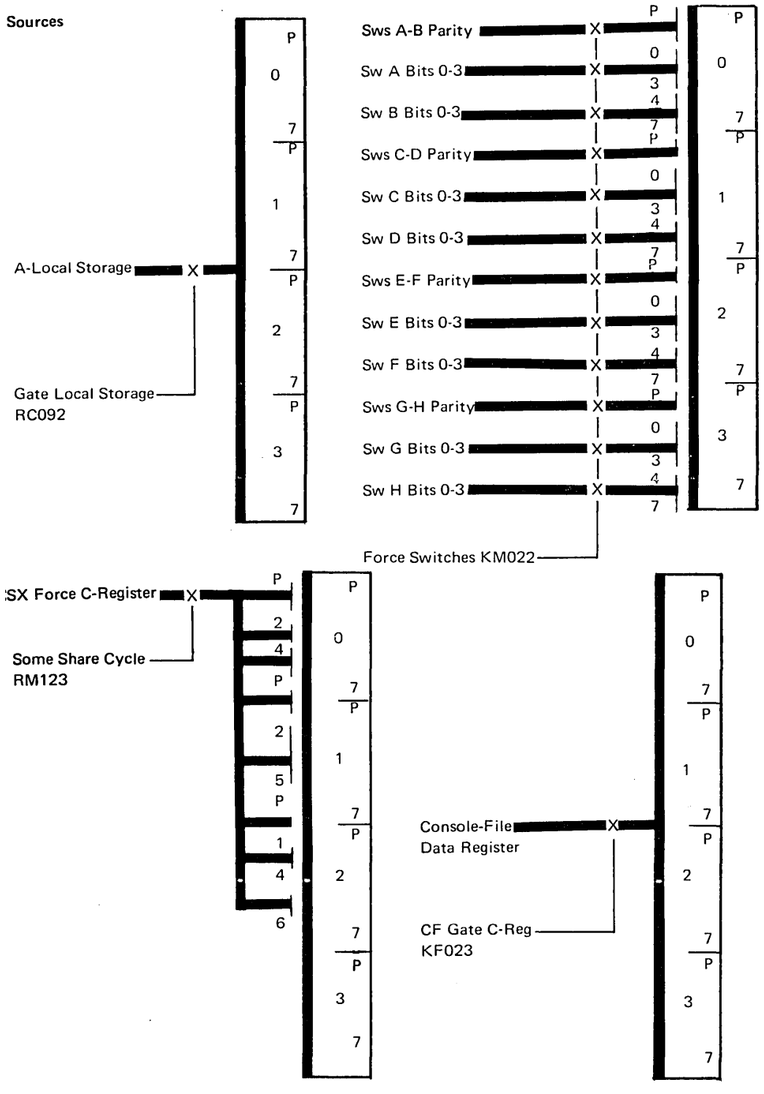


SECONDARY CONTROL ASSEMBLER

The Secondary Control Assembler provides a direct path for the microprogram and the console file to move data and addressing information.

From:	To:
Console-file data register	C-Register
A-Local storage	Local-storage control assembler
Console switches ABCDEFGH	Expanded local-storage address assembler
Selector-channel force C-register	External control assembler

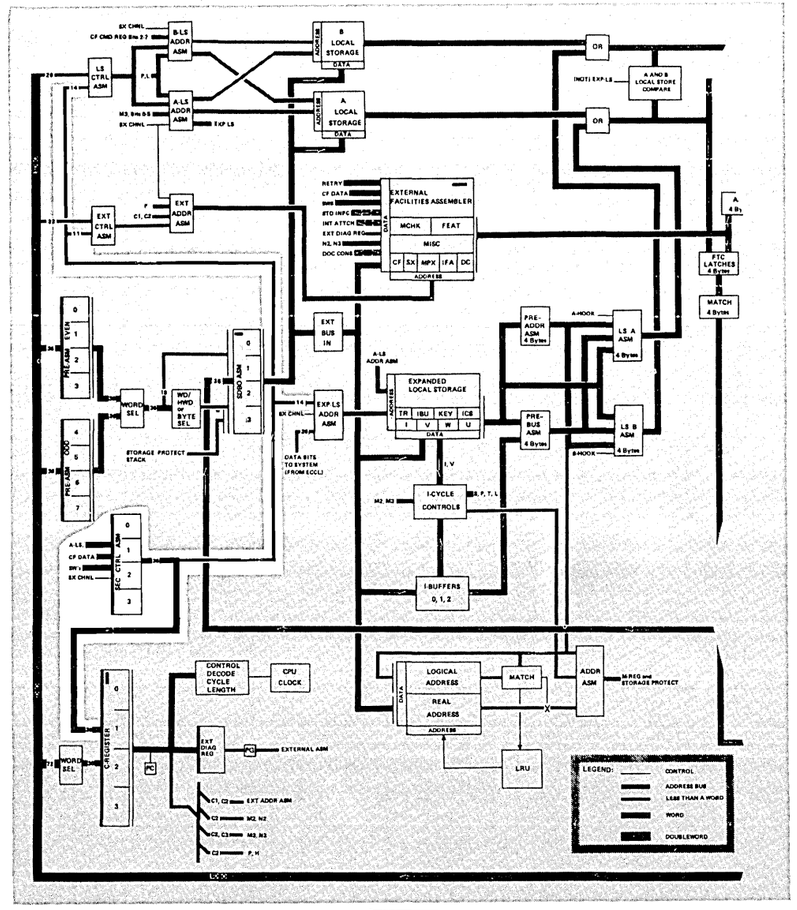
Sources



Note: In Local Storage Control Storage (LSCS) mode, diagnostic hardware forces A-Local Storage to act as control storage. Control words are read out of A-Local Storage, loaded into the C-Reg, and executed.

Refer to Chapter 13, Diagnostic Hardware.

Logic Pages	
Bytes 0, 1, 2, and 3.	
Bit	Page
P	RC112
0	RC122
1	RC132
2	RC132
3	RC152
4	RC162
5	RC172
6	RC182
7	RC192



Destinations

Note: Only the bits used by the microprogram are shown for each application of the Secondary Control Assembler.

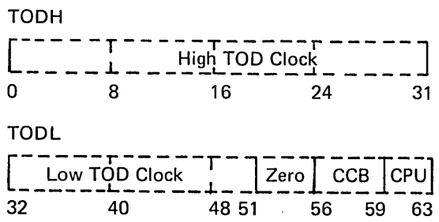
REMEMBER

There is a Reader's Comment Form at the back of this publication.

TIME OF DAY (TOD) CLOCK

The time-of-day clock provides a consistent measure of time suitable for elapsed time and time-of-day indications. The cycle of the clock is about 143 years when started from zero as an elapsed time measure. To provide a consistent time-of-day indication, the zero point must be defined to a calendar date. IBM programming systems have established this date as January 1, 1960, 0 AM Greenwich Mean Time.

Setting the TOD clock on the basis of a synchronization signal given by the operator introduces errors in the fractions of a second. This error is usually of small consequence in defining the time relating to human reaction. The error does not enter elapsed-time calculations because the difference between two time readouts does not consider the initial setting of the clock. For many TOD applications, only the high-order 32 bits need be considered. Position 31 of the counter is advanced every 1.048576 seconds. Operation in this mode still requires entering some value for the TODL destination, or the clock does not start.



CCB = Counter Control Bits
CPU = CPU Identification

The clock is a binary counter with a two-word format (64 bits) numbered 0 to 63 corresponding to the bit positions of a fixed-point number of double precision. Time is measured by incrementing position 51 of the counter every microsecond. Only the high-order 52 positions of the counter are used for this configuration. The remaining low-order positions are not used for time indication and are normally set to zeros except for three positions that define the status of the clock.

The program is not signaled of an overflow condition when the counter is advanced to the point of carry from either position 1 or position 0. At the point of carry out from position 0, the counter goes to zero and continues to count from that value.

The clock can be inspected by means of the instruction store-clock. The current value of the clock counter is stored in main storage. The clock can be set to a specific value by means of the instruction set-clock. The operand specified by the instruction replaces the current value in the clock counter. The set-clock instruction can be executed only when the clock security switch on the system control panel is set to enable changing the clock value.

The operation of the time-of-day clock is not affected or inhibited by any normal activity or event in the system other than turning off the CPU power. The clock runs when the CPU is in wait state, stopped state, or instruction step mode, and its operation is not affected by system reset or the IPL procedure.

The 3145 time-of day clock stops when the CPU power-off switch is operated. It is necessary to execute the set-clock instruction each time the system is started.

Physical Description

The time-of-day clock is driven by a 1 MHz oscillator feeding a binary-coupled trigger to produce a 1 MHz output. These circuits also develop a 75.46 KHz output that is used to drive the interval timer.

The clock counter functions as a binary-connected counter but is modified with a set of adder latches that allow holding the output of the basic latches until such time that the CPU sampling is complete. The counter can be loaded by destining the appropriate binary values to the TODL and TODH externals after executing a loading sequence. The counter starts to advance, immediately following the loading by gating the 1 MHz drive signal to the low-order position of the counter. The counter advance is checked by predicting parity and then comparing the predicted value with the parity generated from the counter. The counter output with parity bits for each byte is available to the CPU through the store clock instruction that causes externals TODL and TODH to be transferred to main storage.

The TODH and TODL externals may be called out as a destination at any time, but the contents of the clock are not changed unless TODL byte 3 bit 0 and the clock-run latch have been reset. The FTC (flush-through check) is blocked by the TODL byte 3 bit 0 to prevent error signals for this condition.

The control bits are associated with the clock readout to convey information to the user as to whether the clock value is a true measure of elapsed time since the last time the clock was set. These control bits are stored as the three high-order bits of byte 3 of the low-order word (TODL). Bit 0 indicates that the clock is running. Bit 1 indicates that the clock was set. Bit 2 indicates that an error occurred. During the processing of the store-clock instruction, the indicators control the condition code to be set.

Clock Security Switch (TOD CLK)

The clock security switch (TOD CLK) provides an interlock with the set-clock instruction as a means of guarding against inadvertent change of the clock value. The switch is spring-returned to the secure position. When the switch is in the enable-set position, execution of the set-clock instruction causes the clock to be set to the value of the designated operand. When the switch is in the secure position, execution of the set-clock instruction does not change the value of the clock. The switch does not have any other effect on the operation of the clock.

Clock Validity Indicator (TOD CLK INVAL)

The clock validity indicator (TOD CLK INVAL) is used to indicate when the time-of-day current clock value is not a true measure of the elapsed time since the last time the clock was set. The validity indicator is turned off when the set-clock instruction is executed with the TOD CLK switch in the enable-set position and no exceptions are encountered. The indicator is turned on whenever the clock misses a time increment or stops. This may result from a power failure or a malfunction in the clock circuits. When the indicator lights for an error condition, the machine-check indicator is set and an interrupt is requested. If the clock is started by the POR routine and used for elapsed-time indications, the indicator remains lighted because it is invalid as a TOD indication. In this case, the indicator being lighted does not mean that an error has occurred.

Error Detection

The time-of day clock checks its advance operation by a check on the progressive parity conditions. The counter value of each byte before the advance is fed into a parity predictor circuit to develop the updated parity bits. These parity bits are compared with parity bits generated from the updated value in the counter. Any failure within the counter advance circuits results in a difference between the predicted parity bits and the resultant parity bits. The difference signal sets the TOD clock check latch. The latch output sets the MCKA byte 3 bit 6 and resets the bit 2 and bit 1 latches as indicators. A class 3 machine-check interrupt is requested. The setting of the MCKA latch resets the TOD clock check latch. The reset of the bit 2 latch lights the TOD invalid indicator on the system control panel.

Clock-Setting Sequence

The clock is set to zero and started by the power-on-reset sequence. With this start the clock output can be used to indicate running time or elapsed time. Under this mode of operation, the TOD-Invalid indicator on the system control panel remains lighted because the clock output is not TOD.

To obtain time-of-day output, the set-clock instruction must be executed to start the clock at the current time. When the set-clock instruction is executed, the correct TOD is assumed and the TOD Invalid indicator on the system control panel is not lighted. If a clock error is detected during operation, the indicator is lighted to warn the operator.

The set-clock instruction must have the TOD CLK switch on the system control panel held in the enable-set position to allow execution. If the switch is not operated, the clock value and its operation are not affected. The first step in the clock-set sequence is to reset TODL byte 3 bit 0 along with the 'clock run' latch and the clock controls. With the 'clock run' latch reset, the TODH and TODL externals can be destined. The value destined for TODL must have byte 3 bit 0 set to 1. When both load latches have been set, the TODL byte 3 bit 0 line is developed, the 'clock run' latch is set, and the 'clock set' latch is set. At this point, the clock is set and starts to run with the next 1 MHz pulse.

The POR start differs only in that the TOD CLK switch does not need to be operated and the first TODL destined microstep is not required.

For microdiagnostic operation the enable-set output of the switch is forced, but the set sequence is the same as for the set-clock instruction.

TOD-Clock-Update Sequence

Because of the asynchronous operation of the TOD clock, the advance pulse from the 1 MHz oscillator cannot be used directly to advance the clock. The clock readout must not change during the A-register set/reset time. The CPU 90-135 time is gated with the 1 MHz oscillator to develop the advance pulse. If the CPU clock is stopped, the CPU oscillator provides the timing. The advance pulse sets the start-update latch whose output provides a series of dalayed outputs that control error sample and advance of the clock.

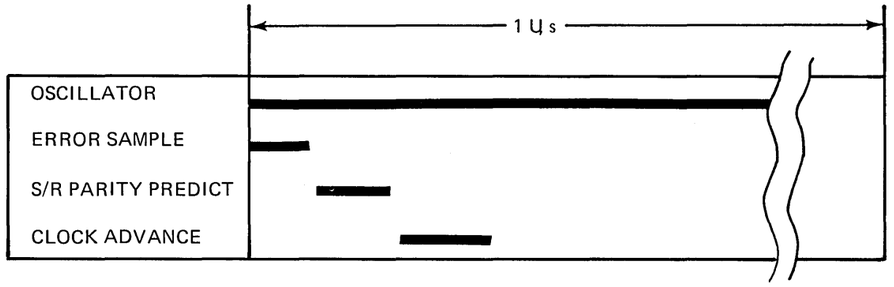
TOD Clock Instructions

The time-of-day clock has two instructions associated with its operation:

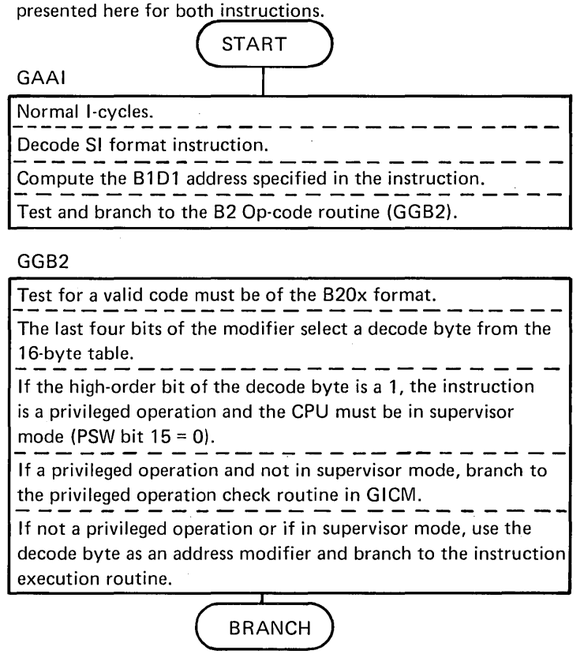
- Set clock used to set the initial time.
- Store Clock used to enter the current clock value into main storage.

Both instructions are SI format instructions modified so that byte 2 is an extension of the operation code (byte 1) instead of the immediate operand. Byte 2 is used to specify the exact function of the operation code.

Both instructions are decoded as operation code B2 in the GAAI routine. The operation branches to the GGB2 routine to decode and validate the modifier. Te st for privileged operation is made when required, and then the operation is branched to the routine of the function specified. The common data flow is presented here for both instructions.



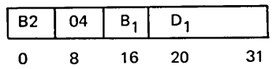
TOD CLOCK UPDATE SEQUENCE



Set Clock Instruction

Set Clock Instruction

SCK D_1 (B_1) [SI]



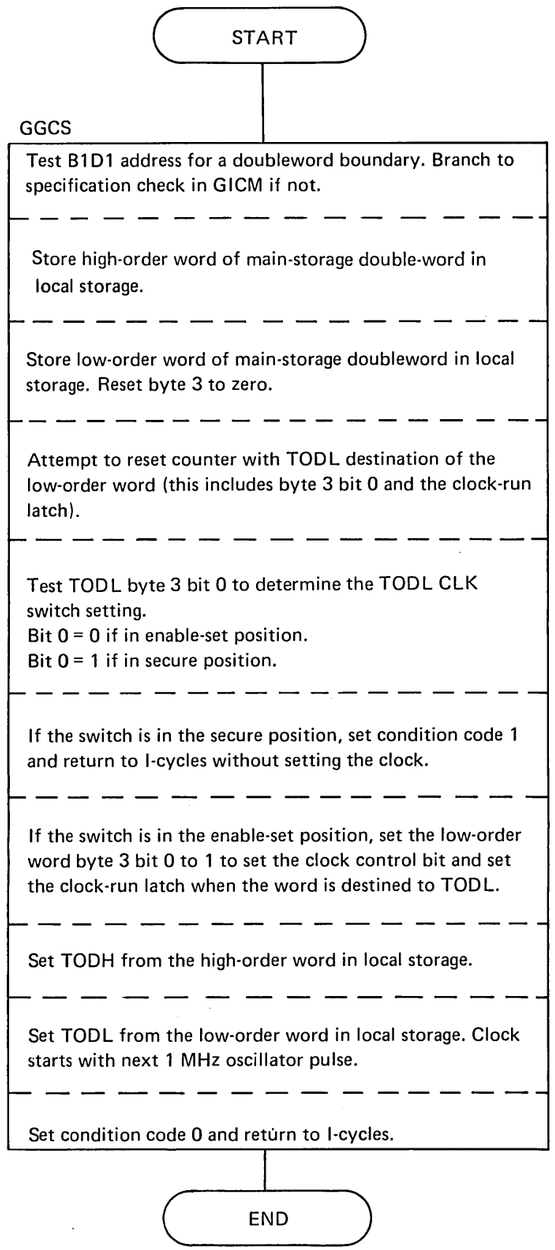
B2 = Operation Code
04 = Set Clock Function
B1D1 = Storage address of an eight-byte field. Must be on a doubleword boundary. Bits 52-63 of the field are ignored and are not used in the clock value.

The set-clock instruction is a privileged operation used to place a value into the time-of-day clock. The location of the value is specified by the B1D1 portion of the instruction. The implied length of the value is eight bytes (two words). The address must be located on a doubleword boundary. Only the high-order 52 bits of the doubleword are used to set the clock counter. The remaining bits (52 to 63) are ignored by the operation in setting the clock value.

The value in the time-of-day clock is replaced by the designated value if the set-clock instruction is executed while the TOD CLK switch is in the enable-set position. If the TOD CLK switch is in the secure position when the set-clock instruction is executed, the value in the clock is not replaced and the condition code is set to 1 to indicate.

Condition code settings:

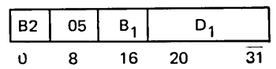
0 = Clock value set.
1 = Clock value secure (TOD CLK switch in secure position; therefore, clock value was not changed).
2 = Not used by the 3145.
3 = Not used by the 3145.



Store Clock Instruction

Store Clock Instruction

STCK D_1(B_1) [SI]



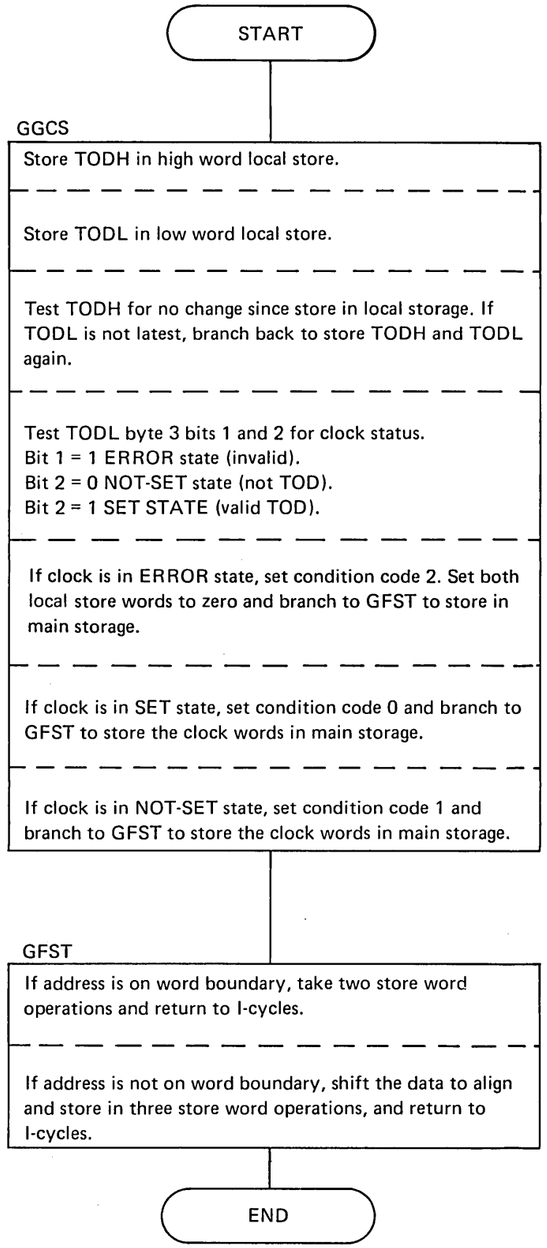
B2 = Operation code
05 = Store-clock function
B1D1 = Storage address of an eight-byte field. May be located on a byte boundary. Bits 52-63 of field are:

52-55 = Set to zero.
56-58 = TODL Ctrl Bits 0-2.
59-63 = CPU Identification bits.

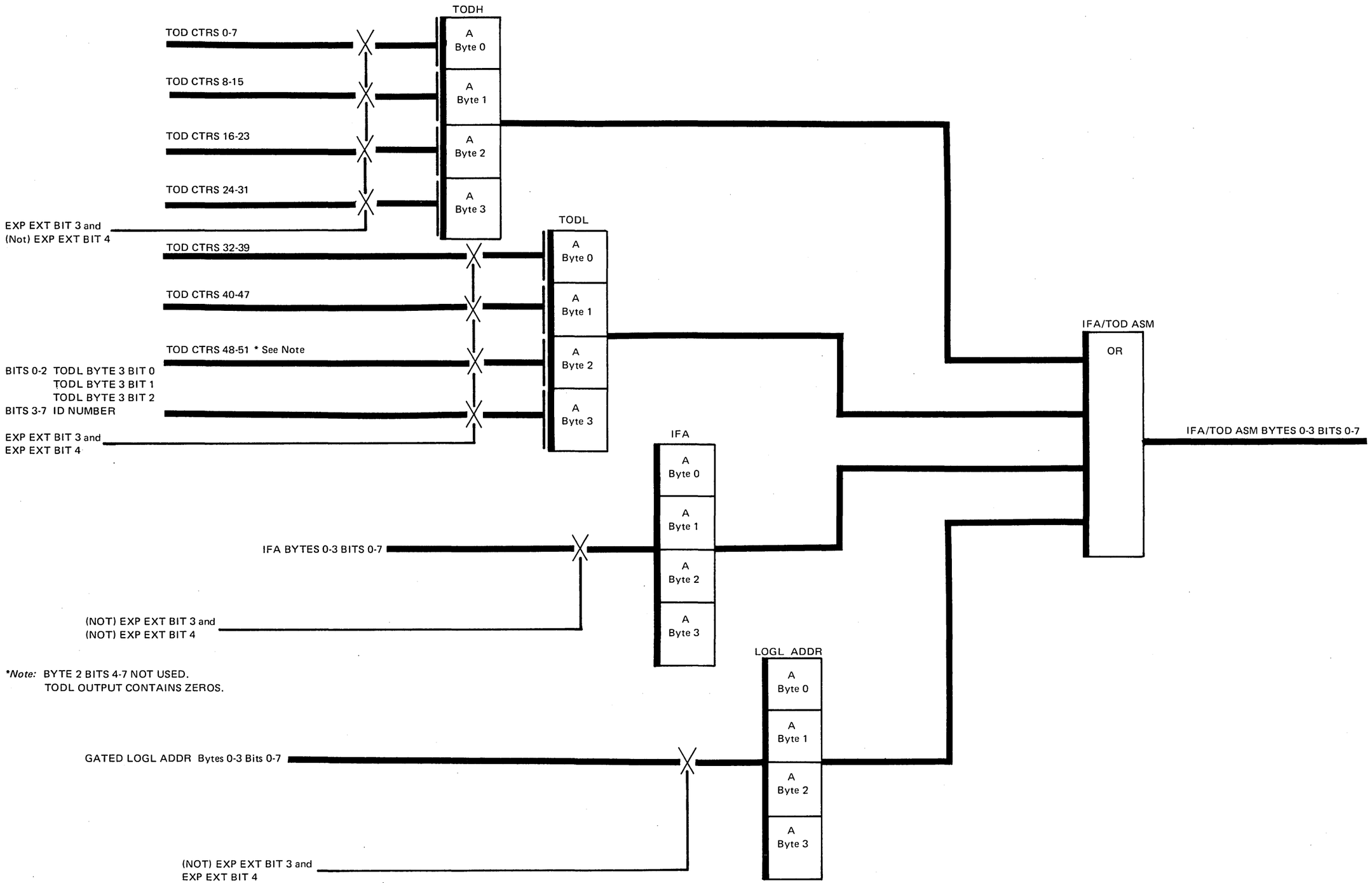
The store-clock instruction is used to place the current time-of-day clock value in the eight-byte field of main storage designated by the B1D1 portion of the instruction. The 52-bit clock value stores in the high-order of the doubleword assignment. The low-order byte stores the three clock control bits and the five-bit CPU pluggable identification. If the clock value is invalid, the doubleword is stored with all zeros.

Condition code settings:

0 = Clock in SET state.
1 = Clock in NOT-SET state.
2 = Clock in ERROR state.
3 = Not used by the 3145.



TOD Clock Output Assembler



Note: BYTE 2 BITS 4-7 NOT USED.
TODL OUTPUT CONTAINS ZEROS.

TOD Circuit Card Locations and Related Logic

CIRCUIT CARD LOCATION: A1L2

LOGIC/ALD PAGE:

CT011
- Osc Drive

CIRCUIT CARD LOCATION: A1N2

LOGIC/ALD PAGE:

CT111
- Time-of-Day Counter Advance Controls
- Interval Timer Osc Drive

CT112
- TOD Counter Set Controls
- Condition Codes
- Identification
- IF A/TOD ASM BYTE 2 BITS 2-3

CT113
- Error Detection

CT114
- TOD Ctrs 18-19
- TOD Ctrs 50-51
- TOD Asm Byte 2 Bits 2-3

CT115
- Lock-Load Timing Generation

CIRCUIT CARD LOCATION: A1P2

LOGIC/ALD PAGE:

CT211
- TOD Ctrs 0-1-2
- TOD Ctrs 32-33-34
- TOD Asm Byte 0 Bits 0-1-2

CT212
- TOD Ctrs 3-4-5
- TOD Ctrs 35-36-37

CT212 Continued
- TOD Asm Byte 0 Bits 3-4-5

CT213
- TOD Ctrs 6-7-8
- TOD Ctrs 38-39-40
- TOD Asm Byte 0 Bits 6-7
- TOD Asm Byte 1 Bits 0

CT214
- IFA/TOD Asm Byte 0 Bits 0 through 7
- IFA/TOD Asm Byte 1 Bit 0
- Gating Controls

CT215
- Parity
- Parity Predict
- Parity Asm for Byte 0

CT216
- TOD Termination

CIRCUIT CARD LOCATION: A1Q2

LOGIC/ALD PAGE:

CT221
- TOD Ctrs 9-10-17
- TOD Ctrs 41-42-49
- TOD Asm Byte 1 Bits 1-2
- TOD Asm Byte 2 Bit 1

CT222
- TOD Ctrs 11-12-13
- TOD Ctrs 43-44-45
- TOD Asm Byte 1 Bits 3-4-5

CT223
- TOD Ctrs 14-15-16
- TOD Ctrs 46-47-48
- TOD Asm Byte 1 Bits 6-7
- TOD Asm Byte 2 Bit 0

CT224
- IFA/TOD Asm Byte 1 Bits 1 thru 7
- IFA/TOD Asm Byte 2 Bits 0-1

CT225
- Parity
- Parity Predict
- Parity Asm for Byte 1

CT226
- TOD Termination

CIRCUIT CARD LOCATION: A1R2

LOGIC/ALD PAGE:

CT311
- TOD Ctrs 20-21-22
- IFA/TOD Asm Byte 2 Bits 4-5-6

CT312
- TOD Ctrs 23-24-25
- IFA/TOD Asm Byte 2 Bit7
- IFA/TOD Asm Byte 3 Bits 0-1

CT313
- TOD Ctrs 26-27-28
- IFA/TOD Asm Byte 3 Bits 2-3-4

CT314
- TOD Ctrs 29-30-31
- IFA/TOD Asm Byte 3 Bits 5-6-7

CT315
- Parity Asm for Bytes 2 and 3
- Gate Buffers

CT316
- Parity Predict for TOD Bytes 2 and 3

CT317
- TOD Asm Byte 3 Bits 3-4-5-6-7
- CPU Identification Number

INTERVAL TIMER

The Interval Timer provides program interruption on a program-controlled time basis. Uses of the interval timer include:

- Job accounting
- Monitoring for perpetual program loops
- Time stamping
- Polling at timed intervals.

The storage word at program-storage locations 80-83 (decimal) is reserved for the interval timer feature. Any value stored at this location is automatically reduced by decrementing, provided the interval-timer switch is in the MORM (normal) position.

The program in process can be automatically interrupted by an external interruption (if PSW system-mask bit 7 and control register bit 24 are on) when the interval-timer word goes from a positive value to a negative value. The interruption is identified by setting the appropriate PSW bit on.

Description

- Contained in hardware
- 3 Cards: A-B2K4 – Controls
  A-C1T2
  A-C1U2
  Timer
  Logic Pages CH031-CH 225
- Uses the same oscillator as the TOD clock (HR Timer 75.46 KHz oscillator)
- Fullword: a 32-position counter is used

Program storage locations 80-83 are actually a 32-position counter, located on three cards (along with the controls).

When the programmer sets a value into the interval timer, lo location 80 is address. (hex 50). (he gets the 32-byte counter instead of actual main-storage position 80)

The M-Reg address is ANDed with a store operation to set the desired value into the counter. [32-position (bits 0-3) and parity].

The value placed into the counter is the complement form of the output of the A-byte Asm (bytes 0-3). The bit condition is inverted from the A-byte Asm to the counter position.

The binary counter ripples through (referred to as decrement, but actually it adds) until a high-order position 0 carry-out occurs. This causes an EXT Timer Interrupt.

The value in the counter may be displayed manually by placing main-storage address 80 in the console switches, and displaying the contents of the external SDBO lines.

Interval Timer Switch

When the interval-timer switch is set to the NORM position, the value stored in the interval-timer word is automatically decremented immediately after being stored.

When the switch is in the DSBL position, no decrementing of the interval-timer word takes place. The four bytes may be used for normal program applications.

NORMAL

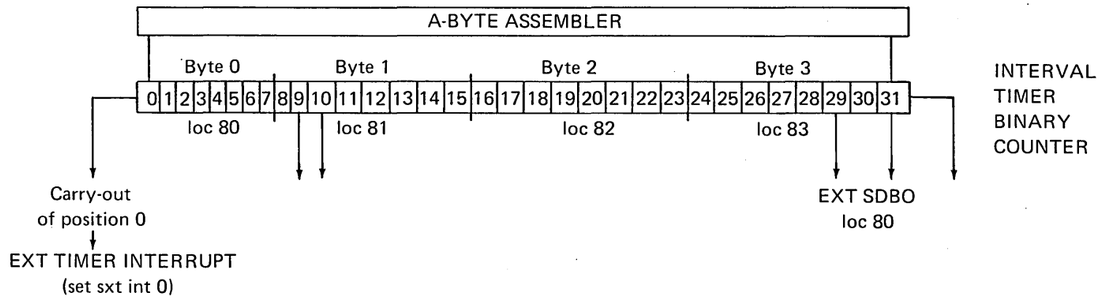
This position enables the timer.

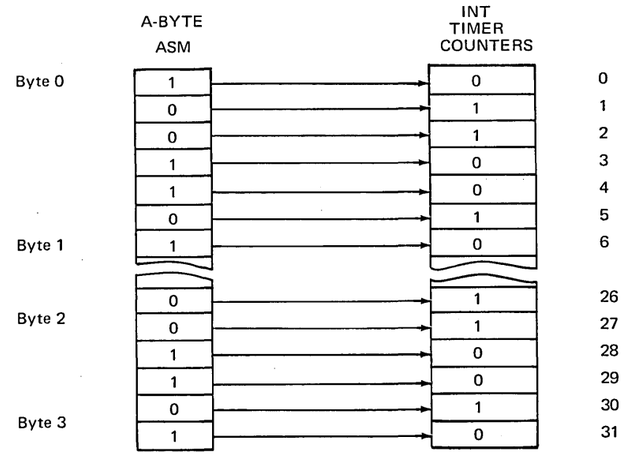
DISABLE

This position disables the interval timer. The content of the timer is not available to the data flow for timer functions.

Soft Machine-Check Interrupt

(INTERVAL TIMER DAMAGE) This interrupt occurs if PSW bit 7 and the external mask bit are on. It indicates damage to the timer. Programmed validation procedures and error logging are required.





Input lines to the binary counter (For a store timer operation)

OS DOS COMPATIBILITY

- Consists of the OS DOS emulator program and the hardware and microprogramming needed for execution.
- Two new instructions: Execute Local (EXL) and Adjust CCW String (ACCW), are used by the Emulator program.
- The DOS Emulator and the DOS system being emulated are located in main storage above the OS area.
- The minimum storage area needed for the DOS Emulator and the DOS system is 38K bytes.
- The OS DOS Emulator operates in the same manner as any OS job.

When operating in local mode (DOS programs being executed), all addresses pertaining to the DOS area are adjusted by the address-adjustment hardware.

When local mode is terminated, addressing is performed in the standard manner.

Refer to the OS DOS functional units description, and the Real Address Computation example for an explanation of the address-adjustment hardware.

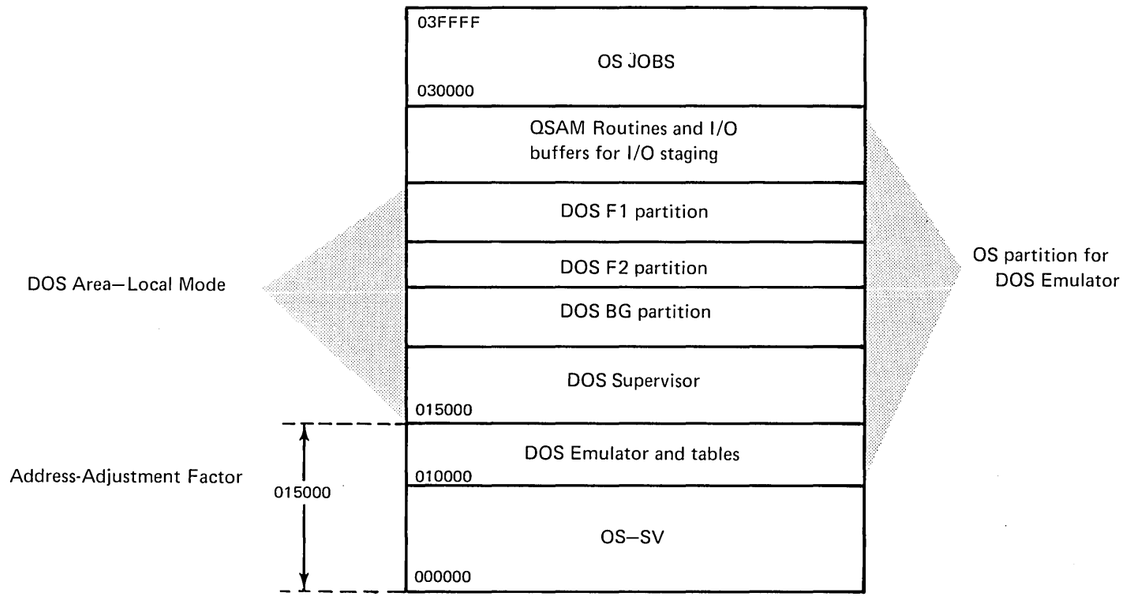
The reason that address translation is needed for DOS emulation is that the addresses of the DOS supervisor (SV) are basically fixed and the DOS supervisor is located in an area of storage not normally used by DOS SV. All references to the fixed addresses of the DOS SV must be adjusted to reflect the real location of these fixed areas.

The EXL instruction operates with the Local List (LEX List), which is a table that the emulator program loads before the execution of the DOS area. This list is located in the emulator area and is used to handle entry to and exit from the local mode of operation.

The ACCW instruction operates with the Adjust CCW list (ACCW List). The ACCW list is loaded and maintained by the emulator program for use in the adjustment of CCW data addresses.

The OS DOS emulator and the DOS system being emulated (DOS supervisor and up to three processing program partitions) execute together in an MFT partition or MVT region, which must be a minimum of 38K. The OS DOS emulator program and tables require 22K plus another 4K if I/O staging is used. Additional OS DOS emulator program storage may be required, depending on the I/O devices used. Up to ten I/O devices are supported in 22K, and 250 bytes are required for each additional device. The I/O staging requirement of 4K supports unblocked reader, printer, and punch records and residence of the required OSAM routines in the OS DOS emulator partition or region.

The DOS system being emulated can be 16K, 24K, or 32K and up, in 4K increments. The OS DOS emulator is scheduled to operate in the same manner as any other OS job, and one or more OS DOS emulator jobs can execute concurrently with OS jobs if enough I/O devices and processor storage are available. In addition, the Model 145 OS 1401/1440/1460 and 1410/7010 Emulator programs can execute concurrently with the OS DOS emulator if enough resources are present.



Example Storage Assignment for 256K Model 145

OS DOS Functional Units

Table Buffer Registers

- Eight 24-bit registers are used to contain the local and real addresses used during the accessing of the local area when operating in local mode.
- The local address and the real address each occupy 12 bits of these registers.
- The local address is gated to the registers from PAA.
- The real address is gated to the registers from EBI.
- The register to be loaded is addressed by the LRU.

Least Recently Used Matrix (LRU)

- The LRU is an address matrix that keeps track of the use of the table buffer registers.
- The LRU addresses the least recently used table buffer register whenever a computed real address must be loaded.
- Composed of 28 latches and associated circuits.
- The LRU is reset to zero before the OS DOS operation.

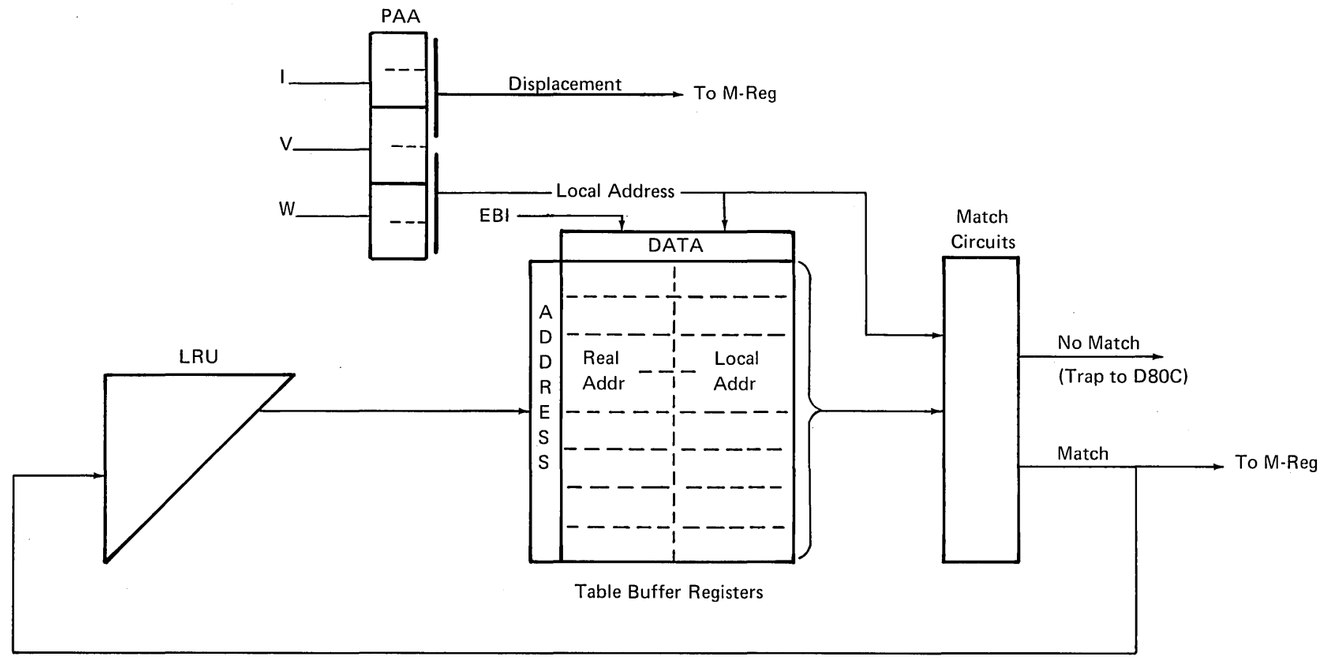
When a mismatch occurs between the local address portion of the PAA and the local address portion of the table buffer registers, the LRU determines the register to be loaded with the computed real address. In the GGST microroutine, the computed real address and the local address causing the mismatch are loaded into the register addressed by the LRU.

The status of the LRU is changed each time there is a match between the local address portion of the PAA and the local address from the table buffer registers. This constant changing assures that the least recently used register is addressed when a mismatch occurs. Refer to the example on the facing page.

Match Circuits

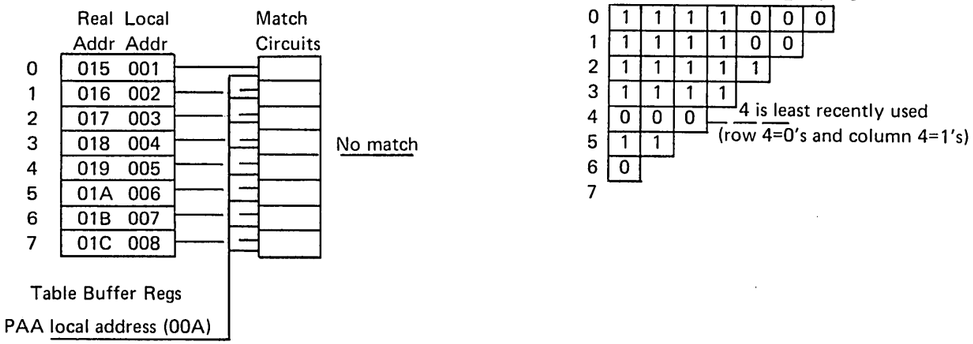
- Perform the comparison of the local address portion of the PAA and the local address portion of the table buffer registers.
- Output of the match circuits sets and resets specified combinations of the LRU.

There is a match circuit for each table buffer register. When a match occurs, the corresponding LRU row is set to ones and the corresponding column is reset to zeros. The resulting status of the LRU provides the means of addressing the least recently used register.



LRU Operational Example

FIRST STORAGE ACCESS ATTEMPT

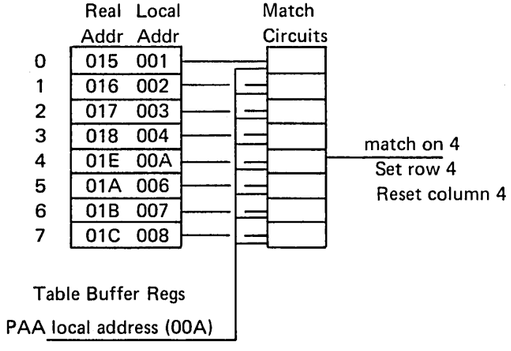


LRU

	7	6	5	4	3	2	1	0
0	1	1	1	1	0	0	0	
1	1	1	1	1	0	0		
2	1	1	1	1	1			
3	1	1	1	1				
4	0	0	0					
5	1	1						
6	0							
7								

4 is least recently used
(row 4=0's and column 4=1's)

RE-EXECUTION OF STORAGE WORD



LRU

	7	6	5	4	3	2	1	0
0	1	1	1	0	0	0	0	
1	1	1	1	0	0	0		
2	1	1	1	0	1			
3	1	1	1	0				
4	1	1	1					
5	1	1						
6	0							
7								

6 now least recently used
(row 6=0's and column 6=1's)

Assume: Storage access attempted for local address 0A000.

Adjustment factor = 14000

Table buffer Regs are set to values indicated.

Objectives: Provide a real address to the M-register to access the local area specified as address 0A000.

Description: When the storage access is attempted in local mode, the local address portion of the PAA and the local address portion of the table buffer Regs are compared.

A no-match condition results from the match circuits. This no-match condition causes a trap to control-storage address D80C. The GGST microroutine is executed. The GGST microroutine:

- Computes the real address,
- Loads the least recently used of the table look-aside buffer register with the real and local address,
- Re-executes the storage word that caused the mismatch.

Re-executing the storage word causes a match to occur from the number 4 table buffer register. The number 4 table buffer register was loaded in the GGST microroutine.

The match line from the number 4 register sets row 4 and resets column 4 of the LRU. The status of the LRU now indicates that the number 6 table buffer register is the least recently used. Should a mismatch occur on the next storage access, the computed real address will be loaded into register 6.

New Instructions for OS DOS Emulator

- The new instructions are Execute Local (EXL), and Adjust CCW String (ACCW).
- The Op code for both these instructions is B2.
- The immediate byte determines which of the two instructions is to be executed.

Execute Local

B2	0E	B1	D1

(EXL)

- This instruction addresses the Local Execution (LEX) list, performs certain initialization functions, and sets Local mode for system operation.

When the EXL instruction is executed, the condition code, program mask, and instruction address in the current PSW are replaced by values from the LEX list. General registers 14 and 15 are loaded from the LEX list and the CPU is placed in Local mode.

During the execution of the EXL instruction, the modified PSW is not checked for program interruptions. Any such checks occur as part of the next instruction execution.

Condition Code

Upon completion of the EXL instruction, the condition code is set according to the condition code loaded from the LEX list.

Program Interruptions

Addressing: The address of the LEX list is invalid. The address formed by the addition of the Origin address and the Local address exceeds the maximum address allocated to the Emulator program.
The operation is suppressed.

Operation: The instruction is not installed. The operation is suppressed.

Protection: The LEX list is protected for fetching or storing. The operation is suppressed.

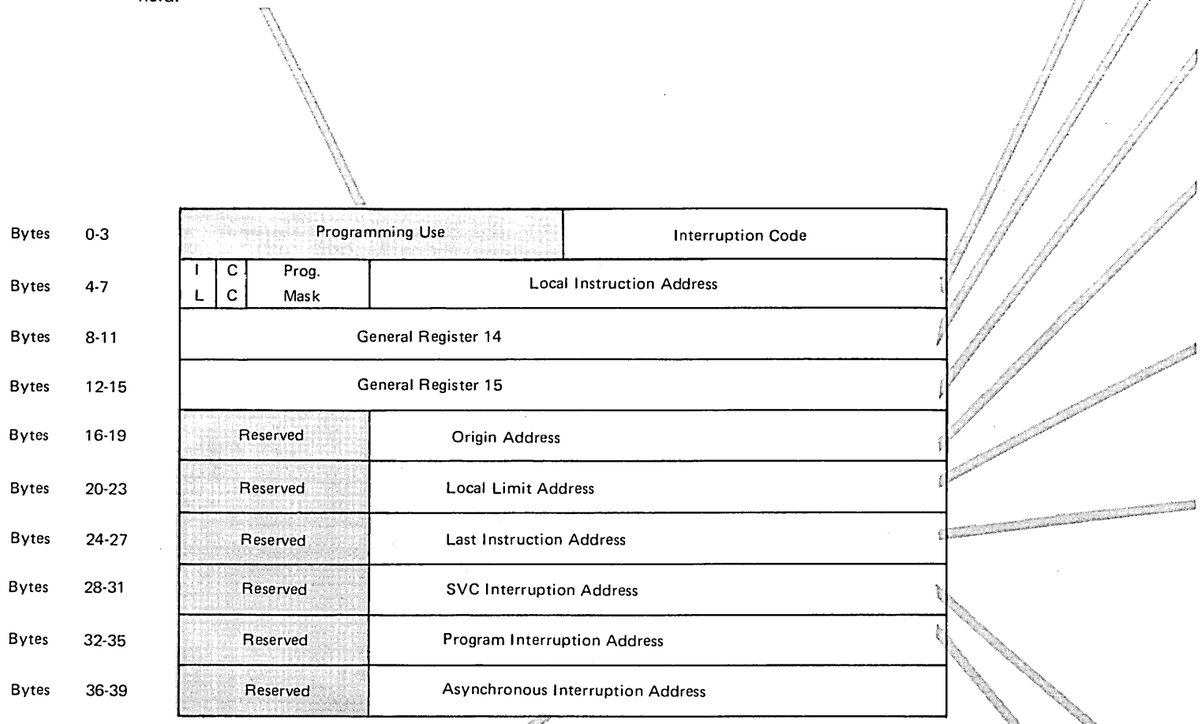
Specification: The first operand address does not specify a 64-byte boundary.
The Origin address is not a multiple of 4096.
The Local Limit address is not one less than a multiple of 4096.
The operation is suppressed.

Special Operations: The EXL instruction is encountered while in Local mode. The operation is suppressed. The interruption is reflected in the LEX list of the program that placed the CPU in Local mode.

Local List Format and Definition

Bytes 0, 1 — Reserved for emulator program use. This area is not addressed by the EXL instruction.

Bytes 2, 3 — Upon termination of Local mode by a program or supervisor call interruption, the 16-bit interruption code describing the interruption is placed in this field.



Bytes 36-39 — When Local mode is terminated by an asynchronous interruption (External, I/O, or machine-check), the address located in bytes 37-39 is placed in the PSW. The high-order byte is reserved and should be set to zero. The adjusted PSW is stored in the corresponding low storage old PSW and an interrupt is taken to the OS Supervisor.

Bytes 4-7 — Used to update the current PSW when the EXL instruction is executed. The instruction address in bytes 5-7 is the address of the next instruction within the DOS area. Whenever Local mode is terminated by an interruption, this area is updated. The ILC field is unpredictable when Local mode is terminated by an asynchronous interruption.

Bytes 8-11 — The value in this field is loaded into general register 14 when the EXL instruction is executed. When Local mode is terminated by an interruption, the current contents of general register 14 are stored into this field.

Bytes 12-15 — The value in this field is loaded into general register 15 when the EXL instruction is executed. When Local mode is terminated by an interruption, the current contents of general register 15 are stored into this field.

Bytes 16-19 — The address contained in bytes 17-19 corresponds to the address adjustment factor and points to the zero address of the emulated DOS supervisor. This address must be a multiple of 4096, or a specification interrupt occurs. The high-order byte of this field is reserved and should be set to zero.

Bytes 20-23 — The address contained in bytes 21-23 specifies the upper address of the emulated environment. The address must specify one less than a 4096 boundary, or a specification interrupt occurs. The high-order byte of this field is reserved and should be set to zero.

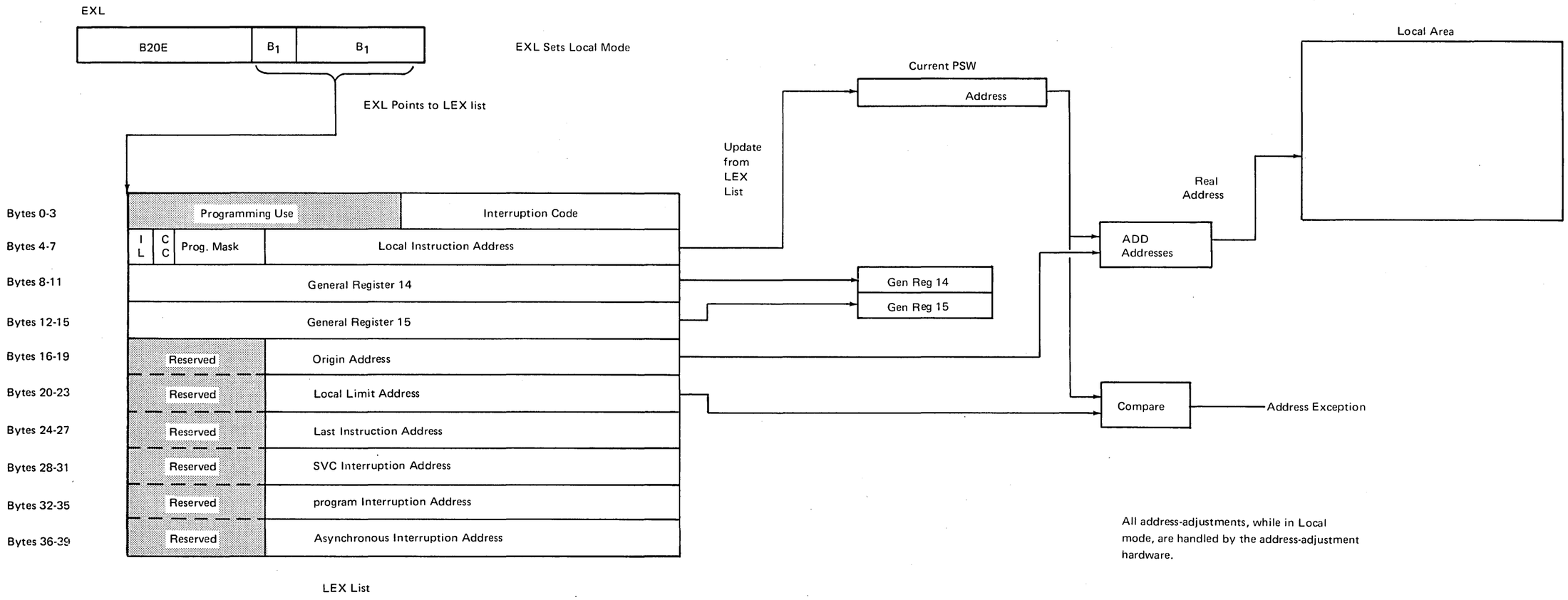
Bytes 24-27 — When Local mode is terminated by a program or supervisor call interruption, this address points to the instruction causing the interruption. This address is within the boundary of the emulator program. If the instruction causing the interruption was the object of an Execute instruction, the address placed in this field would be the address of the Execute instruction.

The contents of the address field are unpredictable when Local mode is terminated by an asynchronous interruption. The high-order byte of this field is set to zero.

Bytes 28-31 — When Local mode is terminated by a supervisor call interrupt, the address located in bytes 29-31 is placed in the PSW. This address is within the boundary of the emulator program. The high-order byte is reserved and should be set to zero.

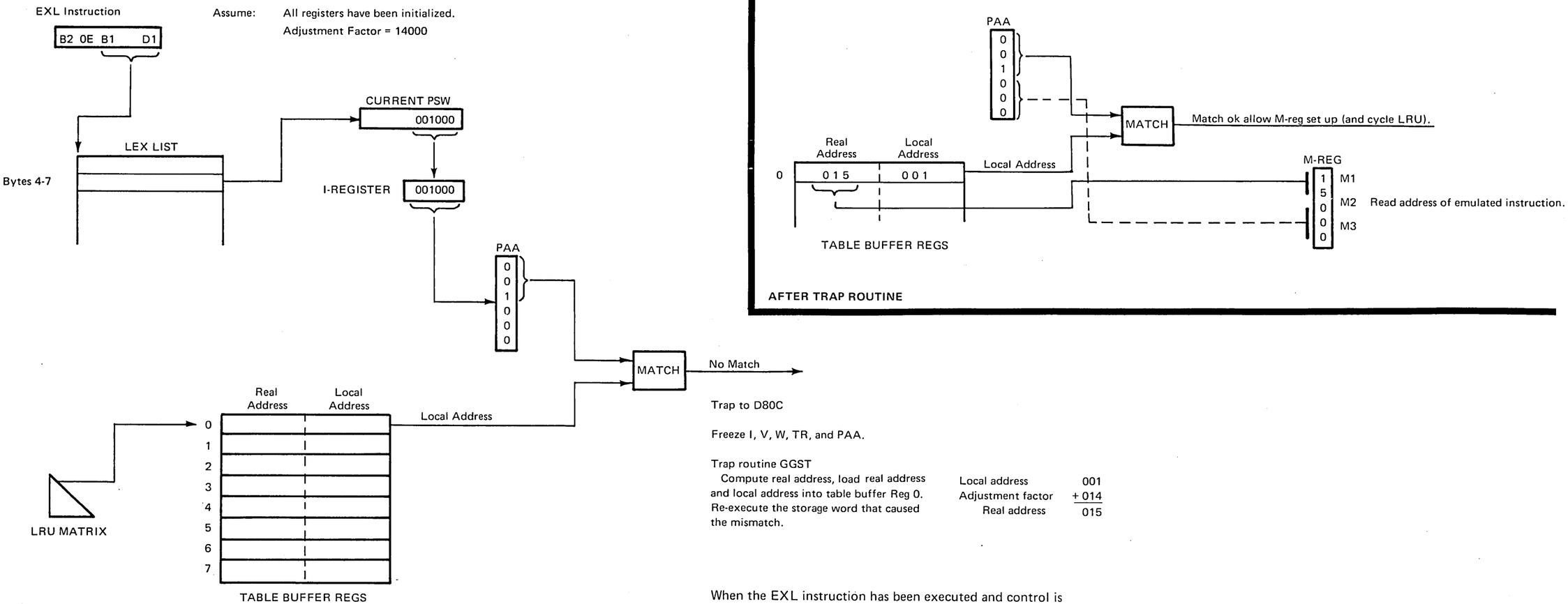
Bytes 32-35 — When Local mode is terminated by a program interrupt, the address located in bytes 33-35 is placed in the PSW. This address is within the boundary of the emulator program. The high-order byte is reserved and should be set to zero.

EXL Example: Execution and Local Mode Operation Start



All address-adjustments, while in Local mode, are handled by the address-adjustment hardware.

Real Address Computation Example



Initially the Table registers are set to zero. This includes the parity bits. The first time the Table registers are accessed, a mismatch occurs.

Trap routine GGST

Compute real address, load real address and local address into table buffer Reg 0. Re-execute the storage word that caused the mismatch.

Local address	001
Adjustment factor	+014
Real address	015

When the EXL instruction has been executed and control is passed to the current PSW, the address-adjustment circuits are activated.

Each time an access is made to main storage while in local mode, the local address portion of the PAA is checked against the local address area of all eight table buffer registers. When a match is detected, the real address from the table buffer register causing the match is gated to the M-register. The displacement value from PAA is also gated to the M-Reg to make up the complete real address of the location to be accessed.

When a no-match condition occurs, a trap is forced and the GGST microroutine is executed. The real address is formed, and the real address and local address are loaded into the least recently used table buffer register. The storage word causing the mismatch is re-executed, and the resulting match condition causes the real address to be gated to the M-register for a main-storage access.

Adjust CCW String

B2	0F	B1	D1

- The operand address designates the ACCW list.
- With the information from the ACCW list, this instruction addresses CCWs and performs adjustment on the data addresses of the CCWs.

The ACCW instruction interprets successive doublewords as CCWs and adjusts their data addresses by algebraiclly adding the adjustment factor to them. This process continues until:

The last CCW adjusted did not specify changing, or

A CCW, whose command code specifies TIC, has been adjusted,or

A CCW, whose data address points outside the emulated environment is encountered, or

The address of the next CCW is outside the limits of the emulated environment or does not specify a doubleword boundary.

Any of these conditions terminates the execution of the instruction and sets the proper condition code to specify the reason for termination.

When the ACCW instruction is completed, the address of the last CCW adjusted +8 is stored in bytes 17-19 of the ACCW list for condition codes 0, 1, or 2. For condition code 3, the address stored is CCW + 0. If data chaining was in progress, the command code and the address of the CCW containing the command code are set in the operation byte and operation pointer fields respectively.

If the last CCW adjusted specified transfer in channel, bytes 21-23 of the ACCW list contain the unadjusted data address from the TIC CCW. If the TIC CCW is encountered in a data-chaining sequence, the operation byte and operation pointer of the ACCW list contain the values set from the first CCW of the chain. When the TIC is not data chained, the operation byte in the ACCW list is set to zero. The CCW address field in the ACCW list is set to the address +8 of the TIC CCW.

Condition Code

0 End of the CCW string. The last CCW adjusted specified neither data chaining nor command chaining.
1 A TIC CCW was the last CCW adjusted.
2 An adjusted data address was encountered that fell outside the area of the emulated environment.
3. The address of the next CCW to be adjusted did not specify a doubleword boundary or fell outside the area of the emulated environment.

Program Interruptions

Addressing: The address of the ACCW list is outside available storage. The operation is suppressed.
The address of a CCW is outside available storage. The operation is terminated..

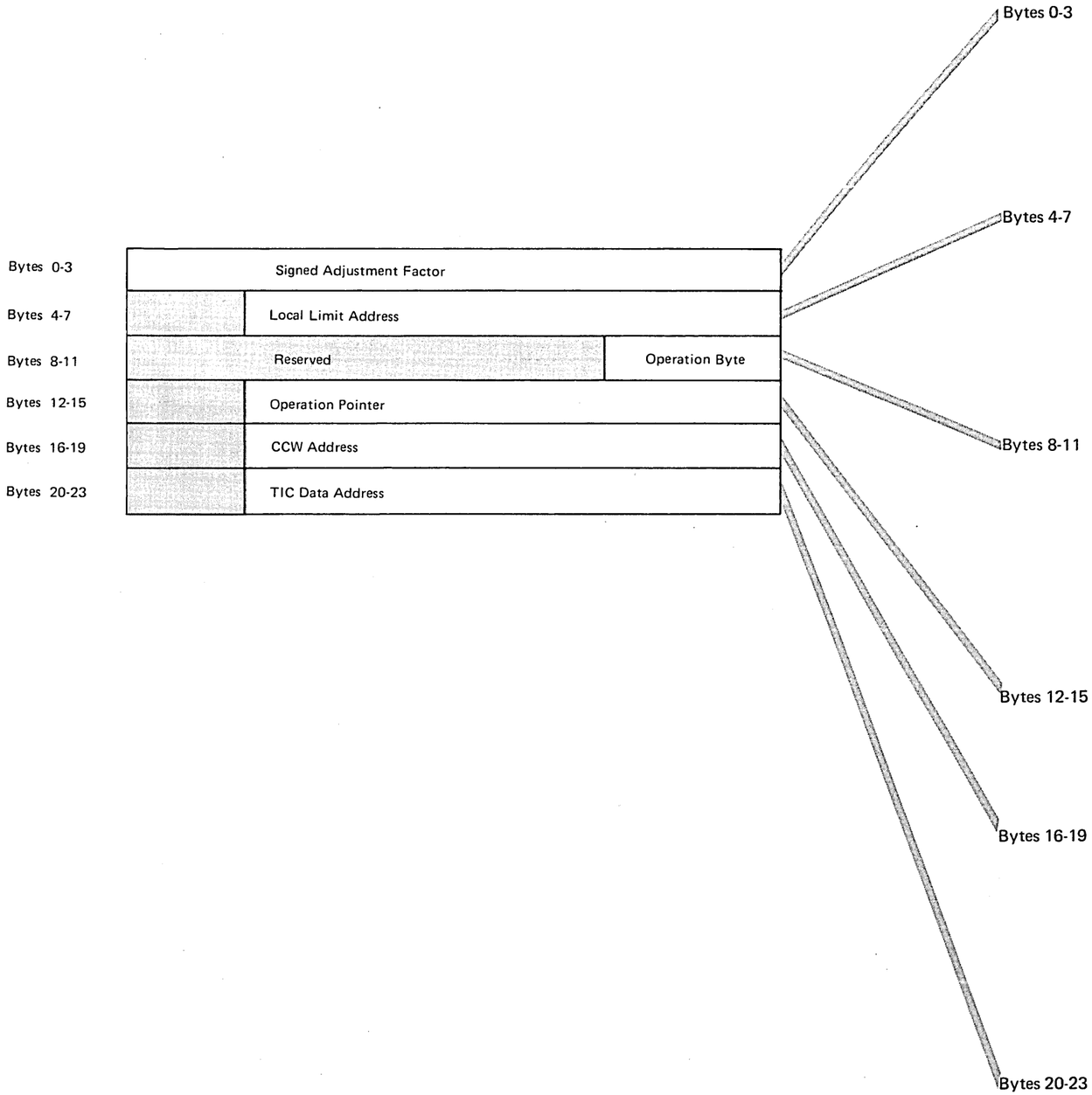
Operation: The instruction is not installed. The operation is suppressed.

Protection: The ACCW list is protected for storing or fetching. The operation is suppressed.
A CCW is protected for fetching or storing. The operation is terminated.

Specification: The first operand address does not specify a 64-byte boundary; the signed adjustment factor is not a multiple of 4096; the local limit address is not one less than a multiple of 4096. The operation is suppressed.

Special Operation: The ACCW instruction was encountered while in local mode. The operation is suppressed. The interruption is reflected to the program that placed the CPU in local mode, by an address in the local mode, by an address in the local list.

Adjust CCW List Format and Definition



Bytes 0-3 The signed binary number located in this field is added to the data address of the CCW addressed by bytes 16-19 of the ACCW list. The 24 low-order bits of the result are set into the data address field of the CCW. The CCW data address, which is local to the emulated environment, is compared against the local limit address. If the comparison indicates that the local address is above the local limit address, a program interruption occurs.

Bytes 4-7 The address contained in bytes 5-7 specifies the upper address of the emulated environment. The address must specify one less than a 4096 boundary, or a specification interrupt occurs. The local limit address is compared with the local CCW address, and the extreme local address of the storage area defined for each CCW by the data address, command code, and unit, to assure that the local address is within the limits of the storage area assigned to the emulated environment.

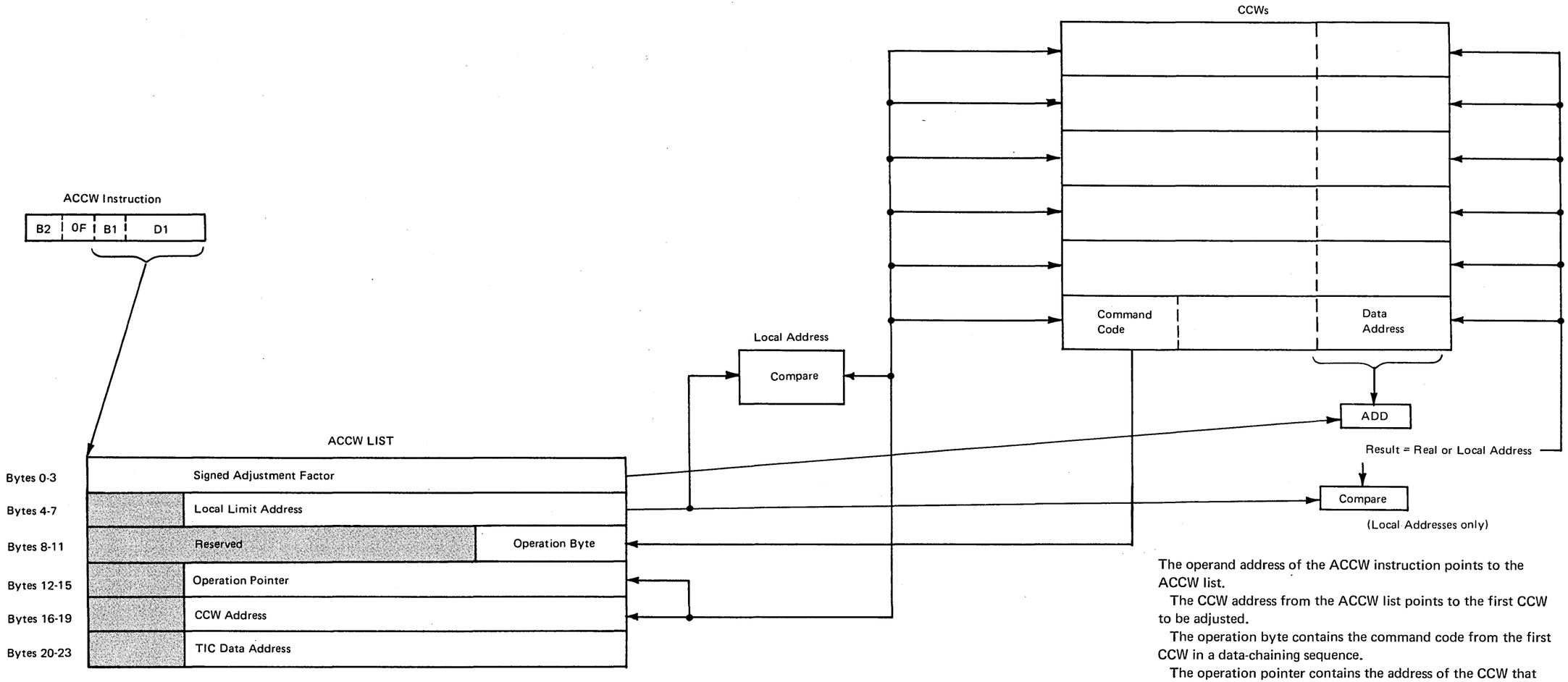
Bytes 8-11 The operation byte (byte 11) carries the command code for CCWs that are data-chained. The operation byte is set to zero if the CCW being processed is not data-chained. When the operation byte is fetched from the ACCW list, the high-order bytes of this field are ignored. When the operation byte is stored in the ACCW list, the high-order bytes are set to zero. A nonzero operation byte encountered upon initiation of the instruction indicates that the first CCW to be adjusted is part of a data-chained sequence.

Bytes 12-15 This field contains the address of the CCW that originated the operation byte for the last non-TIC CCW adjusted. When this address is fetched from the ACCW list, the high-order byte is ignored. When the address is stored in the ACCW list, the high-order byte is set to zero.

Bytes 16-19 The address contained in this field is the address of the first CCW of a string when the ACCW instruction is encountered. When the ACCW is completed, this address points to the doubleword above the last CCW adjusted when terminated with condition 0, 1, or 2. When terminated with condition code 3, the address points to the CCW causing termination. When this address is fetched from the ACCW list, the high-order byte is ignored. When the address is stored in the ACCW list, the high-order byte is set to zero.

Bytes 20-23 Contains the unadjusted data address from the CCW whose command specifies transfer in channel. The high-order byte is set to zero.

Operation Example of the ACCW Instruction



The operand address of the ACCW instruction points to the ACCW list.

The CCW address from the ACCW list points to the first CCW to be adjusted.

The operation byte contains the command code from the first CCW in a data-chaining sequence.

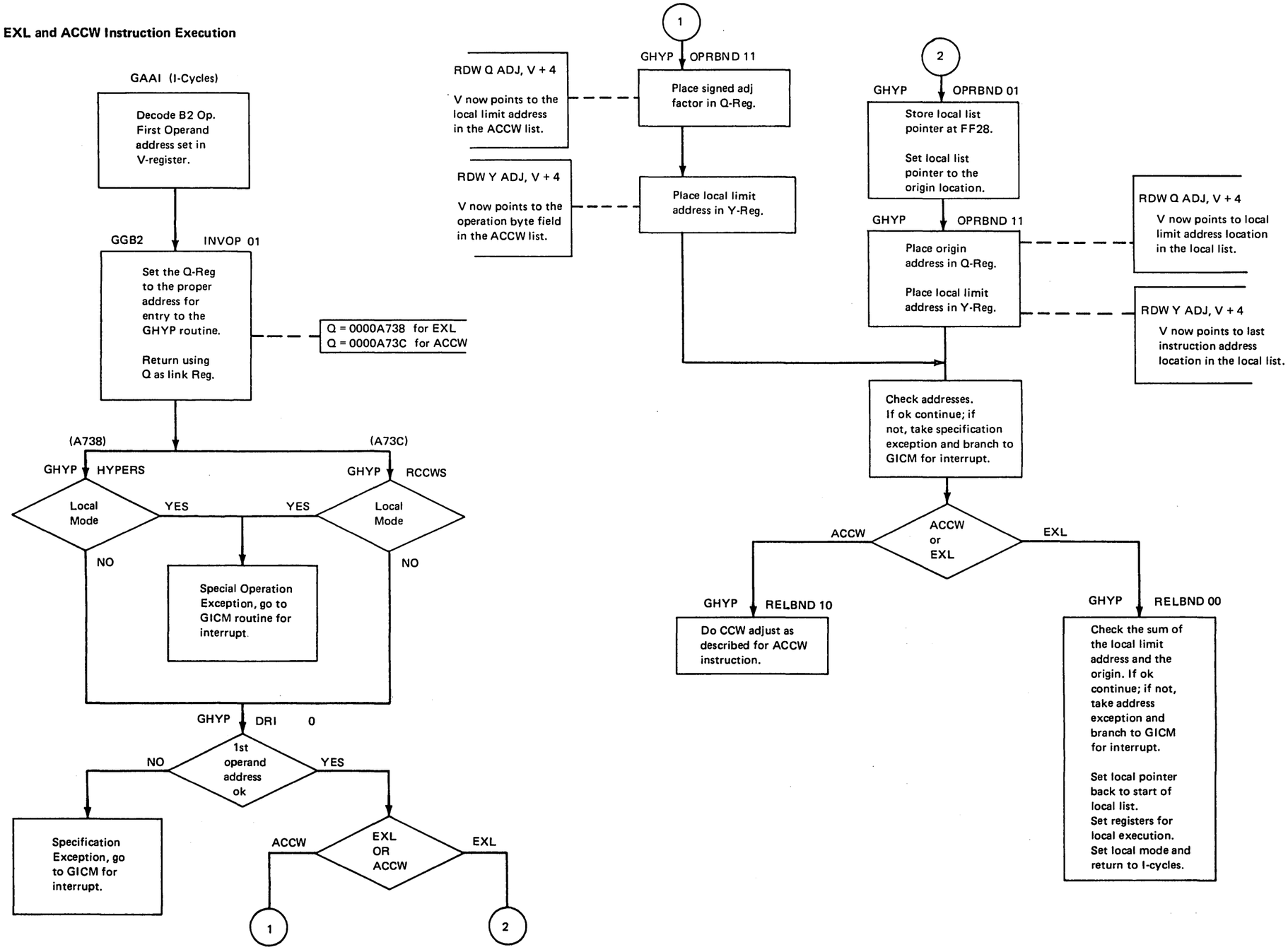
The operation pointer contains the address of the CCW that provided the operation byte.

Each CCW in the chain is addressed from the ACCW list.

The data address from the CCW is added with the signed adjustment factor from the ACCW list. The local CCW address is compared with the local limit address. If the comparison indicates a valid local address, the operation continues. If the comparison indicates an invalid local address, the operation is terminated and a condition code of 3 is set. The extreme local address of the storage area defined for each CCW by the data address, command code, and count are compared with the local limit address and zero. If the comparisons indicate a valid local storage area, the adjusted data address is placed in the data address field of the CCW. If the address compare is invalid, the operation is terminated and a condition code of 2 is set.

The CCW address in the ACCW list is updated +8 to point to the next CCW to be adjusted. When the last CCW in the string has been adjusted, the CCW address points to the next sequential doubleword.

EXL and ACCW Instruction Execution



Interruptions

- Any interrupt removes the CPU from local mode.
- All non asychronous interrupts that occur while in local mode are handled by the emulator program.
- All asynchronous interrupts that occur while in local mode are first handled by the OS Supervisor.

If a Supervisor Call, Program, External, I/O, or recoverable Machine-Check interruption occurs while the CPU is in local mode, the following action takes place.

The 16-bit interruption code associated with the Supervisor Call or Program interruption is stored in the Interruption Code field of the LEX list. The contents of this field after an External, I/O, or Machine-Check interruption are unpredictable.

The ILC, CC, Program Mask, and instruction address of the current PSW are stored in the bytes 4-7 of the LEX list. The value of the ILC after an asynchronous interruption is unpredictable.

The current contents of general registers 14 and 15 are stored into bytes 8-15 of the LEX list.

If the interrupt is a Supervisor Call or Program interrupt, the local address of the instruction causing the interruption is stored into bytes 25-27 of the LEX list. If the instruction causing the interrupt was the object of an Execute instruction, the local address of the Execute instruction is stored.

The corresponding interrupt address (SVC, Program, or Asynchronous) is loaded into the current PSW from the LEX list.

If the interrupt is an asynchronous interrupt, (I/O, External, or Machine-Check) the adjusted current PSW is stored in the corresponding low storage old PSW and an interrupt is then sent to the OS Supervisor.

The CPU is removed from local mode.

LEX List After SVC Interrupt

Bytes	0-3	Programming Use			SVC Interrupt Code
Bytes	4-7	I L	C C	Prog. Mask	Local Instruction Address
Bytes	8-11	General Register 14			
Bytes	12-15	General Register 15			
Bytes	16-19	Reserved			Origin Address
Bytes	20-23	Reserved			Local Limit Address
Bytes	24-27	Reserved			SVC or Execute Address
Bytes	28-31	Reserved			SVC Interruption Address
Bytes	32-35	Reserved			Program Interruption Address
Bytes	36-39	Reserved			Asynchronous Interruption Address

LEX list after Program Interrupt

Bytes	0-3	Programming Use			Pgm Interrupt Code
Bytes	4-7	I L	C C	Prog. Mask	Local Instruction Address
Bytes	8-11	General Register 14			
Bytes	12-15	General Register 15			
Bytes	16-19	Reserved			Origin Address
Bytes	20-23	Reserved			Local Limit Address
Bytes	24-27	Reserved			Addr of inst. causing int. or execute address
Bytes	28-31	Reserved			SVC Interruption Address
Bytes	32-35	Reserved			Program Interruption Address
Bytes	36-39	Reserved			Asynchronous Interruption Address

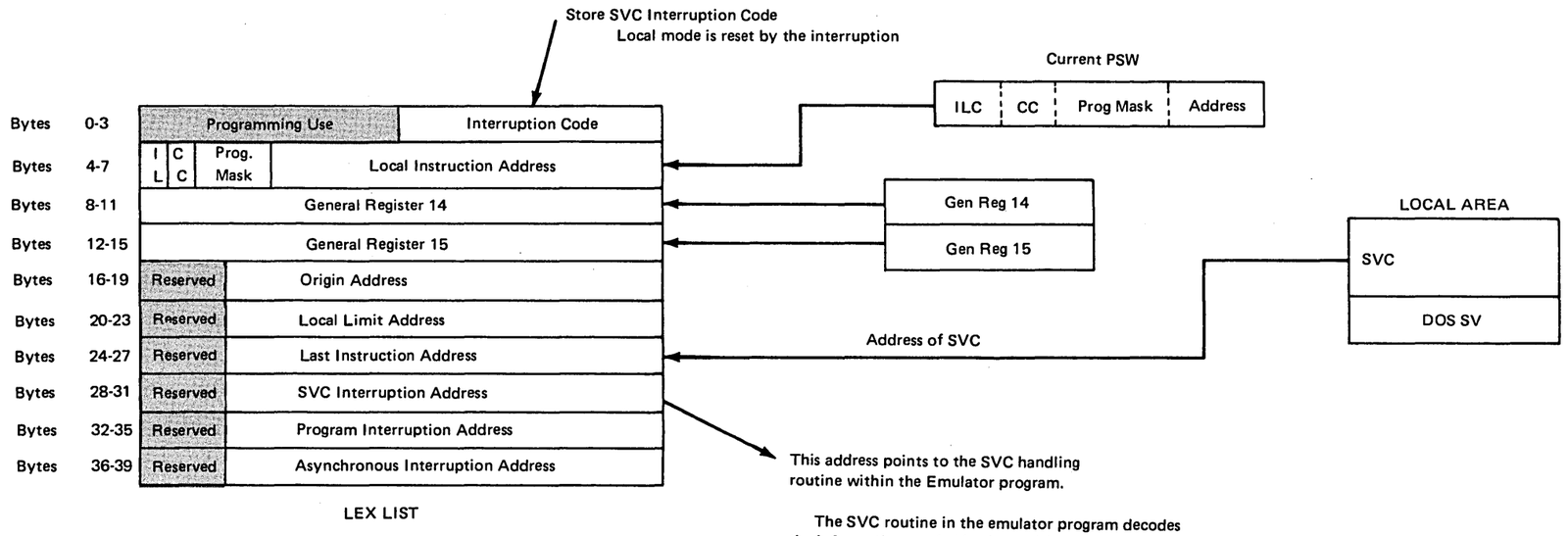
When the interruption is a specification exception due to an odd address, the ILC is unpredictable. The Last Instruction field contains the odd address.

LEX List After an Asynchronous Interrupt

Bytes 0-3	Programming Use			UNPREDICTABLE
Bytes 4-7	I L	C C	Prog. Mask	Local Instruction Address
Bytes 8-11	General Register 14			
Bytes 12-15	General Register 15			
Bytes 16-19	Reserved			Origin Address
Bytes 20-23	Reserved			Local Limit Address
Bytes 24-27	Reserved			UNPREDICTABLE
Bytes 28-31	Reserved			SVC Interruption Address
Bytes 32-35	Reserved			Program Interruption Address
Bytes 36-39	Reserved			Asynchronous Interruption Address

*ILC is unpredictable

SVC Interruption Example



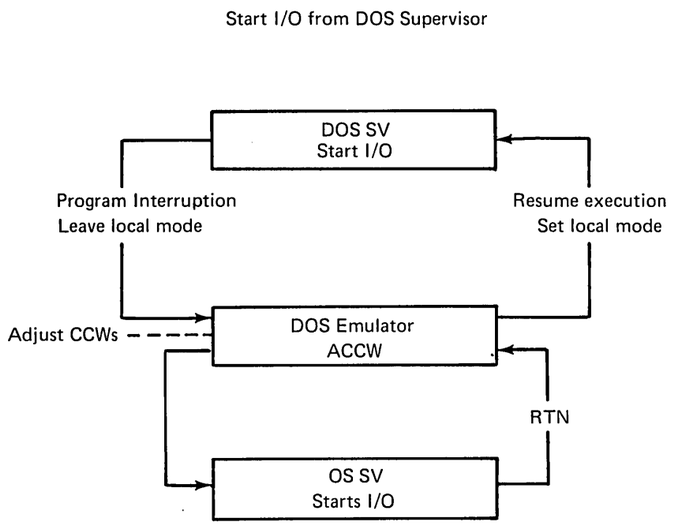
This address points to the SVC handling routine within the Emulator program.

The SVC routine in the emulator program decodes the information carried in the SVC instruction to determine the DOS SV address to branch to.

There are no PSW swaps for this type of interrupt.

Once the DOS return address has been computed, an EXL instruction is executed, local mode is set, and a return is made to an address in the DOS SV.

Start I/O Interruption Example



When a Start I/O instruction is executed in the DOS SV, a program interruption occurs because the DOS SV is being operated on in the problem state. The DOS Emulator intercepts the program interruption, leaves local mode, and performs the following:

1. The privileged operation interruption code is set into bytes 2 and 3 of the LEX list.
2. The selected current PSW information is loaded into bytes 4-7 of the LEX list.
3. The address of the Start I/O instruction is loaded into bytes 25-27 of the LEX list.
4. The current values of general registers 14 and 15 are loaded into bytes 8-15 of the LEX list.
5. The Program Interruption Address, bytes 33-35 of the LEX list, is placed into the Program old PSW of the OS SV.
6. The ACCW instruction is executed to adjust the CCW data addresses back to their original values.
7. Control is transferred to the OS SV to handle the Start I/O.

When the OS SV has handled the Start I/O, a return is made to the DOS Emulator. The Emulator executes an ACCW instruction to adjust the CCW data addresses. The EXL instruction is then executed to restore information prior to restarting the DOS SV at the point of interruption. Refer to the example of EXL execution.

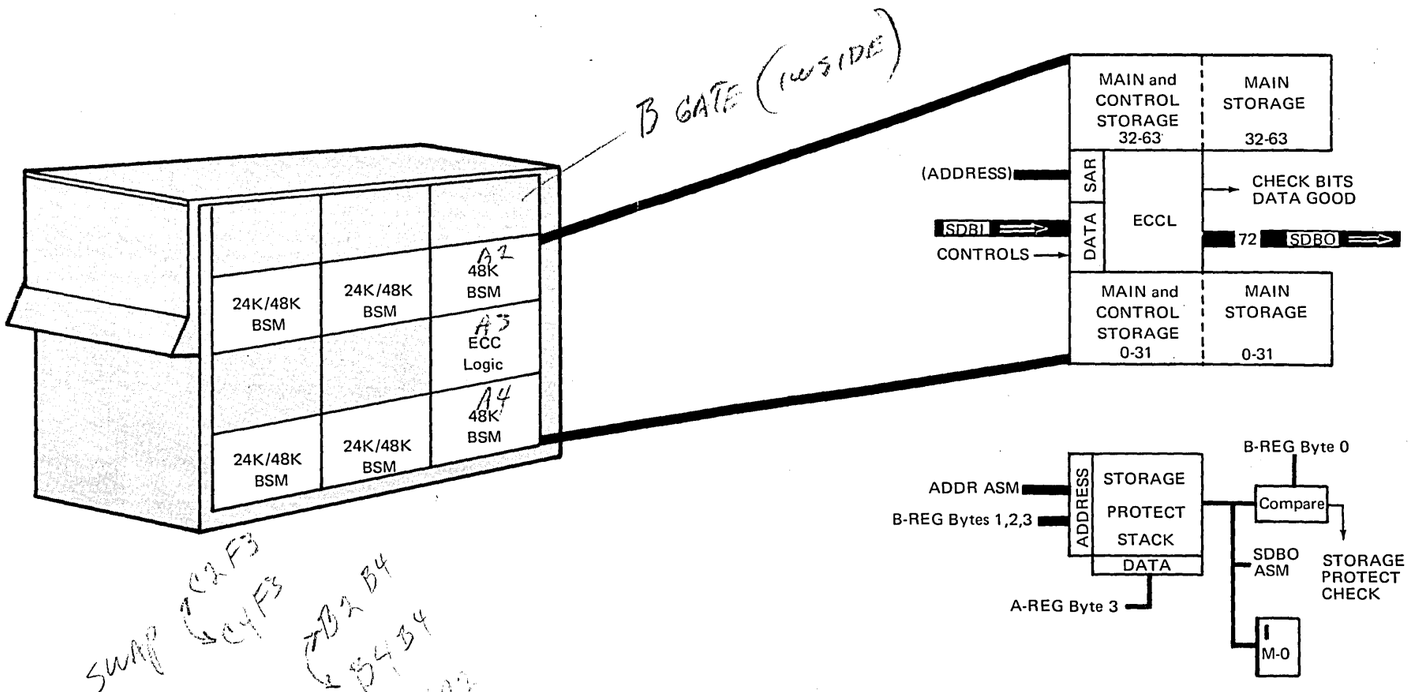
REMEMBER

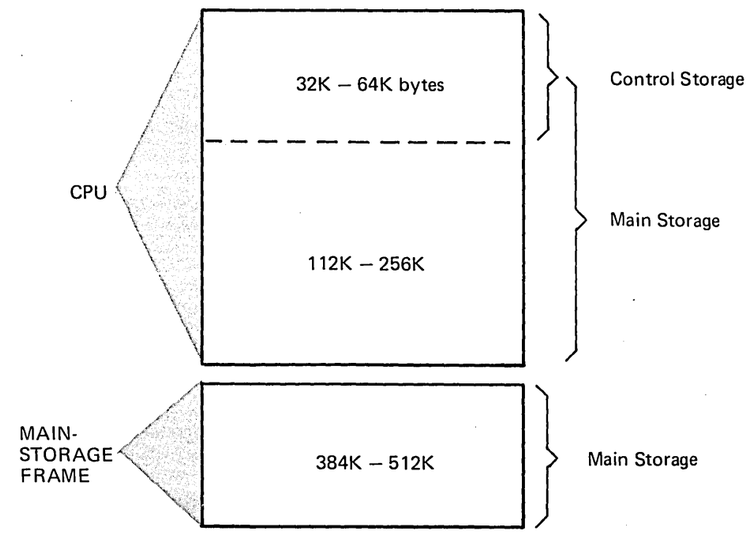
There is a Reader's Comment Form at the back of this publication.

CHAPTER 3. MAIN AND CONTROL STORAGE

REMEMBER

There is a Reader's Comment Form at the back of this publication.





- Main and control storage uses Phase 2I monolithic circuitry.
- Non-destructive readout.
- Main storage is the area of storage assigned for program use.
- Control storage contains the microprogram upon which all system operations depend.
- Both main and control storage are addressed by the M-register.
- The first control-storage-address location always follows the last high-order address of main storage.
- The control-storage area is loaded automatically on a power-up sequence (IMPL).

- Storage cycle times: 540.0 nanoseconds for main-storage read; 607.5 nanoseconds for a main-storage write.
- Control-storage access time: 109 nanoseconds.
- Storage access width: eight bytes (one doubleword).
- Protection features: Both store protection and fetch protection are standard on the 3145.
- Error-checking circuitry and correction circuitry for main and control storage used by the 3145 automatically correct single-bit errors, which reduces the number of system interruptions. Automatic detection of double-bit errors is also provided, stored in MCKB register (byte 2, bit 5).

- Storage size: The main-storage capacity within the CPU frame may be any of the following:

CPU Model	Program Storage	Control Storage
3145FED	114,688 bytes (112K)	32K (See Note)
3145GE	163,840 bytes (160K)	32K
3145GFD	212,992 bytes (208K)	32K
3145H	262,114 bytes (256K)	32K
3345	393,216 bytes (384K)*	32K
3346	524,288 bytes (512K)*	32K

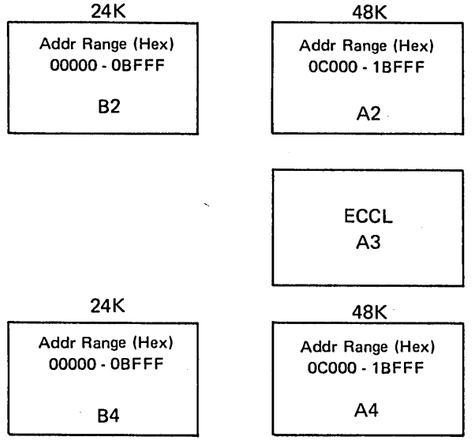
*Main-storage capacity above 256K bytes is contained in a 3345 main storage frame. When a main storage frame is attached, it contains the low-order storage addresses.

Note: The 3145 has movable control-storage boundary that allows up to 64K (65,536) bytes of control storage, depending on the features installed. The additional control-storage capacity above 32K is at the expense of main storage. The storage boundary is determined at the time that the microprogram is compiled.

3145 BASIC STORAGE MODULE (BSM) CONFIGURATIONS

- These are the four main-storage sizes located at the 01 gate that are available to the 3145.
- The storage area interfaces with the CPU via the error-correction and control logic board.
- Depending upon the storage capacity, the 3145 can contain up to 6 BSMs.

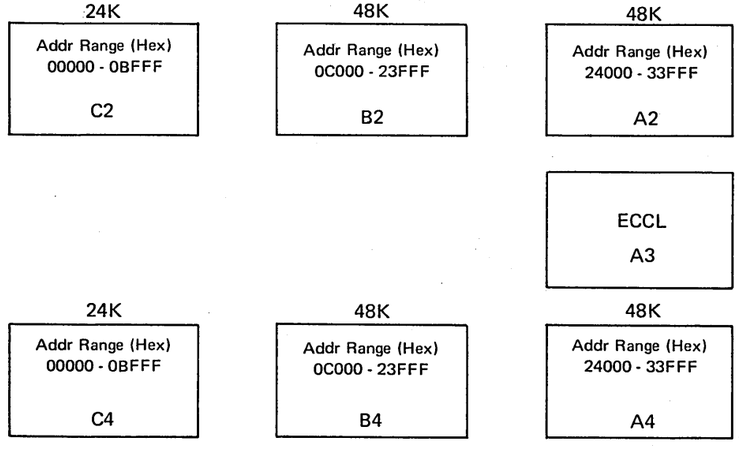
112K BYTES (MODEL FED)



A2 and A4
1 Each address BFR
1 Each terminator
36 Each array cards

B2 and B4
1 Each address BFR
1 Each terminator
18 Each (bottom row) array cards
36 Each jumpers (red)

208K BYTES (MODEL GFD)

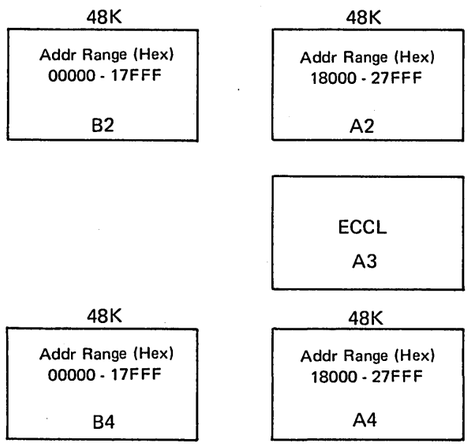


A2 and A4
1 Each address BFR
1 Each terminator
36 Each array

B2 and B4
1 Each address BFR
1 Each terminator
36 Each array card
36 Each jumpers

C2 and C4
1 Each address BFR
1 Each terminator
18 Each (bottom row) array
36 Each jumpers (red)

160K BYTES (MODEL GE)

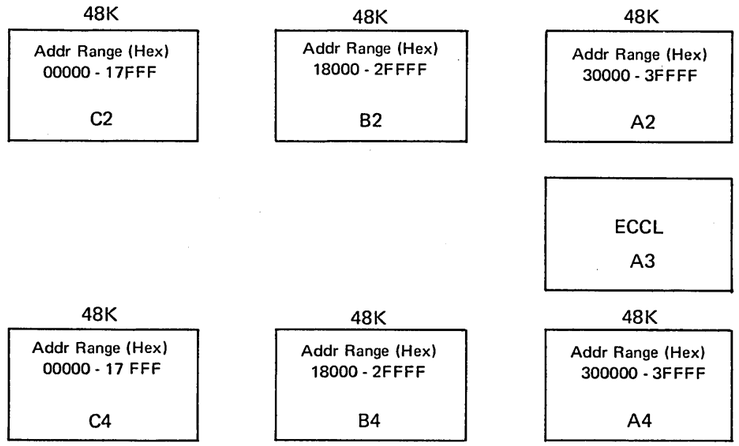


A2 and A4
1 Each address BFR
1 Each terminator
36 Each array cards

B2 and B4
1 Each address BFR
1 Each terminator
36 Each array cards
36 Each jumpers (yellow)

Note:
1. Add BFR (A-socket) terminator card (V-socket)
2. 24K BSM-array cards B4-U4 sockets
3. Jumpers 24K BSM- (red)
   48K BSM- (yellow)

256K BYTES (MODEL H)



A2 and A4
1 Each address BFR
1 Each terminator
36 Each array cards

B2 and B4

C2 and C4
1 Each address BFR
1 Each terminator
36 Each array cards
36 Each jumpers (yellow)

Note:
1. Add BFR (A-socket) terminator card (V-socket)
2. 24K BSM-array cards B4-U4 sockets
3. Jumpers 24K BSM- (red)
   48K BSM- (yellow)

BASIC BSMs

The optional main-storage sizes availabe on the 3145 are 112K, 160K, 208K, 256K, 384K, and 512K bytes. All BSMs on models with main-storage sizes of 256K or fewer are mounted in the CPU frame. An external main-storage frame is required on models with 384K or 512K main storage.

An additional 32K bytes reloadable control storage is incorporated in each system. The reloadable control storage shares certain BSMs with main storage but is not included as part of main storage. In addition to the storage card, each BSM contains an address buffer card and a terminator card.

BSM Sizes

Traditionally the total storage capacity of a given storage element has been expressed in kilobytes such as 32K, 48K, 208K, 256K, and so on. The width of the associated data path in bits received minimum emphasis. The capacity of the BSM is given in total words by the number of bits per word: 12K x 36 bits; for example, a 12K x 36 BSM can store 12000 36-bit words, or 48K bytes. The basic storage interface and data flow are shown on page 3-7.

Two basic BSM sizes, 48K and 24K, are used in varied combinations to provide the range of main-storage options within the CPU.

48K BSM (12K x 36)

Two versions: fast and slow
Fast uses 12K x 1 fast array cards.
Slow used 12K x 1 slow array cards.

12K x 1 fast array card contains:

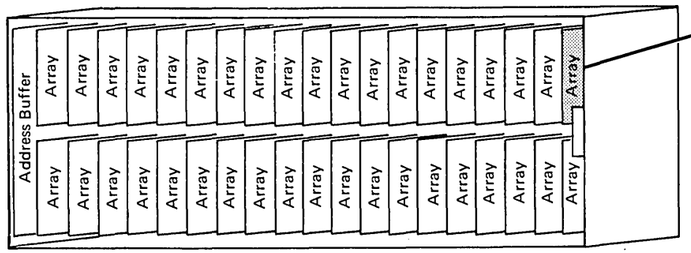
1. Selected array modules that meet the requirements for minimum circuit delay.
2. One sense latch used in parallel to decrease readout time.

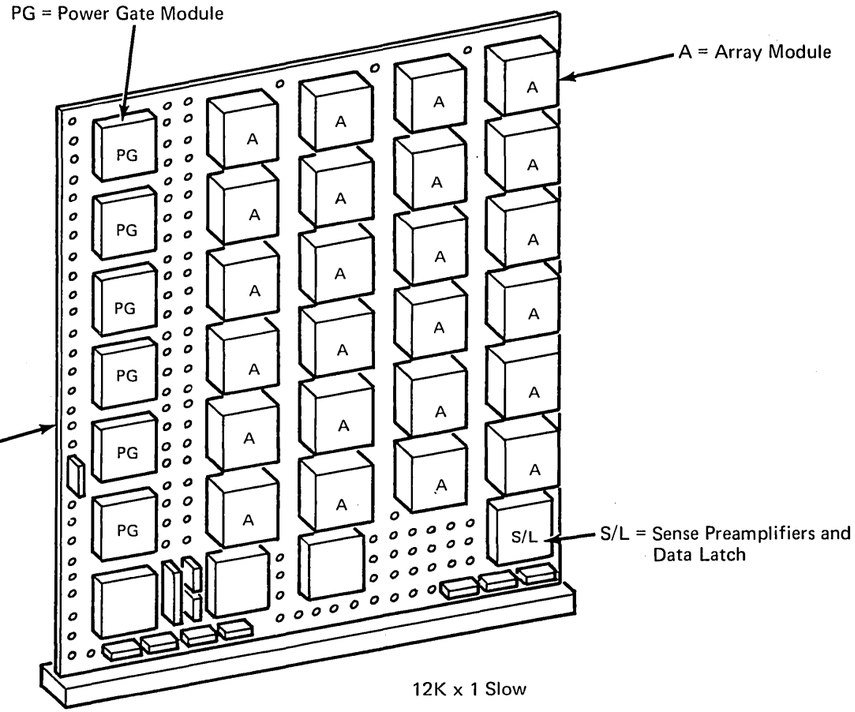
The 48K slow BSM is used for main storage where access and cycle time is less critical.

The 48K fast BSM is used for reloadable control storage where fast access and short cycle time is critical.

Note:
A 12K x 1 fast card can replace a 12K x 1 slow card, but a 12K x 1 slow card cannot replace a 12K x 1 fast card.

PHASE 2 48K BSM



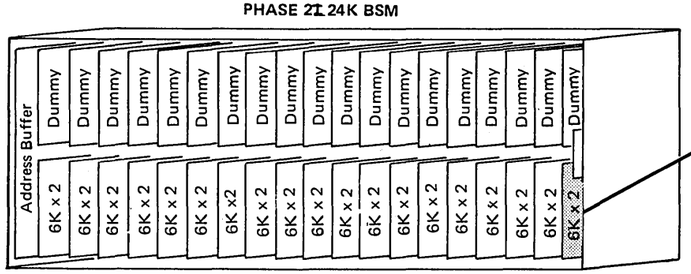


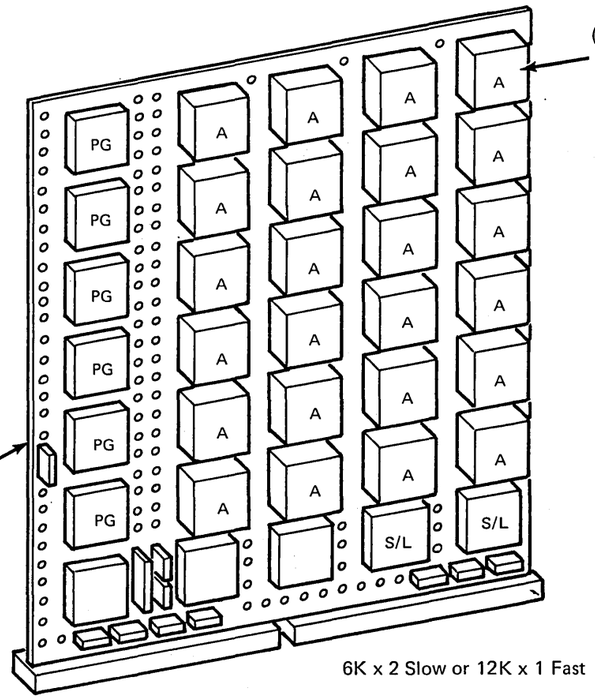
12K x 1 Slow

24K BSM (6K x 36)

Uses 6K x 2 array cards. SAR address is modified to allow reading or writing two bits at a time on each array card. BSM is field-expandable to 48K bytes.

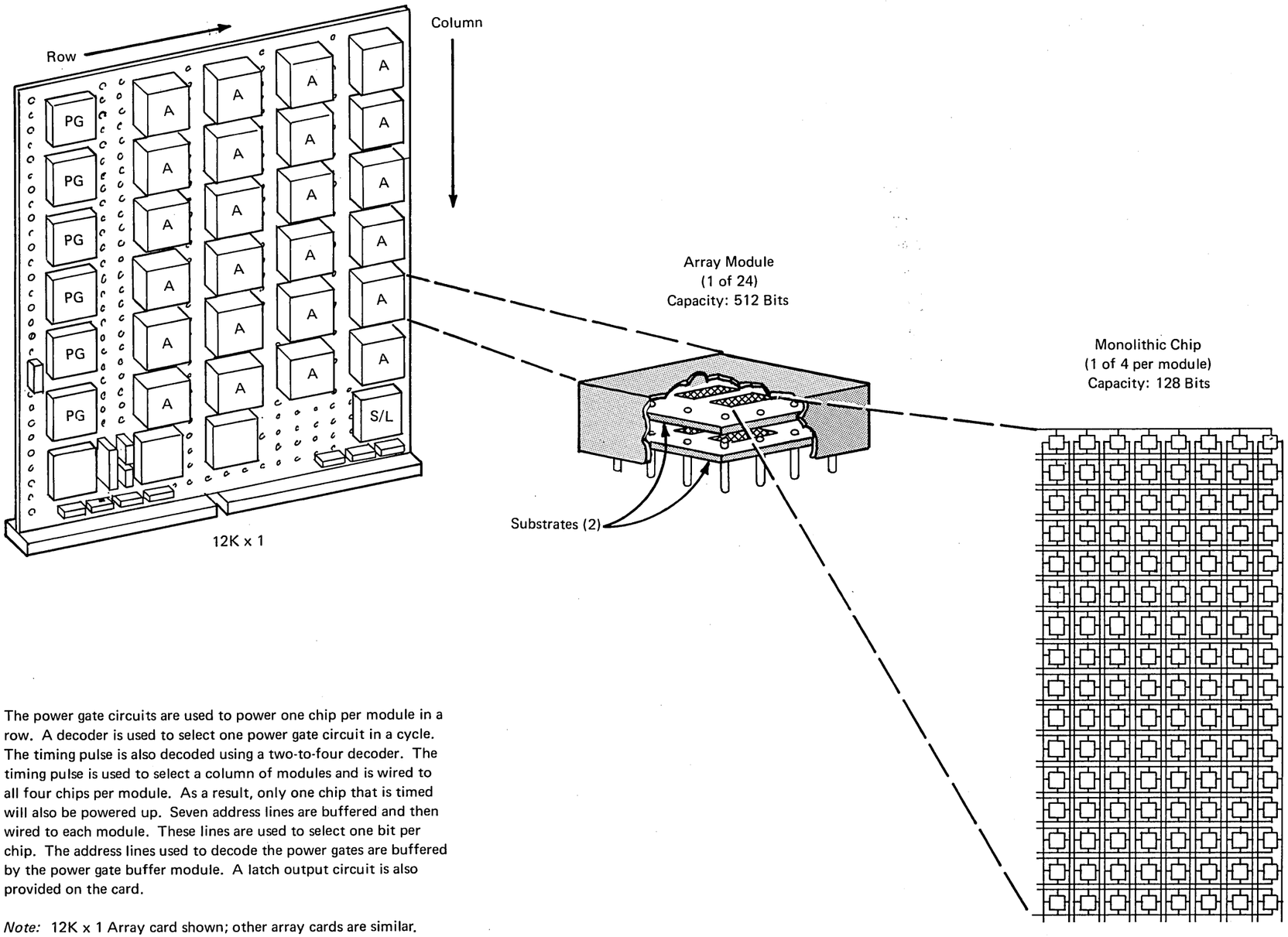
PHASE 2 24K BSM





6K x 2 Slow or 12K x 1 Fast

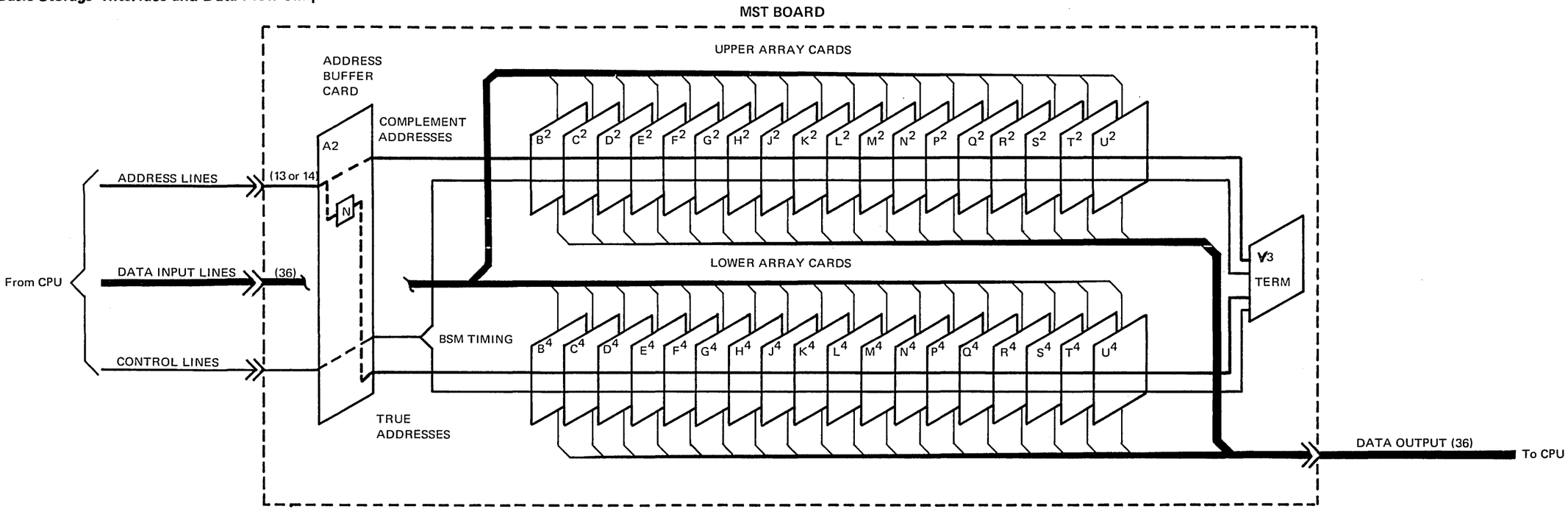
Array Card Storage Elements



The power gate circuits are used to power one chip per module in a row. A decoder is used to select one power gate circuit in a cycle. The timing pulse is also decoded using a two-to-four decoder. The timing pulse is used to select a column of modules and is wired to all four chips per module. As a result, only one chip that is timed will also be powered up. Seven address lines are buffered and then wired to each module. These lines are used to select one bit per chip. The address lines used to decode the power gates are buffered by the power gate buffer module. A latch output circuit is also provided on the card.

Note: 12K x 1 Array card shown; other array cards are similar.

Basic Storage--Interface and Data Flow Simplified



12K x 36 BSM

- All address lines enter the address buffer card, where they are powered and inverted to provide in-phase and out-of-phase addresses to the array cards. The use of complement addresses minimizes power-driver-circuit requirements on the address buffer card.
- The control lines, board select, read/write, and certain address bits are used on the address buffer card to generate the BSM timing pulses used during each store or fetch cycle.
- All array cards participate in each store (write) or fetch (read) cycle.
- During each store cycle, one bit of data is stored in each array card.
- For each array card, data is routed directly across the BSM interface to the system.
- The terminator card provides the correct terminating impedance for the internal address and control lines.

CONTROLS

- The BSM storage area controls consist of:
  1. Clock for timing and
  2. An addressing scheme

Controls for moving data into and out of the storage unit consist of the addressing scheme to locate words in the BSM, and a clock to time operations within the storage cycle. BSM clock operation is asynchronous with the CPU. If not in use, the storage area does not cycle but waits for a storage select pulse to activate the internal timing circuits. To start a storage cycle, the CPU sends an address, read (not write control) or write control, and a storage select pulse to the ECCL board within the storage area. The cycle proceeds under control of the storage unit clock, and data is gated into or out of the BSMs.

BSM Timing

The BSM timing mechanism is composed of tapped delay lines and provides all the timed pulses necessary to control the BSMs. The clock is composed of two delay line cards located in the ECCL board. In addition to the delay lines in the ECCL board, each BSM contains its own delay line within the address buffer card. The ECCL board delay lines are adjustable and operate for approximately 280 nanoseconds (ns) and are in steps of 10-nanosecond increments. These delay lines are basically used to generate control pulses. The delay line in the address buffer card generates the BSM timing signals required by each array card during a store or fetch cycle.

The delay line in the ECCL board is started after the receipt of a 45-ns select pulse generated at the CPU. The delay line in the address buffer card starts about 8 nanoseconds later than the ECCL board delay line. This allows the signals from the ECCL board to propagate to the selected BSMs. For further detail on storage timing, refer to Page 3-22.

BSM Addressing

The main- or control-storage address gated the the M register is transferred directly to the storage address register (SAR) as shown in the illustration. From SAR, the address bits are routed to board select circuits within the ECCL board and to the BSM address buffer card to select the array module and chip, and the proper bit cell. For each storage access, the address is sent to corresponding pairs of BSMs. Address lines 1-14 are used on 48K (12 x 36) BSMs; address line 6 is not used on 24K (6K x 36) BSMs.

The address lines may enter the array board (BSM) at either of two socket locations, A2 and A5. If the board pins in socket location A5 are used as entry pins, then the board pins in socket location A2 becomes exit pins. Depending on the board location in the BSM configuration, the exit pins may be used to:

1. Route the address lines to an adjacent BSM, or
2. Terminate the address with plug on terminators if the array board is the last in a chain of multiple BSMs.

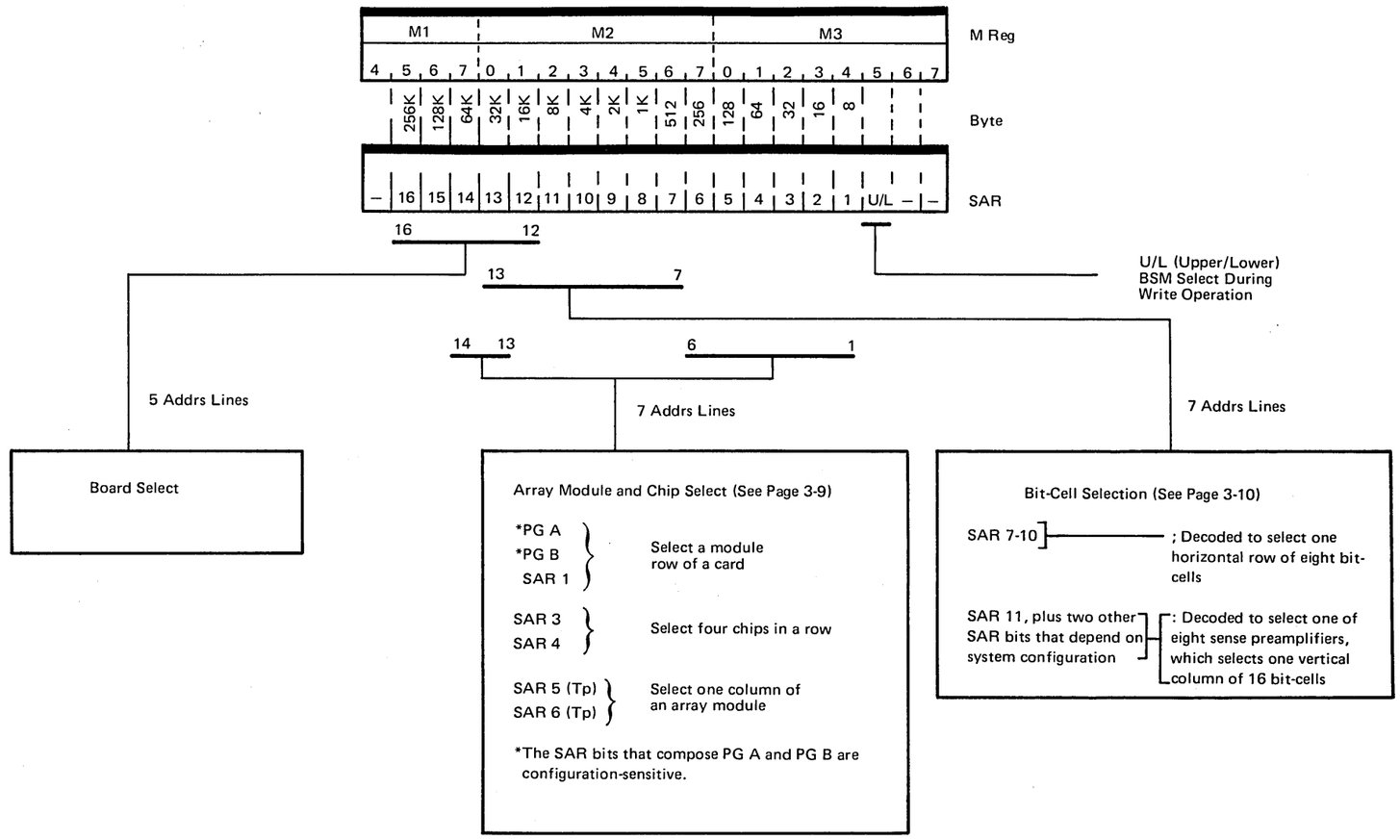
Details pertaining to the addressing scheme are covered on Page 3-15.

MAIN-STORAGE SELECTION

Main-storage addressing values are specified by program instructions. These addresses are three bytes long. The four high-order bits, however, are not used for the actual addressing. (Sixteen address bits provide for addressing up to 512K byte locations, the largest storage size possible in this system). Main-storage addresses are not modified but are sent to the memory-addressing circuitry unchanged from the M-register to the storage address register. If a storage address greater than the assigned main-storage capacity of the system is specified, an address check occurs.

CONTROL-STORAGE SELECTION

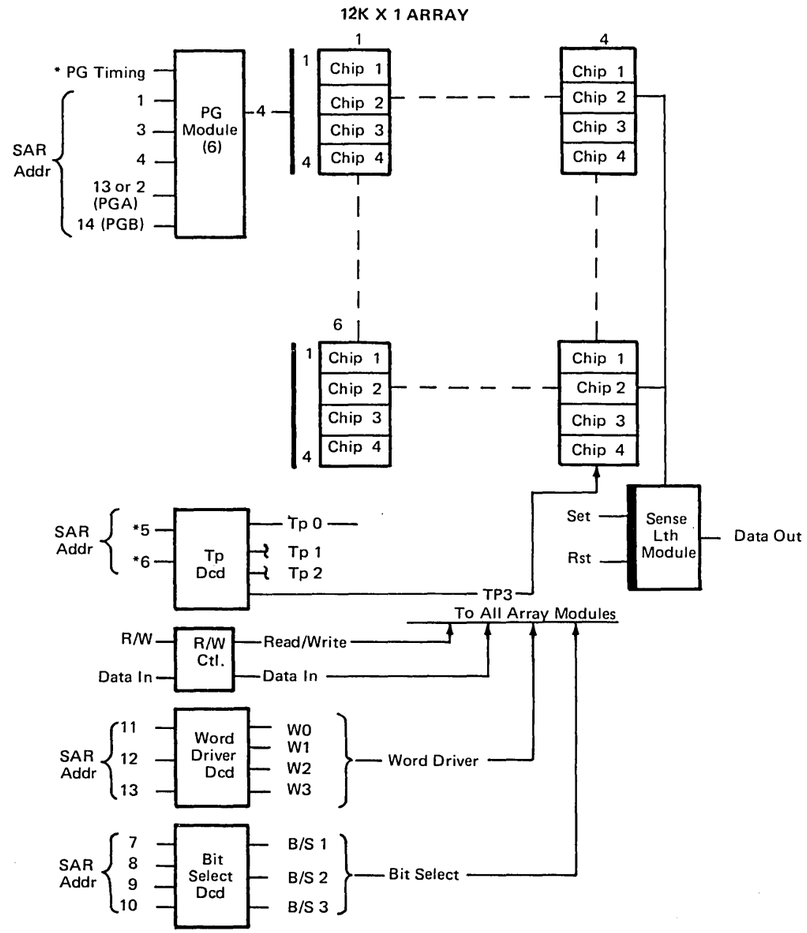
Control-storage selection and addressing is generated within the CPU. When control storage is selected, a control store gate forces selection of only the A2/A4 boards. (These boards always contain the control-storage data.)



ARRAY MODULE and CHIP SELECTION-SIMPLIFIED

All addresses and control inputs to the array card are from the address buffer card. The 'data in' line is from the BSM board interface.

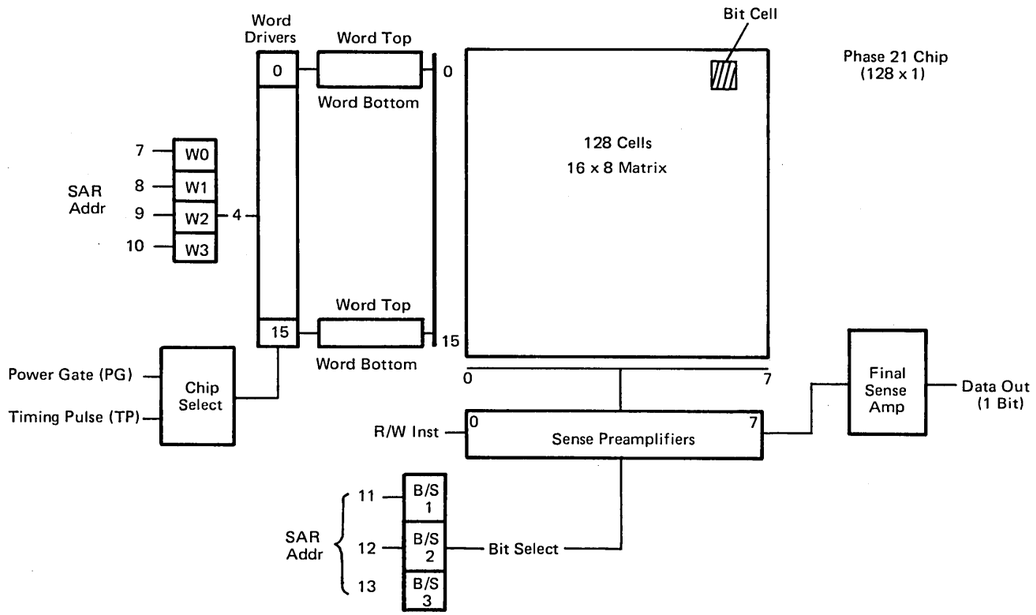
- Power gate signals select one chip in each of the four array modules in a horizontal row.
- Each TP signal is applied to all chips in the four modules in a vertical column.
- The selected chip is the one that receives coincident PG and TP signals.
- One sense latch is used on 12K x 1 slow cards. 6K x 2, and 12K x 1 fast cards each have two sense latches.



*These are the timing lines that control writing and reading on each array card; they are generated on the address buffer card.

Note: The SAR bits shown on this diagram is intended for instructional use only; the actual use of the SAR bits depends upon the main-storage size. The decode of the SAR bits and their use for each configuration is contained on 3-15.

BIT-CELL OPERATION and SELECTION-SIMPLIFIED



Note: The use of the SAR bits shown on this diagram are intended for instructional use only; the actual use of the SAR bits depends upon the main-storage size. The decode of the SAR bits and their use for each configuration is contained on Page 3-15.

- A bit cell is selected when the Word Top (WT) and the Word Bottom (WB) signals coincide.
- The coincidence of the WT and WB signals produces a read 0 or read 1 signal on the sense 0 or sense 1 line during a read.
- The coincidence of WT, WB, and a write signal on the write 0 or write 1 line is required to write a bit into the bit cell.
- The bit cell is the basic storage element used in the phase 2I monolithic memory; it stores one data bit.
- The bit cell is a direct-coupled flip flop that can be set to a 0 or 1 state.
- The coincidence of TP and PG signals select the chip.
- The write signal gates 'data in' through the selected sense pre-amp and activates the write 1 or write 0 line.
- The Absence of a write signal places the chip in read status.
- Word lines (SAR addresses 7, 8, 9, and 10) are decoded to select one horizontal row of eight bit-cells. The selected bit cells are in high-power state. (The bit cell is selected to read or write.)
- Bit select (addresses 11, 12, and 13) are decoded to select one of eight sense preamplifiers and thus one vertical column of 16 bit cells.

BSM/CPU INTERFACE

The main storage and control storage are independent storages located in the 01 frame, B gate and communicate with the CPU via the error-correction and control logic (ECCL) board. Each BSM contains an address buffer card that powers and inverts address lines, and generates the timing signals required by each array card during a fetch or store cycle. The ECCL board contains the storage data register (SDR), storage address register (SAR), BSM clocks, and the logic for error location and correction.

The data and signals transmitted on the CPU/ECCL board, and ECCL/BSM interfaces include: addresses, control information, data, status information, and miscellaneous signals necessary for a successful operation, and synchronization with the CPU. The illustration shows the overall interface between the BSM and the other functional areas. The connecting lines do not represent actual cabling; they represent logical sets of interface lines.

ECCL to Storage Interface

Data to Storage

These lines (64) transfer data to storage on a store cycle.

Check to Storage

Based on the data, these lines (8) are generated to enable detecting and correcting single-bit errors.

Address

The address lines (28) are divided so that half go to the upper storage bound, and half to the lower.

Board Select

The board select lines (5) are decoded to select a particular pair of 24K or 48K boards.

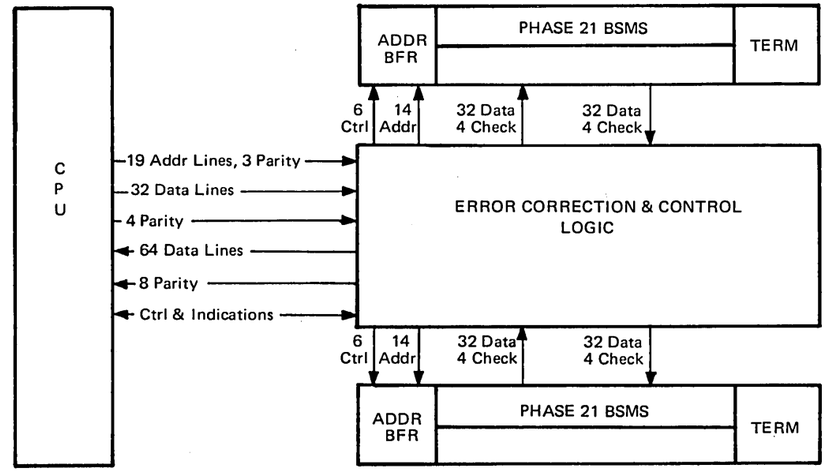
Storage to ECCL Interface

Data from Storage

These lines (64) transfer data.

Check from Storage

The check lines (8) contain single-error-correction information.



BSM FUNCTIONS

- Receives and sends only when signaled by the processor.
- Two general kinds of storage accesses are performed:

  fetching control words;
  fetching (or storing) data from (or into) main or control storage.
- The storage cycle for a fetch consists only of a readout function.
- The storage cycle for a store operation consist of a fetch followed by a store.
- Error correction is performed on all storage functions.

Main storage and control storage provide the system with directly addressable fast-access storage of data. Because the storage elements of the BSMs are monolithic circuits, the storage initially does not contain information when power is turned on. Both data and programs must be loaded into main and control storage (from the CPU) before they can be processed.

To fetch (read-out) information or store (write) information, the system initiates a storage operation. When not performing either function, the storage unit is idle.

Main and Control-Storage Areas

At least 32K bytes of control storage, and up to 256K bytes of program storage, are housed in the CPU. For systems that have a storage capacity above 256K, they require a main storage frame (3345).

The system is equipped with a movable control-storage boundary that allows for up to 64K bytes of control storage, depending upon the mix of features installed for the system configuration. This additional control-storage requirement is at the expense of main storage. The storage boundary is determined when the microprogram is compiled.

Control storage contains the microprogram upon which all system operations depend. Control storage is not available to the user and should not be modified. To do so could make application results unpredictable.

Application-program data is read out of (or into) storage four bytes (a fullword) at a time. To increase performance, however, program instructions are read out eight bytes (a doubleword) at a time.

Error checking and correction (ECC) provides automatic single-bit error detection and correction. It also detects all double-bit errors and most multiple-bit storage errors but does not correct them. Parity checking is used to verify proper data transfers to the storage unit.

Main-storage addressing begins at location 0 and continues up through the highest installed program-storage byte location.

BSM/ECCL Data Flow

There are two operations in storage: fetch and store. All controls, data, and the select pulse come from the CPU, are gated through the ECCL board, and distributed to the proper BSMs. The ECCL board, which has all the controls necessary to communicate between the CPU and the internal storage BSMs, handles all fetch and store operations, as well as error correction and control. The overall relationship and logical interface among the major functional units of the storage area is shown on Page 3-13.

Major Functional Areas

Beside the BSMs, the storage area is composed of an ECCL board that contains storage data register (SDR); storage address register (SAR); BSM clock; delay lines; and the error-correction and control logic. The logic consists of read generator, write generator, syndrome generator, syndrome decoded, parity-out generator, and error-type decoder. Each functional unit is briefly described as follows:

STORAGE DATA REGISTER (SDR)

The SDR buffers and directs the data between the CPU and storage. During a fetch operation, data is received from storage, latched, then sent to the read generator and the CPU as uncorrected data. Some time later, decoded syndromes are used to determine which data bit, if any, is in error. This bit is corrected, relatched, and sent to the CPU as corrected data. On a store operation, the foregoing is repeated, and the SDR gates either new data or old data to the write generator and to storage.

STORAGE ADDRESS REGISTER (SAR) AND STORAGE CLOCK

The SAR logic provides the write instruction, busy signal, 14 address lines, BSM select timing, and the address check.

A write signal is generated if the instruction is received from the CPU and a cancel has not been initiated. The instruction is then timed and latched. The latched signal is split into a write upper and a write lower instruction which are sent to corresponding pairs of storage boards. In a diagnostic mode either the upper or the lower instruction is active.

A busy signal is initiated upon receiving a select from the CPU and is active until the end of the storage cycle that it was activated on. This line is sent to the system.

Up to 17 address lines are received from the CPU and latched. Both phases of most lines are distributed to the storage boards, depending on the configuration. The address check logic compares 17 addresses and 3 parity lines and determines whether an address error has occurred. The result is sent to the system and to the logic cancel circuitry.

BSM CLOCK

The BSM clock generates the BSM set and reset functions that are sent to storage. Address lines in conjunction with a select signal from the CPU are used to determine which BSMs are to be selected.

DELAY LINE

The delay line is approximately 280 ns long and tappable in steps of 10 ns and is used basically to generate control pulses.

READ OR WRITE GENERATOR

Each generator receives 64 data bits and creates the eight check bits and eight parity bits.

SYNDROME GENERATOR

The syndrome generator card contains logic to generate syndromes and hardware check, and provides check bits to storage.

Check bits from storage are compared to the check bits from the read generator. This output is latched and sent to decoder logic as syndromes.

Check bits from the write generator are gated to storage if the diagnostic parity mode is inactive. Parity bits from the system are gated to storage during diagnostic parity mode. Hardware Check is a compare of the check bits from the read generator and write generator. This check is valid only on a fetch instruction. A late bit also brings up Hardware Check.

SYNDROME DECODER

The syndromes are decoded into byte error lines and bit-position lines. These lines are sent to the SDR card, where the last level of decode is performed for bit correction.

PARITY-OUT GENERATOR

The parity-out generator provides the double-error detecting (DED) bit syndromes, correct parity to the system, and byte parity and data in parity checks.

The generation of the DED syndrome bit (CT) on a read cycle is accomplished by exclusive ORing the parity bits from the read check bit generator with the check bits from storage.

Data parity from the read check bit generator are corrected and gated to the CPU. In diagnostic mode, the check bits from storage are gated to the CPU instead.

Data-parity bits from the write generator are compared with data-parity bits from the CPU for data-transmission errors.

The byte lines received from storage are latched and distributed to the control logic within the SDR and syndrome decodes. The byte lines are parity-checked.

ERROR-TYPE DECODER

The syndromes are used to determine which byte contains a failing bit. This information is then used to complement the corresponding parity bit that was generated on erroneous data. Corrected byte parity is then sent to the system. Also, if the failing bit is in a byte that is to be restored in storage, the syndromes are used to complement some of the check bits to be stored. This eliminates the time needed to regenerate corrected bits based on the new updated word for storage. If a bit has failed, the error type and position will be indicated by lines sent to the CPU.

Fetch Operation in the Storage Area

A fetch operation is started when the CPU sends to the storage area a select pulse and an address. The fetch instruction along with the select pulse starts the clock located on the ECCL board. This operation allows the selected input address to be latched and timed. The address lines are then decoded to select a data word from an upper BSM (located at A2, B2, or C2) and a data word from a lower BSM (located at A4, B4, or C4).

Sixty-four data bits, seven check bits, and a double-error detection bit (CT) are accessed from storage. The 64 bits are latched in the SDR; the check bits plus the CT bits are transferred to the syndrome generator. The 64 data bits are forwarded to the system via the SDBO initially as uncorrected data but are prevented from being used by the CPU until it has been verified that the data contains no errors. The data is also gated to both the read generator and write generator. (On a fetch, the write generator receives only the data read out of storage.) The generators produce 7 check bits and 8 parity bits; the generated check bits are compared with the check bits read from storage. The output from this compare is then sent to the syndrome generator, where the check bits are analyzed to determine whether any of the bits issued from storage are in error. The syndrome generator (error-bit decoder circuits) determines which data is in error, if any, and sends these signals as syndromes to the syndrome decoder. The syndrome decoder contains circuits that complement the early data latched in the SDR if any error was detected. This corrected bit is then transferred to the CPU. The CPU does not operate on any data until it is assured that all data from storage is valid.

If a single-bit error arises during a control-storage access, the bit in error is corrected before it is restored into storage.

The eight generated parity bits are transferred to the parity-out generator, which normally sends these parity bits to the CPU. The parity bits can be replaced by check bits during a diagnostic operation.

The check bits from the syndrome decoder are also forwarded as check bit syndromes to the error-type decoder. If an error is detected in any of the bytes, the error-type decoder uses the check bit syndromes to correct the early parity bit associated with the byte that contained the error. Corrected parity is forwarded to the CPU via the parity-out generator. The error-type decoder also provides the CPU with signal lines that indicate the error type. The error-type signals are displayed on the console.

Usually, control-storage data is available to the CPU on the cycle following the request for data. If corrections to any data bit are required, an additional CPU cycle is necessary. Main-storage data becomes available to the CPU approximately 237 nanoseconds after the ECCL board receives the request.

Store Operation in the Storage Area

The store operation requires a select pulse, store instruction, address, data bits, and storage byte control lines. The select pulse and the storage instruction initiate the store operation. The store operation always starts with a read followed by a write. Thus, as the store operation commences, 32 data bits are gated to a set of latches in the SDR from the CPU; the 64 data bits from the same address location at which the data is to be stored are gated to another set of latches in the SDR. A decision is made in the SDR based on the particular bytes to be stored as to what new information is to be stored and what old information is to be retained. Sixty-four data bits (32 old and 32 new) are then transferred from the SDR to the write generator. (The read generator receives the 64 data bits from storage.) The write generator produces 7 check bits, one double-error detection (CT) bit, and 8 parity bits on the new data combination to be stored.

The generated parity bits on the new data from the system are compared with the parity-in bits; and if an error has occurred, a byte parity signal is issued to the cancel circuit. The generated check is forwarded to the syndrome generator, where it is latched. At the syndrome generator, a decision is made whether the check bits or the data parity-in bits from the system are to be stored. (Data-parity bits can replace the check bits during diagnostic tests.)

For the data that is to be restored, under normal operations, the syndrome generator determines whether the data read from storage and the data that is to be restored is the same. If all data bits compare, the check bits are gated to storage unaltered. If a bit error is detected in a byte to be restored during a main-storage access, the check bits associated with the byte are complemented before being transferred to storage. Similarly the CT bit is latched and can be updated, if necessary, before being transferred to storage.

If a single-bit error is detected during a control-storage access, the bit in error is corrected before it is restored. This data-bit correction eliminates the check-bit correction cycle.

When the storage is initially loaded, a validate line is issued to the ECCL board. The validate line allows you to write data into storage with the correct check bits.

STORAGE PROTECTION

Storage protection, composed of the store and fetch protection features, prevents the unauthorized changing or use of the contents of main storage. *Store protection* prevents the contents of main storage from being altered by storage addressing errors in programs or input from I/O devices. *Fetch protection* prevents the unauthorized fetching of data and instructions from main storage. As many as 15 programs (with associated main-storage areas) can be protected at one time.

Protection is achieved by dividing main storage into 2,048-byte blocks and by associating a storage key with each block. Each storage key may be thought of as a lock. Each block of storage, then, has its own lock. Two instructions are provided for assigning and inspecting the key, which contains a four-bit code. The same code may be used by many blocks, using binary codes 0001-1111.

A right of access to storage is identified by a protection key, located in the program status word (PSW) or in a special word used in channel operations. During a main-storage reference (storing or fetching), the storage key is compared with the protection key associated with the reference. Access to the location is granted only when the four leftmost (high-order) bits of the storage key match the protection key, or when the protection key is zero (0000). When both the store- and fetch-protection features are installed, bit 4 of the storage key determines whether fetch protection is operative for the storage block associated with that key. If the bit is 1, fetch protection is operative; if it is 0, it is inoperative.

The storage protect unit has a 64 x 8 protection stack that applies to main-storage locations (in sequential block of 2,048 bytes) zero through 131,072. Additional stacks are provided in the CPU when main-storage capacity exceeds 131,072 bytes. The operation of the storage protect unit is described on page 3-41.

ERROR HANDLING-STORAGE AREA

The handling of errors for the storage area is generally divided into these categories:

- Hardware error-detection system that signals a failure condition.
- Error logging that allows you to retain or display through hard copy pertinent information on failure conditions and,
- Diagnostics that allow rapid analysis and isolation of a failure.

Error Detection

The error-detection circuits are divided into two kinds: control and data. Control errors are those relating to controlling functions such as storage addressing. Data errors are detected by parity checks from the data bus line, and by the error-checking circuits within the storage area. The data path to storage is parity-checked, byte by byte. Both control and data error conditions are displayed on the console.

The error-checking mechanism within the storage area has the ability to detect and correct single-bit errors that occur when data is fetched from storage without disturbing normal operations. If a BSM loses a bit or picks up an extra bit, the error-correction logic detects the error and corrects it before allowing the CPU to use the data. If the error involves more than one bit position, the error will be detected but not corrected. Detection of double-bit failures signals the system to perform the normal retry routine. The indication of double-bit errors, which includes one intermittent bit failure, is removed as a result of the CPU retry routine. In addition, checks are provided on the error-correction circuits.

Parity checking within the storage area is provided for:

- Storage addresses
- Data on SDBI during a store operation
- Command sent to storage
- Byte lines (that determine bytes to be stored)
- Storage protection stack key

Error Logging

Error logout messages are retained in dedicated maintenance storage for later recall. Storing this information into main storage as a result of, or in association with, a machine error is referred to as a machine-check logout. The purpose of these messages is to give you information on intermittent and/or solid failures.

Diagnostics

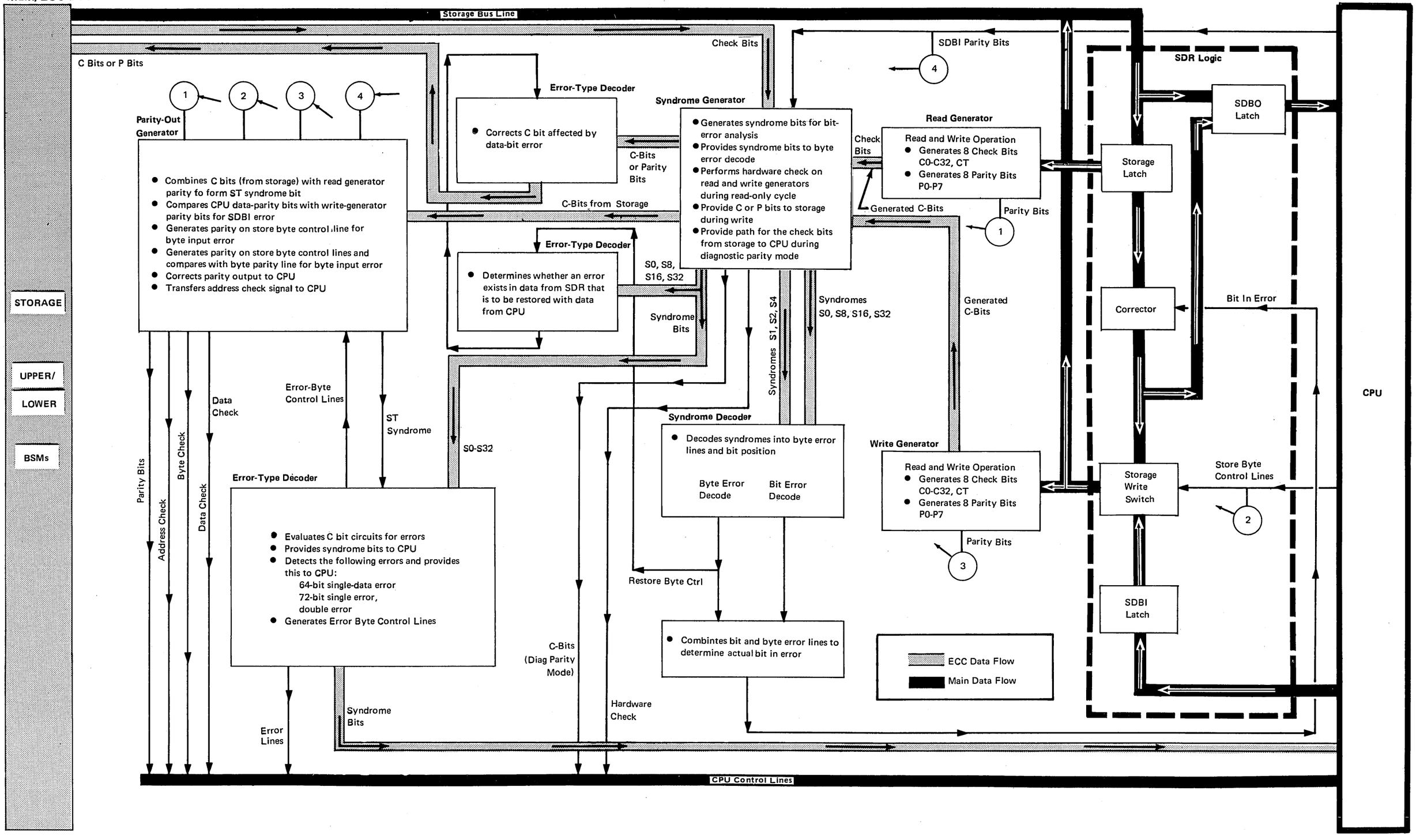
The diagnostics that check the main- and control-storage areas are part of the Basic Group (Microprogram *BAS). The purpose of the group of tests is to detect and locate failures in the basic Model 145 and Console File Attachment controls. This group enables the sections in the extended group to operate under a micromonitor and use the console typewriter as a communication device.

The console file enables initiating this group of tests with a building-block technique from a minimal entity; the console-file attachment circuitry. Initially, control words are loaded directly in the C-register and executed under control of the file. Then control words are loaded into local storage and executed in local store-control store mode. Finally, control storage itself is loaded with the remaining basic group sections and executed under normal CPU control.

The basic CPU groups consist of 12 sections, one of which tests the main and control area and includes these tests:

Test BGA1	Set ACB and test address bits 14-17.
Test BGA2	Test phase 21 address bits 1-13.
Test BGA3	Locate error-free word.
Test BGA4	Write generator test.
Test BGA5	Read generator test.
Test BGA6	C0 - C32 error detect.
Test BGA7	CT and double-error detect.
Test BGA8	Single-error detect and correct.
Test BGA9	Any double errors in control storage.
Test BGB0	Any double errors in program storage.
Test BGB1	Control-storage and program-storage address test.
Test BGB2	Test data for read/write generator test.

NOTE: The extended group of microdiagnostics run under micromonitor EASY, MBA0 through MBA8, comprises the extended group. (See "Microdiagnostic Servicing Handbook" for operating procedures, Part 2641601.)


STORAGE
UPPER/
LOWER
BSMs
Storage Bus Line
C Bits or P Bits
Check Bits
SDBI Parity Bits
SDR Logic
Error-Type Decoder
Syndrome Generator
Parity-Out Generator
Corrects C bit affected by data-bit error
C-Bits or Parity Bits
Generates syndrome bits for bit-error analysis
Provides syndrome bits to byte error decode
Performs hardware check on read and write generators during read-only cycle
Provide C or P bits to storage during write
Provide path for the check bits from storage to CPU during diagnostic parity mode
Read Generator
Read and Write Operation
Generates 8 Check Bits C0-C32, CT
Generates 8 Parity Bits P0-P7
Check Bits
Parity Bits
Generated C-Bits
SDBO Latch
Storage Latch
Corrector
Bit In Error
CPU
Combines C bits (from storage) with read generator parity fo form ST syndrome bit
Compares CPU data-parity bits with write-generator parity bits for SDBI error
Generates parity on store byte control line for byte input error
Generates parity on store byte control lines and compares with byte parity line for byte input error
Corrects parity output to CPU
Transfers address check signal to CPU
C-Bits from Storage
Error-Type Decoder
Determines whether an error exists in data from SDR that is to be restored with data from CPU
S0, S8, S16, S32
Syndrome Bits
Syndromes S1, S2, S4
Syndromes S0, S8, S16, S32
Generated C-Bits
Data Check
Error-Byte Control Lines
ST Syndrome
S0-S32
Syndrome Decoder
Decodes syndromes into byte error lines and bit position
Byte Error Decode
Bit Error Decode
Write Generator
Read and Write Operation
Generates 8 Check Bits C0-C32, CT
Generates 8 Parity Bits P0-P7
Parity Bits
Storage Write Switch
Store Byte Control Lines
Parity Bits
Byte Check
Address Check
Data Check
Error-Type Decoder
Evaluates C bit circuits for errors
Provides syndrome bits to CPU
Detects the following errors and provides this to CPU:
64-bit single-data error
72-bit single error, double error
Generates Error Byte Control Lines
Restore Byte Ctrl
SDBI Latch
C-Bits (Diag Parity Mode)
Combintes bit and byte error lines to determine actual bit in error
ECC Data Flow
Main Data Flow
Syndrome Bits
Hardware Check
Error Lines
CPU Control Lines

FUNCTIONAL AREAS

The main and control area is structured around 2 basic BSMs (12K x 1 and 6K x 2), and an ECCL board. The storage area may consist of both the 12K x 1 and 6K x 2 BSMs or just the 12K x 1 BSMs, depending on the requirement of a particular system. The BSM configuration shown on Page 3-5 illustrates the data, and address and control flow among the BSMs and the ECCL board. The relative location of the BSMs shown on this diagram are viewed from the wiring side of the logic gate.

This chapter discusses storage addressing and timing, BSM interface connectors, and the functional units that compose the ECCL board. The ECCL board layout is shown on Page 3-17.

STORAGE ADDRESS REGISTER and STORAGE CLOCK

The storage address register (SAR) is a 17-bit register that accepts the CPU storage addresses from the M-register and is used for both control- and main-storage accesses. In addition, the SAR logic contains the write control, busy signal, the BSM select timing and the addresses check logic. (See Page 3-8.)

The address contained in the M-register is gated to the SAR by a select powered gate control that is generated as a result of the select command received from the CPU.

The CPU addresses are gated to the SAR under control of the CPU select control line, which is received at the ECCL board after it is generated. This select line can be delayed for an additional 22 ns during the storage I cycle for certain types of CPU operations before it is used in the SAR logic. The width of the select line is chopped from 45 ns to 38 ns by a time delay circuit, forming a select powered control. Each time the select powered control line is active, the contents of the M-register are gated and latched in the SAR. The addresses from the SAR are decoded to select the proper BSM.

The SAR is reset each time the select powered line is degated. A write or read signal is generated each time an instruction is received from the system if a cancel signal is not active. A write signal is also generated if a restore signal is activt and a single error has occurred during control store access. (See Page 3-8.) The write signal is latched in the system write latch at approximately T_0 + 80 ns (T_0 is defined as the time the select powered signal becomes active). The latched signal is forwarded to the storage write latch. The latched signal from the storage write latch is split into a write upper and a write lower instruction which are sent to corresponding pairs of storage BSM.

The busy signal is initiated by the select powered control signal and is active until the end of the cycle on which it was activated. This line is sent to the system.

The BSM timing control is gated to storage clock controls for proper BSM select. The BSM select line starts the timing pulse generator on each address buffer card.

SAR Address Checking

All storage addresses that are gated to the SAR are parity-checked for addressing errors as shown on ALD SQ403. An address error sets an address check latch. This signal is sent to the CPU (MCK B0, bit 0), and also sets the ON cancel latch in the SAR logic.

BSM ADDRESSING

The SAR bits are decoded to select the BSM boards, array module and chip, and the proper bit cell. (See Page 3-8.) In addition, certain address bits are manipulated within the storage area to assure that only the valid portions of an array card are addressed. The BSM address bits (PGA, PGB, chip, and on occasion, address 12) are generated by decode circuits within the SAR card or storage clock located on the ECCL board. The use of these bits depends upon the storage configuration and may not always represent the same SAR bits as they are routed from board to board. Address lines 1, 3, 4, 5, 7 through 11, along with the control lines, are propagated to each board. The address lines are routed to an address buffer card located on each BSM, where they are powered and inverted to provide true and complement addresses to the array cards.

For each storage access, the addresses are sent to corresponding pairs of BSMs. Address lines 1-14 are used on 48K (12 x 36) BSMs; address 6 is not used on 24K (6K x 36) BSMs.

Board Select

Byte locations in storage are consecutively numbered starting with 0; each number is considered the address of the corresponding byte. A group of bytes in storage is addressed by the leftmost byte of the group. The number of bytes in a group is either implied or explicitly defined by the operation. The addresses in the M-register represent the byte locations. The starting address is always located at the board farthest from the ECCL board. For example, in a 256K storage configuration, the low-order address is located in the C2/C4 boards.

Board select circuits are divided into 32K byte or 64K byte selection depending upon the capacity of each board used with a particular storage configuration. Board select is determined by the decode of bits 12 through 16 of SAR or by the control store gate that forces selection of only the A2/A4 board. (These boards contain the control-storage data.) The SAR bits 12 through 16 represent the following byte-address ranges:

Bit	*Range (Bytes)*
12	16K to 32K
13	32K to 64K
14	64K to 128K
15	128K to 256K

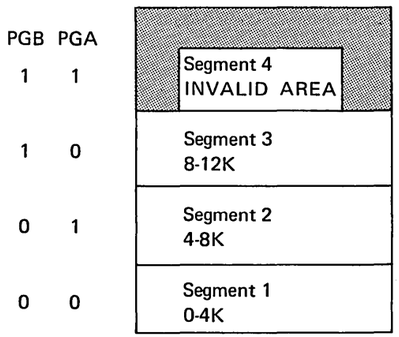
The board select signal is received at the BSM approximately T_0 + 9 ns for main storage; at T_0 + 6 ns for control storage. When the board select signal reaches the address buffer card of the BSM, T_0 of that BSM commences.

Addressing Circuits

The address bits needed to generate the special address lines PGA, PGB, CHIP, and control depend upon storage configuration and also board location. (See Page 3-21.) The selection of what bits are used for PGA, PGB, and CHIP are chosen to match the address range of an array card in a particular BSM. To minimize the wiring between the ECCL board and the BSMs, a chaining technique is used to propagate the special address lines from one board to the next. Thus, where the address CHIP line is used on one board, the same address bit may be propagated to another board and be used as a PGB signal. The addressing scheme also makes use of the special address bits to assure that a false invalid address indication is not generated as the CPU address count is incremented beyond the capacity of an array card.

Array Card Logical Addressing

All BSMs within the storage area have 12K addresses regardless of card size. To address the 12K card that is not on a binary boundary, the card has been effectively subdivided into four segments, with each segment containing 4K addresses. The address control lines PGA and PGB are used to keep track of segment location within the card. PGA selects the 4K segments, and PGB selects upper/lower segment of a card.



Because each segment represents 4K addresses, only three segments are required to address the card properly, leaving one segment in an invalid area. This invalid area is detected by a simultaneous decode of 1, 1 for PGA and PGB. If PGA and PGB are active at the same time at any one BSM, this means that the address exceeded the BSM capacity. When this happens, a double-bit error signal is transmitted to the machine-check circuitry.

To avoid addressing the invalid segment of the array card whenever the CPU addressing input reaches the address capacity of the card (PGA and PGB both equal to a logical I), either PGA or PGB is inverted in the SAR logic before the bits are transferred to storage. The bit that is inverted depends upon the addressing requirements of a particular BSM. The inversion of either PGA and PGB causes the selection of the BSM with the next high-order address. For example, if the CPU were sequentially addressing a 256K storage starting with address 0 at the C2/C4 board, the B2/B4 boards are selected when PGA (address bit 13) and PGB (address bit 14) become active simultaneously. When the SAR logic decodes this condition, PGA bit (bit 13 in this example) is inverted via a cable before it is used in storage. The 256K address flow (ALD SQ011) shows the addressing interface connections among the BSMs and the ECCL board. In the 24K BSMs (used in 112K and 208K storage configurations), the PGA signal becomes address 12 so that proper byte boundaries can be maintained for the 24K storage.

The CHIP control line is gated to every module and is used as one of the seven address lines needed for bit-cell selection.

The control address line is used as a gating control line for the PGB and CHIP address lines whenever the CPU accesses the control-storage area. This gating action effectively changes control-storage locations on the array card.

Array Module and Chip Selects

The array module and chip in an array card are selected as follows:

PGA, PGB, and SAR 1 select the module row of a card. The SAR bits that compose PGA and PGB are shown on Page 3-21.

SAR bits 3 and 4 select one chip in each of the array modules in a horizontal row.

SAR bits 5 (T_p) and 6 (T_p) select one column of an array module.

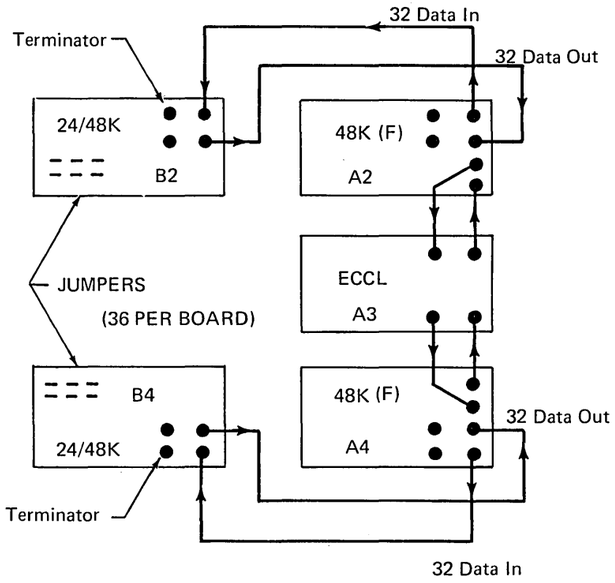
The PGA, PGB, address 1, along with address bits 3 and 4, combine to select the module row of a card being addressed and select one chip in each of the four modules in that horizontal row. These addresses form the power gate select lines (output of power gate drivers) and are controlled by the power gate timing signal generated at the address buffer card. Address 5 (T_p) and 6 (T_p) select one column of any array module. The proper column within an array card is selected by a decode network that makes use of both phases of address 5 and 6. Address 5 is timed and is derived off the delay lines in the address buffer card. This is the T_p pulse. The selected chip within an array module is the one that receives coincident PG and T_p signals. A simplified drawing showing array module and chip is shown on Page 3-6. When observing the address 5 timing pulse on the array card at pin D06 or B10, you will notice that only one of these pulses is timed. The address 5 signal that is timed is dependent upon the input polarity of address 5. The (-) 5 address is the timed pulse. So, if a (+) 5 address is generated, the out-of-phase signal at pin D06 has the timing pulse; when a (-) 5 address signal is generated, the in-phase signal at pin B10 has the timing pulse.

The PG and T_p timing signals are adjusted at the address buffer card. (See maintenance section.)

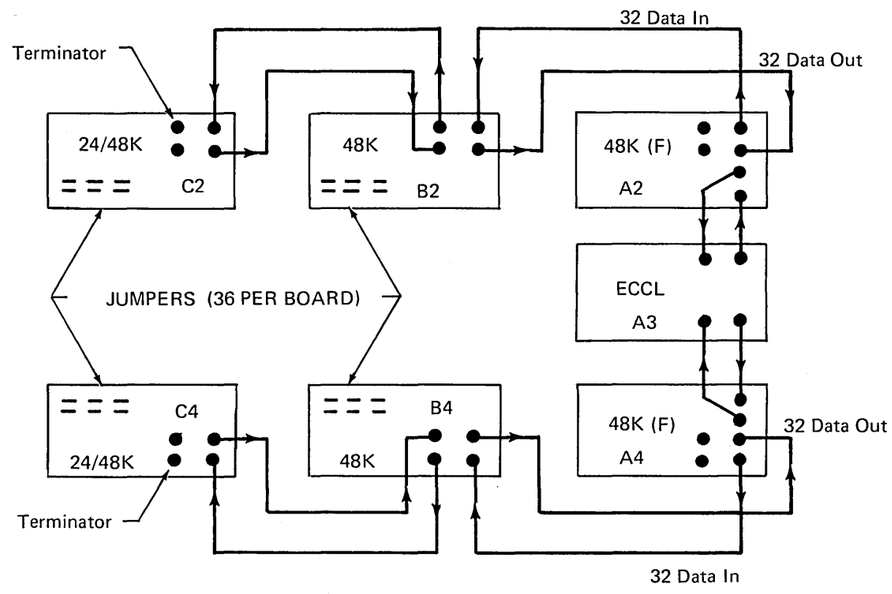
BSM Configuration-Address and Control Interface

Data Interface

112K/160K BSM Storage Size

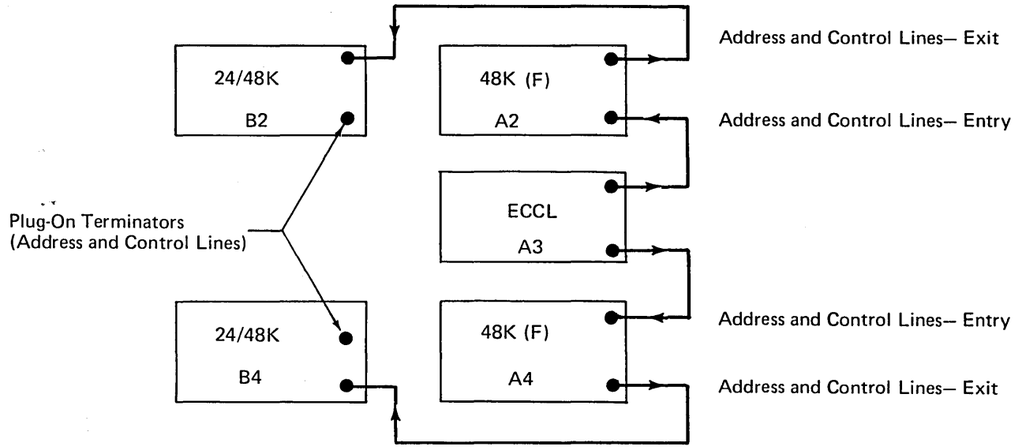


208K/256K BSM Storage Size

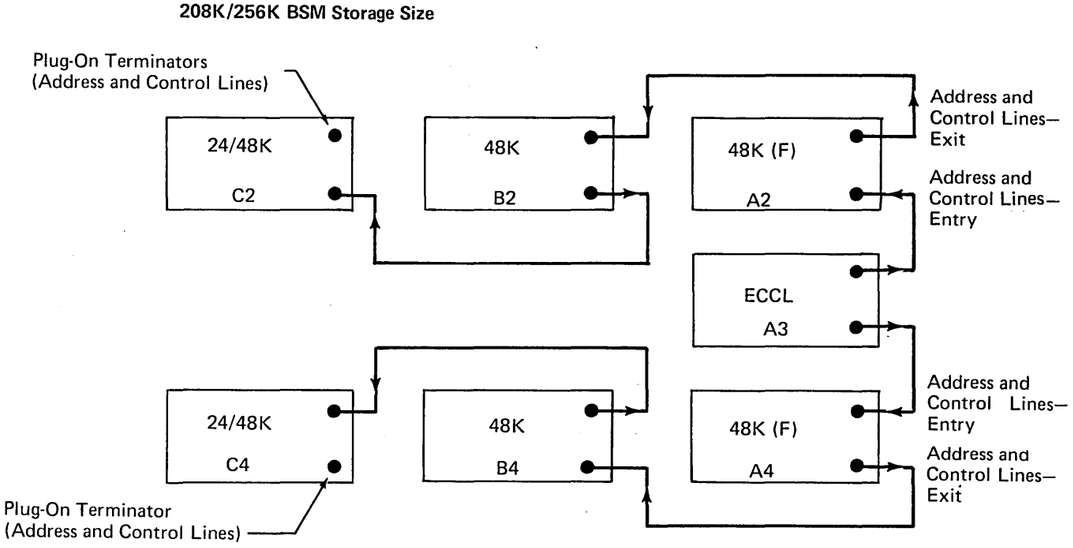


Address Interface

112K/160K BSM Storage Size



208K/256K BSM Storage Size



ECCL Board Layout

V	U	T	S	R	Q	P	N	M	L	K	J	H	G	F	E	D	C	B	A
		1862 SQ232	1862 SQ226	1862 SQ244	1862 SQ238	6031 SQ261 SQ262	1860 SQ301 SQ273	1863 SQ271 SQ273	1861 SQ290 SQ292	1859 SQ280 SQ287	6031 SQ251 SQ257	1862 SQ208	1862 SQ202	1862 SQ220	1862 SQ214	1865 SQ312	1864 SQ401 SQ404	1866 SQ407 SQ410	
DCPL		SDR Bits	SDR Bits	SDR Bits	SDR Bits							SDR Bits	SDR Bits	SDR Bits	SDR Bits				DCPL
		0 1 32 33	8 9 40 41	16 17 48 49	24 25 56 57			Synd Gen # 1				4 5 36 37	12 13 44 45	20 21 52 53	28 29 60 61	Delay Line # 1			
		1862 SQ235	1862 SQ229	1862 SQ247	1862 SQ241			1863 SQ275 SQ277				1862 SQ211	1862 SQ205	1862 SQ223	1862 SQ217	1865 SQ315			
DCPL		SDR Bits	SDR Bits	SDR Bits	SDR Bits			Synd Gen #2				SDR Bits	SDR Bits	SDR Bits	SDR Bits				DCPL
		2 3 34 35	10 11 42 43	18 19 50 51	26 27 58 59							6 7 38 39	14 15 46 47	22 23 54 55	30 31 62 63	Delay Line #2			
	Basic System Card					Read Check Bit Gen	Parity Out Gen		Error Type Decoder	Syndrome Decoder	Write Check Bit Gen						Storage Address Reg	BSM Clock	

SDBI

2/Card
0-31
J05
J13

SDR (EFGH-QRST) 2/4 P/N 1862 (All Cards)

Data In From Stg	Data Out To System
B04	B09
J04	D05
J10	D09
J12	D10

Parity Out Gen (N)

Bytes Latched -Level	0	S02	Add Ck S08
	1	M13	Data Ck S04
	2	M10	Byte Ck U07
	3	P10	(-Levels)

CHECK BIT GENS

RD/WR	(P&J)
C0	M08
C1	D13
C2	S03
C4	P04
C8	M04
C16	G09
C32	G08
CT	U12
+=1	-=0

Data all 1's CK bits all 1's
data all 0's CK bits 00111100

SYND GEN # 1 AND # 2 (M)

+Hdware CK D11 2/4
+ On Synd B07

ERROR TYPE DECODER (L)

-Sing Data Err	U02
-72 Bit Err	P10
-Double Err	J13
-On Cancel	Q11
+Disable	M04

DELAY # 1 D2

0 Time D2D12	Clock Start
50 Time B10	Bsm Reset (Ready)
70 Time B13	Reset Syndromes
80 Time J05	Set Sys Write
100 Time G04	Rst Data Good

STG ADD REG (C) SAR

-BSM Sel Time	B12
-Write From (Sys)	B07
-Select	D02
-Write Upper	B05
-Write Lower	B04

DELAY #2 D4

170 Time B08 Set Synd
180 Time D09 Start Fast Write
210 Time B13 Start Write
275 Time G08 Reset Bsm (Write)
275 Time G07 Write Width

BSM CLOCK (B)

-Ctrl Store	J13
+Diag Parity	P06
+Validate	P07
+Diag Ripple	P07
+On Early Error	D07M13
-Set For Access	B10
-Wr Width Delay	P12
+Maintain Data	M07
-Restore	U05

Bit-Cell Selection

Seven address lines are used to select one of 128 bit cells that form a 8 x 16 matrix on each chip. The address lines are SAR bits 7-11, plus two special address control bits CHIP and either SAR bits 12 or 13. The decode of SAR bits 7-10 select (via word driver) a horizontal row composed of eight bit-cells. The decode of SAR bit 11, CHIP and either SAR bits 12 or 13, depending upon system configuration, select one of eight sense preamplifiers that activate on a vertical column containing 16 bit-cells. The selected preamplifier either gates a write signal (data-in), which activates the write 1 or write 0 line during a write cycle; or gates a read signal, which is generated on the sense 0 and sense 1 line of each bit cell during a read operation. Coincidence of the selected word driver output and the selected preamplifier output selects the bit cell that is to be used in the write or read operation. A simplified drawing of bit-cell selection is shown on Page 3-10.

Address Control Lines

The following address control lines are gated to the BSMs.

SIGNAL NAME	ECCL EXIT PIN	BSM ENTRY OR EXIT PIN	FUNCTION
-BSM Select	(See Addr FlowALDs SQ008 SQ011)		Initiates each store or fetch cycle by starting the timing pulse generator on the address buffer card. A separate 'BSM select' signal is used for each BSM.
+Board Activity	See SQ005		This is an output signal; this line is + when the BSM is selected. Its use is optional.
-Write	A3D04	A2B10 or A5D04	Controls the write or read function of each storage cycle; this line is – for write, + for read.
+Reset	A3B04	A2B07 or A5B04	When active, this line resets the sense latches on each array card.
+Machine Reset	A3D03	A2B09 or A2D13	When active, this signal suppresses the generation of extraneous timing signals in the BSM during a system power-on or when manually activated from the console panel; it resets the timing latches on the address buffer card and the sense latches on each array card.

BSM Address and Control Interface

The identification of the address control lines for each storage configuration is shown in ALD logic pages SQ005, SQ008 through SQ011. The entry and exit pins for a specific interface line are electrically connected in the array card. Their functions may, therefore, be interchanged; an exit pin may be used as an entry pin, or vice versa.

The address and control lines from the ECCL board enter the BSM board at either of two socket locations—A2 or A5. If the board pins in socket location A5 are used as entry pins, the board pins in socket location A2 become exit pins. Depending upon the board location in the storage configuration, the exit pins may be used to:

1. route the address and control signals to an adjacent BSM, or
2. terminate the address and control lines with plug-on terminators in the array board in the last in a chain of multiple BSMs. This termination points are shown on ALD SQ016.

The address flow drawings (ALD SQ007 through SQ011) show the address and control lines relationship between the ECCL board and the BSMs. These drawings (one for each configuration) contain the pin location for the interface lines along with the polarity of each signal. Generally, the polarity of the address gated to the upper BSMs (A2, B2 and C2) on the 2/3 socket location, is negative. The polarity of the addresses gated to the lower BSMs (A4, B4, and C4) on the 4/5 socket locations is positive. All array cards participate in each store (write) or fetch (read) cycle. The address lines from the ECCL board enter the address buffer card on the BSMs as shown on ALD SQ005.

The selection of address bits used for PGA, PGB, and CHIP, and an occasional address 12, are chosen to match the address range of the array card in a particular storage configuration. The special chaining technique used to propagate these special address lines from one board to the next is shown on logic pages SQ008 through SQ011. Address lines 1, 3, 4, 5, 7 through 11, along with the control lines, are propagated to each board.

Address Flow-Example

To illustrate the addressing flow between the BSM and the ECCL board, consider the following example:

Storage Configuration	256K
Access	Main Storage
Storage Address	8000 hex

The address bit configuration in SAR for the address is:

16 15 14	13 12 11 10	9 8 7 6	5 4 3 2	1
0 0 0	1 0 0 0	0 0 0 0	0 0 0 0	0
0	8	0	0	0

ALD page SQ011 shows how the address and control bits are routed to each BSM of the 256K configuration. The BSM logic board interconnections are shown on ALD SQ005.

UPPER BOARD ADDRESS FLOW

ALD SQ011 shows the common address lines, and the W/R, reset and machine reset controls entering the address buffer card of the upper BSM A2 at socket A5 at the pins designated on ALD SQ005. From socket A5, each address is fed to an AND circuit, where the address line is powered and inverted to provide true and complement addresses to the array cards. The 2/3 sockets represent the upper board, which receives the out-of-phase address lines. The addresses leave board A2 from socket A2 at the pin locations shown on ALD SQ005 and are chained to board B2 at socket A2. The addresses exit board B2 at socket A5 and connect to board C2 at socket A5. The addresses from the C2 board are terminated at the pin locations shown on ALD SQ016.

At each board, the addresses are connected to the array cards as shown on ALD SQ005.

The PGA, PGB, CHIP, and CONTROL lines distribution generally follow the same chaining technique. Although these special address lines are not used in the same manner at each board, they enter each board at the same pin location. For example, the PGA address line always enters every BSM board, regardless of configuration, at pin location B2B11.

If you were to follow address 10 from the ECCL board through each BSM and to its termination point, the address path would be:

Upper BSM Boards

	ECC BRD	A2 BRD		B2 BRD		C2 BRD	
ADDR	EXIT	ENTRY	EXIT	ENTRY	EXIT	ENTRY	TERM
10	A3B10	A5B12	A2B03	A2B03	A5B12	A5B12	A2B03

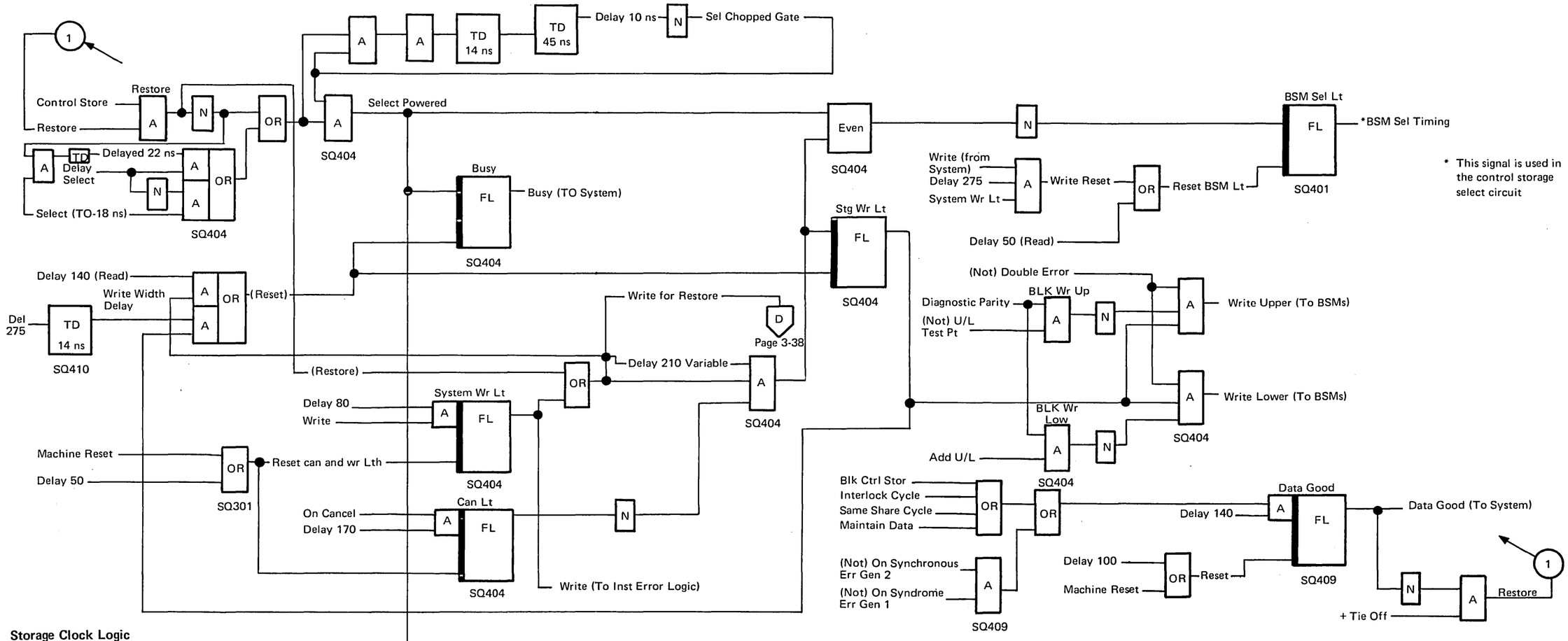
NOTE: Address 10 enters all array cards at pin B08.

LOWER BOARD ADDRESS FLOW

The address flow for the lower BSMs is the same as for the upper boards except that the ECCL board connects to the first board (A4) at the A2 socket. Therefore, the entering and exiting pins on the lower boards are directly opposite to those on the upper boards. Using address 10 for an example, the address propagates through the BSMs as follows:

	ECC BRD	A4 BRD		B4 BRD		C4	
ADDR	EXIT	ENTRY	EXIT	ENTRY	EXIT	ENTRY	TERM
+10	A4B03	A2B03	A5B12	A5B12	A2B03	A2B03	A5B12

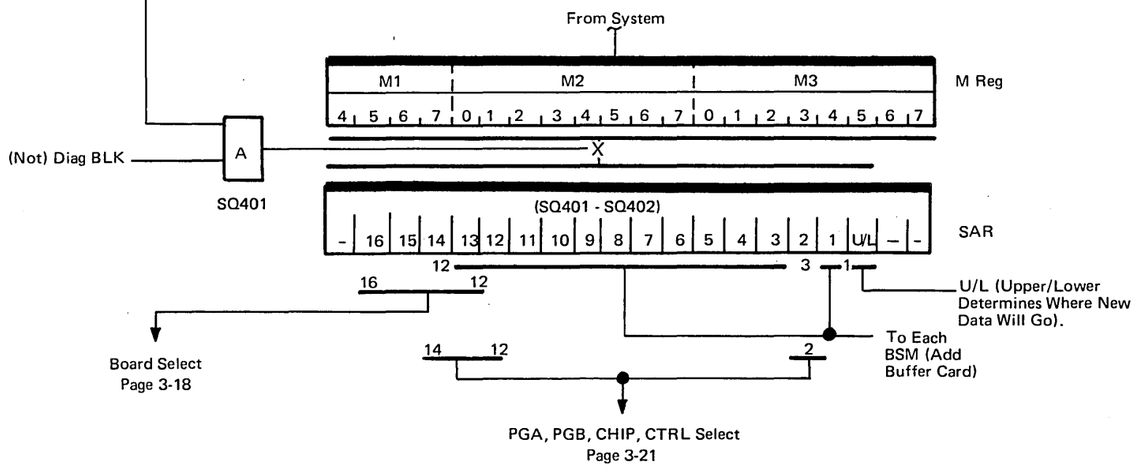
SAR and STORAGE CLOCK



* This signal is used in the control storage select circuit

Storage Clock Logic

SAR



REMEMBER

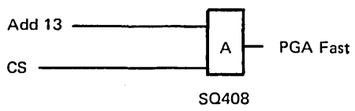
There is a Reader's Comment Form at the back of this publication.

PGA, PGB, CHIP, and CONTROL-BITS SELECTION

STORAGE SIZE 112K, PLUS 32K CTRL

Board A2/A4

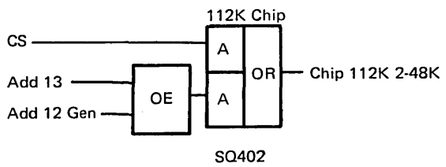
PGA



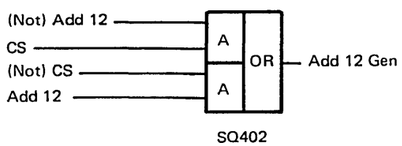
PGB

PGB = Add 2 to Storage (SQ401)

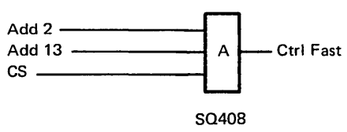
Chip



Add 12



Control



Board B2/B4

PGA = Add 12

PGB = Add 13

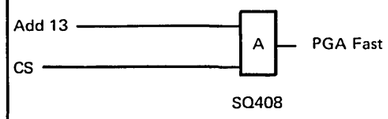
Chip = Add 2

Add 12 = Add 6 (propagated from A2/A4 boards)

STORAGE SIZE 160K, PLUS 32K CTRL

Board A2;A4

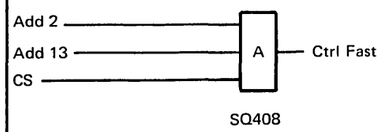
PGA



PGB = Add 14

Chip = Add 2

Control



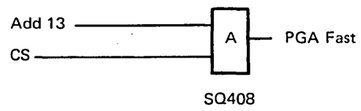
Board B2/B4

PGA = +13
PGB = 14
CHIP = 2

STORAGE SIZE 208K, PLUS 32K CTRL

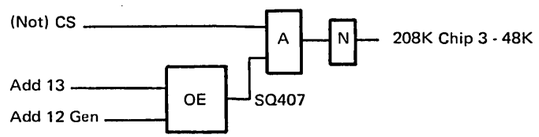
Board A2;A4

PGA

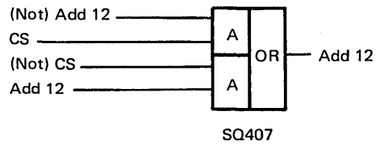


PGB = Add 2

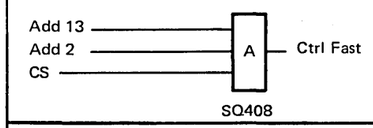
CHIP



Add 12

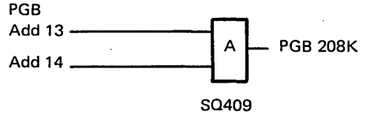


Control



Board B2/B4

PGA= Add 14
PGB



CHIP= Add 2

Board C2/C4

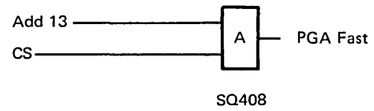
PGA = Add 12
PGB = Add 13

CHIP = Add 2
Add 12 = Add 6 via Add 12

STORAGE SIZE 256K, PLUS 32K CTRL

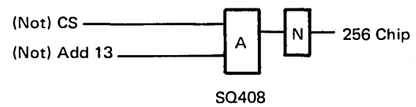
Board A2/A4

PGA

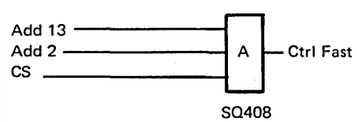


PGB = Add 2

CHIP



256 Control



Board B2/B4

PGA = Add 13 +
PGB = ADD 14 –
CHIP = Add 2 –

Board C2/C4

PGA = Add 13 –
PGB = Add 14 –
CHIP = Add 2 –

TIMING-ECCL BOARD and BSM

The timing mechanism in the ECCL board and the BSMs is composed of a tapped delay line and provides all the timed pulses necessary to control the storage operations. The ECCL board contains two delay-line cards, which are basically used for generating control pulses for functional operations performed on the ECCL board. These delay lines are adjustable and operate for approximately 275 nanoseconds (ns), and are in steps of 10-nanosecond increments. The delay line in the address buffer card generates the BSM timing signals required by each array card during a store or fetch cycle. The BSM select (board select) is the signal that starts each store and fetch cycle. This signal is generated in the BSM clock circuits located in the ECCL board. (See page 3-17.) Address lines in conjunction with the BSM select signal determine which BSMs are to be selected.

The ECCL board also contains a BSM clock card, which is primarily used for generating the set/reset function for the storage data register (SDR), and for generating the clock pulse for resetting the busy and storage write latches as shown on Diagram 3.9.

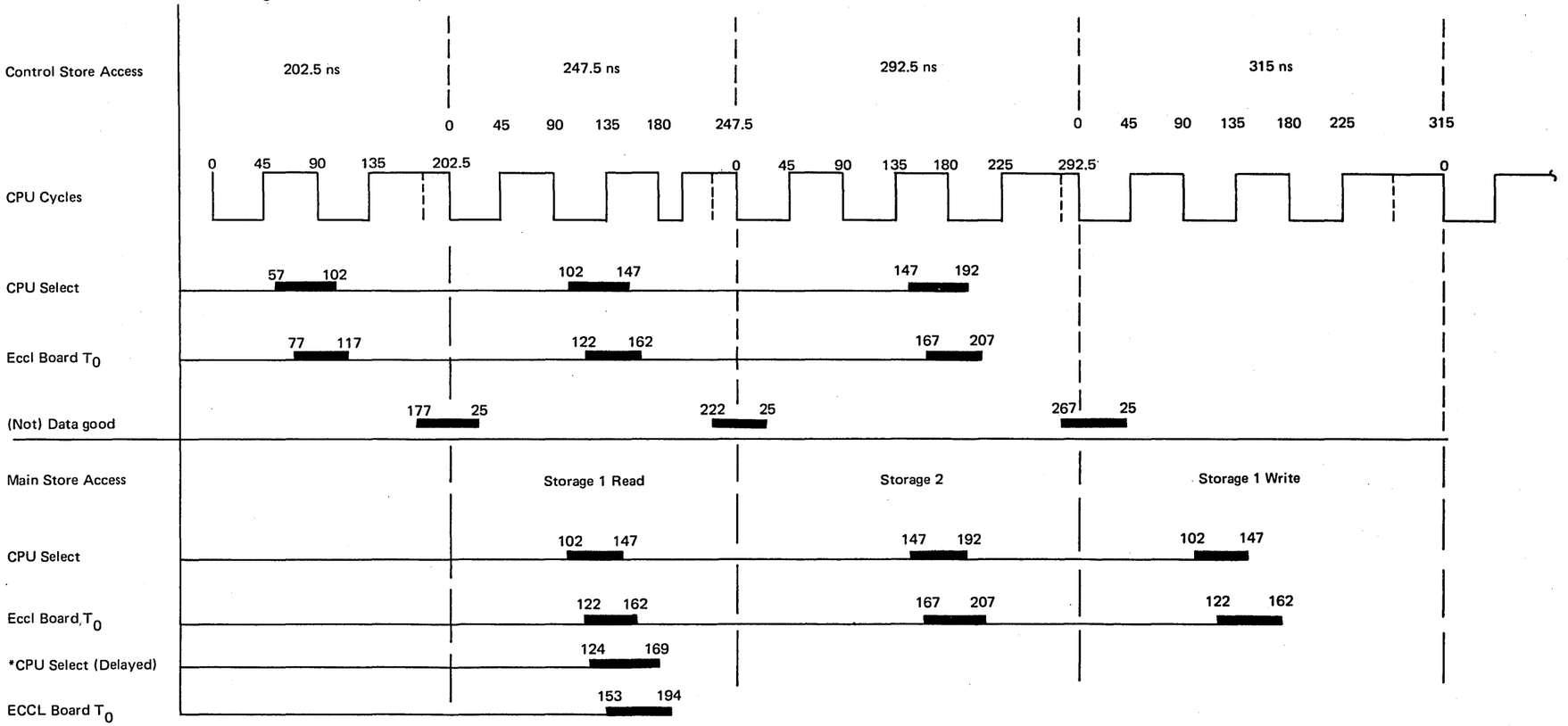
CPU Select Pulse/ECCL Board T_0 Relationship

The CPU has a variable-cycle clock that produces timing durations of 202.5, 247.5, 292.5, and 315 nanoseconds. The CPU control-word type determines the time duration of the CPU cycle. For control-storage accesses, the 315 ns time duration is not used; for main-storage accesses, the 202.5 ns time duration is not used. To address either main or control storage, the CPU must generate a CPU select pulse that starts the delay-line clock within the ECCL board. The time at which the CPU select pulse is generated is dependent upon the CPU cycle length.

The width of the select pulse is dropped from 45 ns to approximately 38 ns by a time-delay circuit, forming a select powered control line that starts the delay-line clock.

During control-storage accesses, data becomes available at the SDBO at the same time it is being analyzed for errors. This simultaneous action permits the CPU to operate on the SDBO data on the subsequent cycle if it contains no errors. To prevent the use of the SDBO data before the error-checking circuits complete their analysis, a (Not) data good line is activated at T_0 + 100 ns and is reset at T_0 + 150 ns if no error is detected. The reset of this control line means that the control-storage data on the SDBO is correct and can be used by the CPU. Because the CPU samples the data on the SDBO at the same time regardless of CPU cycle length, the (Not) data good/CPU time relationship for each CPU cycle length must be the same. To maintain this relationship, the generation of the CPU select pulse differs for each CPU cycle length. The (Not) data good control line/CPU cycle relationship is shown on page 3-22.

PU Select Pulse/Eccl Board T_0 Relationship



* Delayed 22 ns for address-adjust operation

Note: The timing shown on this page is intended for instructional purposes only; the actual timings may differ (see "Storage Timing Adjustments").

ECCL Delay Lines

The ECCL delay lines produce all the timed pulses necessary to control the operation of the ECCL board. The clock consists of a serial network of delay lines. Physically, there are two 14 ns and two 45 second delay lines mounted on each of two delay-line cards. A total delay of 275 ns is available from the delay line network, and an output is available from tags at 10-nanosecond intervals after input time.

The logic for the delay-line network is shown on Page 3-23. The select powered pulse enters delay-line card No. 1, which furnishes a pulse to delay-line card No. 2 at approximately 120 ns after the receipt of the select powered pulse.

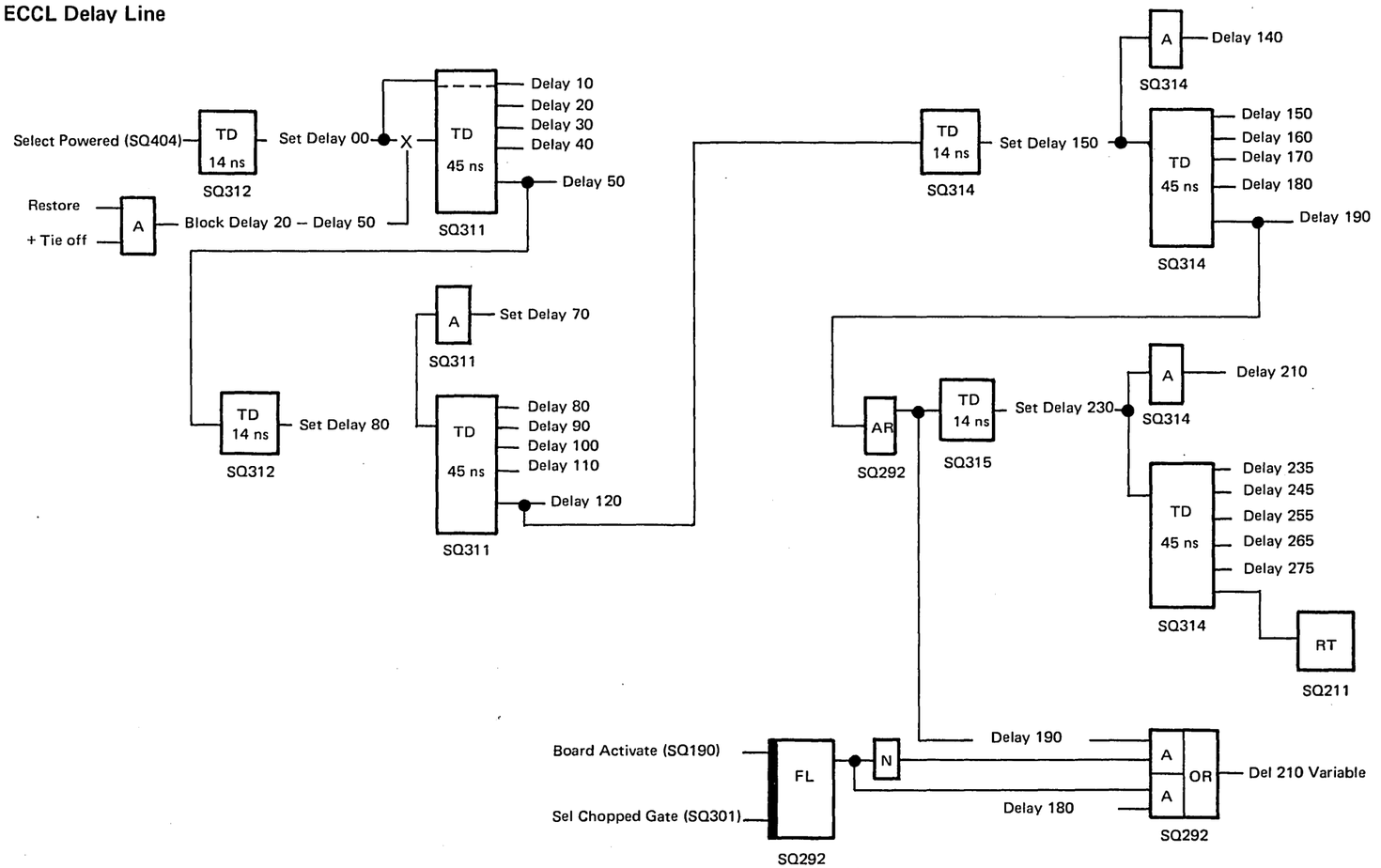
If a single-bit error arises during a control-storage access; the storage bit in error is corrected before the CPU makes use of the SDR data, and the bit is also corrected in the BSM from which it was fetched. To correct a storage bit in error in the BSM, a restore function is made active that changes a control-storage read cycle into a control-storage write cycle. This BSM restore function can be inhibited by using tie-off on 01B-A2-B2D09.

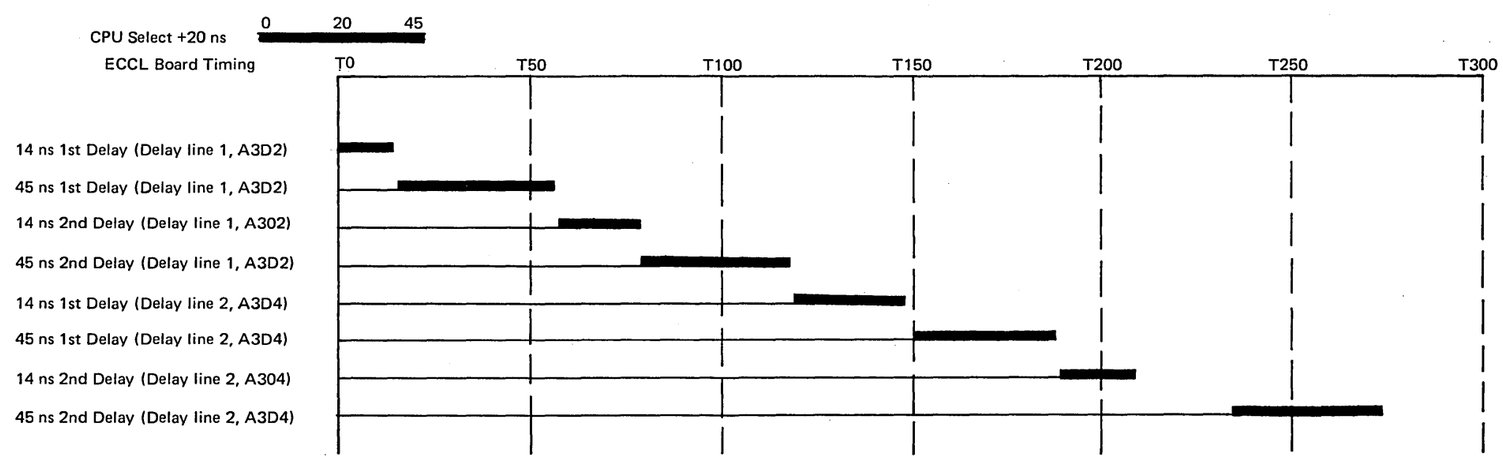
All timings within the logics of the ECCL board are in relation to the select powered pulse which, by definition, is referred to as T_0 of the ECCL board. Each timing pulse has a time duration of approximately 35 ns.

BSM Timing Specification

The timing specifications for the BSMs used with the 3145 are shown on Pages 3-24 and 3-25. The actual timing for the BSM address buffer card is contained on ALDs SQ002 and SQ003.

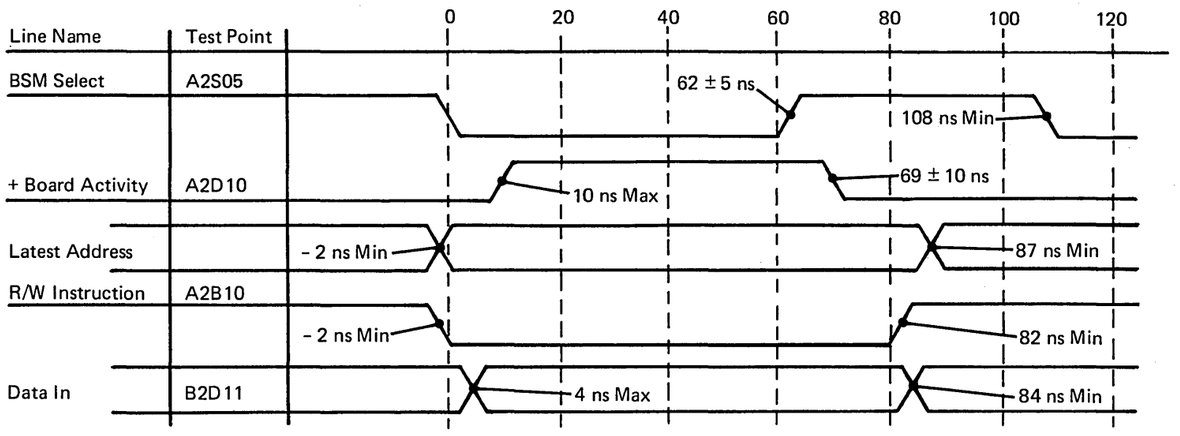
ECCL Delay Line



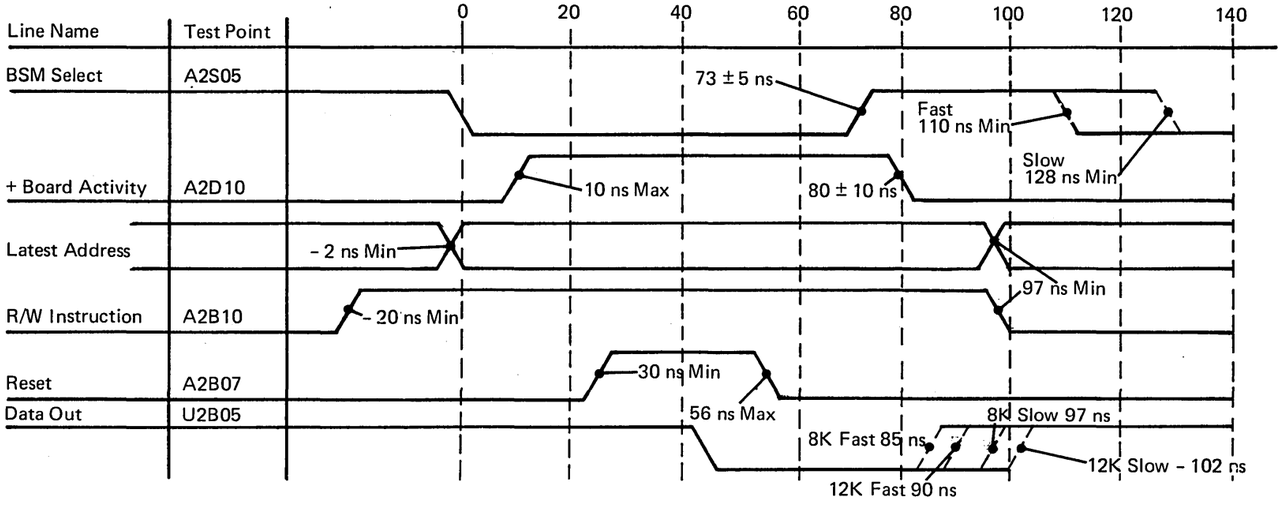


BSM Interface Timing

Write Cycle Interface



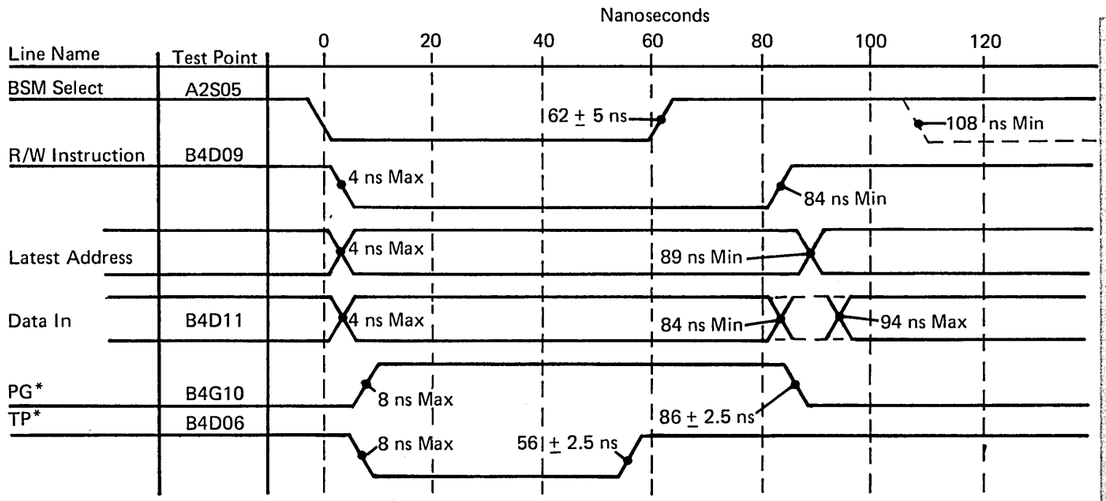
Read Cycle Interface



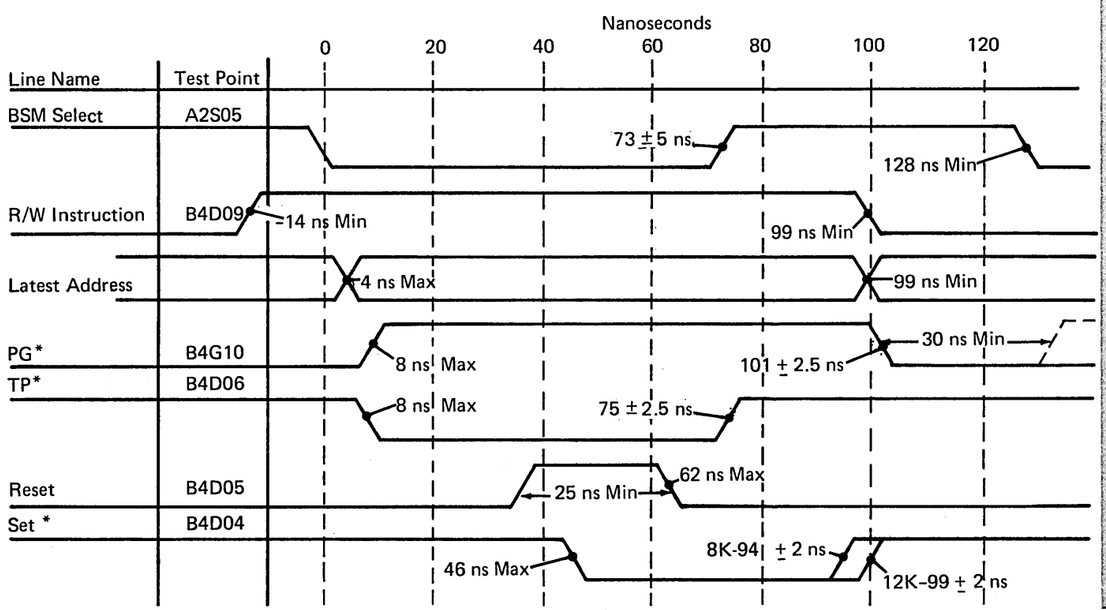
Note: The timing shown on this page is intended for instructional purposes only; the actual timings may differ. (See ALD SQ002 and SQ003.)

BSM Internal Timing

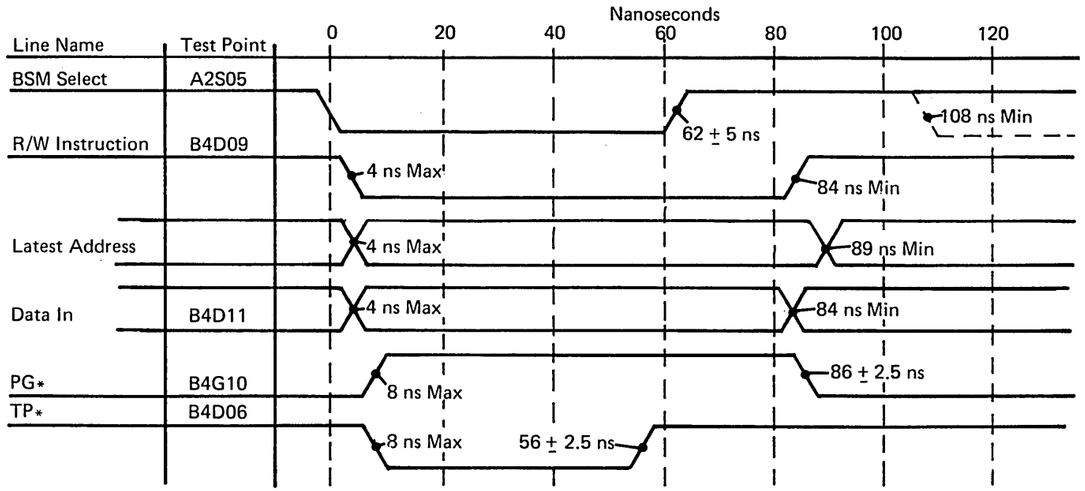
Write Internal Timing—Slow



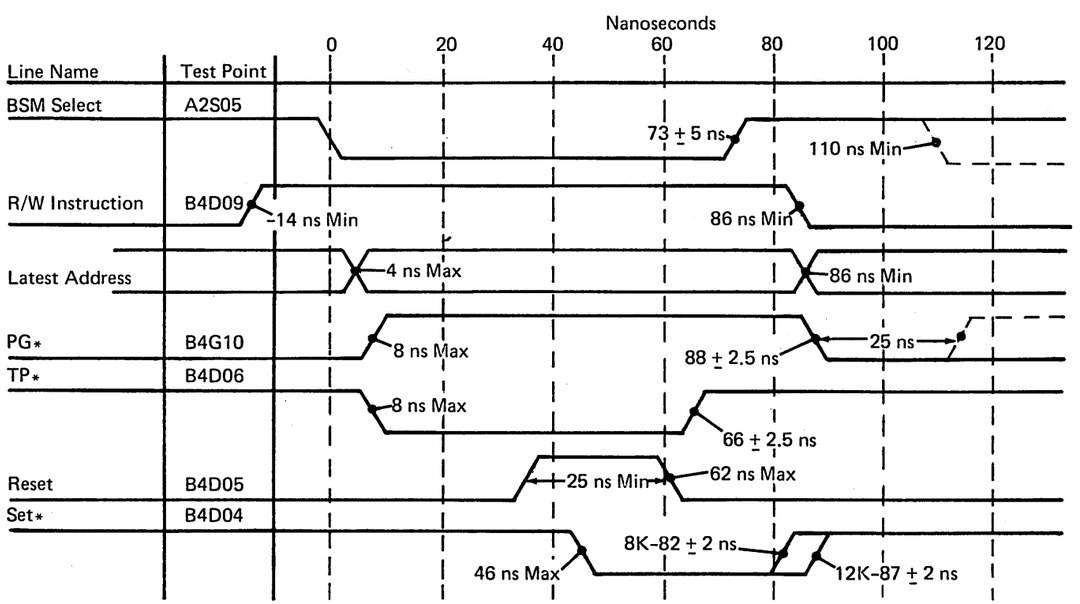
Read Internal Timing—Slow



Write Internal Timing—Fast



Read Internal Timing—Fast



* The timing of these signals is adjusted by altering delay-line taps on the address buffer card (See ALD SQ002 and SQ003). The timing of the other signals is adjusted on the ECC board (see "Storage Timing Adjustments").

Storage Timing Adjustments and Checks

Storage-timing adjustments and checks include the following.

STORAGE-TIMING ADJUSTMENTS

1. CPU Select Pulse
2. Delay-Line Card 1
3. Delay-Line Card 2

STORAGE-TIMING CHECKS

1. Store Cycle 1-Read
2. Store Cycle 1-Write
3. Store Cycle 2
4. Write Pulse

For valid results, the storage-timing adjustments must be performed in the defined sequence (1 through 9). Readjust a delay line only when it is outside the checking tolerance. When plugging delays, always plug vertically or horizontally rather than diagonally.

STORAGE-TIMING-ADJUSTMENT PREREQUISITES

A. Make certain that the following wire-wrapped jumpers are installed:

From	*To*
B-B2A5D02	B-B2A5D07
B-B4A5D02	B-B4A5D07
B-C2A5D02	B-C2A5D07
B-C4A5D02	B-C4A5D07

B. Make certain that the following wire-wrapped jumpers are removed:

From	*To*
B-A2A5D02	B-A2A5D07
B-A4A5D02	B-A4A5D07

C. Do not plug or remove address buffer card (A2) from any BSM while power is on. If checking or retiming of the address buffer card is necessary, refer to ALDs SQ002 and SQ003.

D. Make certain that 'data good' line is activated.

E. Oscilloscope setup (Tektronix type 454 or equivalent):

Channel 1 Volts/Div: 50 mv/cm

Channel 2 Volts/Div: 50 mv/cm

Time/Div: 0.1 usec with X10 MAG

Select Center Pulse

1. CPU SELECT-PULSE ADJUSTMENT
   A. Perform control word 30 00 0000 using CPU console as follows:
      - Set the rate switch to SINGLE CYCLE HARDSTOP.
      - Set DIAGNOSTIC/CONSOLE FILE CONTROL to EXE CTRL WORD SWS A-H.
      - Set switches A-H to 30 00 0000.
      - Set rate switch to PROCESS.
      - Press start key.

   B. Attach channel 1 probe as sync at A-B3H4J07 (CPU clock -0 time).

   C. Attach channel 2 probe to B-A3C2D02 (select pulse).

   D. Verify that the minus select pulse is 45 $\pm$4 ns wide and occurs 57 -60 ns after CPU clock 0 time. If the pulse is not within tolerance, adjust the programmable delay line tap at A-C1F2 to obtain the specified timing. (See CPU clock and timing adjustment for layout of the programmable delay line.)

2. DELAY 00 ADJUSTMENT
   A. Attach channel 2 probe to B-A3D2D12 to monitor delay 00.

   B. Verify that a minus pulse occurs 74-76 ns after sync goes plus. If necessary, adjust the delay tap at B-A3D2 to obtain specified timing. (Refer to Page 3-26 for delay block layout.)

3. DELAY 50 CHECK
   A. Attach channel 1 probe (sync) to B-A3D2D12 (delay 00).

   B. Attach channel 2 probe (delay 50) to B-A3D2D10.

   C. Verify that a minus pulse occurs 48-55 ns after sync pulse.

4. DELAY 80 ADJUSTMENT
   A. Keep channel 1 probe (sync) at B-A3D2D12 (delay 00).

   B. Attach channel 2 probe (delay 80) to B-A3D2J05.

   C. Verify that a minus pulse occurs 79-81 ns after the sync pulse. If necessary, adjust delay tap at B-A3D2 to obtain the specified timing.

5. DELAY 120 CHECK
   A. Attach channel 1 probe (sync) to B-A3D2J05 (delay 80).

   B. Attach channel 2 probe (delay 120) to B-A3D2G08.

   C. Verify that a minus pulse occurs 38-45 ns after sync pulse.

6. DELAY 150 ADJUSTMENT
   A. Keep channel 1 probe (sync) at B-A3D2J05 (delay 80).

   B. Attach channel 2 probe (delay 150) to B-A3D4D07.

   C. Verify that a minus pulse occurs 69-71 ns after sync pulse. (Refer to Page 3-27 for delay block layout.)

7. DELAY 190 CHECK
   A. Attach channel 1 probe (sync) to B-A3D4D07 (delay 150).

   B. Attach channel 2 probe (delay 190) to B-A3D4D10.

   C. Verify that a minus pulse occurs 38-45 ns after sync pulse.

8. DELAY 275 ADJUSTMENT
   A. Keep channel 1 probe (sync) at B-A3D4D07 (delay 150).

   B. Attach channel 2 probe to A-A3D4G08.

   C. Set oscilloscope time/div control to 0.2 us with X10 mag.

   D. Verify that a minus pulse occurs 124-126 ns after sync pulse. If necessary adjust the delay tap at B-A3D4 to obtain the specified timing. (Refer to Page 3-27 for delay block layout.)

9. TERMINATE TEST BY DOING THE FOLLOWING
   A. Set rate switch to SINGLE CYCLE HARDSTOP.

   B. Return DIAGNOSTIC/CONSOLE FILE CONTROL switch to PROCESS/IMPL.

   C. Set rate switch to PROCESS.

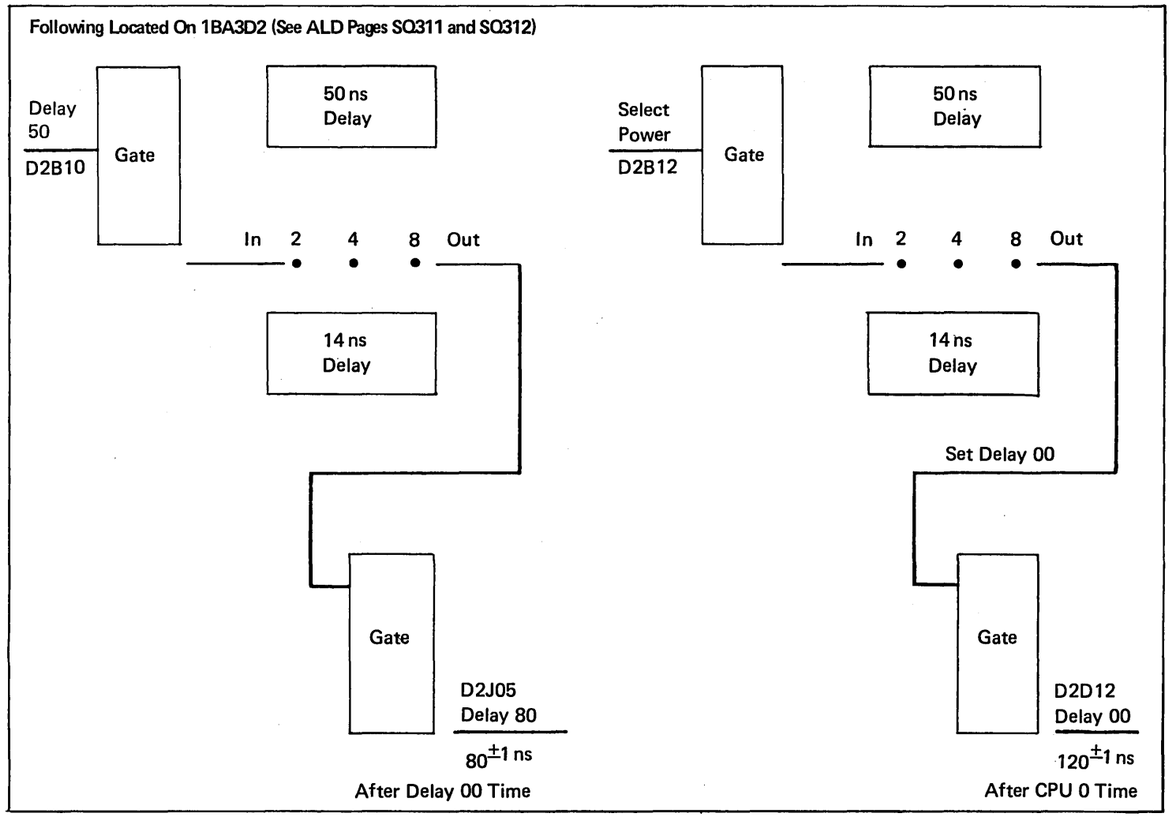
   D. Remove any external jumpers used to activate 'data good line'.

   E. Remove test equipment.

   F. Press start key.

NOTE: Delay jumpering same as card 2 example.

Delay Line 1 Card Layout



STORAGE-TIMING CHECKS

For valid results, the storage timing checks should be done in the defined sequence (1 through 6). Before doing these timing checks, perform items specified under the heading "Storage-Timing-Adjustment Prerequisites."

1. STORE CYCLE 1—READ CHECK
   - A. Place check control switch to DISABLE.
   - B. Perform storage word 40 04 1400 using CPU console as follows:
     - Set rate switch to SINGLE CYCLE HARDSTOP.
     - Set DIAGNOSTIC/CONSOLE FILE CONTROL to EXE
     - CTRL WORD SWS A-H.
     - Set switches A-H to 40 04 1400.
     - Set rate switch to PROCESS.
     - Press start key.
   - C. Set oscilloscope Time/Div to 0.5 us with X10 mag.
   - D. Attach channel 1 probe (sync) to A-C3G2D07 (plus) store cycle 1 line.
   - E. Attach channel 2 probe (0 time) to A-B3H4G05. Align pulse to first traticule on the oscilloscope.
   - F. Verify that storage cycle I occurs approximately 0 to 20 ns after CPU 0 time. The duration of storage cycle I pulse should be approximately 250 ms.
   - G. Attach channel 2 probe (select) to B-A3C2D02. Set oscilloscope Time/Div control to 0.2 us.
   - H. Verify that a minus pulse occurs at approximately CPU 102 time.
   - J. Attach channel 2 probe delay 00 to B-A3D2D12. Set oscilloscope Time/Div to 0.5 us.
   - K. Verify that a minus pulse occurs at approximately CPU 153 time.

2. STORAGE CYCLE-1-WRITE CHECK
   - A. Assume that the check control switch is at DISABLE and perform storage word 48 04 1400 using CPU console. (See 1-B for setup procedure.)
   - B. Attach channel 1 (sync) to A-C3G2D07 (plus). (Logic reference-DC031.)
   - C. Attach channel 2 (CPU -0 time) at A-B3H4J07. Align pulse to first graticule on the oscilloscope.
   - D. Verify that storage cycle 1-write occurs approximately 0 to 20 ns after CPU 0 time. The time duration of storage cycle 1-write pulse should be approximately 315 ns.
   - E. Attach channel 2 probe (select) to B-A3C2D02. Set oscilloscope Time/Div control to 0.2 us.
   - F. Verify that a minus pulse occurs approximately CPU 102 time.
   - G. Attach channel 2 probe (delay 00) to B-A3D2D12. Set oscilloscope Time/Div control to 0.5 us.
   - H. Verify that a minus pulse occurs at approximately CPU 153 time.

3. STORAGE CYCLE 2 CHECK
   - A. Attach channel 1 probe (sync) to A-C3G2D07 (storage cycle 1 plus).
   - B. Attach channel 2 probe (storage cycle 2) to A-C3G2J07.
   - C. Verify that a plus signal approximately 295 ns in duration occurs when storage cycle 1 (sync pulse) goes off.

4. SET SDR CHECK
   - A. Attach channel 1 probe (sync) to B-A3D2D12 (delay 00).
   - B. Attach channel 2 probe (set SDR) to B-A3K2D12.
   - C. Verify that trailing edge of 'Set SDR' goes positive 121-130 ns after delay 00 pulse. (Logic reference SQ287 and SQ410.)

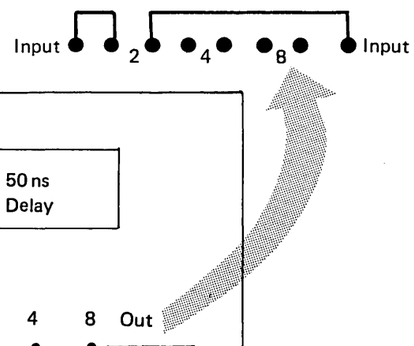
5. WRITE PULSE CHECK
   - A. Attach channel 1 probe (sync) to B-A3D4D07.
   - B. Attach channel 2 probe (delay 150) to B-A3D4D07. Align delay 150 pulse on the first graticule on the oscilloscope.
   - C. Attach channel 2 probe to B-A3C2B04.
   - D. Verify that the positive-going trailing edge occurs approximately 160 ns after delay 150 time.

   NOTE: The write pulse is controlled by 'Write Width Delay' (ALD SQ140) and is located on B-A3B4.

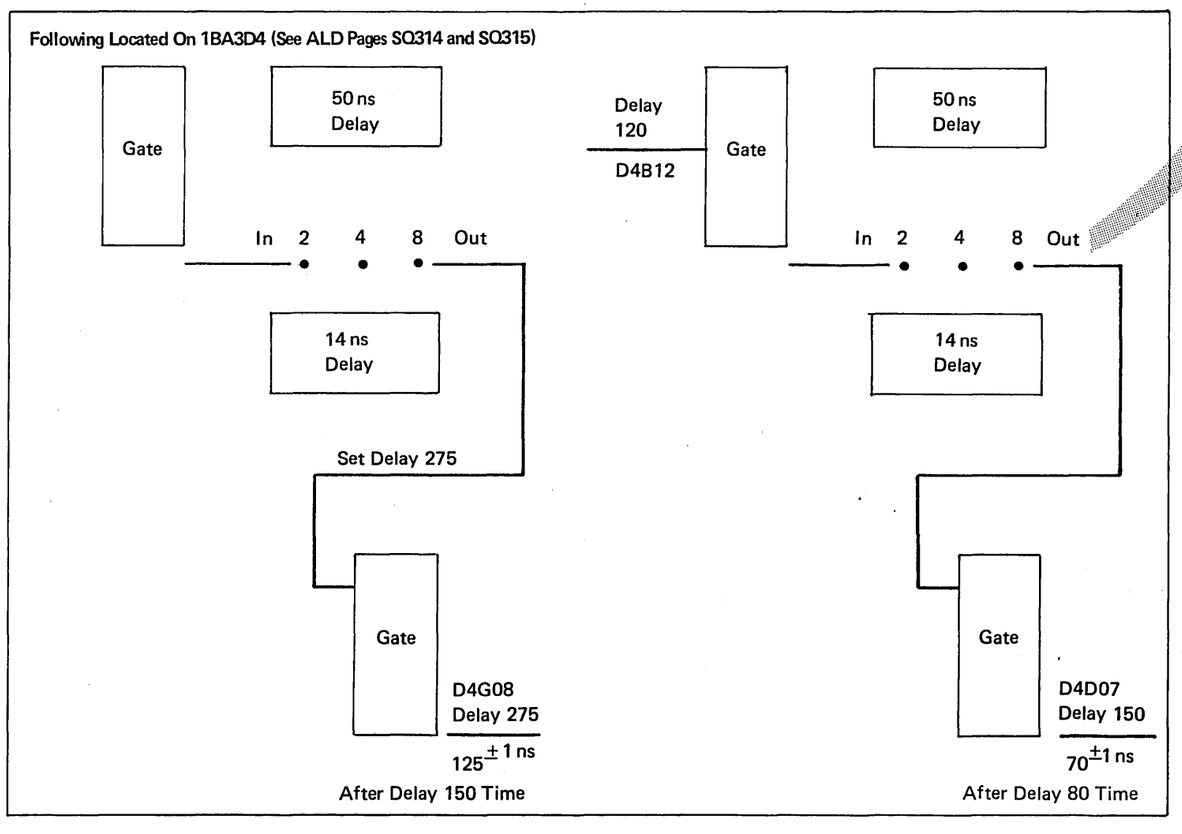
   All delays should be plugged to obtain this signal. See note at bottom of ALD SQ410.

6. TERMINATE TEST BY DOING THE FOLLOWING
   - A. Set the rate switch to SINGLE CYCLE HARDSTOP.
   - B. Return the CPU console switches to their normal operating position.
   - C. Return the rate switch to PROCESS.
   - D. Remove any external jumpers used to activate 'data good' line.
   - E. Remove test equipment.
   - F. Press the start key.

Example:
2-Nanosecond Delay-Card Jumpering



Delay Line 2 Card Layout



BSM DATA FLOW

Main storage and control storage have one condition of data flow for a storage-fetch operation and have two conditions of data flow during each store (write) operation. (Putting data into storage involves a read operation followed by a write.) Data transfer to and from storage is handled by the storage data register (SDR). The SDR interfaces with the BSMs by means of data cables as shown on Page 3-16.

The 3145 uses two basic BSMs: a 48K (12K x 36) BSM that contains 36 array cards, and a 24K (6K x 32) BSM that contains 18 array cards. Because data is fetched and stored on a double-word boundary, a pair of BSMs (upper and lower) are always used for each storage operation. The lower BSMs (boards A4, B4, and C4) contain data bits 0 through 31, and check bits C4, C16, C32, and CT. The upper BSMs (board A2, B2, and C2) contain data bits 32 through 64, and check bits C0, C1, C2, and C4. The data-bit location chart is shown on Page 3-29.

The 48K (12K x 36) BSM has 36 array cards, each of which contains a single data or check bit. The 24K (6K x 36) BSM has 18 cards, each of which contains two bits. Every data and check bit is defined the same in each BSM regardless of the BSMs position within the B gate. For example, bit 60 is always in the array card located in the column B of any BSM board.

Data and Check Bits Cabling

The data and check bit cabling for the upper and lower BSMs is shown on ALD SQ014 and SQ015 respectively. These ALDs show the data and check bit cabling relationship between the ECCL board and the BSMs and contain the pin location for each bit on the BSM.

The entry and exit pins for a specific interface line are electrically connected in the array card. The function may, therefore, be interchanged—an exit pin may be used as an entry pin or, vice versa. The initial interface between the ECCL board and the BSMs is always at the A2/A4 boards. The data and check bits to and from the ECCL board enter the upper BSM (A2 location) at card row 6, and enter the lower BSM (A4 location) at card row 1. At the A2 board, the bits are propagated to card row 1. From this point, the bits are chained to the same card row in each of the subsequent boards (B2 and C2, if used). At the A4 board, the bits are propagated to card row 6. From this point, the bits are chained to the same card row in each of the remaining lower boards (B4 and C4, if used). The board wiring for the B and C board differs from the A board wiring because the B and C board locations can contain either a 48K (12K x 36) or a 24K (6K x 36) BSM. The wiring difference within the B and C boards involves jumper wires that are used to provide either one bit per array card (48K BSM), or a two bits per array card (24K BSM). The 24K jumper assembly is colored red; the 48K jumper assembly is colored yellow. The manner in which the jumpers are wired is shown on ALD SQ004, as well as on the data cable ALDs SQ014 and SQ015. ALD SQ004 illustrates the jumper assembly connections for both a 24K and a 48K BSM.

This ALD shows the wiring side of an upper board and is intended for instructional purposes. The board effectively is split into three parts, showing the wiring of check bits 1 and 2, and data bits 32 and 33, plus illustrating the array card with its associated four lines. From ALD SQ014 and SQ015, we recall that every data and check bit is defined the same in each BSM. To satisfy this requirement, jumper wires are used as illustrated on the bottom portion of ALD SQ004, enabling the BSM to provide one or two bits per array card, depending upon the BSM size.

Let's first assume that the BSM in location B2 or C2 is 24K. Now, if one follows the data input for bit 32, it enters the board on T1E11 and feeds through the board to T6E2. From T6E2, the bit is jumpered to U6C2, enters the bottom array on J02 (as shown on card input/output tab layout), and then terminates at U1C13. Bit 33 enters the board at U1B11 and feeds through the board to U6B2, jumpers to U6A2, and terminates at U1A13. The data enters the bottom array card at D11. This jumper assembly allows data or check bits to feed only the 4/5 sockets of the BSM board because the 2/3 sockets are not used. Now let us follow the same bits (32 and 33) in a 48K BSM. Bit 32 enters the board at T1E11 (same as for the 48K BSM) and enters the array card located at the 2/3 sockets at D11. The data path continues through the board to T6E2, is jumpered to U6A2, feeds back through the BSM, and terminates at U1A13. Bit 33 enters the board at U1B11 and enters the array card located at the 4/5 socket at D11. The data path continues through the board to U6B2, jumpers to U6C2, feeds back through the BSM, and terminates at U1C13. This jumper assembly allows the selection of only one bit per array card.

Terminators are always located adjacent to each data or check bit.

BSM Data Bit Flow Example

To illustrate the propagation of a single bit from the ECCL board through the BSMs, the following example is provided.

Data in 256K Lower Boards

BIT	LEAVE 018-A3 PAGE NET	ENTER 01B-A4	LEAVE 01B-A4	ENTER 01B-B4	JUMPER		LEAVE 01B-B4	ENTER 01B-C4	JUMPER		TERMINATE 01B-C4
30	SQ701 AB7	B1C11	B6C04	B6C04	B1C11	B1D11	B6D04	B6C04	B1C11	B1D11	B6D04

Data Bit Location Chart

Upper Board

	V	U	T	S	R	Q	P	N	M	L	K	J	H	G	F	E	D	C	B	A
		34	32	42	40	50	48	58	56	CK BIT 1	CK BIT 8	38	36	46	44	54	52	62	60	
Terminator Card		(34) 35	(32) 33	(42) 43	(40) 41	(50) 51	(48) 49	(58) 59	(56) 57	(CK Bit 1) CK BIT 2	(CK Bit 8) CK BIT 0	(38) 39	(36) 37	(46) 47	(44) 45	(54) 55	(52) 53	(62) 63	(60) 61	Address Buffer Card

Wiring Side

Lower Board

	V	U	T	S	R	Q	P	N	M	L	K	J	H	G	F	E	D	C	B	A
		2	0	10	8	18	16	26	24	CK BIT 16	CK BIT 32	6	4	14	12	22	20	30	28	
Terminator Card		(2) 3	(0) 1	(10) 11	(8) 9	(18) 19	(16) 17	(26) 27	(24) 25	(CK Bit 6) CK BIT 4	(CK Bit 32) CK BIT CT	(6) 7	(4) 5	(14) 15	(12) 13	(22) 23	(20) 21	(30) 31	(28) 29	Address Buffer Card

Wiring Side

Notes:

1. The 24K BSM has 2 bit/card location for 18 cards in the lower row.
2. The 48K BSM has 1 bit/card location for 36 cards.
3. Data In
   - 24K – data bits J02
     - normal D11
   - 48K – D11

   Data Out
   - 24K – data bits B03
     - normal B05
   - 48K – B05

STORAGE DATA REGSITER

The storage data register (SDR) buffers and directs data between the CPU and storage. The SDR logic (Page 3-31) is composed of:

- 64-bit storage data bus-out (SDBO) register A , and a 64-bit storage latch register B . These registers receive data from storage during both the fetch (read) and a store (write) operation. The SDBO register provides the data to the CPU; the storage latch register provides data to the correction circuits and to the read and write generators.
- 32-bit storage data-bus-in (SDBI) register that receives data from the CPU during a write operation C
- Corrector circuits that change to its correct status D a specific bit that is in error within the SDBO register.
- Storage write switch circuits that are used during a write operation to gate new data to the BSMs E

SDR logic functions as follows:

- Data from storage is always accompanied by check bits that are sent directly to the syndrome generator, where they are used for error-checking analysis.
- Gating control for all registers is derived from the ECCL board delay-line clocks. Data to the SDBO, storage latch, and SDBI registers is gated at approximately T_0+80 ns. Corrected data is gated to the SDBO register at approximately T_0+210 ns.
- During a storage-write operation, byte marks indicate which bytes of data are to be stored. There are four byte marks that accompany the data from the CPU.

Data Flow-Fetch

Each time the CPU decodes a storage word fetch (read) operation, data is received from storage and latched into the SDBO and storage latch registers at approximately T_0+80 ns. If the storage operation fetches data from the control-storage area, data from the SDBO register becomes available on the SDBO at approximately T_0+109 ns as uncorrected data. This uncorrected data is accompanied by a 'data good' control line, which inhibits the use of any data until the ECC control circuits complete their error-checking routine. Reset of the 'data good' control line means that the control-storage data on the SDBO is correct and can be used by the CPU subsequent cycle. For a main-storage access, data is made available to the CPU, only after error checking has taken place. Main-storage data or corrected control-storage data is available on the SDBO at approximately T_0+237 ns. The data that is gated to the storage latch register is immediately distributed to the corrector circuits, to the read generator, and to the write generator via the storage write circuits. From the read and write generators, the data is gated to the other checking circuits (syndrome hardware) for error detection. The decoded syndromes determine which data bit, if any, is in error. If a bit is in error, the on-error line to the corrector circuit becomes active. The corrector circuit is composed of EVEN logic circuits. If a bit is in error, the on-error line to the corrector circuit becomes active and is always opposite in polarity with the bit in error in the storage latch register. Because the corrector circuit is composed of EVEN logic blocks, the bit in error is inverted by the corrector circuit. This inverted bit is gated into the SDBO register at approximately T_0+210 ns, making corrected data available to the CPU at approximately T_0+237 ns.

If a single-bit error arises during control-storage access, the storage bit in error is corrected before the CPU makes use of the SDR data, and the bit may also be corrected in the BSM from which it was fetched. A restore function is available for correcting the status of a BSM bit that is failing intermittently. This BSM restore function can be inhibited by activating the disable switch on the CPU console.

Data Flow-Store

The store operation always starts with a read operation followed by a write operation. Thus, as the store operation commences, 32 data bits from the CPU are gated into the SDBI register. The data bits (64) from the same address location at which the data is to be stored are ingated to the SDR, and storage latch register. The data from the storage latch register is gated to the read generator and to the write generator under the command of the byte-control circuits. The byte-control circuits determine what new information is to be stored and what old information is retained. For example, assume that the upper/lower (U/L) control line and byte 0 control lines were both active. For these conditions, new data is stored in the upper BSMs (location of bytes 4 through 7), and new data is written into byte 4. The byte-control circuits illustrating this operation is shown on Page 3-31. The old and new data to be stored are gated through the storage-word switch circuits, which forwards the data to storage and also to the write generator. The write generator produces seven checks bits, one double-error detection (CT) bit, and eight parity bits on the new data to be stored.

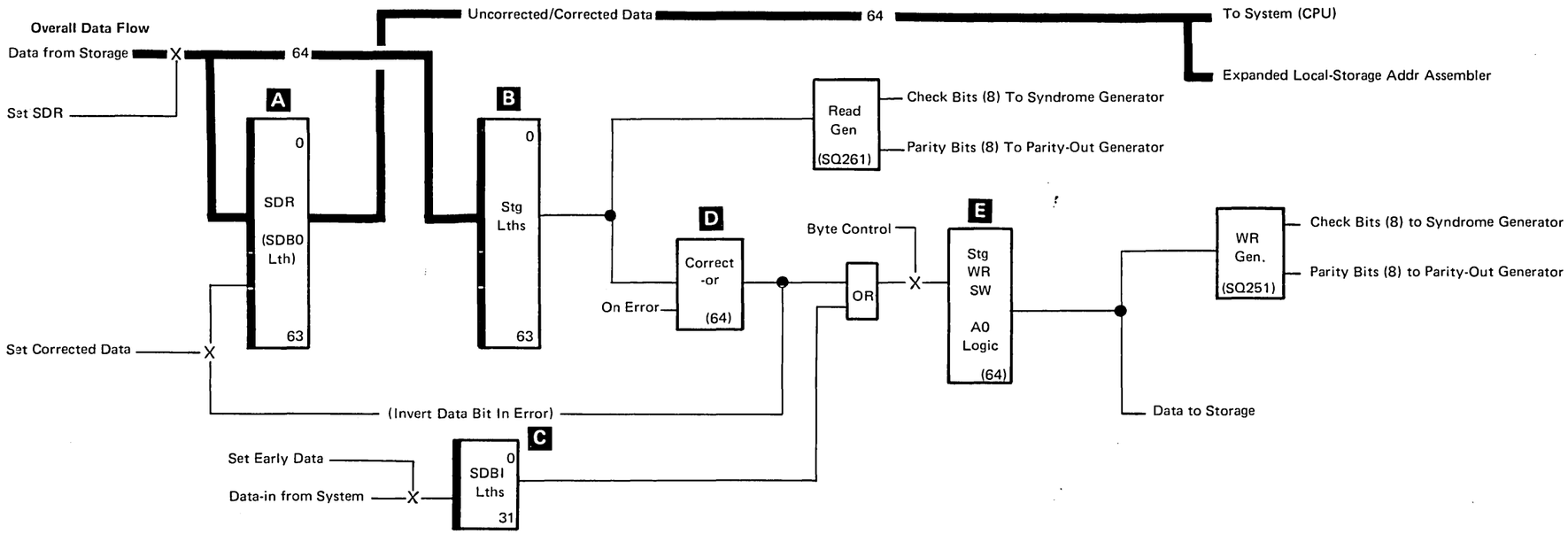
BYTE-MARK REGISTER

Byte marks select the bytes to be stored. The byte marks are sent from the CPU with the data and address bits. A byte mark must be active for every byte of data that is to be stored.

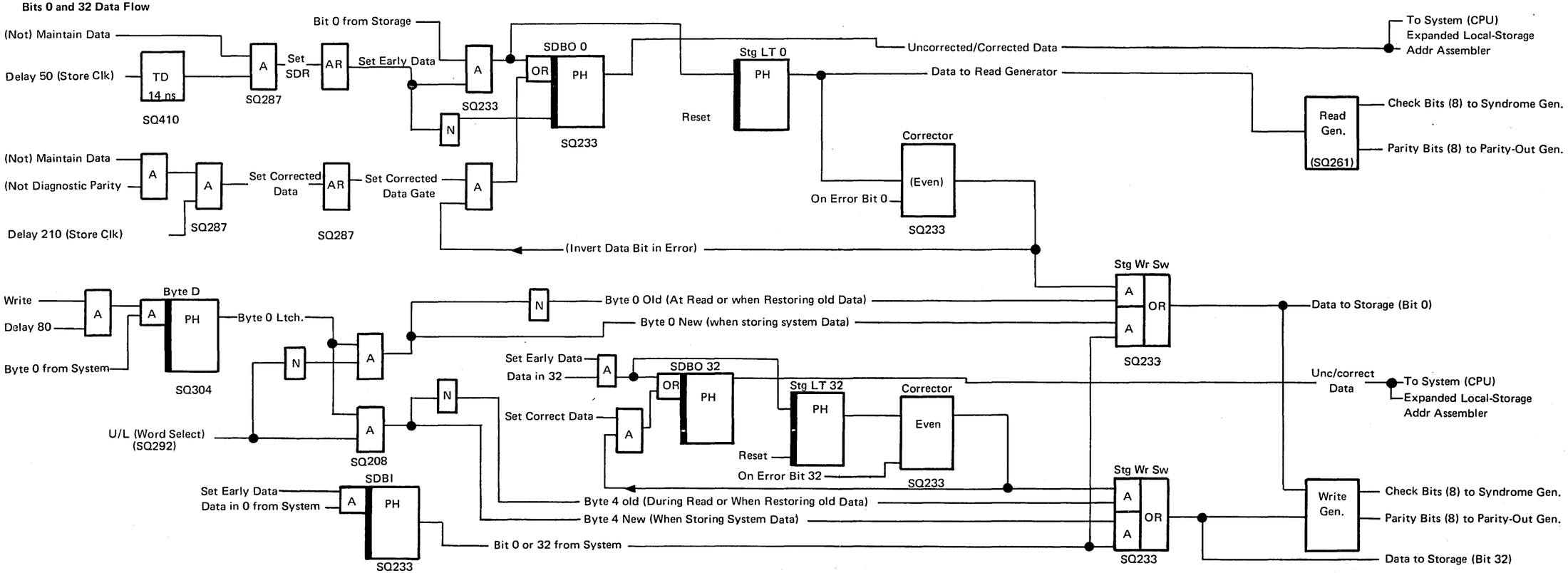
The byte marks are generated within the storage control logic (Diagram 26 of the 3145 *Diagrams Manual*) and sent to the byte mark register, only during a write operation. (Byte 0 latch of this register is shown on Page 3-31.) A maximum of four byte marks (plus parity) comes from the CPU. Each byte mark is then ANDed with the U/L control line to select one of two bytes. For example, byte 0 latch is used to select byte 0 when the U/L control line is inactive; byte 0 latch selects byte 4 when the U/L control line is active.

The byte-mark register is parity-checked (see ALD SQ304) in the parity-out generator. A parity error activates the 'byte check' latch which sets the cancel latch on in the storage clock logic, sets the MCKB, byte 0, bit 3, and turns on the Stor Byte Marks Parity Check light on the CPU console.

Storage Data Register, Data Flow



Bits 0 and 32 Data Flow



ERROR CHECK and CORRECTION

- Error check and correction (ECC) logic detects and corrects single-bit errors.
- Error check and correction logic converts parity bits to ECC bits.

Error checking and correction is the normal mode of operation for the 3145 storage area. Within the ECCL board, the ECC network checks on a doubleword basis for dropped or picked bits (errors). The ECC logic enables correcting single-bit errors. However, multiple-bit errors (though detected) are uncorrectable. Either type of error is indicated to allow the CPU to determine any necessary action.

The error-correction bits constitute a special type of parity checking. In a normal parity scheme, each data bit contributes to the status of the associated parity bit. Recall that each data bit exhibits an equal amount of influence on the status of the parity bit. With error-check and correction (ECC) logic, the parity bits are replaced by a group of seven bits that record a parity sum. Each data bit contributes to this sum. No two data bits contribute the same amount; however, the amount contributed by any one bit is equivalent to the position of that bit in the doubleword. Thus, bit position 1 contributes to ECC bit 1, position 2 contributes to ECC bit 2, and position 3 contributes to both ECC bits 1 and 2. Additionally, bits 0-32 contribute to ECC bit 0, and bit 0 also contributes to ECC bit 32.

Eight check bits are identified as C0, C1, C2, C4, C8, C16, C32, and Ct. The last check bit, Ct, acts as a parity bit for the other check bits. (Check bits are the result of the exclusive ORing of specific combinations of data bits of a doubleword. C-bits are stored in parity positions of bytes, but they are related to the entire doubleword rather than to the byte above.) The C-bits detect errors in the data and develop syndrome bits to decode and correct single-bit errors.

The syndrome bits, identified as S0, S1, S2, S4, S8, S16, S32, and St are decoded to indicate and correct the changed bit (even a change of the check bit) if only one bit has changed. If two or more bits have changed, the syndrome bits decodes a double-bit error.

ECC Data-Flow Concepts

The function of the ECC logic is to detect and correct single-bit errors (correctable errors) and to detect double-bit errors (uncorrectable errors) in storage data or check bits (C-bits). The ECCL board signals the CPU when there is a correctable error or an uncorrectable error.

SINGLE-BIT ERRORS

Single-bit errors are detected in the ECC logic circuit. Bit and byte error signals are sent to the SDR error-bit decoder where these signals are ANDed. The output of the AND is then exclusive ORed with the bit in error and the SDR corrector to complement (correct) the bit in error.

DOUBLE-BIT ERRORS

Double-bit errors are detected in ECC compare and error-detection circuits, (error-type decoder), but the bits are not complemented (an attempted correction would probably be unsuccessful).

ECC Data Flow and Functional Areas

Page 3-14 illustrates the overall relationship and logical interface among the functional areas that compose the ECC logic. The logic circuits that carry out error checking and correction are: read and write generators, syndrome generator, syndrome decoder, error type decoder, parity-out generator, and error bit decoder and corrector circuits within the SDR logic. The description of the functional areas is written in the sequence in which they take part in error checking and correction.

ECC Read and Write Generators

Two generators (read and write) are each composed of even and odd circuits that produce parity bits by byte and eight check bits (C-bits). Input and output for the read generator A are shown on Page 3-32. (The bit generation for the write generator is similar.) The ECC decode chart (Page 3-34), which applies to both generators, shows the bits used to develop each individual C-bit.

The generation of parity bits and C-bits for the read and write generators is contained on ALDs SQ251, SQ252, (write) and SQ261 and SQ262 (read). Regardless of how the bit is generated, an even bit count input produces a bit for the respective parity or check bit. For an even count, a bit equals 1; for an odd count a bit equals 0. Sixty-four (doubleword) data bits from the SDR enter the read and write generators during a read operation. (For a write operation, the write generator receives 32 data bits from the SDR latches (old data) and 32 data bits from the CPU via the SDBI latches.) Eight new C-bits are generated A C0, C1, C2, C4, C8, C16, C32, and CT) that are gated to the syndrome generator B for syndrome bit generation, and to a hardware check circuit shown on Page 3-39. Parity by byte is also generated and passed on to the parity-out generator **C** for ST syndrome generation, and to the parity-correction circuits shown on Page 3-33.

ECC Comparison and Error Correction

The circuit for this discussion is shown on Page 3-33. ECC comparison is a combination of even circuits within the read generator **A** . C-bits produced by the write generator and stored are compared with C-bits produced by the read generator from the data gated to it from the SDR. Because the C-bits from each generator were produced from the same data (before and after storage), they should be the same; if they do not compare equally, one or more bits has been dropped or picked in error by the syndrome generator **B** . The result of the comparison is syndrome bits S0-S32, plus ST which is used in the syndrome error circuit on ALD SQ275.

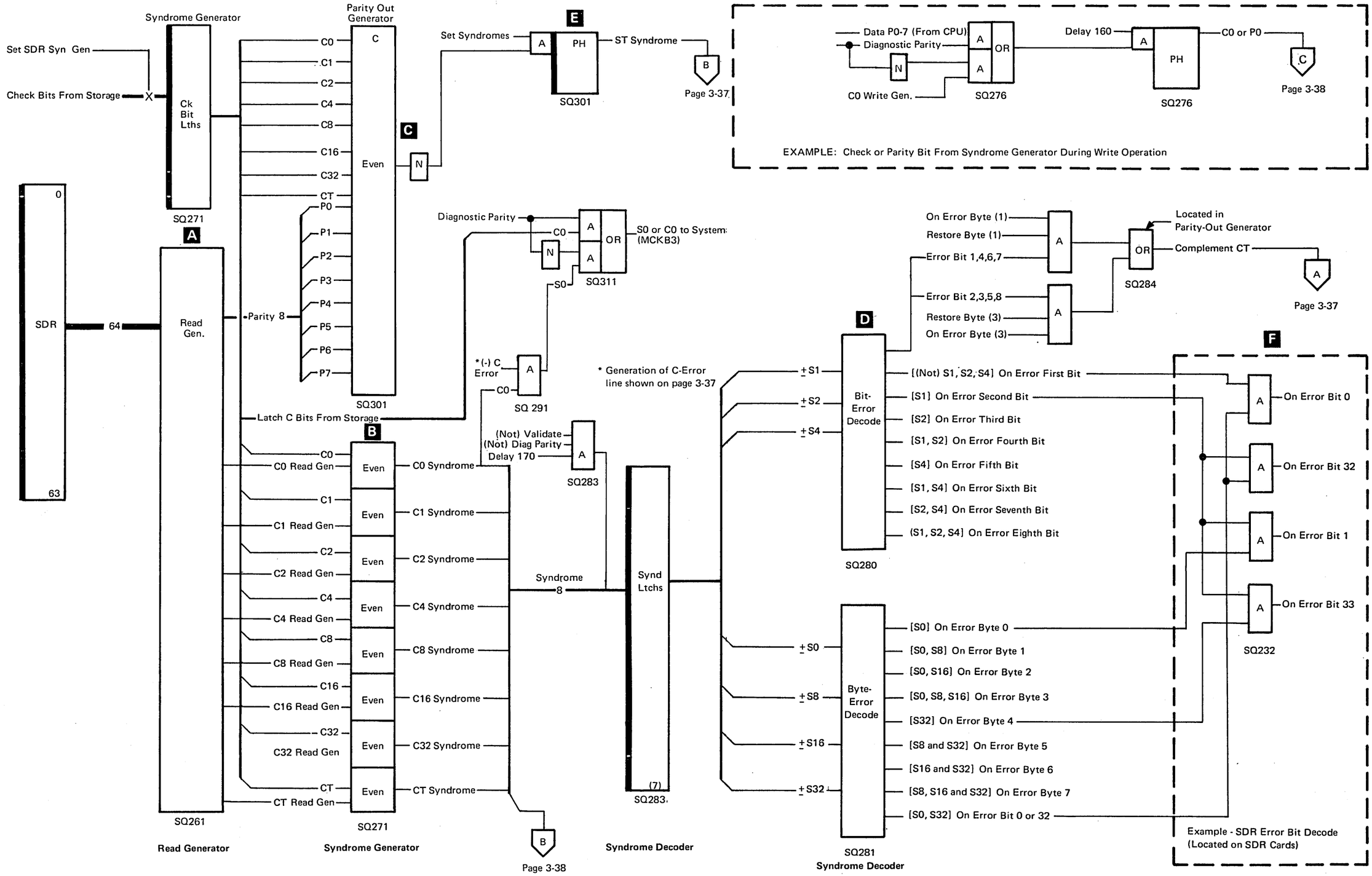
When there is no error, the syndrome bits are 0's. When there is an error, the resulting syndrome bits are 1's and they are gated to the syndrome decoder **D** .

Parity signals for bytes 0-7 produced by the read generator (from fetched data) and all fetched C-bits are compared for an even count in the parity-out generator **C** . An even count (pulse) indicates no error or multiple (uncorrectable) errors. An odd count (minus) indicates an odd (correctable) error. It sets the ST syndrome latch **E** , which is gated to the error-type decoder.

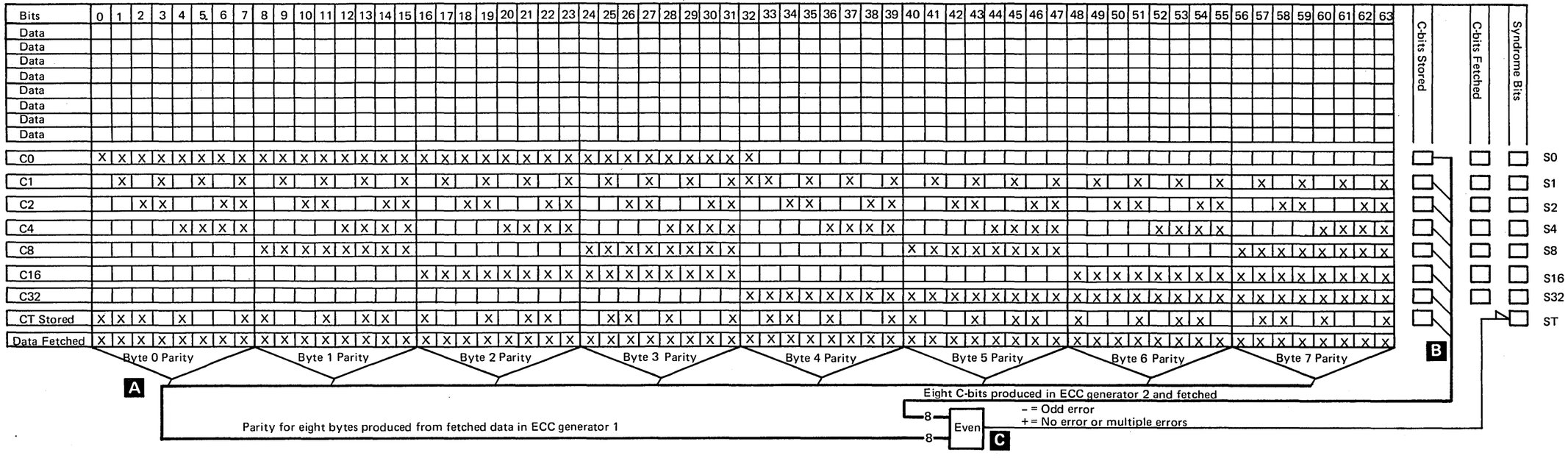
Syndrome Decode

On Page 3-33, syndromes from the syndrome generator **B** are gated to the syndrome decoder **D** , where they are analyzed by type of error and gated to the SDR error-decode network **F** for bit-in error detection. Syndrome bits are also gated to the error-type decoder for internal checking of the check-bit circuitry (Page 3-37). The syndrome decode chart is shown on Page 3-35.

ECC Compare, Error Detect, and Decode



ECC Decoder Chart



Check bits (C-bits) are developed in read or write generators; then stored with data. When storage is cycled, these C-bits and data are fetched; C-bits are developed in the read generator from the fetched data and compared with the fetched C-bits for dropped or picked bits. The accompanying decoder chart, shows how C-bits are developed and used to pick syndromes that point to the bit in error.

To develop C-bits:

1. Insert binary data (simulated or actual) under bits 0-63. Do not use parity bits.
2. For fetched data or data to be stored, proceed as follows.
   a. Whenever X appears in the C-bit line, and 1 in the corresponding data line, add 1 to a count.
   b. When all of the X positions have been checked in every C-bit line, generate the check bits as follows.
      C0, C1, C32, CT: develop even parity (count is even, C-bit=0; count is odd, C-Bit=1) and insert the results in the respective C-bit positions.
      C2, C4, C8, C16: develop odd parity (count is odd, C-bit=0; count is even, C-bit=1) and insert the result in the respective C-bit positions.

Note: The CT bit developed from the fetched data goes only to the hardware checked, but the fetched CT bit is used to develop the ST syndrome bit. (See Page 3-33).

To develop syndrome bits (S-bits):

1. For S0-S32, exclusive OR each C-bit (C0-C32) fetched with the corresponding C-bit developed from fetched data. With both equal, the S-bit=0; with either at 0 and the other at 1, the S-bit=1.
2. For ST, follow the circuits on the chart:
   Develop parity **A** by byte (8) for each bit of data inserted. (Even count =1.) Use these as inputs to the even block.
   Use the eight C-bits **B** fetched as inputs to the even block.
   With a total count even **C** , the output of the block is plus, indicating no error or multiple errors.

With a total count odd, the output of the block is minus and sets ST to indicate an odd, or correctable error. (Other inputs may inhibit ST; these are shown on Page 3-37).

To find the bit in error, add the S-bits or use the syndrome decoder chart Page 3-35. Bits 00 and 32 are special cases, as follows.

1. Bit 00 is in error with ST, S0, and S32 on, and all other syndromes off.
2. Bit 32 is in error with ST, S0, S1, and S32 on, and all other syndromes off.

Syndrome Decoder Chart

ST syndrome in the ON state, indicates a single (correctable) error condition. If ST alone is on, it indicates an error in bit CT. The other syndrome bits, S0-S32, locate the bit in error for a correctable error (ST on), which may be a check bit or a data bit. The check bit, the bit position, or the bit position within the byte may be determined by using the decoder chart.

Note that the byte is equal to the sum of syndromes S0, S8, S16, and S32 if these are assigned a value of 0, 1, 2, and 4, respectively. The bit position within the byte is equal to the sum of syndromes S1, S2, and S4 if these are assigned a value of 1, 2, and 4, respectively.

Syndrome Decoder Chart Use

Assume syndromes S0, S1, and S8 are 1's (indicated by shaded portion). The point of intersection of the coordinates is at bit 9 (bit position 1 of byte 1), indicating an error in that position.

Syndromes					S1/1	0	1	0	1	0	1	0	1
					S2/2	0	0	1	1	0	0	1	1
					S4/4	0	0	0	0	1	1	1	1
S32 /4	S16 /2	S8 /1	S0 /0	Bit Byte	0	1	2	3	4	5	6	7
0	0	0	0		CT	C1	C2		C4			
0	0	0	1	0	C0	1	2	3	4	5	6	7
0	0	1	0		C8							
0	0	1	1	1	8	9	10	11	12	13	14	15
0	1	0	0		C16							
0	1	0	1	2	16	17	18	19	20	21	22	23
0	1	1	0									
0	1	1	1	3	24	25	26	27	28	29	30	31
1	0	0	0	4	C32	33	34	35	36	37	38	39
1	0	0	1		0	32						
1	0	1	0	5	40	41	42	43	44	45	46	47
1	0	1	1									
1	1	0	0	6	48	49	50	51	52	53	54	55
1	1	0	1									
1	1	1	0	7	56	57	58	59	60	61	62	63
1	1	1	1									

Data Bits and Check Bits

ST	S0-S32	Error
Off	All off	None
On	All off	CT
On	One or more on	Correctable error (decode)
Off	One of more on	Uncorrectable error

Single-Bit Error Correction

Assume syndromes (S0 and S1 on Page 3-37) decode, indicating an error in byte 0 and bit position 1 (See F). This combination of syndromes bits enables the on error bit 1 signal. The on error bit 1 signal is forwarded to the corrector circuit within the SDR logic, which causes the invertion of bit 1 in the SDR register. An example of the corrector circuit is shown on Page 3-31. Each bit in a byte is handled in a similar manner.

Single-Bit Detection

Correctable errors are detected by a plus output at B (Page 3-37), which is a C-error detector. A plus output for the C-error detector indicates that a check bit in both syndrome error decode circuits A that feed the detector have at least one C-bit active, indicating that a data bit is in error. This signal is ANDed with the ST syndrome bit. If both signals are active, a single-bit error is detected C (64 single data error line active). If the ST syndrome bit is inactive when the syndrome decoder indicates a single-bit error, the result is equal to a double-bit error. D (Two errors reset ST syndrome latch.)

A minus output from the C-error detector represents a check-bit error. This signal is also ANDed with the ST syndrome bit for double-error detection. A check-bit error, along with an ST syndrome, equals a single check-bit error E and activates the 72 bit single error line to the CPU. The error byte 0 and error byte 1 control lines F are used along with a second error decoder (Page 3-38) for check-bit correction.

Double-Bit Errors

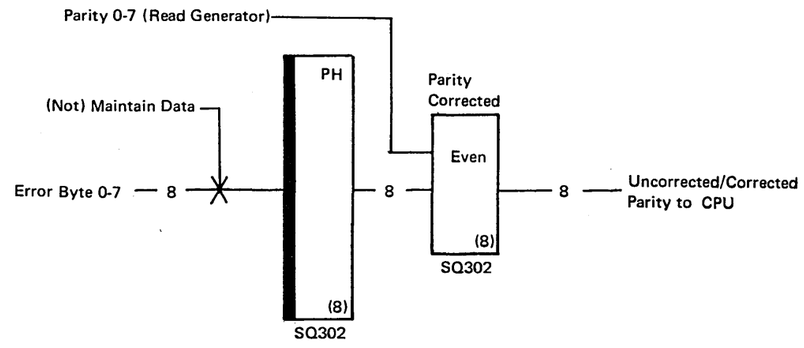
Double-bit errors are detected by a combination of any syndromes S0-S32 on, (Page 3-37), while the ST syndrome bit is in the OFF state. A double-bit error sets the cancel latch, but the manner in which this signal is used within the CPU depends upon the storage-word operation.

Check-Bit Correction

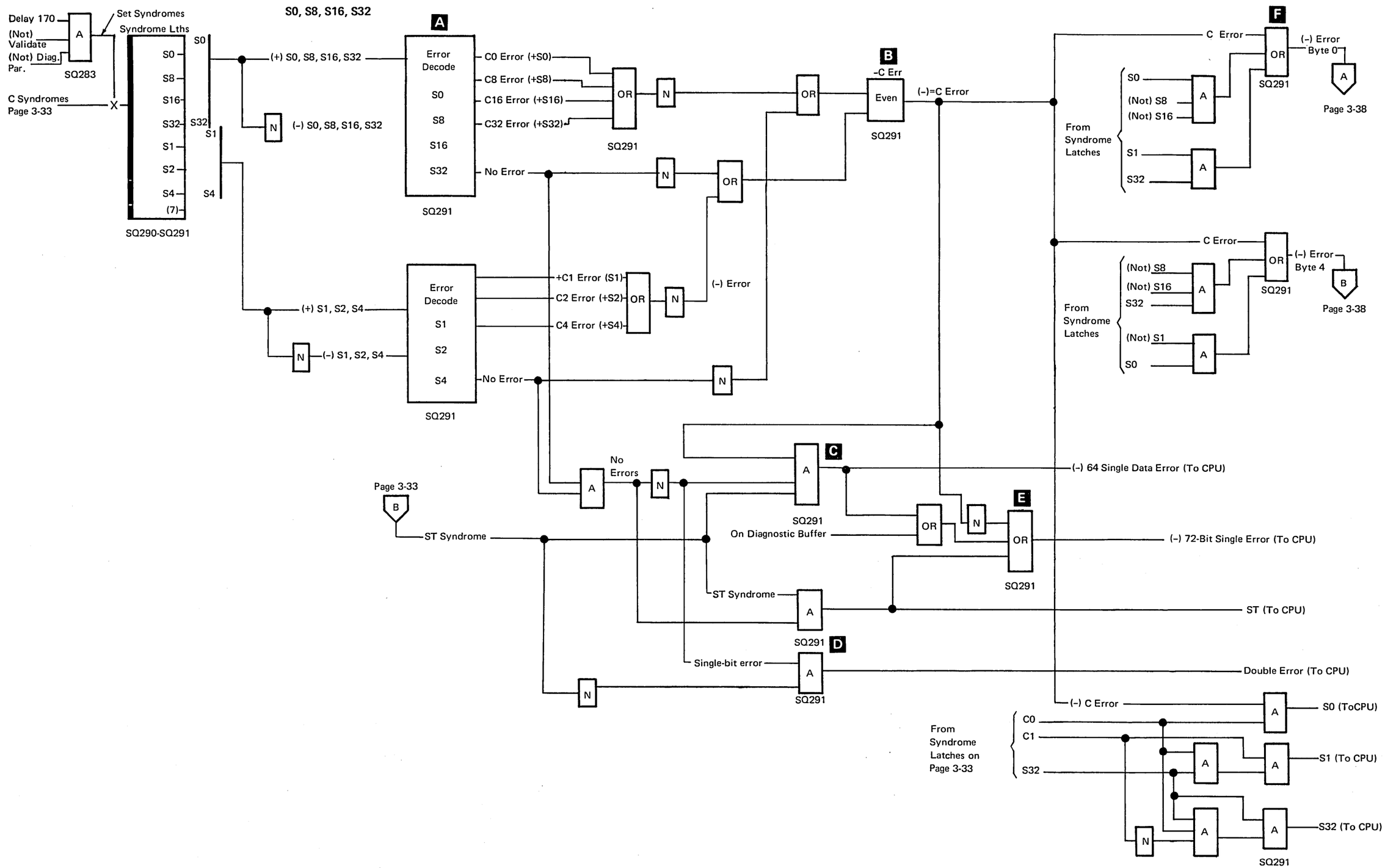
During a write operation, the 32 data bits originally fetched from storage remain in storage unchanged. If an error is detected on one of the bytes that is to be retained, the check bit associated with that bit must be inverted before being returned to storage. This operation is accomplished in the check-bit correction circuit G , which inverts the check bit in error. This circuit is shown on Page 3-38.

Parity Correction

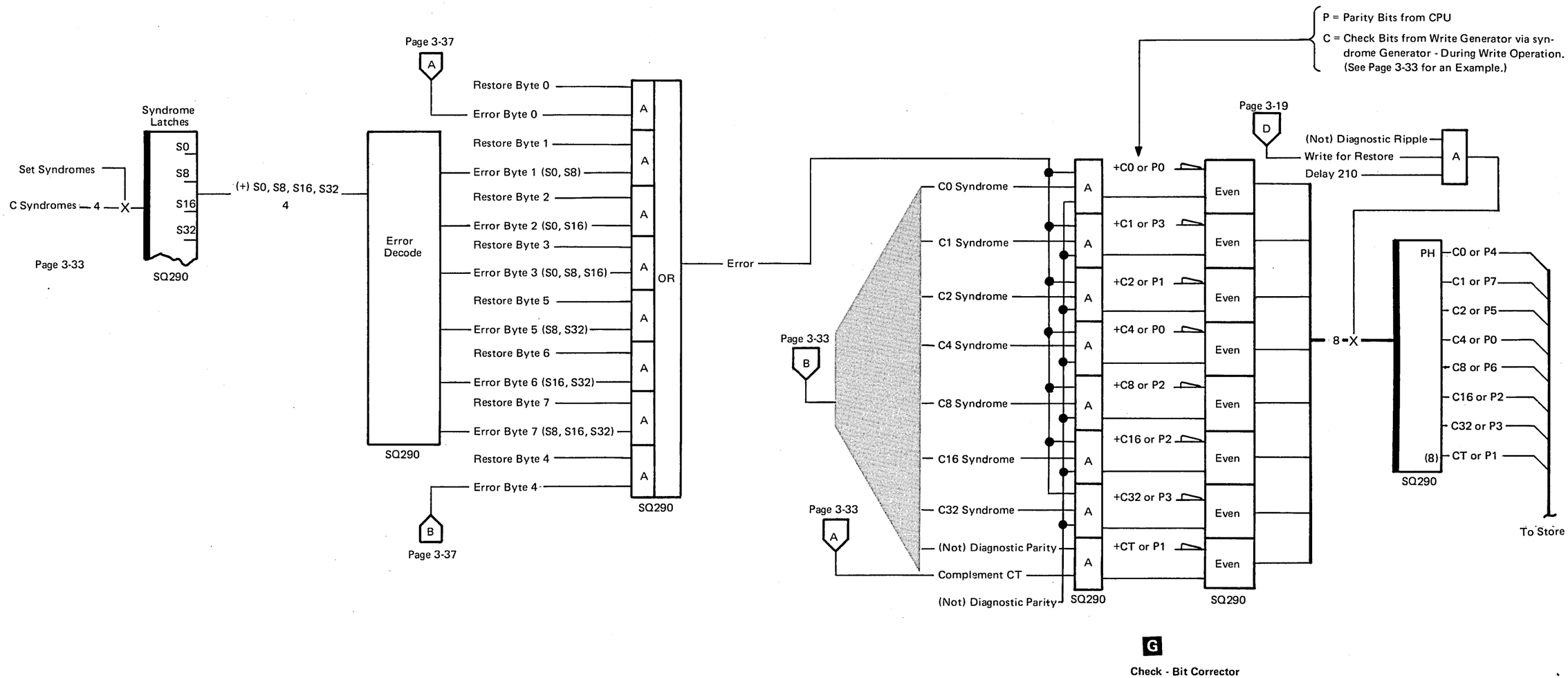
Parity by byte is produced in the read generator from the same data that passed through the syndrome generator and decoder; therefore, an error corrected in a byte required that parity for that byte be changed. Parity correction takes place in the parity-out generator.



Error-Type Decode (Part 1 of 2)



Error-Type Decode (Part 2 of 2)



G

Check - Bit Corrector

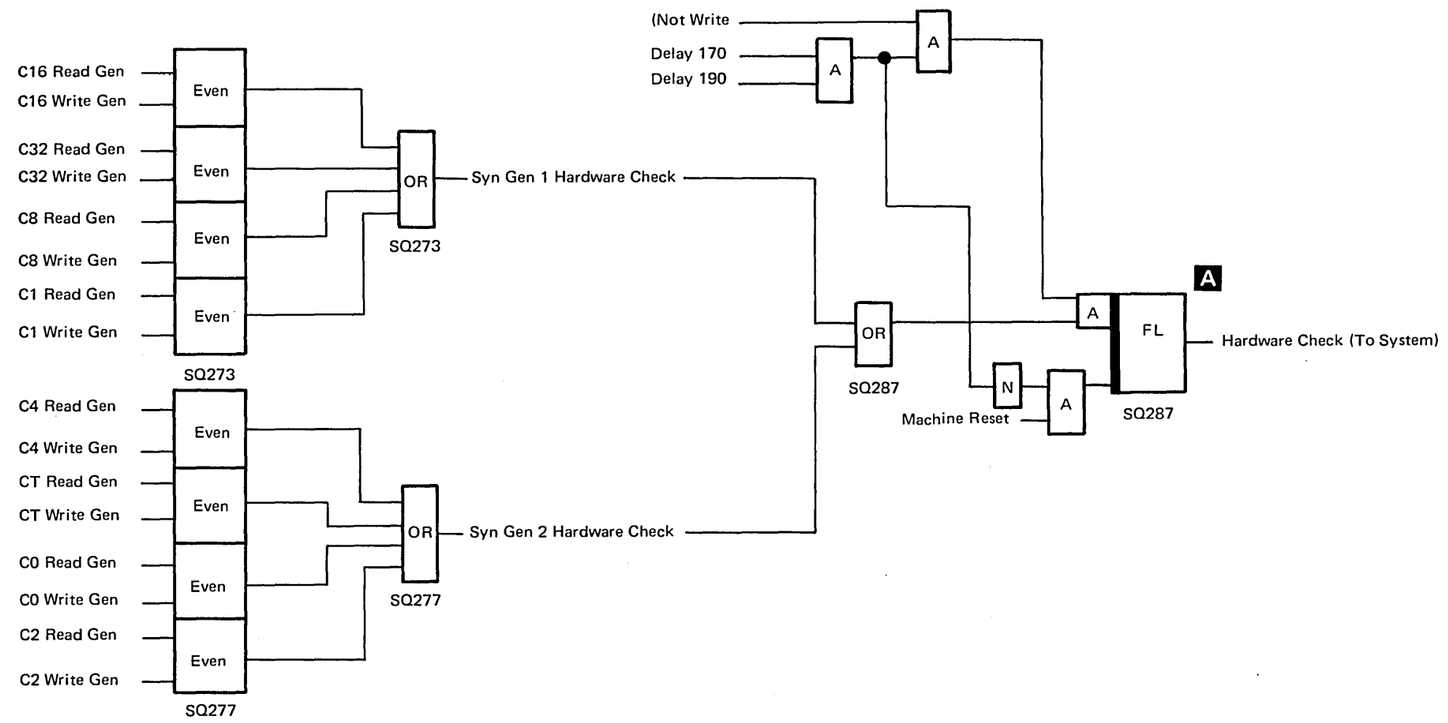
HARDWARE CHECKS

The hardware checking circuits check both data and control lines, plus the ECC hardware. The five checks are: ECC hardware check, byte check, data check, address check, and storage control line parity check. The logic that generates these control lines is here.

Hardware Check

Hardware Check **A** is activated when a logical error in either the read or write generator is detected. The check is performed only during the read cycle of a storage operation. If a hardware check in the ECC logic is detected; the MCKB-byte 2-bit 4 is set, and the ECC HDW light on the CPU console is lit.

Hardware Checks (Part 1 of 2)



Byte Check

On the byte check control line, B is activated when the parity generated on the four byte lines gated to the parity and generator do not compare to the byte-control line issued from the CPU. This kind of error sets MCKB byte 0, bit 3 the cancel latch in the storage clock logic and turns on the Stor byte marks parity check light on the CPU console.

Data Check

Data check control line C is activated when a parity error is detected on data received from the CPU during a store operation. The data check is performed in the parity-out generator. A data check sets the cancel latch within the storage clock logic, MCKB byte 0 bit 1, and turns on the SDBI parity check light on the CPU console.

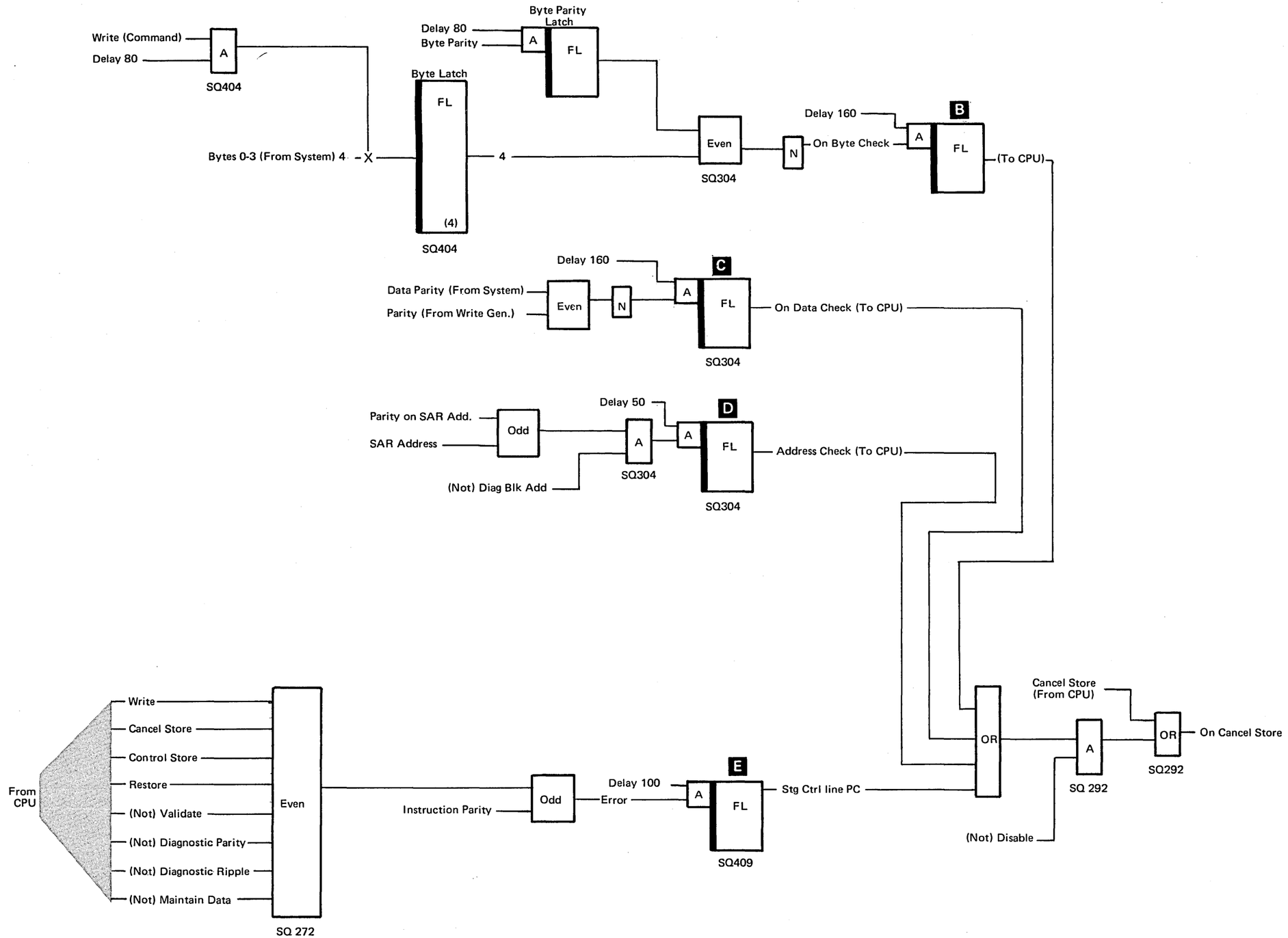
Address Check

Address check control line D is activated when a parity error is detected on the address lines received from the M-register. The address check is performed on the output of the storage address register. An address check sets MCKB, byte 0, bit 0, and turns on the M-Reg parity light on the CPU console. An address check also sets the cancel latch within the storage check logic.

Storage-Control-Line Parity Check

Storage control line parity check control line E is generated on the following instruetions issued to storage: Diagnostic Ripple; Write; Cancel Store; Validate; Diagnostic Parity; Control Store; Restore; and Maintain Data. If the generated parity bit does not compare with the instruction parity bit gated from the CPU, the MCKB, byte 2 bit 2, latch is set and the Stor Ctrl lines parity check light on the CPU console is lit, and the cancel latch within the storage clock logic is set.

Hardware Checks (Part 2 of 2)



STORAGE PROTECT

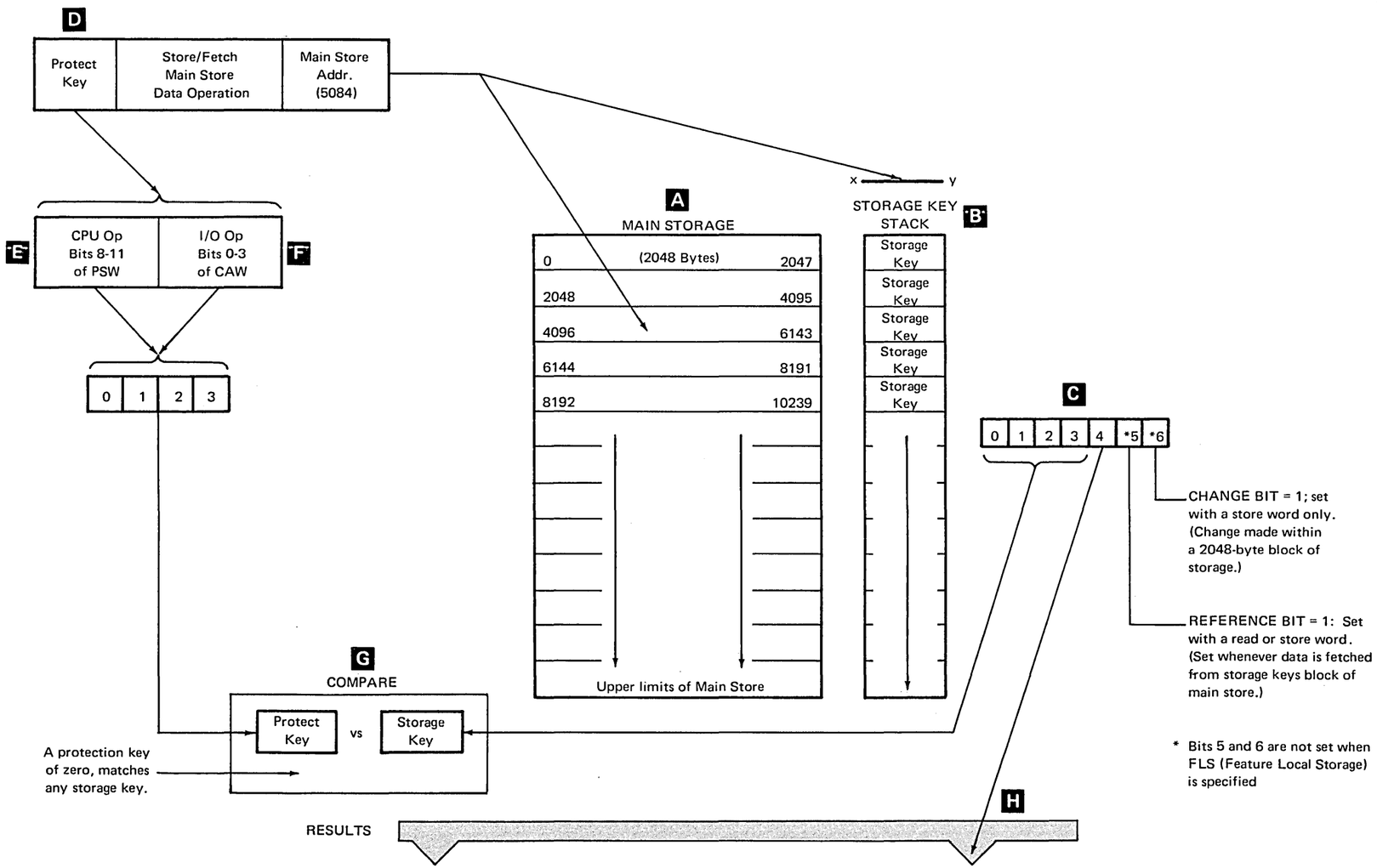
Storage-protect circuits safeguard sections of storage by preventing inadvertent or illegal access to the protected section during either store or fetch operations. To protect specific areas of storage, key bits are first stored in the array of the storage protect circuit by a write key operation. During a subsequent store or fetch operation, one of the prestored keys is accessed and compared with the key provided by the user. If the keys match, access to data storage is granted; if not, access is denied.

One storage key protects a 2,048-byte block of data within main storage **A** . The storage key **B** stored within the storage protect stack consists of seven bit positions **C** bits 0-3 are the four key bits; bit 4 indicates fetch protection; bit 5 is the reference bit; and bit 6 is the change bit. The storage keys are located in a storage-protect stack in monolithic storage. Each 64 x 8 stack holds 8-bit storage keys (seven information bits and one parity bit).

The protection key **D** is used as a comparand against the storage key to determine whether a match exists. When a store operation is specified by a CPU instruction, the protection key is in the current PSW (bits 8 through 11) **E** . When the reference is specified by a channel operation, the protection key is supplied by the channel address word (CAW) and is recorded in bits 0-3 of the channel status word **F** .

The first four bits of the storage key are compared bit by bit with the protect key in a compare circuit **G** . If the two keys do not match during a store operation, a protect violation is signaled and the store operation is prevented. For a fetch operation, the two four-bit keys are compared and, in addition, bit 4 is checked **H** . Bit 4 must be a 1 for a fetch protect. If the keys do not match and bit 4 is a 1, a protect violation is signaled and the addressed data is not fetched.

A key of all 0s acts as a master key and allows access to any block regardless of the stored key.



Functional Description

A drawing illustrating the overall data flow is shown on Page 3-43.

The 64 x 8 protection stack applies to main-storage locations (in sequential blocks of 2,048 bytes) zero through 131,072. Additional stacks are provided in the CPU when main-storage capacity exceeds 131,072 bytes. The particular stack-key byte referenced during a main-storage access is determined by address bits from the M-register, or address assembler. The bits used are M1-B7, M2-B0 through B4, (or address assembler byte 1-B7, M2-B0 through B4).

When protection applies to a storage reference, the key in the stack for that block of storage is compared with the protection key. Access to storage is permitted only when the keys match or when the protection key is zero. The matching mechanism depends on the system mode.

When a protection mismatch is encountered in an attempt to place information in main storage, the contents of the protected location remain unchanged. On fetching, the protected information is not loaded into an addressable register or moved to another storage location.

Protection mismatch due to an I/O operation is indicated in the channel status word stored as a result of that operation.

In macroprogramming the protection system is always active. It is independent of the problem, supervisor, and disabled states of the CPU and the type of instruction or I/O command being executed. In microprogramming, key matching is done only by a microcontrol word of the storage word (non K-addressable) type when C2 bits 4 and 5 of that control word are on. NOTE: following microword does not activate storage key: STW X W, *NOP*.

A key can be inspected by use of the insert storage-key instruction. A storage control word in the microprogram for this instruction reads the key from the stack position specified (by the main-storage address in the address-source location). The key is then stored in byte 3 of the location specified by the data-register field. Note that for this operation, the storage word must specify a read-key subform and the feature-local-storage special status set.

A storage control word with a store-key subform and a feature-local-storage special status set must be used to set a key in the storage-protect stack. The key is taken from byte 3 of the data-register location (specified in the storage word) and set into the stack at the location specified by the main-storage address used (from the location specified by the address-source field in the storage word).

A mismatch of keys during a storage-word operation, in which CPU protect mode is specified:

1. Cancels any store operation to main storage.
2. Prevents setting up the next control-word address.
3. Causes a protection trap to be initiated if no higher-priority traps occur at the same time.
4. Prevents address modification.
5. Prevents the contents of the data-register location (specified by the storage word) from being changed.

Insert Storage Key (ISK)

ISK (RR)

09	R_1	R_2

09 = Operation code in bits 0 through 7
R_1 = Register 1 in bits 8 through 11
R_2 = Register 2 in bits 12 through 15

The key of the storage block addressed by the register designated by R_2 is inserted in the register designated by R_1.

The storage block of 2,048 bytes, located on a multiple of the block length, is addressed by bits 8-20 of the register designated by the R_2 field. Bits 0-7 and 21-27 of this register are ignored. Bits 28-31 of the register must be zero; otherwise, a specification exception causes a program interruption.

The four high-order bits of the storage key and the fetch protection bit are inserted in bits 24-27 and 28 respectively of the register specified by the R_1 field. Bits 0-23 of this register remain unchanged, and bits 29-31 are set to zero.

Condition Code: The code remain unchanged.

Program Interruptions: Operation (if protection feature is not installed), Privileged operation, Addressing, Specification.

Set Storage Key (SSK)

SSK (RR)

08	R_1	R_2

08 = Operation code in bits 0 through 7
R_1 = Register 1 in bits 8 through 11
R_2 = Register 2 in bits 12 through 15

The key of the storage block addressed by the register desginated by R_2 is set according to the key in the register designated by R_1.

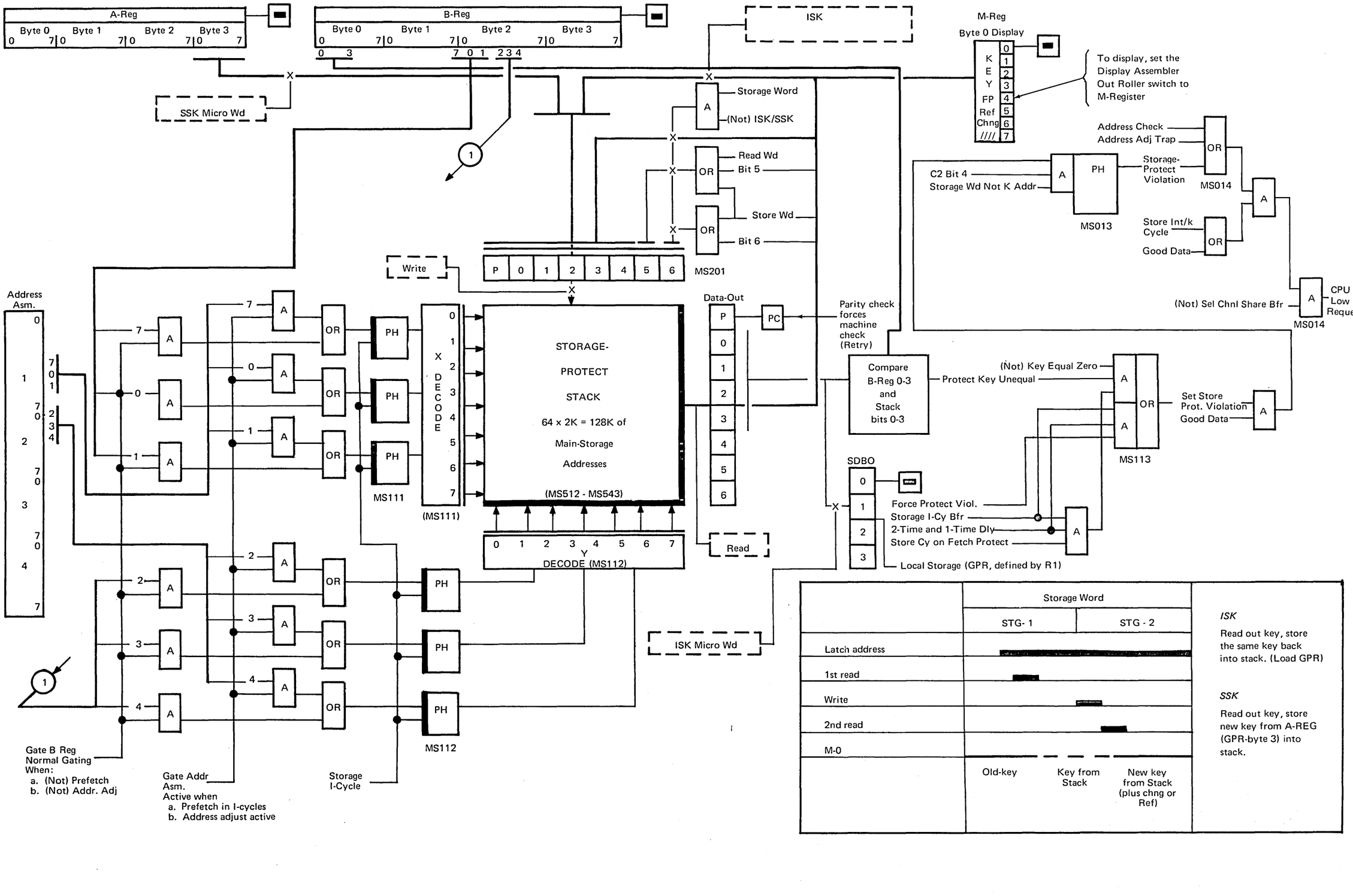
The storage block of 2,048 bytes, located on a multiple of the block length, is addressed by bits 8-20 of the register designated by the R_2 field. Bits 0-7 and 21-27 of this register are ignored. Bits 28-31 of the register must be zero; otherwise, a specification exception causes a program interruption.

The three low-order bits of the storage key are set to zero, the high-order four bits are obtained from bits 24-27 of the register designated by the R_1 field, and the fetch protection bits are obtained from bit 28 of the same register. Bits 0-23 and 29-31 of this register are ignored.

Condition Code: The code remains unchanged.

Program Interruptions: Operation (if protection feature is not installed), Privileged operation, Addressing, Specification.

Storage Protect



	Storage Word	
	STG- 1	STG - 2
Latch address		
1st read		
Write		
2nd read		
M-0		
	Old-key	Key from Stack / New key from Stack (plus chng or Ref)

ISK

Read out key, store the same key back into stack. (Load GPR)

SSK

Read out key, store new key from A-REG (GPR-byte 3) into stack.

FUNCTIONAL OPERATIONS

- The CPU select pulse starts a storage operation (read or write).
- Storage timing commences approximately 20 nanoseconds after the generations of the select pulse.
- The CPU specifies a read or write operation.
- The CPU provides the storage address along with the select pulse.
- A read (fetch) operation obtains data or instructions for the CPU.
- A write (store) operation puts data or instructions into the storage (BSMs) and is divided into two parts. A read operation is always followed by a write.

The CPU provides the ECCL board with read (not write) or write, an address, and a select (start) pulse. Select starts both the read and write operations. Once started, the ECCL board and storage proceed under control of the timing circuits.

The timing mechanism in the ECCL board and the BSMs is made of tapped delay lines and provides all the timed pulses necessary to control the storage operation. The ECCL board delay lines are basically used for generating control pulses for functional operations performed on the ECCL board. The delay line within the address buffer card, located in each BSM, generates the BSM timing pulses required by each array card during a read (fetch) or write (store) operation. The BSM select (board select) is the signal that starts the read or write function within the BSMs. The signal is generated in the BSM clock circuits located in the ECCL board (Page 3-24). Address lines in conjunction with the BSM select pulse determine which BSMs are to be selected. Parity check is made in the address lines received from the CPU in the parity-out generator. The address-check circuit is shown on Page 3-40.

To address either main or control storage, the CPU must generate the CPU select pulse that starts the delay line within the ECCL board. The time at which the CPU select pulse is generated is dependent upon the CPU cycle as shown on Page 3-22. The delay line in the ECCL board is always started after the generation of the CPU select pulse.

This CPU select pulse forms the ECCL select powered control that starts the delay line. Remember, all timing within the logic of the ECCL board is in relation to the select powered pulse, which by definition is T_0 of the ECCL board. The BSM select pulse is generated at approximately T_0+6 ns or T_0+8 ns, depending upon the address buffer card used with a particular BSM.

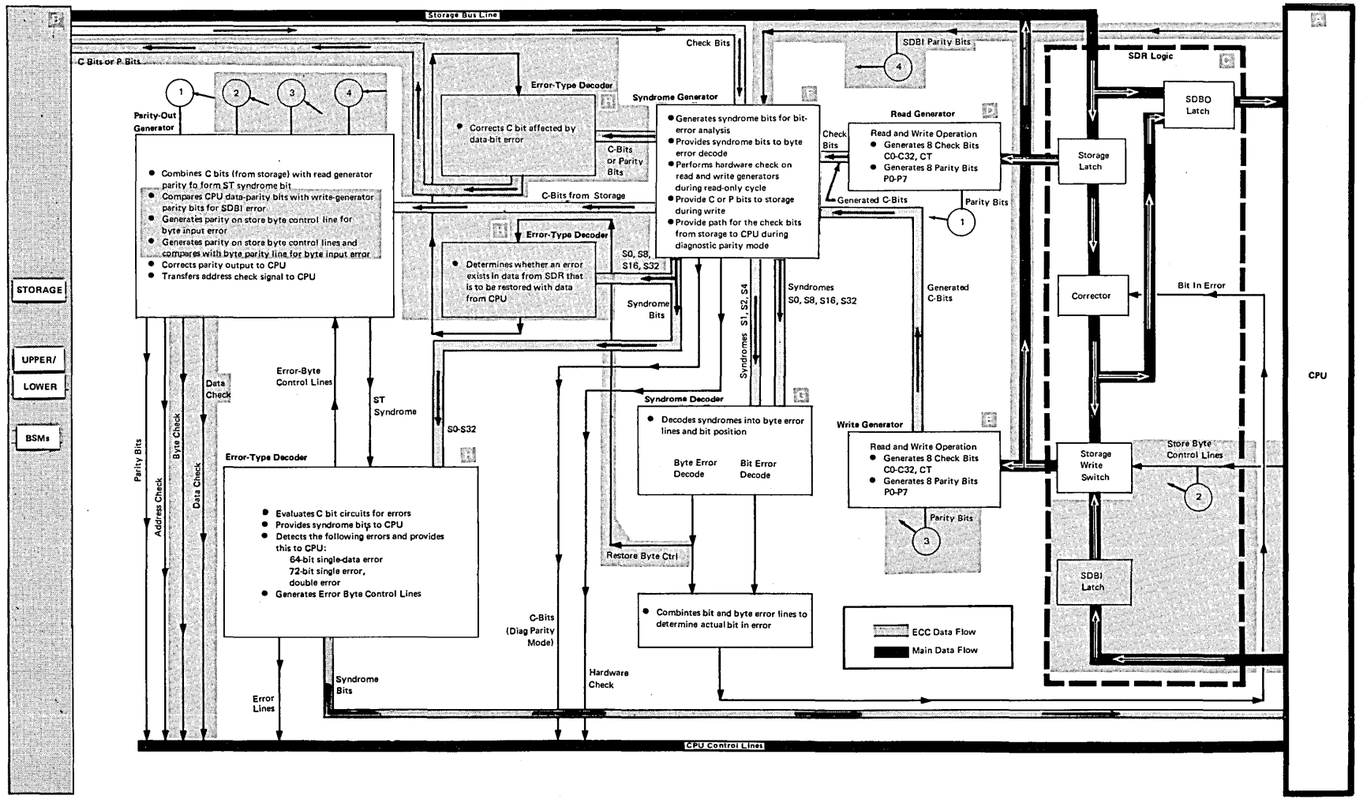
FETCH (READ) OPERATION-MAIN STORAGE

- A fetch operation, also referred to as a *read function*, fetches data from a pair of BSMs to the CPU by way of the SDR.
- ECCL timings are shown on Page 3-22.

A fetch (read) operation is started when the CPU sends to the ECCL board a select pulse and an address. The select pulse starts the ECCL delay-line pulses; the address lines are decoded to select a data word from an upper BSM (located at A2, B2, or C2), and a data word from a lower BSM (located at A4, B4, or C4). The description for a fetch follows the paths and reviews the events that occur during a normal fetch (read) operation.

1. The data (64 bits) from the storage location addressed by the SAR is read from storage **B** into the SDBO latches and to the storage latches **C** at approximately T_0+80 ns.
2. Eight check bits for the data fetched are read from storage into the syndrome generator **F** for bit-error analysis. The check bits from storage are also flushed through the syndrome generator to the parity-out generator **J**, as one of the inputs for ST syndrome generation.
3. The 64 data bits in the storage latches are gated to the read generator **D**, and to the write generator **E** via the storage write switches.
4. The read generators produce eight check bits (C0-C32, plus CT) that are forwarded to the syndrome generator **F** and eight parity bits that are forwarded to the parity-out generator **J**. The write generator produces eight check bits on the same data as the read generator and forwards these bits to the syndrome generator. The parity bits produce by the write generator during a read function are not used.
5. The syndrome generator performs the following.
   a. Hardware check: Compare the check bits generated in the read generator against the check bits generated in the write generator. Because the check bits in each generator were formed on the same data, an equal compare results if both circuits are functioning properly. An unequal compare represents a logical error in either the read or write generator and activates the hardware check line. A hardware check sets MCB byte 2, bit 4 and turns on the ECC HDW light on the CPU console.

Fetch (Read) Operation Data Flow



b. Compares the check bits from storage against the check bits from the read generator. If a check bit does not match, a syndrome is generated. Single errors are represented by syndromes S0, S1, S2, S4, S8, S16, S32, and ST. Double errors are represented by syndromes S0 through S32 and not ST. S0 through S32 are sent to the syndrome decoder G , and also to the error-type decoder H , where they are used to determine the kind of error. The syndromes are set at approximately T_0+170.
c. Latches the check bits from storage and forwards these bits to the CPU during diagnostic parity mode.

6. The syndrome bits gated to the syndrome decoder G are decoded into byte-error and bit-error lines. (S0, S8, S16, S32 decode the byte that contains a failing bit; S1, S2, S3 decode the bit in error.)
7. The byte-error lines and bit-error lines are gated to an SDR bit-error decoder located in the SDR logic. This decoder determines the actual bit in error and inverts this bit in the SDBO latches C via the corrector circuit associated with each bit.
8. ST syndrome bit is formed in the parity-out generator by comparing the C-bits from storage against the parity bits formed by the read generator. The ST bit is gated to the error-type decoder, where it is used for double-error detection.
9. If a single-bit error is corrected, the parity generated for the byte that contained the error must also be corrected. This parity correction occurs in the parity-out generator. Correct parity is gated to the CPU along with the data.
10. The syndrome bits formed at the syndrome generator that result from the read generator/storage check-bit comparison are available to the CPU (MCKB-byte 3) via the error-type decoder.
11. The error type decoder also detects the type of error (data, check bit, double bit) and provides this information to the CPU (MCKB byte 2).
12. Data from the SDBO latches, with correct parity is available on the SDBO at approximately T_0+237 ns.

FETCH OPERATION-CONTROL STORAGE

During control-storage accesses, data becomes available at the SDBO at the same time it is being analyzed for errors. This simultaneous action permits the CPU to operate on the SDBO data during the subsequent cycle if the data contains no errors. To prevent the use of the SDBO data before the error-checking circuits complete their analysis, a (Not) data good line is activated at T_0+100 ns and is reset at T_0+150 ns if no error is detected. Reset of the control line means that the control-storage data on the SDBO is correct and can be used by the CPU.

When a single-bit error exists in the control-storage area, the CPU waits an additional cycle to allow time for the ECC logic to correct the bit-in error before the SDBO data is used in the CPU. When the CPU detects that an error does exist in control storage, (Not) data good line remains active when CPU samples SDBO, a maintain data line is set, blocking setting or resetting of the SDBO latches during the period in which error correction occurs. Corrected data from control-storage becomes available in the SDBO at the same time as main-storage data (approximately T_0+237 ns).

For single-bit control-storage errors, a restore function is available, allowing correction of a control-storage bit that may be failing intermittently. A restore control line (Page 3-19) sets the storage write latch and the BSM select timing pulse at approximately T_0+210. This action allows a write function to occur during the cycle in which the CPU waits for the error correction to occur. This restore operation can be inhibited by using the tie-off on 01B-A2-B2D09.

STORE (WRITE) OPERATION

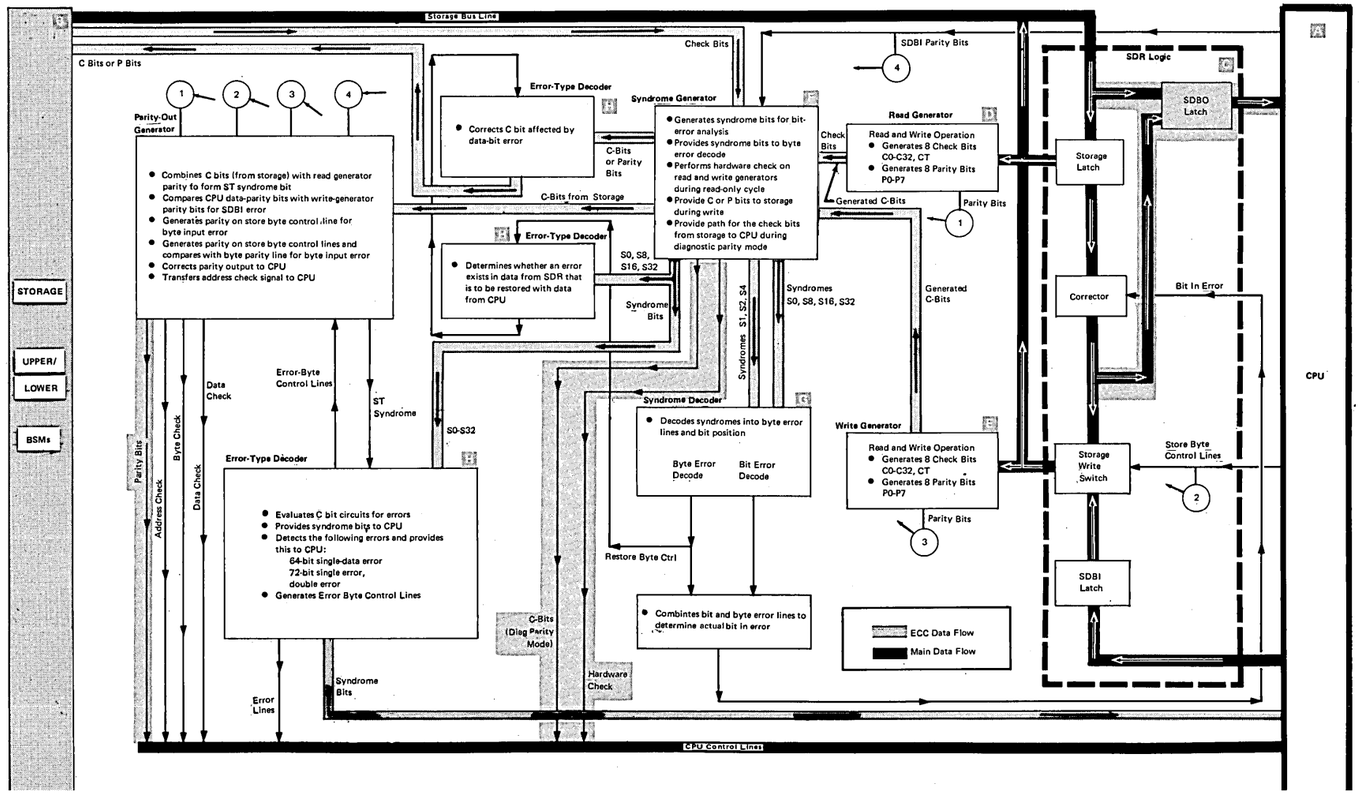
- A store operation (putting data into storage) referred to as a *write function,* fetches data from storage, uses new data from the CPU for bytes with byte marks, assembles a doubleword, and sends the data to the BSMs.

The store (write) operation requires a select pulse, write pulse, storage address, data bits, and storage byte-control lines. The select pulse and the write pulse initiate the store operation. The store operation always starts with a read function followed by a write. The description for a store follows the data paths and reviews the events that occur during a normal store (write) operation.

1. An address is sent from the CPU to the SAR, along with a select and a write pulse that enters the ECCL board at approximately T_0+80 ns and sets the system write latch within the storage clock logic (see Page 3-19). The output of the system write latch that initiates the write function is not used until T_0+210 ns. Therefore, with a select pulse active and the write pulse retained, a read function to the BSM is initiated by activating the BSM select pulse. This pulse is reset at approximately T_0+50 ns, and the SDBO latches and storage latches are reset at approximately T_0+150 ns.
2. Data sent by the CPU (up to four bytes) is received on the SDBI and gated to the SDBI latches **C** at approximately T_0+80 ns, and then transferred to the storage write switches. Byte-control lines from the CPU are also gated to the storage write switches. SDBI parity for the CPU data is gated directly to the parity-out generator **J** and syndrome generator **F**.
3. The 64 bits from the same address location at which the data is to be stored are gated into the SDBO latches and to the storage latches at approximately T_0+80 ns.
4. The data bits from the storage latches are merged with the data from the SDBI latched in the write switches, under byte control.
5. Eight check bits for the data fetched are also read from storage into the syndrome generator **F** for bit-error analysis. The same check bits are also flushed through the syndrome generator to the parity-out generator **J** as one of the inputs for ST syndrome generation.
6. The 64 data bits in the storage latches are gated to the read generator **D**; the data bits in the storage write switches are gated to the write generator **E**.
7. The read generator produces eight check bits (C0-C32, plus CT) that are forwarded to syndrome generator **F**, and eight parity bits that are forwarded to the parity-out generator **J**

   The write generator produces eight check bits (C0-C32, plus CT) on the data to be stored and forwards these bits to the syndrome generator. These check bits are stored with the data. Parity bits are also formed on the combination of old and new data and transferred to the parity-out generator **J** for comparison with CPU data parity bits for SDBI data check.
8. The syndrome generator performs the following.
   a. Compares the check bits from storage against the check bits from the read generator. If a check bit does not match, a syndrome is generated. Single errors are represented by syndromes S0, S1, S2, S4, S8, S16, S32, and ST. Double errors are represented by syndromes S0 through S32, and not ST. S0 through S32 are sent to the syndrome decoder **G** and also to the error-type decoder **H** where they are used to determine the kind of error and whether an error exists in data from the SDR that is to be restored with data from the CPU. The syndromes are set at approximately T_0+170.
9. The syndrome bits gated to the syndrome decoder **G** are decoded into byte-error and bit-error lines (S0, S8, S16, and S32 decodes the byte that contains a failing bit; S1, S2, and S3 decodes the bit in error).
10. The byte-error lines and bit-error lines are gated to an SDR bit-error decoder located in the SDR logic. This decoder determines the actual bit in error and inverts this bit in the write switches **C**, via the corrector circuit associated with each bit. This operation is valid if the error occurred in a byte that contains data to be restored along with the data from the SDBI latches.
11. Syndrome bits S0, S8, S16, S32 are also used to form byte-error decode lines, which are used to correct C-bits that have been previously generated in the syndrome generator on data that contains a bit in error **H**.
12. An ST syndrome bit is formed in the parity-out generator by comparing the C-bits from storage against the parity bits formed by the read generator. The ST bit is gated to the error-type decoder, where it is used for double-error detection.
13. The syndrome bits formed at the syndrome generator that result from the read generator/storage check-bit comparison are available to the CPU (MDKB-byte 3) via the error-type decoder.
14. The error type decoder also detects the kind of error (data, check bit, double bit) and provides this information to the CPU (MCKB-byte 2).
15. With corrections completed, the write cycle within the ECCL board is initiated at approximately T_0+210 ns and the entire contents from the storage write switches, plus the C-bits from the error-type decoder, are gated to storage. Reset of a storage control latch occurs at approximately T_0+275 ns.
16. During diagnostic parity mode, the parity bits from the CPU are gated to storage instead of the C-bits.

Store (Write) Operation Data Flow

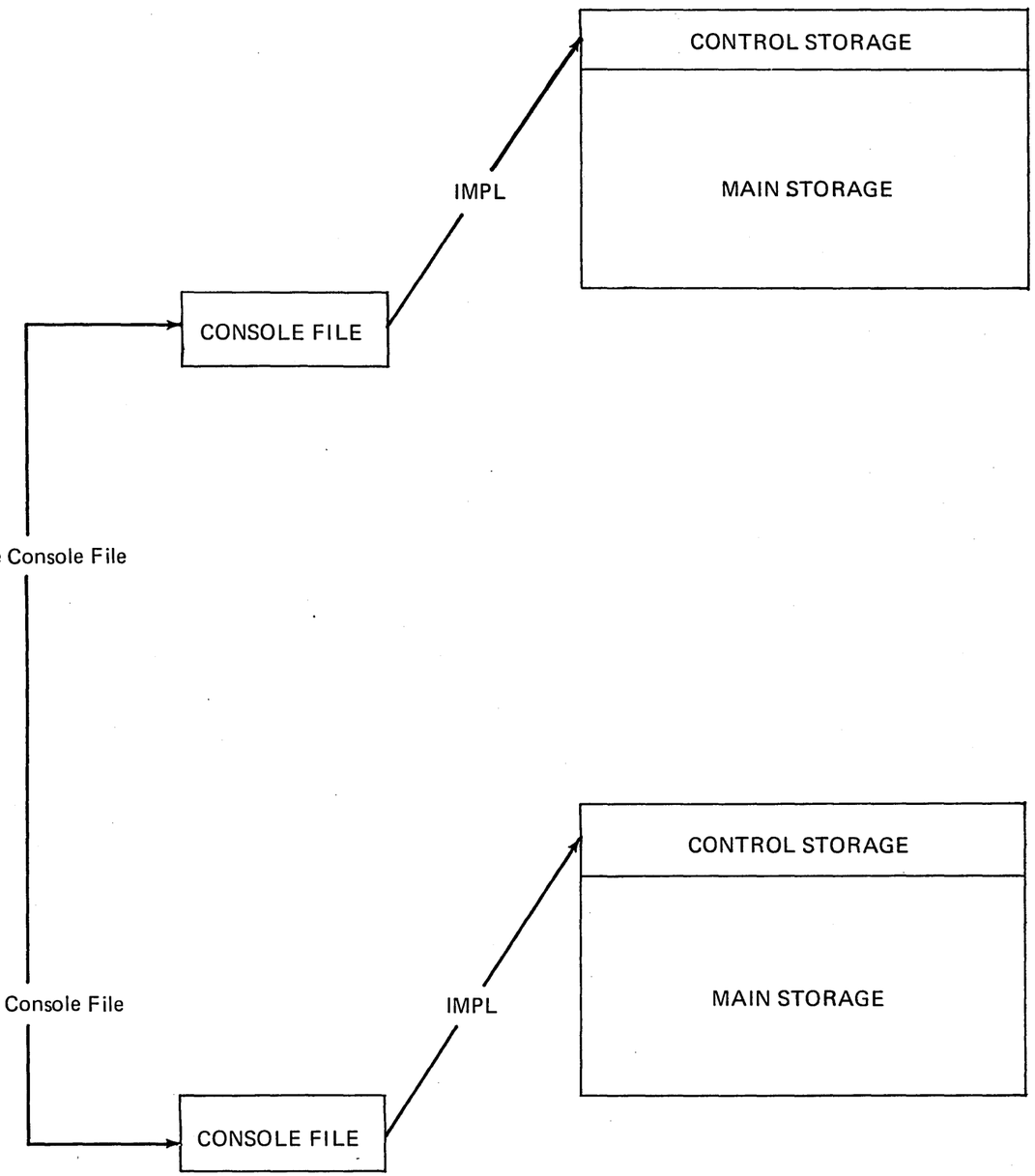


CHAPTER 4. MICROPROGRAM CONCEPTS

MICROPROGRAM CONCEPTS

- All functions performed by the 3145 are controlled by the 370 microprogram.
- Before any processing may begin, the 370 microprogram must be loaded into the control-storage area.
- The 370 microgrogram is loaded into control storage from a disk that is read by the console file.
- This loading process is called Initial Microprogram Program Load (IMPL).

  Refer to "Chapter 6, Console File" for details of the IMPL operation.

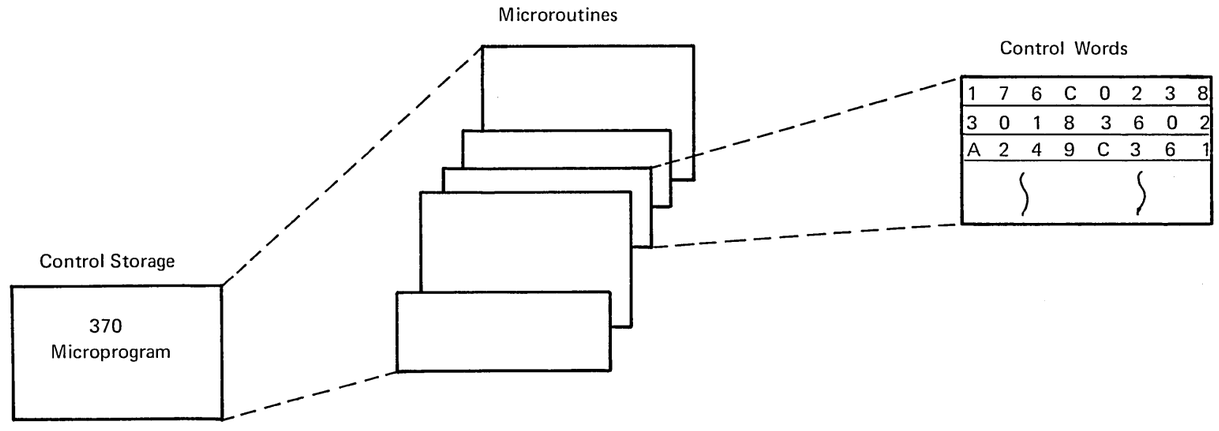


370 MICROPROGRAM

- The 370 microprogram is composed of *microroutines* of varying sizes, each having a specific task to perform.
- The 370 microprogram handles the processing of the instructions and data that are located in the main-storage area.
- Channel operations and the operations of the integrated devices are also handled by the 370 microprogram.
- Each microroutine is composed of bit-significant *control words* that handle particular functions that result in the execution of the specified task of the microroutine.

Each 370 microgrogram is customized to represent the system configuration that it is to control. These customized microprograms are assembled at the manufacturing area, written on a console file disk, and shipped with the system.

During the assembly of the 370 microprogram, a 370 microprogram listing is generated. This microgrogram listing is a representation of the 370 microprogram that is shipped with the system. Refer to "370 Microprogram Listing" for a detailed description.

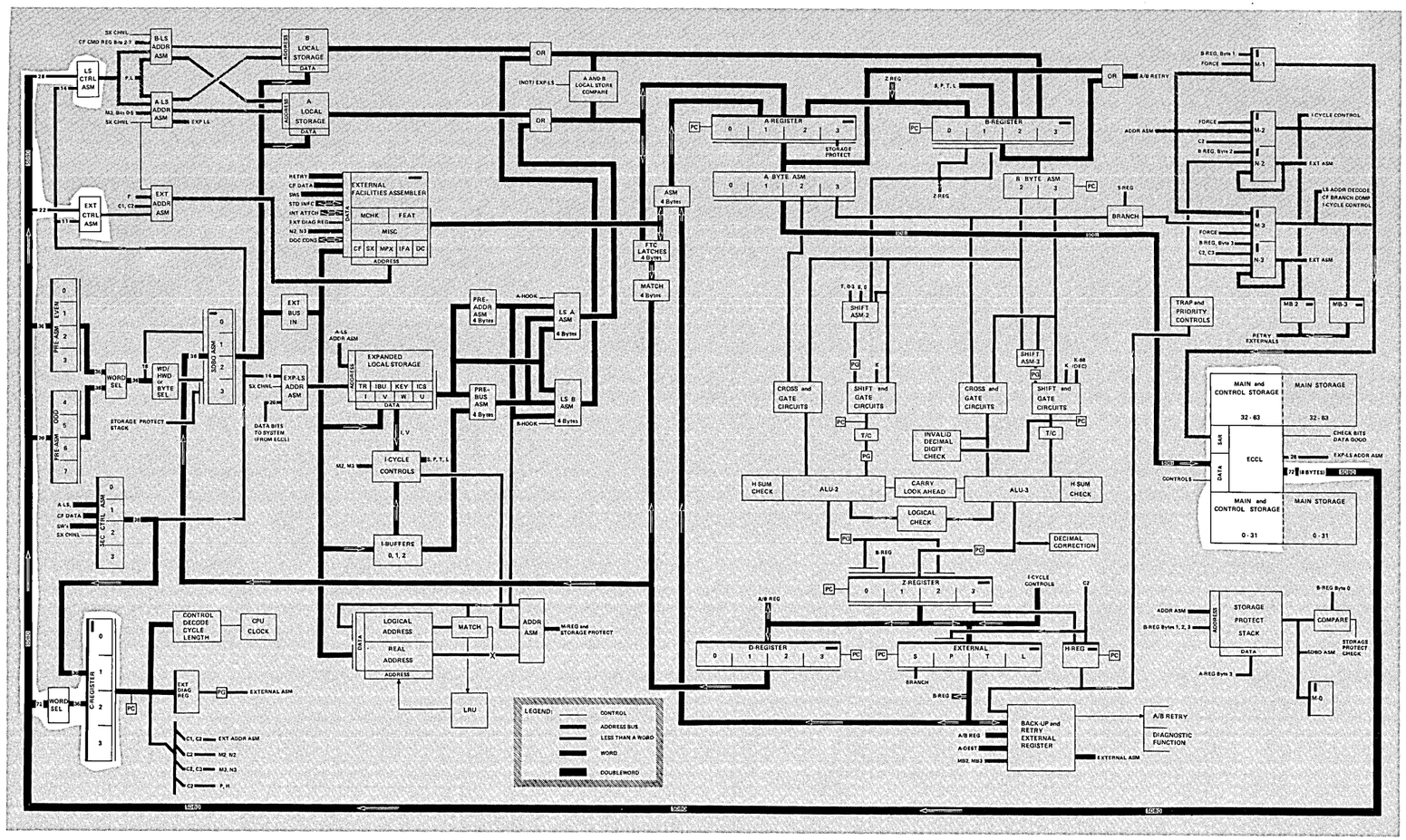


Control-Word Access

Once the 370 microprogram is in the control-storage area, a load operation (IPL) can be performed to place programs and data into main storage.

Assume that control and main storage have been loaded and that processing has begun;

- Control words are read out of control storage and gated out on the Storage Data Bus Out (SDBO).
- Portions of the control words are gated from the SDBO to either the local storage control assembler or the external control assembler. This is done to set up source addresses for these facilities early in the cycle.
- The control words are gated into the control register (C-reg), where they are decoded. The decoding of the control words brings up control and addressing lines that access and execute the programs located in main storage.

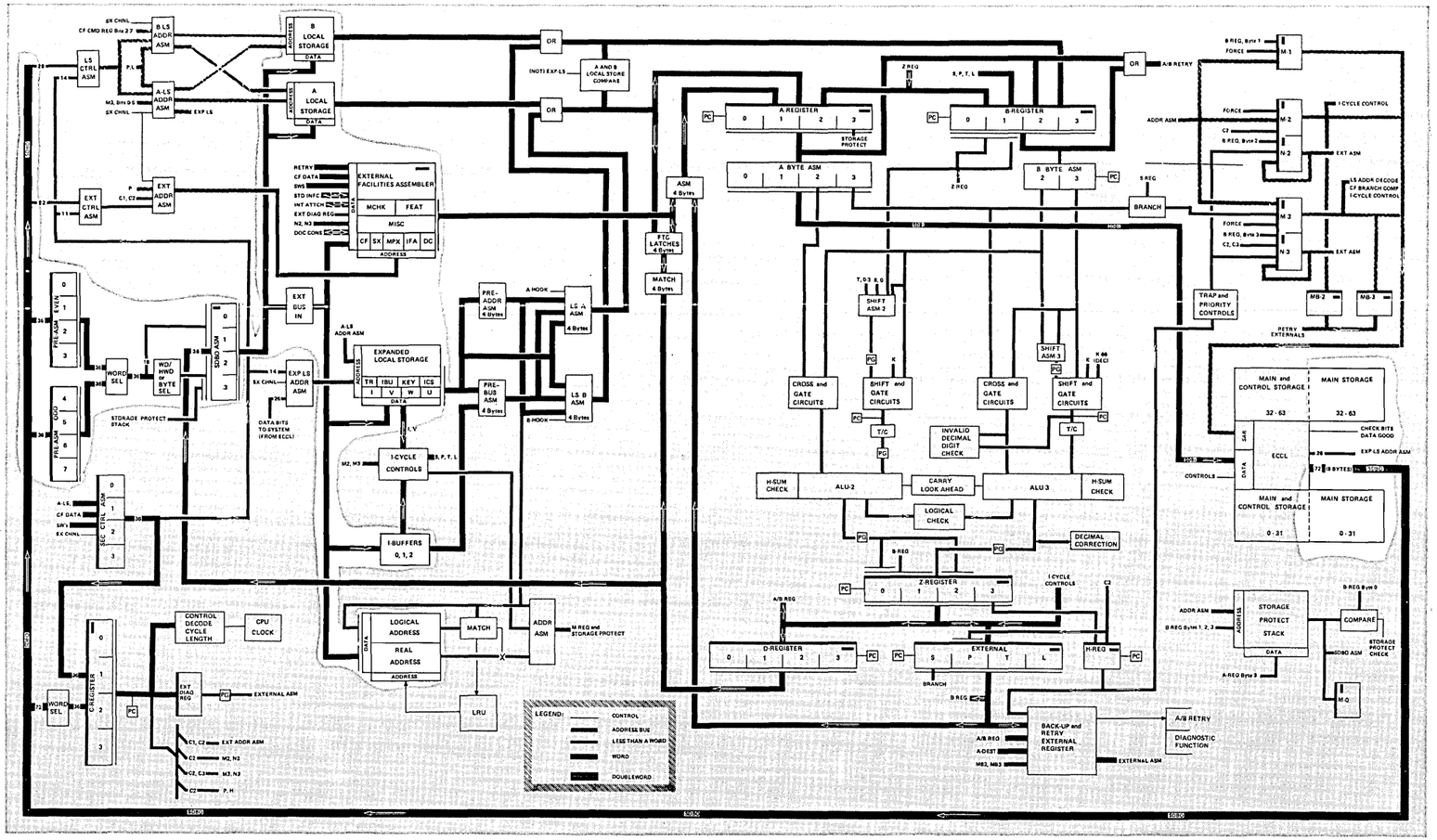


Main-Storage Access

When a control word performs a read operation on main storage, either instructions or data are accessed. All read operations, for control or main storage, result in a doubleword being accessed from storage.

Assume that a control word is performing a read operation on main storage;

- The doubleword from main storage is gated out on the SDBO to the SDBO pre-assembly latches.
- The odd or even address word, of the doubleword, is selected and gated to the SDBO assembler.
- If the word selected is a data word, it is gated to local storage or to some external facility.
- If the word is an instruction, it is gated to the I-buffers, expanded local storage, and in some cases, to the address adjustment circuits.



370 MICROPROGRAM LISTING

- The 370 microprogram listing is generated during the assembly of the 370 microprogram.
- This listing is a representation of the 370 microprogram that is located in the control-storage area after an IMPL is performed.

Symbolic Microprogram Input

The highlighted area of the sample page is the symbolic input written by the microprogrammer. This symbolic input is used by the microprogram assembler to generate the remaining portions of the microprogram listing.

```
GKDJ              DOCUMENTARY CONSOLE TEST I/O OR INTR ROUTINE          *370     EC 128603    PAGE   305/ OVERLAY  4

ADDR  WORD      NEXT  SEQ  I LABEL  LEG  NEXT SEQUENCE  NEXT LABEL  STAT STATEMENT          COMMENTS                    WA1-WA2
86E4  101E01F0  86F0  0096    DECYC  1    0104           STUNST           MBS3,OR,K10        DE CYC,SET BSY UN.STA       L19
86F0  F3606CF4  86F4  0104    STUNST      0104           STUNST           MW3=MW0          ****MOVE UNIT STA TO MW3      LIE LIE
86F4  01D0785B  7858  0105                0106,0111      CONSTA B3        SYSO               CONSOLE PTR 1 OR 2          E05
7858  69626061  786C  0106    CONSTA 0    0106,0111      CONSTA B3        STB MW3 DC,60      STORE UNIT STATUS OF        LIE
                    0107 *                                                                  CONSOLE NO. 1
786C  F3646C64  7864  0108                                                 MW3=MW1            MOVE SENSE                  LIE LIE
7864  69626460  7868  0109                0117           SWITCH           STB MW3 DC,64      STORE SENSE BYTE OF         LIE
                    0110 *                                                                  CONSOLE NO. 1
785C  68626851  7854  0111    CONSTA 1                                                       STB MW3 DC,68      STORE UNIT STATUS OF
                    0112 *                                                                                     REMOTE CONSOLE
7854  F3646C60  7860  0113                                               MW3=MW1            MOVE SENSE                  LIE LIE
7860  69626C60  7868  0114                0117           SWITCH           STB MW3 DC,6C      STORE SENSE BYTE OF         LIE
                    0115 *                                                                  REMOTE CONSOLE
7868  0000B690  B690  0117    SWITCH      GMSW 0002      START                               STORE CSW

                                CROSS REFERENCE FOR ROUTINE GKDJ

GKDJ 0028   GKDI 0164
GKDJ 0029   GKDJ 0028
GKDJ 0030   GKDJ 0029
GKDJ 0031   GKDJ 0030
GKDJ 0032   GKDJ 0031
GKDJ 0033   GKDJ 0032
GKDJ 0034   GKDJ 0033
GKDJ 0052   GKDJ 0030
GKDJ 0054   GKDJ 0052
GKDJ 0055   GKDJ 0029
GKDJ 0056   GKDJ 0028   GKDJ 0054  GKDJ 0055
GKDJ 0057   GKDJ 0055
GKDJ 0076   GKDJ 0031
GKDJ 0077   GKDJ 0032
GKDJ 0078   GKDJ 0076   GKDJ 0077
GKDJ 0079   GDKJ 0056   GKDJ 0078
GKDJ 0094   GDKJ 0033
GKDJ 0096   GKDJ 0052
GKDJ 0104   GKDJ 0057   GKDJ 0079  GKDJ 0095  GKDJ 0096
GKDJ 0106   GKDJ 0105
GKDJ 0111   GKDJ 0105
GKDJ 0117   GKDJ 0109   GKDJ 0114
```

SAMPLE PAGE FROM 370 MICROPROGRAM LISTING

Microprogram Assembler Output

The highlighted area of the sample page is the generated output of the microprogram assembler. This output is the result of reading and decoding the symbolic information submitted to the assembler by the microprogrammer.

GKDJ DOCUMENTARY CONSOLE TEST I/O OR INTR ROUTINE *370 EC 128603 PAGE 305/ OVERLAY 4

ADDR	WORD	NEXT	SEQ	I	LABEL	LEG	NEXT SEQUENCE	NEXT LABEL	STAT	STATEMENT	COMMENTS	WA1-WA2
86E4	101E01F0	86F0	0096		DECYC	1	0104	STUNST		MBS3,OR,K10	DE CYC,SET BSY UN.STA	L19
86F0	F3606CF4	86F4	0104		STUNST					MW3=MW0	****MOVE UNIT STA TO MW3	LIE LIE
86F4	01D0785B	7858	0105				0106,0111	CONSTA B3		SYSO	CONSOLE PTR 1 OR 2	E05
7858	69626061	786C	0106		CONSTA	0				STB MW3 DC,60	STORE UNIT STATUS OF	LIE
			0107	*							CONSOLE NO. 1	
786C	F3646C64	7864	0108							MW3=MW1	MOVE SENSE	LIE LIE
7864	69626460	7868	0109				0117	SWITCH		STB MW3 DC,64	STORE SENSE BYTE OF	LIE
			0110	*							CONSOLE NO. 1	
785C	68626851	7854	0111		CONSTA	1				STB MW3 DC,68	STORE UNIT STATUS OF	
			0112	*							REMOTE CONSOLE	
7854	F3646C60	7860	0113							MW3=MW1	MOVE SENSE	LIE LIE
7860	69626C60	7868	0114				0117	SWITCH		STB MW3 DC,6C	STORE SENSE BYTE OF	LIE
			0115	*							REMOTE CONSOLE	
7868	0000B690	B690	0117		SWITCH		GMSW 0002	START			STORE CSW	

CROSS REFERENCE FOR ROUTINE GKDJ

GKDJ 0028	GKDI 0164			
GKDJ 0029	GKDJ 0028			
GKDJ 0030	GKDJ 0029			
GKDJ 0031	GKDJ 0030			
GKDJ 0032	GKDJ 0031			
GKDJ 0033	GKDJ 0032			
GKDJ 0034	GKDJ 0033			
GKDJ 0052	GKDJ 0030			
GKDJ 0054	GKDJ 0052			
GKDJ 0055	GKDJ 0029			
GKDJ 0056	GKDJ 0028	GKDJ 0054	GKDJ 0055	
GKDJ 0057	GKDJ 0055			
GKDJ 0076	GKDJ 0031			
GKDJ 0077	GKDJ 0032			
GKDJ 0078	GKDJ 0076	GKDJ 0077		
GKDJ 0079	GDKJ 0056	GKDJ 0078		
GKDJ 0094	GDKJ 0033			
GKDJ 0096	GKDJ 0052			
GKDJ 0104	GKDJ 0057	GKDJ 0079	GKDJ 0095	GKDJ 0096
GKDJ 0106	GKDJ 0105			
GKDJ 0111	GKDJ 0105			
GKDJ 0117	GKDJ 0109	GKDJ 0114		

SAMPLE PAGE FROM 370 MICROPROGRAM LISTING

A GKDJ DOCUMENTARY CONSOLE TEST I/O OR INTR ROUTINE *370 EC 128603 PAGE 305/ OVERLAY 4

B

ADDR	WORD	NEXT	SEQ	I	LABEL	LEG	NEXT SEQUENCE	NEXT LABEL	STAT	STATEMENT	COMMENTS	WA1-WA2
86E4	101E01F0	86F0	0096		DECYC	1	0104	STUNST		MBS3,OR,K10	DE CYC,SET BSY UN.STA	L19
86F0	F3606CF4	86F4	0104		STUNST					MW3=MW0	****MOVE UNIT STA TO MW3	LIE LIE
86F4	01D0785B	7858	0105				0106,0111	CONSTA B3		SYSO	CONSOLE PTR 1 OR 2	E05
7858	69626061	786C	0106		CONSTA	0				STB MW3 DC,60	STORE UNIT STATUS OF	LIE
			0107	*							CONSOLE NO. 1	
786C	F3646C64	7864	0108							MW3=MW1	MOVE SENSE	LIE LIE
7864	69626460	7868	0109				0117	SWITCH		STB MW3 DC,64	STORE SENSE BYTE OF	LIE
			0110	*							CONSOLE NO. 1	
785C	68626851	7854	0111		CONSTA	1				STB MW3 DC,68	STORE UNIT STATUS OF	
			0112	*							REMOTE CONSOLE	
7854	F3646C60	7860	0113							MW3=MW1	MOVE SENSE	LIE LIE
7860	69626C60	7868	0114				0117	SWITCH		STB MW3 DC,6C	STORE SENSE BYTE OF	LIE
			0115	*							REMOTE CONSOLE	
7868	0000B690	B690	0117		SWITCH		GMSW 0002	START			STORE CSW	

C

CROSS REFERENCE FOR ROUTINE GKDJ

GKDJ 0028	GKDI 0164			
GKDJ 0029	GKDJ 0028			
GKDJ 0030	GKDJ 0029			
GKDJ 0031	GKDJ 0030			
GKDJ 0032	GKDJ 0031			
GKDJ 0033	GKDJ 0032			
GKDJ 0034	GKDJ 0033			
GKDJ 0052	GKDJ 0030			
GKDJ 0054	GKDJ 0052			
GKDJ 0055	GKDJ 0029			
GKDJ 0056	GKDJ 0028	GKDJ 0054	GKDJ 0055	
GKDJ 0057	GKDJ 0055			
GKDJ 0076	GKDJ 0031			
GKDJ 0077	GKDJ 0032			
GKDJ 0078	GKDJ 0076	GKDJ 0077		
GKDJ 0079	GDKJ 0056	GKDJ 0078		
GKDJ 0094	GDKJ 0033			
GKDJ 0096	GKDJ 0052			
GKDJ 0104	GKDJ 0057	GKDJ 0079	GKDJ 0095	GKDJ 0096
GKDJ 0106	GKDJ 0105			
GKDJ 0111	GKDJ 0105			
GKDJ 0117	GKDJ 0109	GKDJ 0114		

SAMPLE PAGE FROM 370 MICROPROGRAM LISTING

A The first and last lines of every page of a microroutine contain the name and title of that microroutine. The microprogram name, EC number, page number, and overlay instruction are also found on these lines.

Overlays are specified by number on the applicable pages in the microlistings. Overlay 1 is the first group of control words loaded with control storage and executed. After Overlay 1 is completed, Overlay 2 is loaded into control storage and executed. This process continues until the final overlay (370 microprogram) is loaded.

B The second line of a microroutine page contains the column titles for the information that follows.

ADDR

This is the actual address in control storage that the control word listed under the WORD column is located. This address is assigned to the control word at the time the microprogram is assembled.

In some diagnostic-type microroutines, the address column is labeled ADDR
M/LS
appearing in this column can be either the 2 hex digit console file command, or the M3 register value and local-storage word address of the control word loaded from the console file for execution in local-storage control-storage mode (LSCS). For example, in the ADDR column, an entry of B8/2E is noted. The B8 value is the M3-register setting that addresses LS location 2E when operating in LSCS mode. The control word located in LS2E is read out, placed in the C-register, and executed.

WORD

This is the hex representation of the control word that was developed by the microprogram assembler from symbolic information that appears in the Label, Next Label, Stat, and Statement columns.

NEXT

This address is normally the actual address being branched to by the control word located on this line. If the branch is being made to a branch set, the address in this column is the address of the zero leg of that branch set.

SEQ

The number appearing here is the actual sequence number of the statement in the routine. Sequence number 1 is the title line number and does not appear. The numbers start with 2 and may continue up to 9999.

Occasionally there are missing sequence numbers in this column. This indicates that some control word or words have been selected out of this microroutine because the feature mix of the system does not require them.

I

This column identifies the printed line. Blanks identify control words, and asterisks identify full-line comments. For additional items that are identified in this column, see "Unique Instructions."

LABEL LEG

This column contains the symbolic name and leg identifier of the statement that appears on the same line. The leg identifier may be 1,2, or 3 characters depending on the size of the branch set the control word belongs to. If the word is not a member of a branch set, no leg identifier appears.

If words in the listing have nothing in this column, they are to be executed in sequence. However, these words do not necessarily have sequential addresses. The microprogram assembler assigns Labels and Next Labels to words that do not have them, in order to accomplish the function of address assignment. These assembler-assigned labels do not appear on the listing; they are used only during the assembly of the microprogram.

NEXT SEQUENCE

The sequence number of the word being branched to by the control word located on this line is printed in this column. If the control word is branching to a branch set, multiple sequence numbers must be shown. Up to two sequence numbers can be printed in this column on the same line as the control word performing the branch. If more than two sequence numbers are needed to show the branch possibilities, the sequence numbers are printed on the next line. This additional line is flagged by a row of asterisks leading the line.

If the branch is being made out of the current microprogram routine, the name of the microprogram routine being branched to appears in the next sequence column. If additional lines were needed, the name of the microprogram routine being branched to appears there also.

NEXT LABEL

The Next Label column contains the symbolic information that produces the next control-word address bits for branch operations. The symbolic name of the word being branched to, and the branch test fields used to select the leg in the branch set, appear in this column. If the word being branched to is not a part of a branch set, the branch test symbols do not appear. The symbols that are used in these fields may be the fixed type (0 or 1) or the type that designates a test of some kind. There may be up to three branch test fields, separated by commas, to the right of the name in this column. If only one branch field appears, it is the branch low field, and it affects bit 5 of M3 during the next address formation. If there are two branch fields, the rightmost is the low and affects bit 5 of M3; the other is the branch high field and affects bit 4 of M3. If three fields are present, the rightmost affects bit 5, the next affects bit 4, and the other affects bits 2 and 3 of M3. X's are used in the branch fields for alignment only. The X means nothing to the next address, but allows the microprogram assembler to assign the proper bits in the branch fields of the control word.

STAT

The stat column identifies status-set information. This column is used only with the arithmetic and storage control words.

STATEMENT

The statement column contains the symbolic information that indicates the action to be taken on data contained in local storage, main storage, or an external register. In some cases only a functional action is indicated by the statement. In other statements both data-handling and functional action are designated. The statement indicates the the microprogram assembler the control-word type that must be generated to accomplish the specific function the microprogrammer wishes to perform.

The manner in which the statement is written, the symbols used, and the Next Label field, determine the control-word type is easily recognized in some statements, but others are not as easily distinguishable. The first hex digit of the control word can be used to determine the control-word type produced by the assembler. In some cases the bit chart for that word type must be used to determine the total function performed by the word.

COMMENTS

The Comments field is used to describe the function being performed by the control word as it pertains to the execution of a major function. A double asterisk appearing in this field allows the microprogrammer more comments space by starting in the area reserved for the statement. Full-line comments are flagged by an asterisk to the right of the sequence number.

WA1-WA2

The WA1-WA2 fields are used to designate the word addresses of the symbols used in the statement field. WA1 contains the address of the leftmost addressable symbol found in the statement field. WA2 contains the rightmost addressable symbol found in the statement field. If a local-storage register is specified as a source or destination, the symbol printed in either field is Lhh. Where L specifies local storage, and hh is the hexadecimal address of the local-storage register. If an external is specified as a source or a destination, the symbol Ehh is printed. Where E specifies external, and hh is the hexadecimal address of that external. If indirect word addressing is specified, the symbol printed in either field is IND.

Certain selector channel routines are shared by all selector channels. The addresses, both local storage and external, found in the WA1 and WA2 columns are for one selector channel. The setting of selector-channel hardware determines the actual address used.

C At the end of each microroutine, there is a cross-reference listing for that microroutine that is useful in determining the entry points for this microroutine.

The first column of this list contains the sequence number of every control word in the microroutine that has a name in the Label column.

Adjacent to each entry in the first column is the microroutine name and sequence number of any control word that could branch to the control word whose sequence number is in the first column.

If there are multiple references to a control word, they are listed on the same line as the referenced word.

If there are too many references to fit on one line, multiple lines are used. The additional lines start with the first entry aligned with the second column of the cross-reference listing.

Assembler Instructions Found in the Microprogram Listings, (Part 1)

These instructions are used by the microprogram assembler during the assembly of the 370 microprogram. Although these instructions appear in the microprogram listings, their primary purpose is for use by the 370 microprogrammers.

The first letter of all these instructions appears in the I column of the microprogram listings.

AWORD	ADDR=hhhh
AWORD	ADDR=xx,hh
AWORD	ADDR=hh,xx
ATABLE	ADDR=hhhh
ATABLE	ADDR=xx,hh
AEND	
ASEQ	ADDR=hhhh
ASEQ	
H	hhhhhhhh
Maaaaaa	ssssbbbbbb
L	hhhh
RESERVE	
RESERVE	hhhh
RESERVE	hhhhTHRUhhhh
RESERVE	FROMhhhhTHRUhhhh
K	CPGM'cccccc---------c'
K	XPGM'hhhh----------h'

This instruction assigns a complete address of **hhhh** to the control word or hex data that appears below this instruction in the microprogram listings.

This instruction assigns only the word address **hh** to the control word or hex data that appears below this instruction in the microprogram listings. The X's allow the microprogram assembler to assign the word any module address. The two **hh digits must specify a word boundary address.**

This instruction assigns only the module address **hh** to the control word or hex data that appears below this instruction in the microprogram listings. The X's allow the **microprogram assembler to assign any word address.**

This instruction starts a string of address assignments, starting with the control word or hex data that follows this instruction. The address **hhhh** is assigned to the first word, and the address is incremented by 4 for each subsequent word assigned.

This instruction starts a string of address assignments starting with the control word or hex data that follows this instruction. The address **hh** is assigned to the first word, and the address is incremented by 4 for each subsequent word assigned. The **X's** allow the microprogram assembler to assign the word any module address.

This instruction terminates any **ATABLE** assignment.

This instruction is similar to the **ATABLE** instruction in that all words following the **ASEQ** instruction are assigned starting at address **hhhh** and incremented by 4 until the **AEND** instruction is reached.

This instruction starts a string of address assignments starting with the next address of the LAST **ASEQ** table. This instruction would be used to locate data in different routines.

This instruction reselts in the data word **hhhhhhhh** being stored in control storage. The address that the data word is stored into is assigned by a preceding **AWORD ADDR=hhhh** or **ATABLE ADDR=hhhh** instruction.

This is a Module Equate instruction. This instruction appears on the listing, but is not a part of control storage. The Module Equate is used to indicate to the microprogram assembler that the two control words whose labels are given must be assigned addresses in the same module. In the instruction format, the labels **aaaaaa** and **bbbbbb** are assigned the same module address. If the label **bbbbbb** was in another routine, its name would appear in the **ssss** field.

The Link instruction sets up a return word in control storage. An Assign instruction with full address precedes this kind of instruction. This return information may be moved into local storage and used by a standard Return Control Word. The **hhhh** field represents the value that is set into the S and P registers.

The **Reserve** instructions are used to reserve blocks of control storage that are to be used by the microprogram.

These instructions are used by the microprogrammer to indicate a constant to be stored in main storage by the microprogram assembler. The **K CPGM cc---c** type must be printable characters. The **K XPGM hh---h** type must be valid hex characters.

Assembler Instructions Found in the Microprogram Listings, (Part 2)

VLABEL1	LABEL2	
FORMAT=LOCSTG		This instruction is used to obtain the address of constants in program storage.
FORMAT=DSECT		This instruction causes the assembler to format the output for loading into local storage instead of control storage.
FORMAT=DSECT-END		This instruction creates a dummy section, by ignoring all hex, assign, and control-word statements. This function is used only by the selector-channel microdiagnostics.
FORMAT=CONS-FILE		This instruction terminates the FORMAT=DSECT instruction.
FORMAT=STANDARD		This instruction causes the assembler to change the page header on the microprogram listing to represent instructions to be executed from the console file.
FORMAT=NO-REF		This instruction terminates the control of the FORMAT=CONS-FILE instruction.
FORMAT=XREF		This instruction is used to suppress the printing of the cross-reference list at the end of each microroutine.
OVERLAY		This instruction terminates the FORMAT=NO-XREF instruction.
OVERLAY	±N ADDR=hhhh,ACB+LOW-CS-ADDR	This instruction is used only in the microprogram index. Its function is to separate microroutines that are to be assembled separately.
OVERLAY	±N ADDR=hhhh,HIGH-MS-ADDR	In each of the following OVERLAY instructions, the address may be either a local-storage or control-storage location. Local storage is specified by the symbol LShh, and control storage is specified by hhhh, where the symbol h is a hexadecimal digit.
OVERLAY	±N ADDR=hhhh,CHECKSUM	This instruction places the two-byte ACB register in the two high-order bytes of the address indicated, and the low control-storage address in the two low-order bytes of the address indicated. The overlay in which the data is stored is specified by the ±N symbol. Where N is the number of overlays to be skipped in either the forward or backward direction, depending on the + or - sign.
OVERLAY	±N ADDR=hhhh,EC-LEVEL	This instruction operates the same as the one above except that the 20-bit main-storage address is stored in the low 20 bits of the address specified.
OVERLAY	±N ADDR=hhhh,CPU-ID	This instruction operates the same as the other overlay instructions except that the checksum is stored at the address specified.
W		This instruction operates the same as the other overlay instructions except that the EC-level of the microprogram is stored at the address specified.
C		This instruction operates the same as the other overlay instructions except that the CPU-ID is stored at the address specified.
		This instruction, used with console-file operations, indicates that the word on this line must be translated from the symbolic statement. Refer to "Chapter 6, Console File," for descriptions of the various console-file commands.
		This instruction, used with console-file operations, indicates that the word of data on that line is a hexadecimal word that requires no translation.

Control-Word Address Generation

The 3145 processing unit does not have an automatic means to increment control-storage addresses during the execution of the 370 microprograms. This means that each control word must set up the address of the next control word to be executed. Each control word therefore has the capability to branch.

Some control words are capable of branching to any address in control storage. The other control words can branch only within the 64-word address module in which they are located.

The procedure for setting the next control-word address is covered in detail in the descriptions of the individual control words.

Some of the terminology used in the control-word descriptions is identified and described here.

Branch Set

A branch set is composed of 2-16 control words having the same symbolic name. Each control word in the branch set is identified by a leg identifier that is assigned to the control word by the microprogrammer. The leg identifier fixes a portion of the control-word address. All legs of a branch set are assigned addresses in the same address module.

Examples:

ADDR	LABEL
C2F0	MPYINV 00
C2F4	MPYINV 01
C2F8	MPYINV 10
C2FC	MPYINV 11

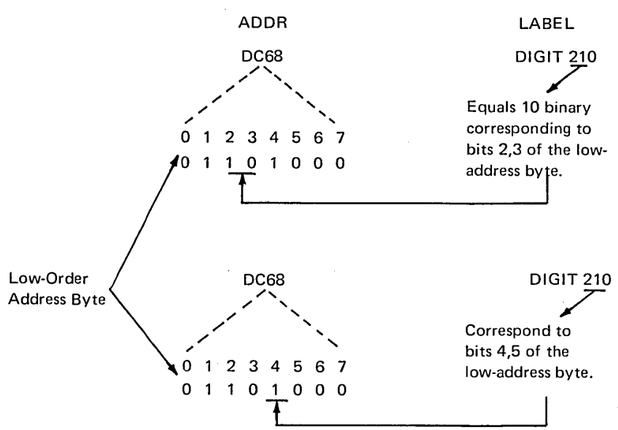
Branch Set Name — Leg Identifier

This is defined as a four-leg branch set. The leg identifier in this example represents the assignment of bits 4 and 5 of the control-word address. The 00 leg of the branch set is assigned the lowest address of the set; the other members are assigned addresses corresponding to their leg identifiers.

X's appearing in the leg-identifier field indicate that the micro-programmer does not care what the status of the bit corresponding to the X is. X's are used for alignment only.

ADDR	LABEL	
DC40	DIGIT	000
DC10	FTST	0
DC14	FTST	1
DC44	DIGIT	001
DC48	DIGIT	010
DC4C	DIGIT	011
DC50	DIGIT	100
DC54	DIGIT	101
DC58	DIGIT	110
DC5C	DIGIT	111
DC60	DIGIT	200
DC2C	SEVEN	11
DC20	EIGHT	0
DCA4	ADD8	
DC24	EIGHT	1
DC28	SEVEN	10
DC64	DIGIT	201
DC68	DIGIT	210
DC6C	DIGIT	211
DC70	DIGIT	300
DC74	DIGIT	301
DC78	DIGIT	310
DC7C	DIGIT	311

The branch set identified by the name DIGIT, is a 16-leg branch set. In this example, a branch set does not have to appear in sequential order in the microlisting but may be separated by other control words or other branch sets within that routine. The leg identifiers in the DIGIT branch set indicate the status of bits 2, 3, 4, and 5 of the control-word address. The number adjacent to the name is a hex number that designates the status of bits 2 and 3 of the control-word address. For example:

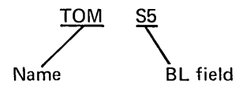


BRANCH SYMBOLS (BX, BH, BL)

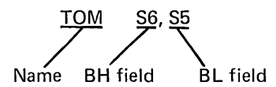
The branch symbols are found in the Next Label column to the right of the name of the control word being branched to. Three branch fields are available in the Next Lable column, the BX, BH, and BL.

BX=special branch field
BH=branch high field
BL=branch low field

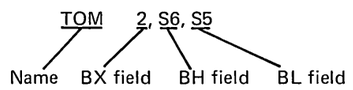
If only one branch field appears in the Next Lable column, it is the BL field. Example;



If two branch fields appear in the Next Label column, they are the BH and BL fields. Example;



If three branch fields appear in the Next Lable column, they are the BX, BH, and BL fields. Example;



The BL field affects the setting of bit 5 or M3 when the next control-word address is formed.

The BH field affects the setting of bit 4 of M3 when the next control-word address is formed.

The BX field normally affects the setting of bits 2, 3 of M3. There is an exception, which is covered in the word-move description.

Control-Storage Maps

The control-storage maps are located at the back of the 370 microprogram listings.

The maps are used for tracing directly back to a sequence number in a microroutine from any control-storage address.

How to use:

Given the control-storage address 8AB8?

- Find the 8A module number in the control-storage maps.
- Find address 8AB0.
- Count by word address to 8AB8.

Entry is GSIN0170. This specifies that the control word at address 8AB8 is located in microroutine GSIN at sequence number 0170.

HOUT=Microroutine name.
0071=Sequence number in HOUT microroutine.

Module Starting Addresses

CONTROL STORAGE MAP *370 EC 128603 PAGE 6/ OVERLAY 4

MODULE NO. 89

-8900-	HOUT0075	HOUT0099	HOUT0076	HOUT0086	-8910-	HOUT0070	HOUT0069	HOUT0068	HOUT0067
-8920-	HOUT0073	HOUT0085	HOUT0072	HOUT0083	-8930-	HOUT0082	HOUT0084	HOUT0077	HOUT0074
-8940-	HOUT0071	CSRD0384	CSRD0378	CSRD0369	-8950-	CSRD0366	CSRD0370	CSRD0367	CSRD0368
-8960-	CSRD0373	CSRD0377	CSRD0381	CSRD0380	-8970-	CSRD0375	CSRD0370	CSRD0374	CSRD0371
-8980-	CSRD0372	CSRD0379	GHLI0253	GHLI0254	-8990-	GHLI0255	GHLI0259	GHLI0256	GHLI0260
-89A0-	GHLI0258	GHLI0257	GHLI0242	GHLI0239	-89B0-	GHLI0251	GHLI0252	GHLI0244	GHLI0243
-89C0-	GHLI0249	GHLI0250	GHLI0240	GCCB0011	-89D0-	GCCB0016	GCCB0018	GCCB0017	GCCB0019
-89E0-	GCCB0022	GCCB0021	GCCB0013	GCCB0015	-89F0-	GCCB0008	GCCB0009	GCCB0006	GCCB0007

MODULE NO. 8A

-8A00-	GMSR0014	GMSR0013	GMSR0016	GMSR0017	-8A10-	GMSR0021	GMSR0022	GMSR0026	GMSR0027
-8A20-	GMSR0024	GMSR0023	GMSR0019	GMSR0018	-8A30-	GMSR0025	GMSR0020	GMSR0030	GMSR0015
-8A40-	GMSR0029	GMSR0028	GDMD0457	GDMD0366	-8A50-	GDMD0302	GDMD0303	GDMD0304	GDMD0305
-8A60-	GDMD0309	GDMD0456	GDMD0373	GDMD0358	-8A70-	GDMD0365	GDMD0361	GDMD0360	GDMD0359
-8A80-	GDMD0367	GDMD0362	GDMD0363	GDMD0364	-8A90-	GSIN0181	GSIN0178	GSIN0169	GSIN0191
-8AA0-	GSIN0184	GSIN0187	GSIN0177	GSIN0182	-8AB0-	GSIN0183	GSIN0185	GSIN0170	GSIN0171
-8AC0-	GSIN0172	GSIN0173	GSIN0174	GSIN0175	-8AD0-	GSIN0176	GSIN0015	GSIN0005	GBAC0004
-8AE0-	GBAD0011	GBAD0012	GBAD0013	GBAD0014	-8AF0-	GBAC0008	GBAC0006	GBAC0009	GBAC0007

MODULE NO. 8B

-8B00-	GSOP0487	GSOP0501	GSOP0486	GSER0407		GSER0405	GSER0414	GSER0416	
	GSOP0502	GSER0406	GSOP0493				GSOP0504	GSOP0489	
	GSOP0492	GSOP0505	GSOP0488				GKDT0537		
		GKDT0532	GKDT0548		-8D10-				GKLA0034
-8D20-		GKDT0542			-8D30-				GKLA0036
-8D40-				GKLA0046	-8D50-	GELL0218		GELL0208	GELL0206
-8D60-				GELL0204	-8D70-	GELL0197	GELL0200	GELL0207	GELL0205
-8D80-	GELL0198			GELL0217	-8D90-	GHLI0261	GHLI0262	GELL0213	GELL0215
-8DA0-	GKLA0051	GHLI0267	GHLI0265	GHLI0271	-8DB0-	GHLI0217	GHLI0218	GELL0212	GELL0214
-8DC0-	GHLI0212	GHLI0264	GHLI0214	GHLI0215	-8DD0-	GHLI0219	GHLI0220	GHLI0268	GHLI0269
-8DE0-	GHLI0245	GHLI0246	GHLI0263	GHLI0248	-8DF0-	GHLI0247	GSOP0538	GSOP0539	GSOP0537

MODULE NO. 8E

-8E00-	GGAD0044	GGAD0046	GGAD0041	GGAD0037	-8E10-	GGAD0048	GGAD0052	GGAD0043	GGAD0022
-8E20-	GGAD0036	GGAD0033	GGAD0035	GGAD0054	-8E30-	GGAD0083	GGAD0078	GGAD0079	GGAD0073
-8E40-	GGAD0026	GGAD0028	GGAD0030	GGAD0031	-8E50-	GGAD0074	GGAD0068	GGAD0063	GGAD0073
-8E60-	HMIO0069	HMIO0072	HMIO0070	HMIO0071	-8E70-	HMIO0097	HMIO0076	HMIO0096	HMIO0075

Address XXX0 Address XXX4 Address XXX8 Address XXXC Address XXX0 Address XXX4 Address XXX8 Address XXXC

REMEMBER

There is a Reader's Comment Form at the back of this publication.

CONTROL WORDS

- Control the accessing of data from:
  - Local Storage
  - External Registers
  - Main Storage
  - Control Storage
  - Expanded Local Storage
- Control the movement of data from one area to another.
- Perform arithmetic and logical operations on accessed data.
- Perform the setting and resetting of certain functional circuits.

Before a control word can perform any of its functions, it must be set into the four-byte control register (C-Reg). The outputs of the C-Reg activate circuitry that causes the execution of specified data-flow functions.

Control words are normally read from control storage and set into the C-Reg. However, control words can be set into the C-Reg directly from the console file, local storage, console switches, and certain control-word bit combinations may be forced into the C-Reg by circuitry.

The control words and their high-level functions are:

BRANCH AND MODULE SWITCH

Functions:

- Branch
- Module Switch
- Destine Data to the S-, T-, or L-registers.

BRANCH WORD

Functions:

- Branch
- Module Switch (Special Function)
- Set/Reset Bits in local-Storage or external-Registers.

BRANCH AND LINK OR RETURN

Functions Branch and Link:

- Store S, P, N2, N3 into a link register
- Set P with a value, or, Module Switch
- Branch.

Functions Return:

- Restore S, P, N2, N3
- Reset H-register bits
- Alter the link address in some cases.

WORD MOVE

Functions:

- Move a full word or selected bytes from one local storage/ external location to another.
- Branch.

STORAGE WORD

Functions:

- Read data from or store data into:
  - Local Storage
  - Main Storage
  - Control Storage
  - External Registers
  - Storage Protect Stack
- Branch and Link
- Branch.

ARITHMETIC WORD

Functions Type 10:

- Perform a variety of arithmetic and logical operations.
- Operate on full words for arithmetic or shifting operations.

Functions Type 11:

- Operates on bytes only
- Exclusive OR, or, true ADD only
- A-register input crossing provided.

Word-Type Recognition

First Hex Digit in Word Column	Control Word
0	Branch and Module Switch
1	Branch
2	Branch and Link or Return
3	Word Move
4–7	Storage
8–B	Arithmetic Type 10
C–F	Arithmetic Type 11

Next Control-Word Address Formation

- Every control word provides all or part of the next control-word address.
- Control-word addresses may also be formed by:
  Hardware circuits (Trapping)
  Hardware circuits (I-cycles)
  Operators Console Switches

Some control words have the capability of setting both the M2- and M3-registers. This provides for the addressing of any control word in the control-storage area. This capability is called module-switching. Other control words can set only the M3-register. This restricts the addressing capability to control words within the current address module.

For details of M-register setting, refer to the M-register description in Chapter 2.

For control-word setting of the M-register, refer to the operational diagrams associated with each word type.

Local-Storage, External-Address Formation

The Local Storage and External Address Formation chart on the facing page specifies the bit combinations used in source and destination address formation.

If the bit structure of the control word is known, and the contents of the P-register are determined, use the flow chart on the facing page to determine the line in the chart.

Two general types of addressing are defined: direct and indirect. *By definition* direct word addressing means that the word address is obtained from C- and P-register bits only. Indirect word addressing means that bits from the L-register are used with C- and P-register bits to form the address.

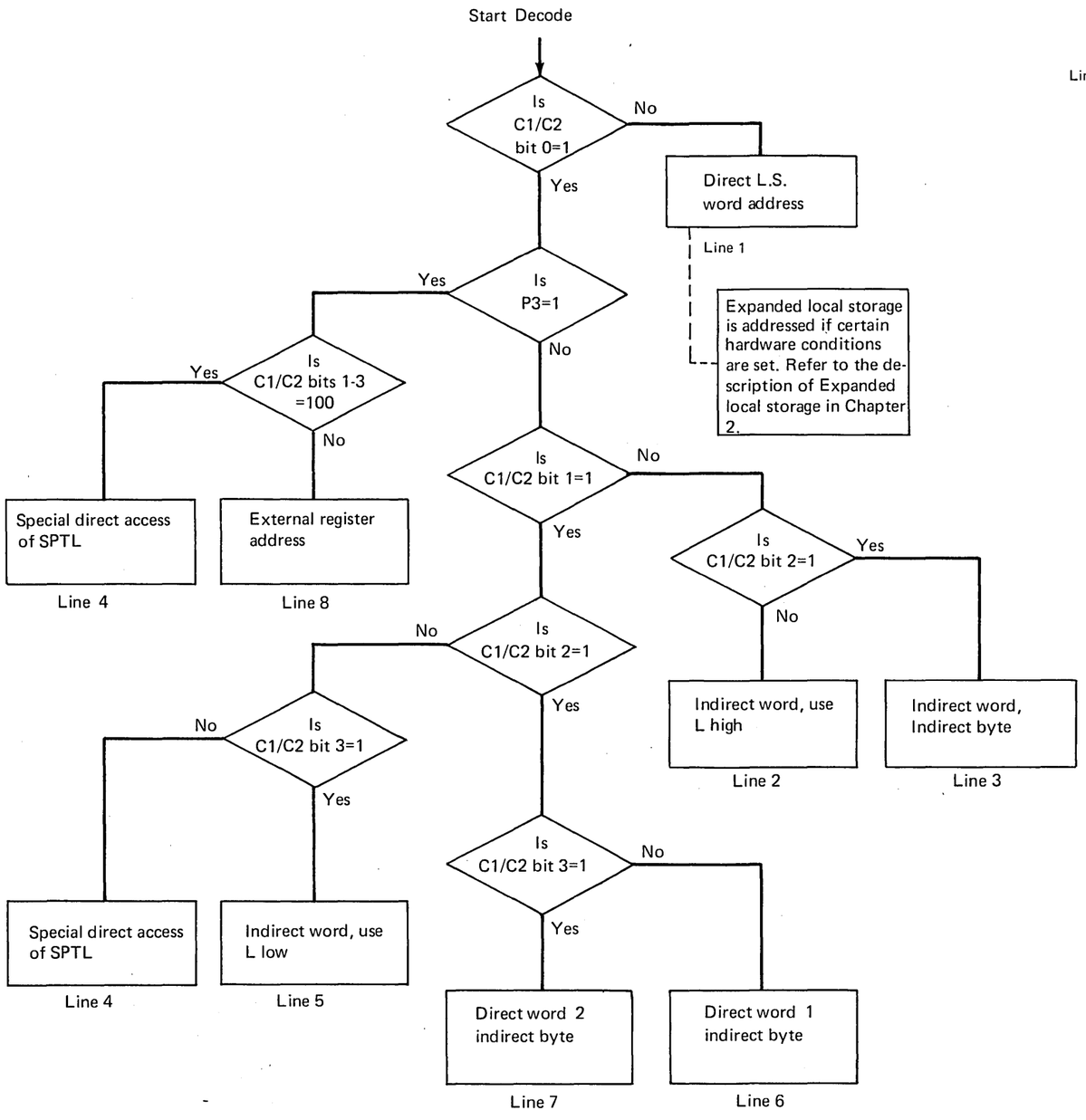
Direct byte addressing, again *by definition*, means that a byte of an addressed word facility is addressed by C-register byte 1, bits 4 and 5; or C-register byte 2, bits 4 and 5. Indirect byte addressing is specified by T-register bits 4 and 5 or bits 6 and 7.

The X's in the address formation chart indicate positions not used to determine the type of addressing used.

Under the heading Address Decode in the address formation chart, columns 2, 3, 4 determine the X line, and columns 5, 6, 7 determine the Y line.

The control-word operational diagrams that follow in this chapter illustrate the address formation of actual local-storage and external registers.

Local Storage and External Address Formation



The flow chart indicates which line in the address formation chart to use.

Line No.	Byte C1 or C2				P3	Defines	Local Storage or External Word * Address Decode								Byte*	
	0	1	2	3	--		0	1	2	3	4	5	6	7	0	1
1	0	X	X	X	X	Direct Local Storage Word Address	0	Note 1	P5	P6	P7	C1 or C2 1	C1 or C2 2	C1 or C2 3	C1 or C2 4	C1 or C2 5
2	1	0	0	X	0	Indirect Local Storage Word Address	0	0	P1	P2	L0	L1	L2	Note 2	C1 or C2 4	C1 or C2 5
3	1	0	1	X	0	Indirect Word – Indirect Byte	0	0	P1	P2	L0	L1	L2	Note 2	T4 or T6	T5 or T7
4	1	1	0	0	X	Special External Register Set: S P T L									C1 or C2 4	C1 or C2 5
5	1	1	0	1	0	Indirect Local Storage Word Address	0	0	P1	P2	L4	L5	L6	L7	C1 or C2 4	C1 or C2 5
6	1	1	1	0	0	Direct Word 1 – Indirect Byte	0	0	P5	P6	P7	C1 or C2 1	C1 or C2 2	C1 or C2 3	T4 or T6	T5 or T7
7	1	1	1	1	0	Direct Word 2 – Indirect Byte	0	0	P5	P6	P7	C1 or C2 1	C1 or C2 2	C1 or C2 3	T4 or T6	T5 or T7
8	1	X	X	X	1	External Registers: 8 groups of seven words	0	0	P0	P1	P2	C1 or C2 1	C1 or C2 2	C1 or C2 3	C1 4	C1 5

X-Line Y-Line

Note 1 Refer to the Expanded local-storage description in Chapter 2.

Note 2 C1 or C2 bit 3 is ORed with bit 3 of the L-register.

For source addressing, C1 bits apply to A local storage, Expanded local storage, or External registers.

For source addressing, C2 bits apply to B local storage, or Expanded local storage.

For destination addressing, C1 and C2 bits apply to A and B local storage, Expanded local storage, or External registers.

Branch and Module-Switch Word

- Provides for addressing any word in control storage.
- May perform up to four-way branching.
- Provides the function of setting the S-, T-, or L-Registers with the contents of any addressable local-storage or external byte source.

The primary function of this word is to change the entire control-word address in the M2 and M3 address registers. This capability allows for branching to any word in the control-storage area.

The S-, T-, and L-registers can be set with the contents of the branch source byte.

Whenever bits of the branch source are being tested in the BH field, the cycle time is 247.5 ns. If only S-register bits are being tested, or the branch test fields are fixed, the cycle time is 202.5 ns.

The following types of statements may appear in the statement field of the microlistings:

BS
S=BS
T=BS
L=BS
blank

Where BS is any valid local-storage or external byte source.

Next Label

The Next Label field specifies the name of the next control word to be executed, and in the case of a branch test, the manner in which bits 2, 3, 4, and 5 of M3 are to be set.

The format of this field is:

XXXXXX BX, BH, BL

Where XXXXXX is the name of the word or branch set being branched to. BX, BH, BL specify the leg of the branch set being branched to.

The following table indicates the types of symbols that may be found in the branch fields:

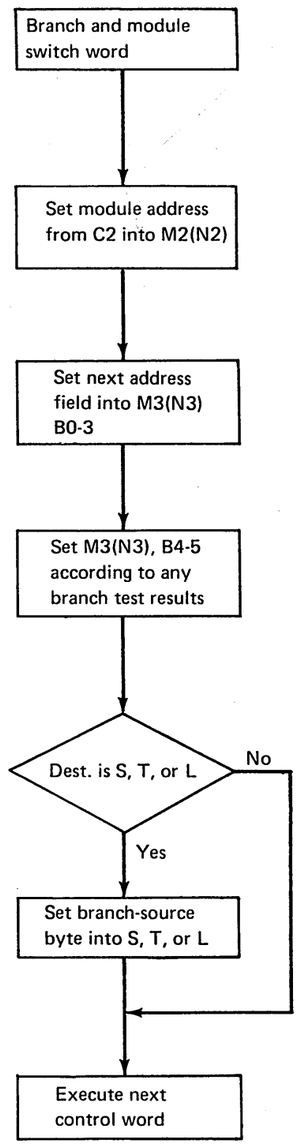
BX	BH	BL
0	0	0
1	1	1
2	S1	Z0
3	S0	NZ
	S2	S3
	S4	S5
	S6	S7
	BH	BL
	B0	B0
	B0	B0
	B7	B7

BX – In the BX field, the hex numbers fix the status of bits 2 and 3 of M3 when the next address is formed.
For example: the value 1 = 01 binary; bit 2 of M3 is set to 0 and bit 3 of M3 is set to 1 when the branch address is formed.

BH – In the BH field, the numbers 0 and 1 fix the status of bit 4 of M3 when the next address is formed. The S symbols refer to S-register bits. Bit 4 of M3 is set to the status of the S bit tested. The symbol BH refers to bits 0-3 of the branch source byte. Bits 0-3 are tested for a non-zero condition; and if non-zero, bit 4 of M3 is set to 1; otherwise set to 0. The remaining symbols refer to bits in the branch source byte directly. Bit 4 of M3 is set to the status of the bit tested.

BL – In this field, the S and B bit symbols are the same as the BH field. Z0 causes S4 and S5 to be tested for a status of 11. If both S4 and S5 are 1, set bit 5 of M3 to 1, otherwise set bit 5 of M3 to 0. NZ causes the test of the branch source byte for a non-zero condition. If the branch source is non-zero, set bit 5 of M3 to 1, otherwise set bit 5 to 0. The BL symbol tests bits 4-7 of the branch source for non-zero; if non-zero, set bit 5 of M3 to 1; otherwise set bit 5 to 0.

Branch and Module-Switch Word Execution



Bit Definition of the Branch and Module -Switch Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch and Module Switch				Branch High				Branch Source						Branch Source Dest		Module Address								Next Address				Branch Low			
								Word				Byte																			
0	0	0	0	0000=0 00001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6 0111=BH 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7						00=0 01=1 10=2 11=3		00=— 01=(S=Branch Source) 10=(T=Branch Source) 11=(L=Branch Source)										0000=0 0001=1 0010=Z0 0011=NZ 0100=S3 0101=S5 0110=S7 0111=BL 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7							

Co Bits 0-3

This field is a value of 0000 to designate the branch and module switch word. In the microlistings, the branch and module switch word can be recognized by the hex digit 0 in the high-order position of the word.

C0 Bits 4-7

This field specifies the branch test or fixed value that determines the setting of bit 4 of the M3-address register. The fixed values 0 and 1 designate directly the setting of bit 4 of M3. The S1, S0, S2, S4, and S6 symbols refer to bits of the S-Register to be tested. Bit 4 of M3 is set to the value of the tested S bit. BH is specified by a value of 0111 in this field and refers to the high 4 bits of the branch source that is addressed by bits 0-5 of C1. The high 4 bits are tested for a non-zero condition. If this test is met, bit 4 of M3 is set to 1. The remaining symbols in this field refer to specific bits of the branch source. M3 bit 4 is set to the value of the branch source bit tested.

C1 Bits 0-5

This field specifies the local storage or external byte to be accessed for branch testing. Bits 0-3 are used to form part of the word address at which the byte to be tested is located. Bits 4 and 5 specify the byte of the word addressed that is to be tested.

C1 Bits 6, 7

6,7=00 indicates no branch source destination.
6,7=01 destine the branch source to the S-register.
6,7=10 destine the branch source to the T-register.
6,7=11 destine the branch source to the L-register.

C2 Bits 0-7

This field contains the module address that is placed in M2 when the next address formation takes place.

C3 Bits 0-3

This field is part of the word address that is gated to M3. When the next address formation takes place, bits 0-3 of C3 are gated to bits 0-3 of M3.

If the BX field is indicated in the Next Label field, the fixed value specified by the BX field is carried in bits 2, 3 of C3. For example, if BX=2, bits 2, 3 of C3 would equal 1, 0.

C3 Bits 4-7

This field specifies the branch test or fixed value that determines the setting of bit 5 of M3. The fixed values 0 and 1 designate directly the setting of M3 bit 5. The Z0 symbol causes bits 4 and 5 of the S-Register to be tested. If S4, 5=11, M3 bit 5 is set to 1. If either S4 or S5=0, M3 bit 5 is set to 0. The NZ symbol causes the branch source to be tested for a non-zero status. If the branch source is non-zero, M3 bit 5 is set to 1. If the branch source is zero, M3 bit 5 is set to 0. S3, S5, and S7 refer to bits of the S-register to be tested. Bit 5 of M3 is set to the value of the tested S bit. The BL symbol refers to the low 4 bits of the branch source. If bits 4-7 of the branch source are non-zero, bit 5 of M3 is set to 1. If bits 4-7 of the branch source are zero, bit 5 of M3 is set to 0. The symbols B0-B7 refer to specific bits of the branch source. Bit 5 of M3 is set to the value of the bit tested.

Branch and Module-Switch Word Example (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
FORMAT B0, B1		**S = DTX2**

Starting Values:

P-Reg = 02
DTX Reg = 08 D6 5A 2C
Address of FORMAT 00 = CE00

Objectives:

Test bits 0 and 1 of DTX2
Set DTX2 into the S-register

Description:

Read the contents of the DTX register from local storage A.

Gate byte 2 of DTX to the branch circuits for testing. Bit 0, as designated by the branch high field is tested and its value is gated to bit 4 of M3. Bit 1, as designated by the branch low field is tested, and its value is gated to bit 5 of M3.

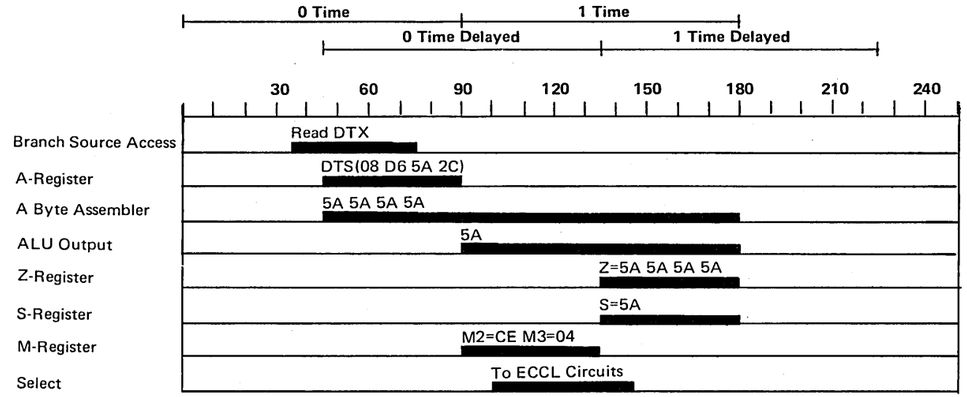
The next Address field (C3 Bits 0-3) is gated to bits 0-3 of M3.

The contents of C2 are gated to M2.

Byte 2 of DTX is gated to ALU 2 and 3 and then to the Z-register.

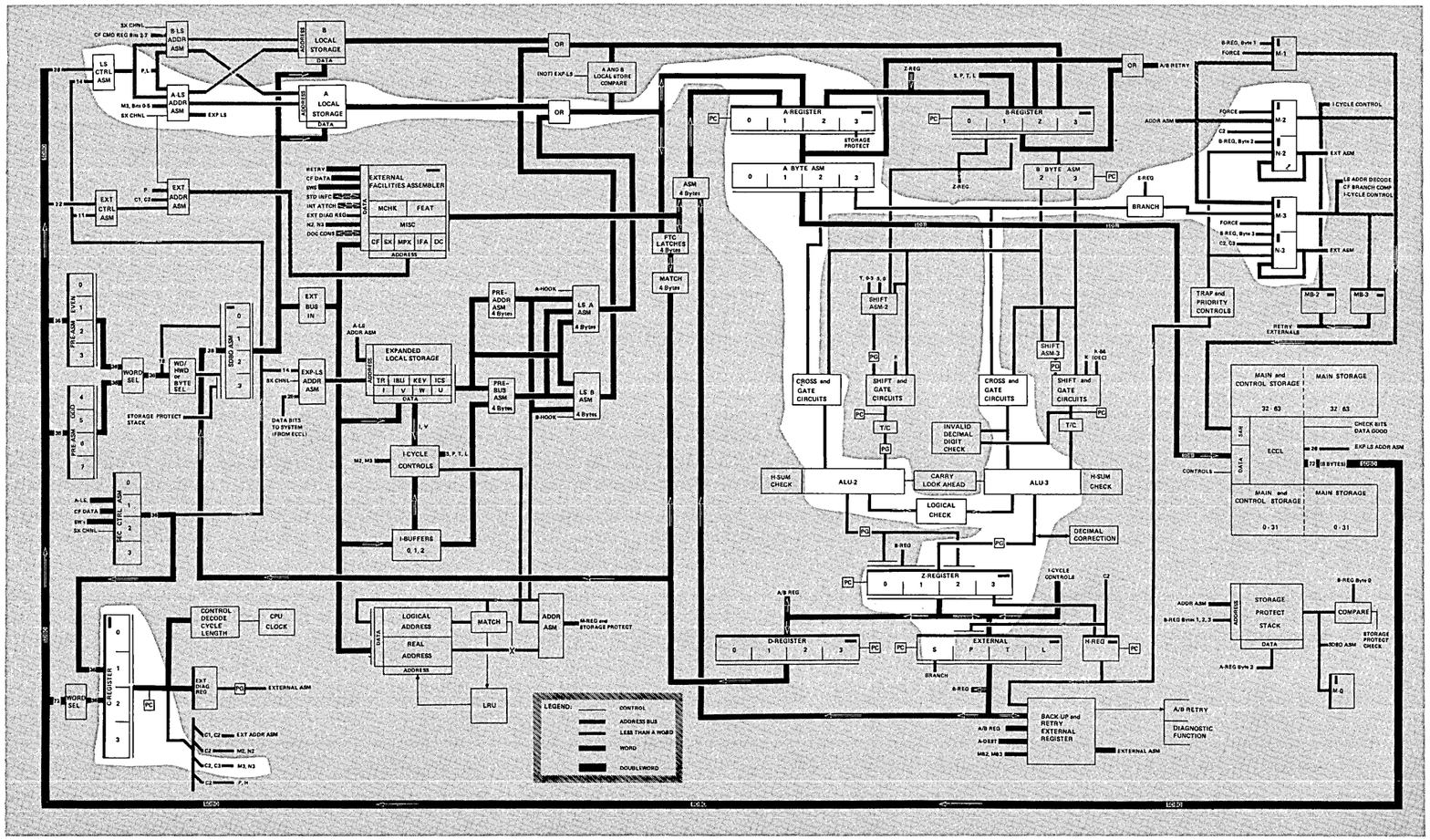
The contents of byte 0 of the Z-register are gated to the S-register to complete the function of this control word.

CPU cycle time for this word is 247.5 ns.

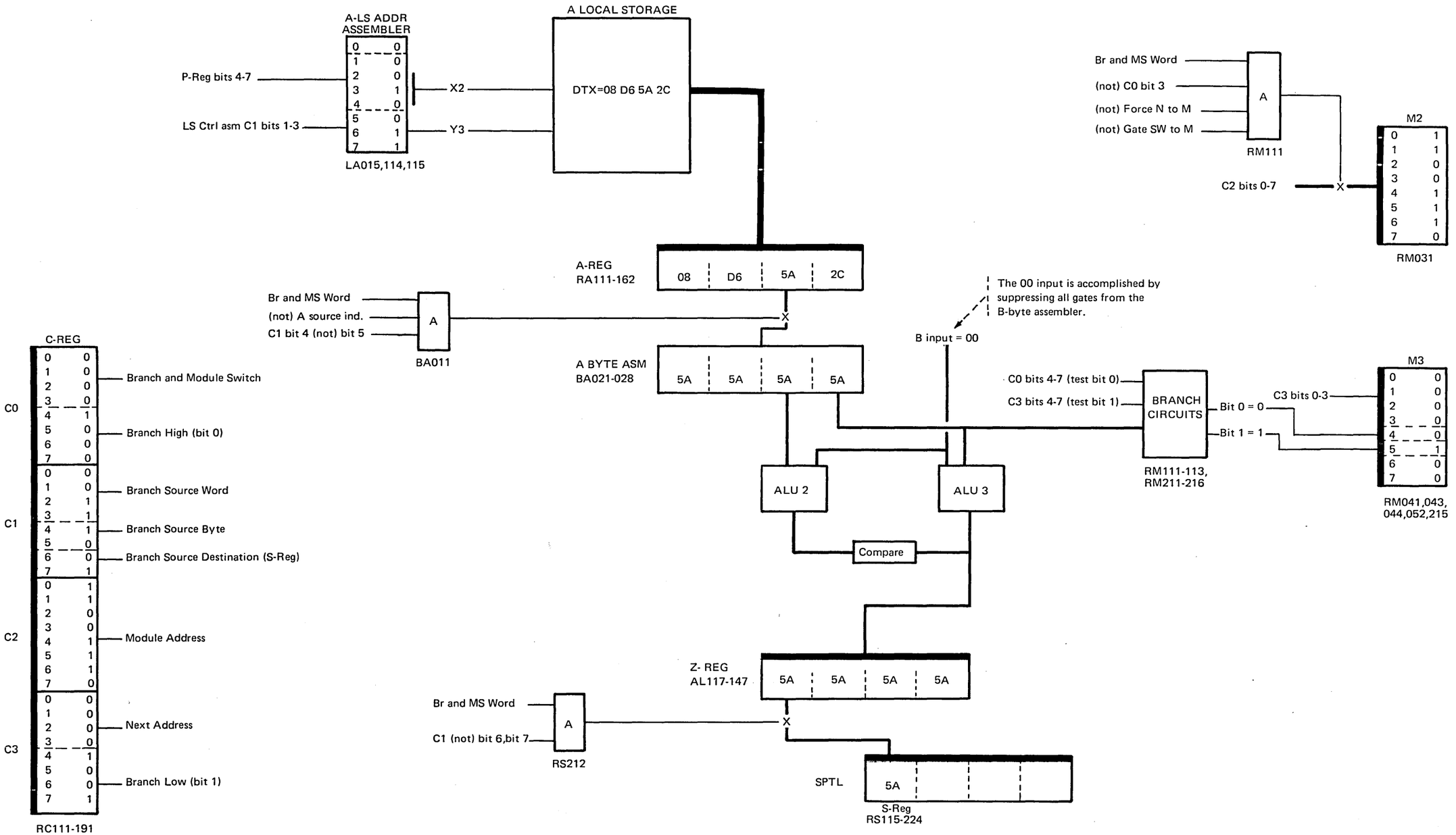


Cycle time = 247.5 ns

Portions of the CPU data flow directly affected by this control word.



Branch-Word Example, Special Module-Switch Function (Part 2 of 2)



Branch Word

- Provides a branching function to address a control word in the current module address.
- May set or reset bits of a specified local-storage or external byte location.
- Provides for module switching, allowing any word in control storage to be accessed.

The primary function of the branch word is to branch to another control word in control storage. The control-word branch address may be selected by testing the status of certain S-register or branch-source bits and forming the branch address accordingly, or specifying the branch address directly.

The branch-test function, performed on the S-register or branch-source bits, results in setting bits 4 and 5 of the M3-address register. For example; if S6 is specified in the branch high field (C0 bits 4-7=0110), and S7 is specified in the branch low field (C3 bits 4-7=0110), and S6=1, S7=0, then M3 bit 4 is set to 1 and M3 bit 5 is set to 0. Any branch testing is done before setting or resetting bits in the byte specified by the S/R source field.

The following kinds of branch word statements may appear in the statement field of the microlistings:

```
BS (,A-,/,OR,) hh
S (,A-/,OR,) hh
P (,A-,/,OR,) hh
GA (,A-,/,OR,) hh
BS S (,A-,/,OR,) hh
BS P (,A-,/,OR,) hh
BS GA (,A-,/,OR,) hh
```

Where BS is any valid local-storage or external byte source, and hh represent hexadecimal digits. Because the field in the branch control word that contains the hex value is only 4 bits wide, the hex value in the statement is restricted to these combinations:

```
hh must be equal (77,CC,22,etc)
h0 or 0h or 00
```

The format of the hex digits determines the bit structure of the K HI-LO field (C1 bits 6, 7).

The operators, in parentheses represent the arithmetic function to be performed:

,A-, specifies the complement AND function. The hex value is complemented, then ANDED to the byte that is to the left of the ,A-, symbol. This function resets any bit of the source that corresponds to a bit in the hex number in the statement.

For example:

If the hex number written is 05, bits 5 and 7 of the source are affected by the operation.

```
0000 0101
0123 4567
```

If the hex number written is 77, bits 1, 2, 3, 5, 6, and 7 of the source are affected.

```
0111 0111
0123 4567
```

,OR, specifies the OR function. Bits of the source that correspond to bits in the hex number in the statement are set to 1. See example.

Next Label

The Next Label field specifies the name of the next control word to be executed, and in the case of a branch test, the manner in which bits 2, 3, 4, and 5 of M3 are to be set.

The format of this field is:

```
XXXXXX BX, BH, BL
```

Where XXXXXX is the name of the word or branch set being branched to. BX, BH, BL specify the leg of the branch set being branched to.

The following table indicates the kinds of symbols that may be found in the branch fields:

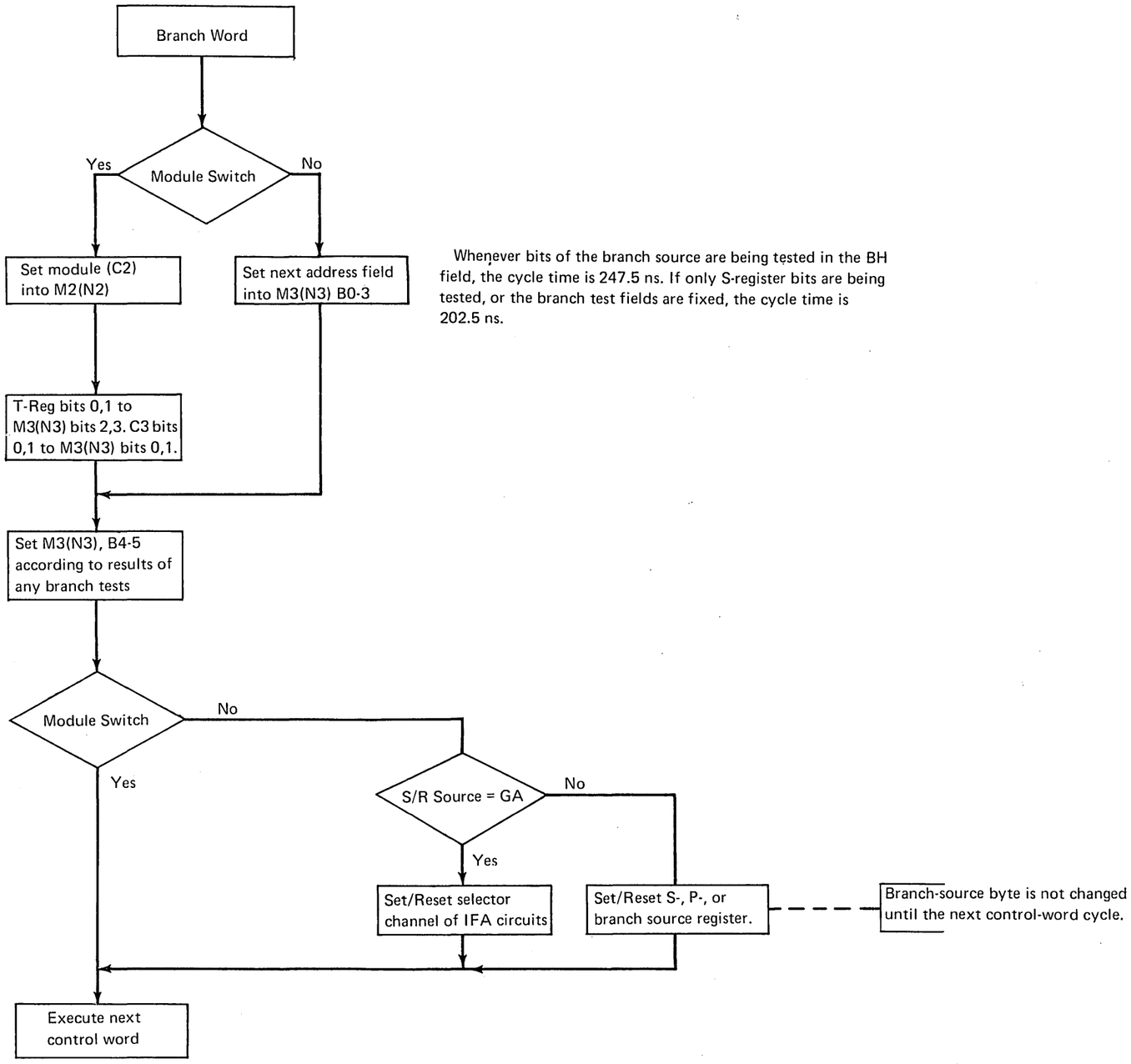
BX	BH	BL
0	0	0
1	1	1
2	S1	Z0
3	S0	NZ
TH	S2	S3
	S4	S5
	S6	S7
	BH	BL
	B0	B0
	B0	B0
	B7	B7

BX — In the BX field, the numerics 0-3 fix the status of bits 2 and 3 of M3 when the next or branch address is formed. For example; the value 2=10 binary, bit 2 of M3 is set to 1 and bit 3 of M3 is set to 0 when the branch address is formed. When TH is specified in the BX field, the special module-switch function is indicated. Module switching is indicated by C1 bits 6, 7=00. The module-switch function sets bits 0-7 of C2 into M2. No setting or resetting of bits can occur when the module-switch function is indicated. The module switch also causes bits 0 and 1 of the T-Reg to be gated to bits 2 and 3 of M3 when the branch address is formed.

BH — In the BH field, the numbers 0 and 1 fix the status of bit 4 of M3 when the next address is formed. The S symbols refer to S-register bits. Bit 4 of M3 is set to the status of the S bit tested. The symbol BH refers to bits 0-3 of the branch-source byte. Bits 0-3 are tested for a non-zero condition; and if non-zero, bit 4 of M3 is set to 1; otherwise set to 0. The remaining symbols refer to bits in the branch-source byte directly. Bit 4 of M3 is set to the status of the bit tested.

BL — In this field, the S and B bit symbols are the same as in the BH field. Z0 causes S4 and S5 to be tested for a status of 11. If both S4 and S5 are 1, set bit 5 of M3 to 1; otherwise set bit 5 of M3 to 0. NZ causes the test of the branch-source byte for a non-zero condition. If the branch source is non-zero, set bit 5 of M3 to 1; otherwise set bit 5 to 0. The BL symbol tests bits 4-7 of the branch source for non-zero; if non-zero, set bit 5 of M3 to 1; otherwise set bit 5 to 0.

Branch Word Execution



Whenever bits of the branch source are being tested in the BH field, the cycle time is 247.5 ns. If only S-register bits are being tested, or the branch test fields are fixed, the cycle time is 202.5 ns.

Branch-Word Examples

NEXT LABEL	STATEMENT
MCHST S2, S3	S,OR,K80

The function indicated by the statement is to OR the value 80 with the contents of the S-register. This OR operation results in setting bit 0 of S to 1.

The Next Label field specifies a branch to the branch set named MCHST. The leg in that branch set is determined by the status of S-Register bits 2 and 3 as indicated by the BH and BL fields. Assu Assume that S2=1 and S3=0, the branch is made to MCHST 10.

NEXT LABEL	STATEMENT
RETURN TH, B2, B5	T

The TH in the BX field of the Next Label specifies the special module switch function of the branch word. The statement field indicates the T-register as the branch source.

The next control-word address is formed in the following manner:

The contents of byte 2 of the control word (C2) is gated to M2.

Bits 0 and 1 of C3 are gated to bits 0 and 1 of M3.

Bits 0 and 1 of the T-register are gated to bits 2 and 3 of M3.

Bit 2 of the T-register determines the setting of bit 4 of M3.

Bit 5 of the T-register determines the setting of bit 5 of M3.

Refer to the Branch Word (Module Switch) operational diagram for a detailed description of this word.

Bit Definition of the Branch Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch								Branch Source						K HI-LO			S/R	S/R Source		K				Next Address				Branch Low			
								Word				Byte																			
0	0	0	1	0000=0 0001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6 0111=BH 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7										00=0 01=1 10=2 11=3		00=MS 01=L 10=H 11=St		0=--- 1=DK	0=,OR. 1=,A,	00=BS 01=S 10=P 11=GA										0000=0 0001=1 0010=Z0 0011=NZ 0100=S3 0101=S5 0110=S7 0111=BL 1000=B0 1001=B1 1010=B2 1011=B3 1100=B3 1101B5 1101=B5 1110=B6 1111=B7			
																M2															
																← Module Address →															

C0 BITS 0-3

This field is a value of 0001 to designate the branch word. In the microlistings, the branch word can be recognized by the hex digit 1 in the high-order position in the Word column.

C0 BITS 4-7

This field specifies the branch test or fixed value that determines the setting of bit 4 of the M3-address register. The fixed values, 0 and 1, designate directly the setting of bit 4 of M3. The S1, S0, S2, S4, and S6 symbols refer to bits of the S-register to be tested. Bit 4 of M3 is set to the value of the tested S bit. BH is specified by a value of 0111 in this field and refers to the high 4 bits of the branch source that is addressed by bits 0-5 of C1. The high 4 bits are tested for a non-zero condition. If this condition is met, bit 4 of M3 is set to 1. The remaining symbols in this field refer to specific bits of the branch source, M3 bit 4 is set to the value of the branch-source bit tested.

C1 BITS 0-5

This field specifies the local-storage or external byte to be accessed for branch testing. Bits 0-3 are used to form part of the word address at which the byte to be tested is located. Bits 4 and 5 specify the byte of the word addressed that is to be tested.

C1 BITS 6, 7

These two bits designate the gating of the K value to the ALU for the set or reset function. L specifies that only the low 4 bits of the ALU are to receive the value in bits 4-7 of C2. H specifies that only the high 4 bits of the ALU are to receive the K value. ST means a gate of the K value into both the high and low 4 bits of the ALU. If C1 bits 6, 7 are 00, the special module-switch function is designated. This function specifies the following action:

No set/reset can occur.
Gate C2 to M2.
Gate T-Reg bits 9, 1 to M3 bits 2, 3.
Branch high and low set bits 4, 5 of M3 in normal fashion.

C2 BIT 0

When =0, designates normal operation. When = to 1, the diagnostic key is set or reset depending on the setting of bit 1 of C2.

C2 BIT 1

This bit=0 specifies the ,OR, function for the ALU operation to be performed on the source that is designated by bits 2 and 3 of C2. The ,OR, function effectively sets the bits of the source that correspond to bits in the K field.

This bit=1 specifies the ,A-, function for the ALU operation. The ,A-, is a complement AND function that resets bits of the source that correspond to bits in the K field.

C2 BITS 2, 3

This field specifies the source that is to be set or reset by the K field.

When this field=00 the branch source addressed by bits 0-5 of C1 is also the source to be set or reset. When 2, 3=01 the S-register is specified as the set/reset source. When 2, 3=10 the P-register is specified as the set/reset source. When 2, 3=11 the GA special function is specified.

C2 BITS 4-7

This field contains the value to be used for setting or resetting bits of the source specified by bits 2 and 3 of C2. The hex value 0-F can be designated in this field.

C2 BITS 0-7

When the special module switch function is specified by bits 6 and 7 of C1, bits 0-7 of C2 contain the module address that is gated to the M2 register for the next control-word address formation.

C3 BITS 0-3

When not module-switching, this field represents part of the next control-word address and is gated to bits 0-3 of the M3-register.

When module-switching, bits 0 and 1 of C3 are gated to bits 0 and 1 of M3 and bits 0 and 1 of the T-Reg are gated to bits 2 and 3 of M3.

If the BX field was used, the fixed value specified by the BX field is carried in bits 2 and 3 of C3. For example, if BX=2, bits 2, 3 of C3 would equal 1, 0.

C3 BITS 4-7

This field specifies the branch test or fixed value that will determine the setting of bit 5 of the M3-address register. The fixed values 0 and 1 designate directly the setting of M3 bit 5. The Z0 symbol causes bits 4 and 5 of the S-register to be tested. If S4, 5=11, M3 bit 5 is set to 1. If either S4 or S5=0, M3 bit 5 is set to 0. The NZ symbol causes the branch source to be tested for a non-zero status. If the branch source is non-zero, M3 bit 5 is set to 1. If all bits of the branch source are 0, refer to bits of the S-register to be tested. Bit 5 of M3 is set to the value of the tested S-bit. The BL symbol refers to the low 4 bits of the branch source. If bits 4-7 of the branch source are not = to 0, bit 5 of M3 is set to 1. If bits 4-7 of the branch source are = to 0, bit 5 of M3 is set to 0. The symbols B0-B7 refer to specific bits of the branch source. Bit 5 of M3 is set to the value of the tested branch-source bit.

GA Special Functions

The special function GA is used to set or reset hardware in either the selector channel or IFA circuits.

If the IFA feature is not installed, the GA function chart for selector channels is used.

If the IFA feature is installed, and selector channel 1 is specified, the GA function chart for IFA is used. If selector channel 2 or 3 is specified, use the selector channel GA function chart.

Example:
Selector channel 1 is operational,

Branch Word----- GA, A-, K0A----- is executed,

Looking at line 2 of the selector-channel chart, find GA, A-, K0h **A**

Find the K value A in column 1 of the chart. **B**

The entry in the GA, A-, K0h column, and the A row is, Reset Supp Out.

Example:
IFA is operational,

Branch Word----- GA, OR, KE0----- is executed,

Looking at line 2 of the IFA chart, find GA, OR, Kh0. **C**

Find the K value E in column 1 of the chart. **D**

The entry in the GA, OR, Kh0 column, and the E row is, Set Data Field Lch.

h (K Field)	Set GAL GA, OR, K0h	Reset GAL GA, A-, K0h	Set GAH GA, OR, Kh0	Reset GAH GA, A-, Kh0
1	Set Poll Control (Soft)	Reset Poll Control	Set Channel 1	Channel Reset
2	Set Poll Control (Hard)	Reset Retry Holdup	Set Channel 2	Chain Reset
3	Set Command Retry		Set Channel 3	Machine Reset
4			Set Channel 4	
5	Set Count Ready	Start I/O Reset	Set Channel Loaded	Reset Channel Loaded
6	Set Protect Check	Set Control Check	Set CC	Diag Buffer Shift
7	Set Program Check	WLR Sample	Set PCI	Reset Interrupts
8	Set Intr Latch	Set DCC Mode	Diag Block Share Req	Reset DCC Mode and Diag Block Share Req
9	Set Select Out	Reset Sel Out and Primed	Set Diag Stat	Reset Diag Stat and Intr Latch
A	Set Supp Out	Reset Supp Out	Set Channel Primed	Set Channel Tried
B	GI into GR		Set Data Out	
C	Set OP Out	Reset OP Out and Diag Set GR Full	Set Command Out	Set Addr Out
D	Infce Ctrl Check	Diag Serv Signal	Set Service Out	Reset PCI
E	Set Diag Mode	Reset Diag Mode	Set Halt I/O	Reset Halt I/O
F				

GA Function Chart for Selector Channels

h K Field	Set GAL GA, OR, K0h	Reset GAL GA, A-, K0h	Set GAH GA, OR, Kh0	Reset GAH GA, A-, Kh0
1	Set Inc Length	Reset FCS	Set IFA Chan Gate	Rst Command Overrun
2	Set Prog Check	Reset PCI	Set Channel 2 Gate	
3	Set Prot Check	Rst Trap Req	Set Channel 3 Gate	Machine Reset
4	Set Chan Ctrl Chk	Rst CCW 0 and WLR		
5	Set Allow Restart	Rst Lo Prior Req	Set Write Clk Gate	Rst Orientation Lch
6		Chain End Reset	Set CS,CR, In Lchs	Rst Cnt Rdy,In,Out
7	Set Contingent Con	Rst Contingent Con	Set CS,CR,Out Lchs	Set Halt I/O
8			Set MS,CR,In Lchs	CE End Op SS
9	Set Chan Busy	Reset Chan Busy	Set MS,CR,Out Lchs	Diag Index
A	Set Intrp Latch	Rst Intrp Latch	Set Control Pulse	Diag Raw Data Pulse
B	Set CUB	Reset CUB	Diag Read Data	Set Diag Read Gate
C	Set DCC	Reset DCC	Diag Clk Gap Sense	Bit Ring Advance
D	Set Lo Prior Req	Rst H/L Comp,CC Er	Diag Data Gap Sense	Set Diag Mode Latch
E	Set IFA Inh Traps	Rst IFA Inh Traps	Set Data Field Lch	Rst Diag Mode Latch
F				

GA Function Chart for IFA

Branch-Word Example (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
TMODE B7		R1 P,OR,K10

Starting Values:

R-Reg = 0C FD 13 58
SPTL = 00 02 00 00
Address of this word = 0C10
Address of TMODE 0 = 0C40
Address of TMODE 1 = 0C44

Objectives

Test bit 7 of R1
Set bit 3 of the P-Reg

Description

The branch control word (R1 P, OR,K10), is read from control storage and gated to the C-register. C0,C1, and C2 are gated to the local-storage control assembler before the C register is set. This allows a fast access of the branch-source byte from local storage.

The R-register is read out of local storage A, and gated to the A-register.

Byte 1 of the A-register is gated to bytes 0-3 of the A byte assembler.

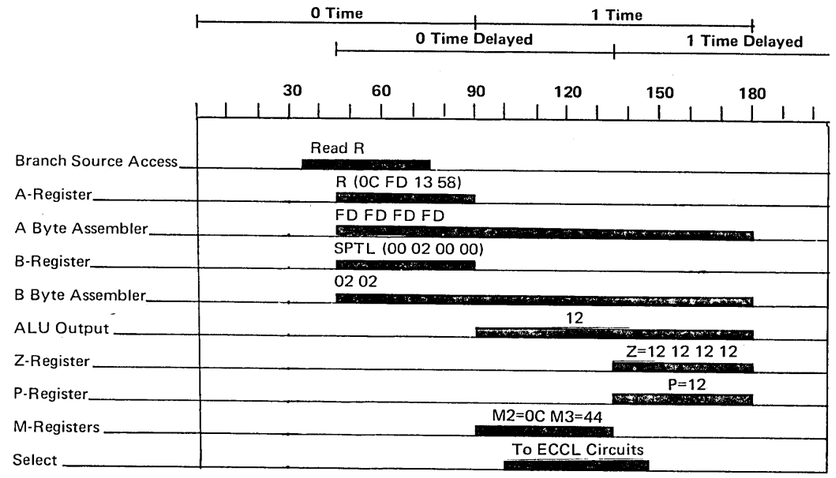
Byte 3 of the A byte assembler is gated to the branch circuits for the test of bit 7.

The next control-word address is formed in the M-registers.

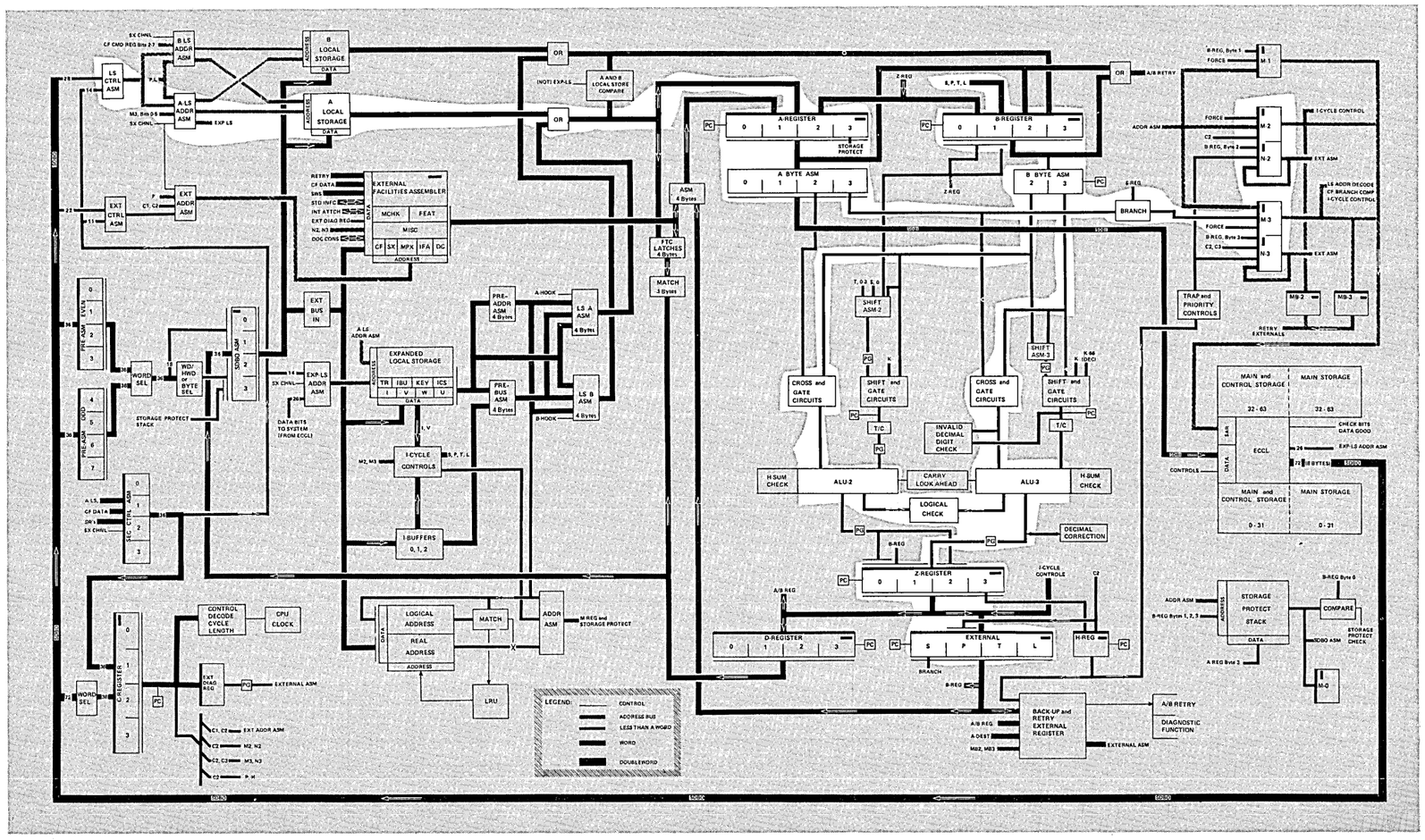
SPTL is gated to the B-register. Byte 1 (P-Reg), of the B-register is gated to the B byte assembler.

The K value (10) and the contents of the P-Reg are ORed in the ALUs. The result (12) is gated to the Z-register.

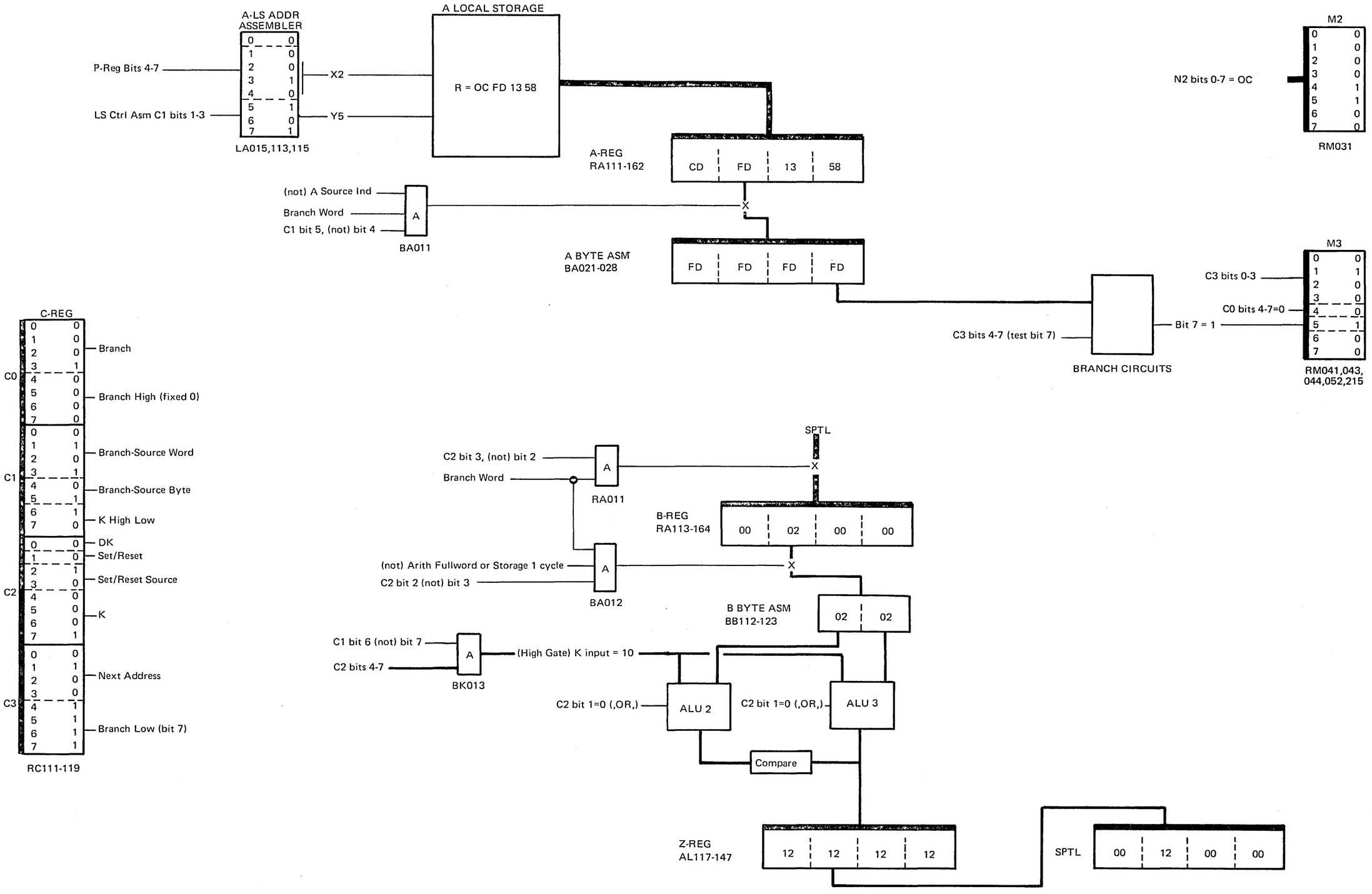
Byte 1 of the Z-register is gated to the P-register. P-Reg now equals 12.



Cycle time = 202.5 ns



Branch-Word Examples (Part 2 of 2)



Branch-Word Example, Special Module-Switch Function (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
RXOPS TH,B2,B3		T

Starting Values

SPTL = 00 02 B0 33
Address of RXOPS 000 = F3C0

Objectives

Perform special module-switch test bits 0-3 of the T-register

Description

The branch control word is read from control storage and gated to the C-register.

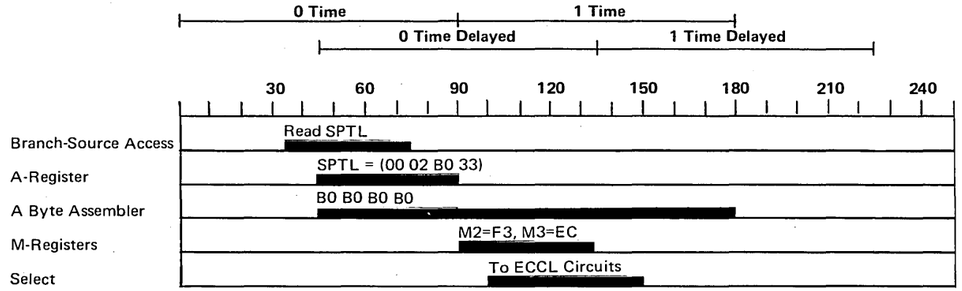
SPTL is gated to the A-register. Byte 2 (T-Reg), of the A-register is gated to bytes 0-3 of the A byte assembler.

Byte 3 of the A byte assembler is gated to the branch circuits where the branch tests are made.

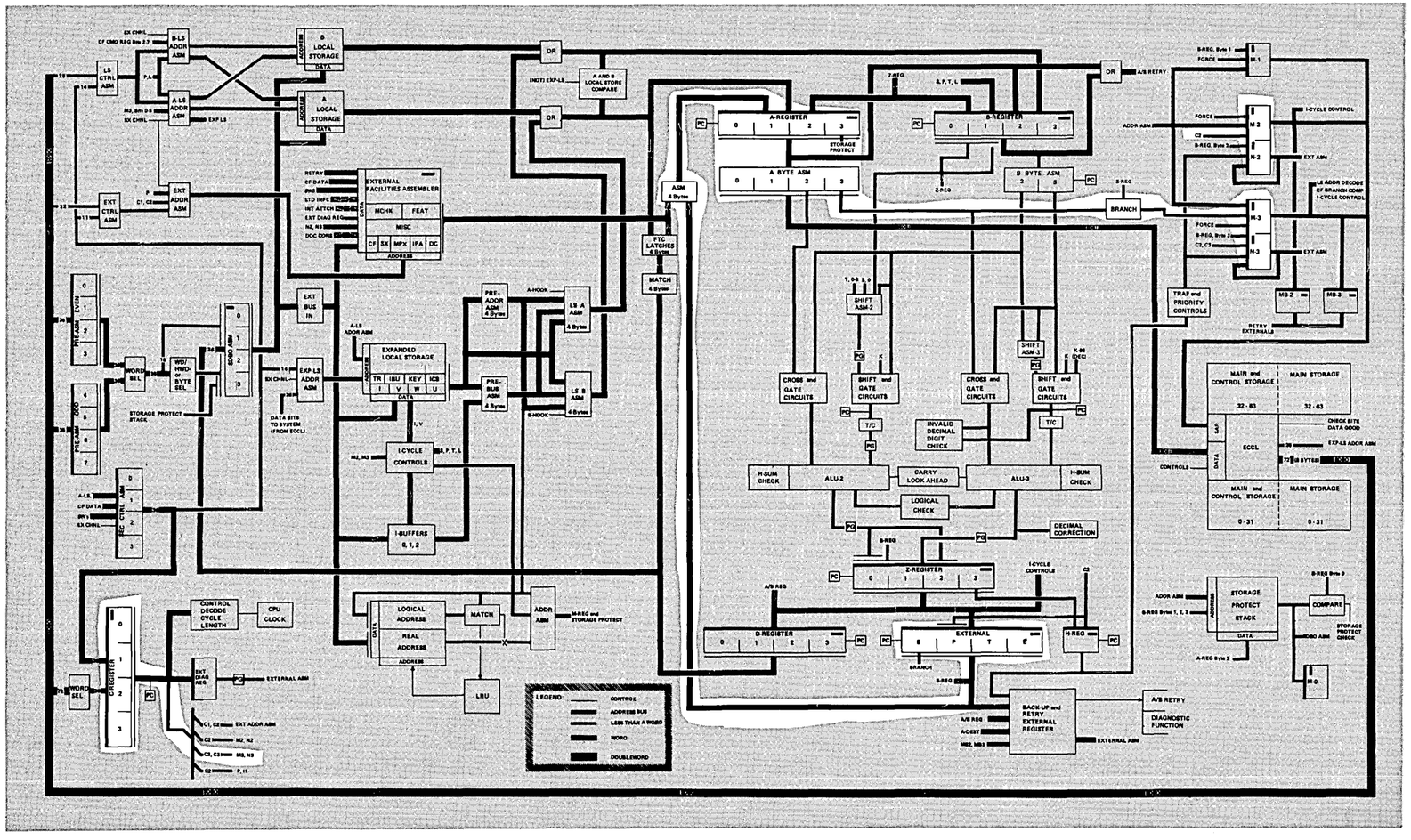
The results of the branch tests are gated to M3 along with next address bits 0, 1 from C3.

C2, the module address, is gated to M2.

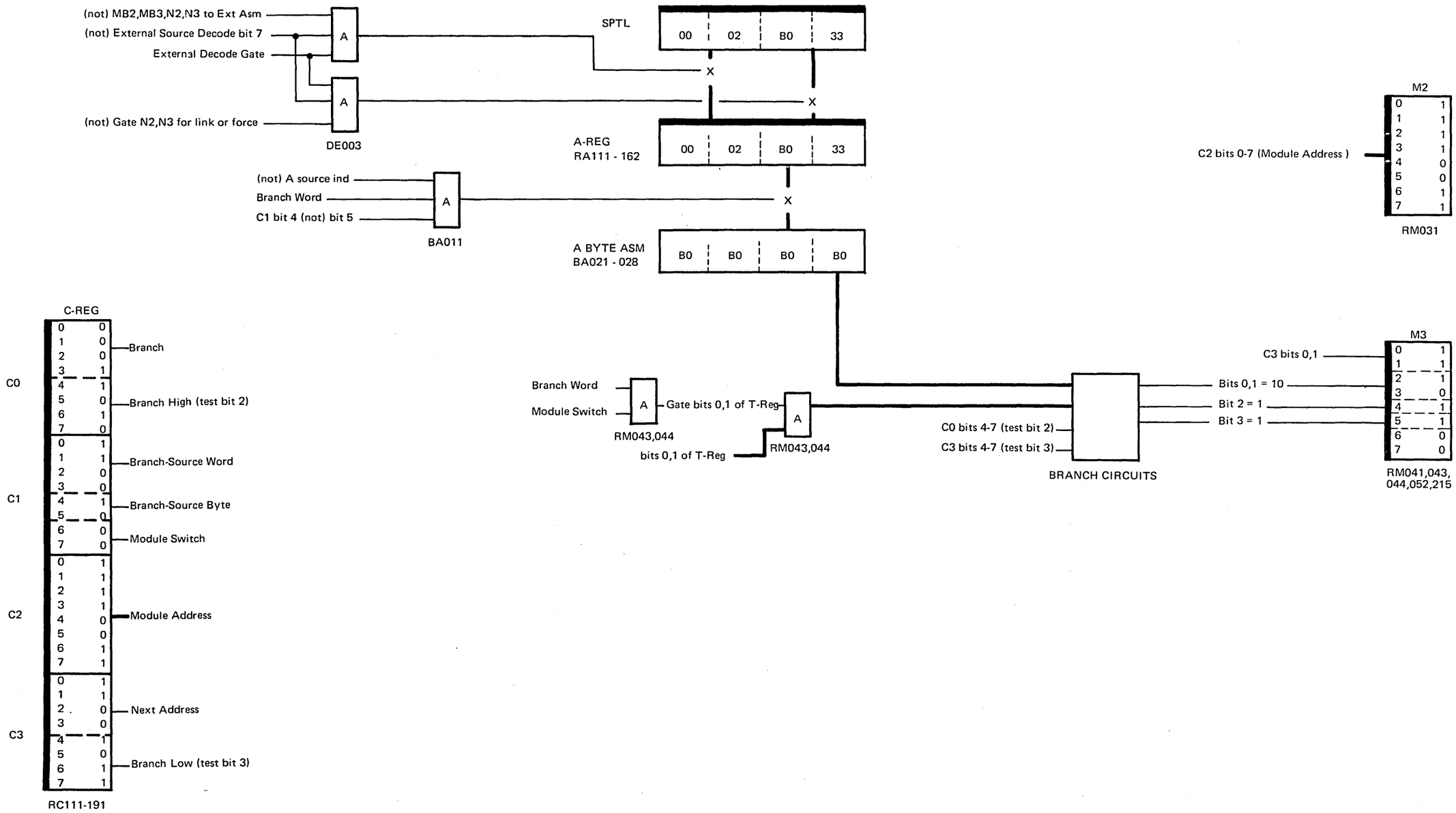
Because of the T-register value, and the module address of the branch set RXOPS, the address branched to is F3EC.



Cycle Time 247.5 ns



Branch-Word Example, Special Module-Switch Function (Part 2 of 2)



Branch and Link or Return Word

BAL

- The branch and link function stores the S-, P-, N2-, and N3-registers into the specified link register.
- Can set the P-register to some designated value or perform a module-switch.
- Can perform up to four-way branching.

RTN

- The return function restores the S, P, M2(N2), and M3(N3) registers with the link-register value.
- Can reset bits of the H-register.
- Can alter the address from the link register.

Branch And Link Statements

The branch-and-link function is specified by either one of two statements:

```
BAL WS P=Khh
BAL WS
```

Where BAL designates branch and link, and WS is any valid word source.

In the first statement, the link information is stored at the word location specified by WS. The P-register is set to the value specified by hh (any hex number 00 through FF).

In the second statement, the link information is stored at the word location specified by WS. A module-switch function is performed when the next address is set up. The lack of a P-register set designation enables the module switch to be performed.

The return function is specified by either one of two statements:

```
RTN WS H, A-, Khh
RTN WS
```

Where RTN designates the return function, and WS is any valid word source.

In the first statement, the data from the word source specified is gated to the following registers:

Byte 0 to S-Reg
Byte 1 to P-Reg
Byte 2 to M2(N2)
Byte 3 to M3(N3)

The H-register bits corresponding to bits in the field specified by hh are reset. For example:

If hh were 02, the bit reset would be bit 6.
If hh were C0, bits 0 and 1 would be reset.
If hh were FF, the entire H-register would be reset.

In the second statement, the data from the word source is gated in the same manner as for statement one.

If any value is specified in the BX, BH, BL columns of the next label field, M3(N3) is set up by the control word rather than from the low byte of the link register.

BX, BH, BL

These fields are used in the same manner as in the branch and module switch word. BX, however, can be 0-F (in return words only) to specify the bits 0-3 of C3 that are used to form part of the low-address byte. If any value is specified in the BX, BH, BL fields for a return word, the next address and branch high and branch low fields are used to set M3 bits 0-5.

BX	BH	BL
0	0	0
1	1	1
2	S1	Z0
3	S0	I0
4	S2	S3
4	S4	S5
F	S6	S7
		I1

In the BX field, the fixed values 0-3 can be used in the branch and link function to set bits 2 and 3 of the M3-register when the next address is set up. The values 0-F can be used in the return function to set bits 0-3 of M3 when the next address is set up. These bits are carried as next address bits in C3, bits 0-3.

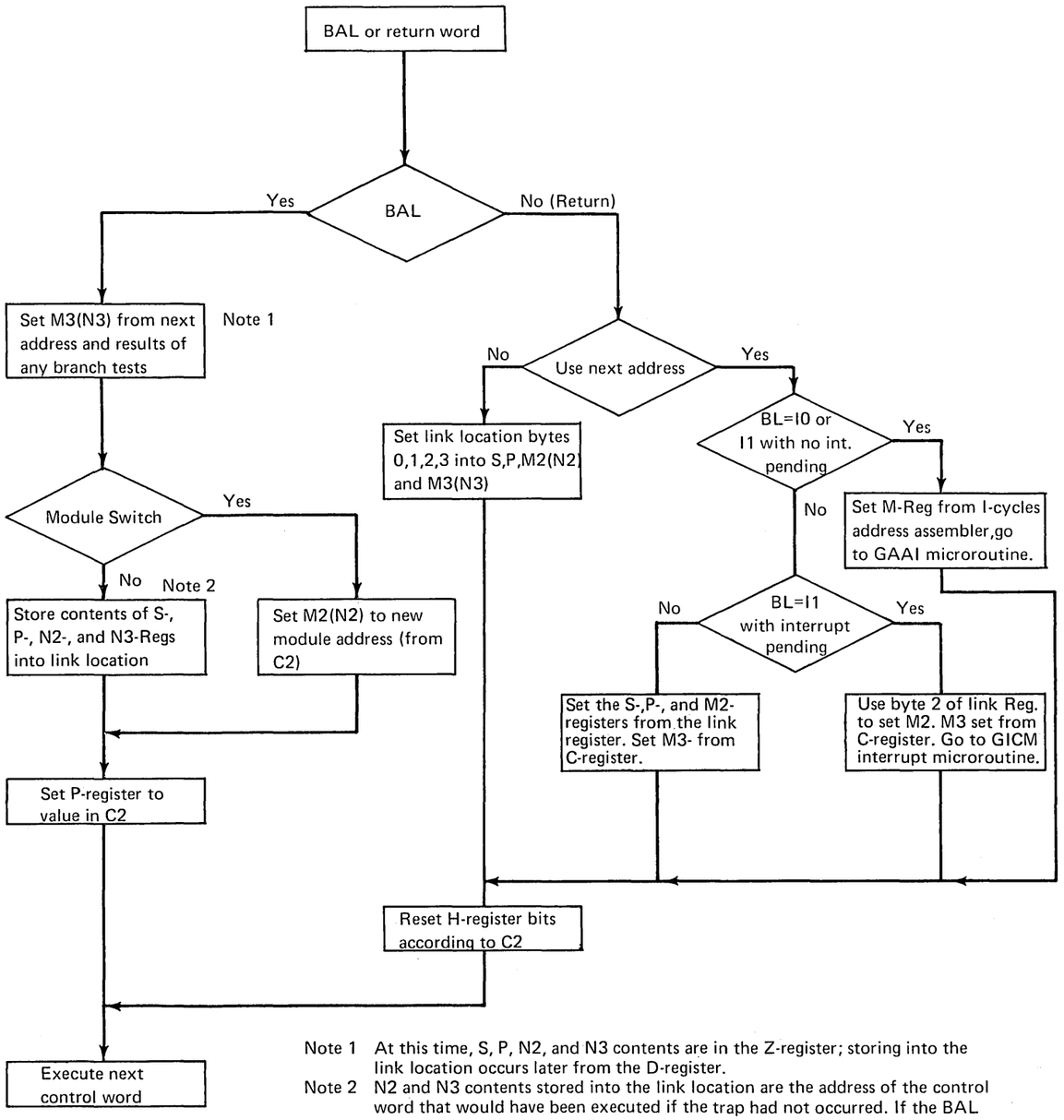
The BH field sets the status of bit 4 of M3, and BL sets the status of bit 5 of M3.

The S symbols refer to specific bits of the S-register to be tested. The symbol Z0 in the BL field causes bit 5 of M3 to be set to 1 if S4, S5=11.

I0 forces an unconditional return to I-cycles.

I1 forces a return to I-cycles if no interrupt is pending.

Bal or Rtn Execution



Note 1 At this time, S, P, N2, and N3 contents are in the Z-register; storing into the link location occurs later from the D-register.

Note 2 N2 and N3 contents stored into the link location are the address of the control word that would have been executed if the trap had not occurred. If the BAL word is not the first word of a trap routine, N2 and N3 contents stored are the address of the BAL word.

Examples

NEXT LABEL	STATEMENT
FMPTP	BAL LNK P=K33

This branch-and-link function stores the S-, P-, N2-, N3-registers in the local-storage register LNK.

The P-register is set to the value 33, and a branch is made to the word at label FMPTP.

NEXT LABEL	STATEMENT
RNDS S6, S7	BAL X

This branch-and-link function stores the S-, P-, N2-, N3-registers in the local-storage register X.

Because no P-register set is given, a module switch takes place. The branch to RNDS is made, and the leg of that branch set is determined by the status of bits 6 and 7 of the S-register. If S6=0 and S7=1, the branch is made to the word of the branch set whose label is RNDS 01.

NEXT LABEL	STATEMENT
	RTN LNK

This statement indicates the return function. The contents of the LNK register are read out and gated in the following manner:

LNK byte 0 to S-Register
LNK byte 1 to P-Register
LNK byte 2 to M2 (N2)
LNK byte 3 to M3 (N3)

NEXT LABEL	STATEMENT
0, S0, S5	RTN X H, A-, K08

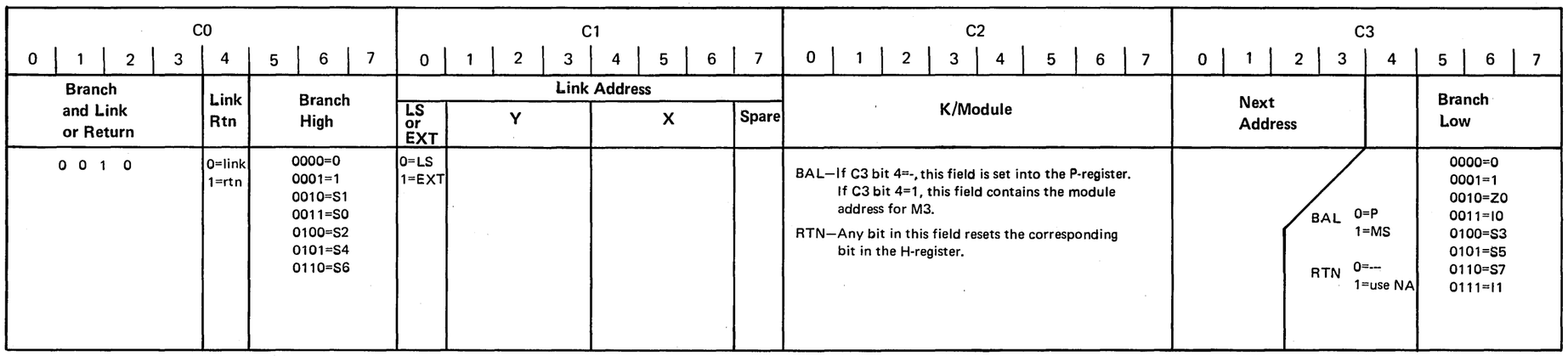
This return function causes the link information to be read from the link register X. Because there is an entry in the next label field, the link address is altered.

Bytes 0 and 1 of the link register X are gated to S and P. Byte 2 of the link register is gated to the M2 (N2) registers.

M3 bits 0-3 are set from C3 bits 0-3, which are 0000 as specified by the 0 in the BX field. M3 bit 4 is set to the value of S0. M3 bit 5 is set to the value of S5.

H-register bit 4 is reset as specified by the 08 in the return statement.

Bit Definition of the Branch and Link or Return Word



C0 BITS 0-3

This field has a value of 0010 to designate the branch and link or return word.

C0 BIT 4

Bit 4=0 specifies the branch-and-link function.
Bit 4=1 specifies the return function.

C0 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 4 of the M3-address register.

The fixed values 0 and 1 designate directly the setting of M3 bit 4. The S symbols refer to S-Register bits to be tested. Bit 4 of M3 is set to the value of the S bit tested.

C1 BITS 0-6

This field directly addresses the link register.
Bit 0=0 indicates the link register to be in local storage.
Bit 0=1 indicates the link register to be an external.
Bits 1-6 contain the X- and Y-lines that access the link register.

C2 BITS 0-7

For the branch-and-link function, this field is either a value to be set into the P-register or a module address to be gated to M2 (N2) when the next address is formed.

If bit 4 of C3=0, this field is set into the P-register.

If bit 4 of C3=1, this field is the module address and is gated to M2(N2) for next address formation.

For the return function, this field provides the bit pattern for resetting the H-Register. Any bit in this field that is on for a return function causes the corresponding bit in the H-register to be reset.

C3 BITS 0-3

This field contains part of the next address. When the branch and link function is performed this field is gated to bits 0-3 of M3(N3) for the next address set up. For the return function, this field is used only when bit 4 of C3=1. Otherwise the next address bits are gated from the link register.

C3 BIT 4

Bit 4=0 for a branch and link causes C2 bits 0-7 to be gated to the P-register.

Bit 4=1 for a branch and link causes C2 bits 0-7 to be gated to the M2(N2)-registers during next address set up.

Bit 4=0 for a return causes the next address to be set from the link register.

Bit 4=1 for a return causes C3 bits 0-3 and the results of any branch tests to be used to set up M3(N3), when the return address is gated from the link register.

C3 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of M3 bit 5 when the next control-word address is formed.

The fixed values 0, 1 designate directly the setting of M3 bit 5. The S symbols refer to S-register bits to be tested. Z0 causes the test of bits 4, 5 of the S-register. If S bits 4, 5 are both = 1, bit 5 of M3 is set to 1.

If I0 is specified in the BL field for a RTN word, an unconditional branch is made to the I-cycles microroutine.

If I1 is specified in the BL field for a RTN word, a branch to the I-cycles microroutine is made if there is no interrupt pending. If an interrupt is pending, the return is to the interrupt-handling microroutine.

If I1 is specified in the BL field for a BAL word, the interrupt-pending condition is tested, and the normal branching procedure is executed.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

Branch and Link Word Example (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
		BAL CX P = K33

Starting Values

SPTL = 00 02 45 22

Address of control word that was to be executed = C3 80

Address of BAL word = D2 08

Objectives

Store S, P, N2, N3 into the link register CX.

Set the P-register with the new value 33.

Description

S, P, N2, N3 are gated to the A-register.

The A-register is gated to both the A byte assembler and the B-register.

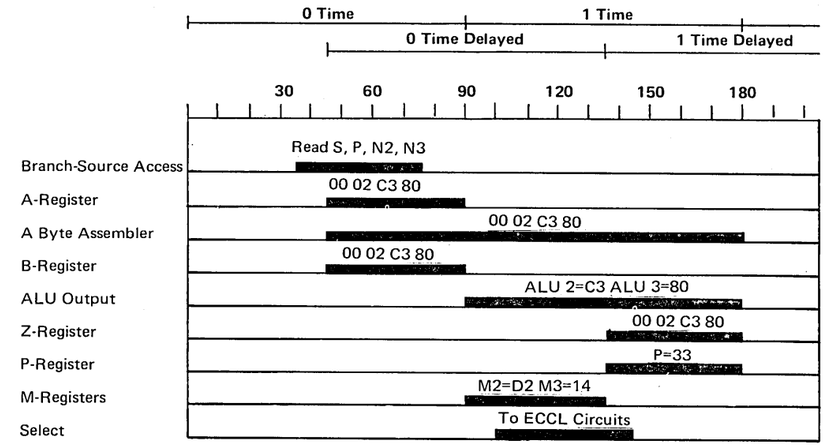
Bytes 2, 3 (N2, N3) of the A byte assembler are gated through the ALUs to bytes 2, 3 of the Z-register.

Bytes 0, 1 (S, P) of the B-register are gated to bytes 0, 1 of the Z-register.

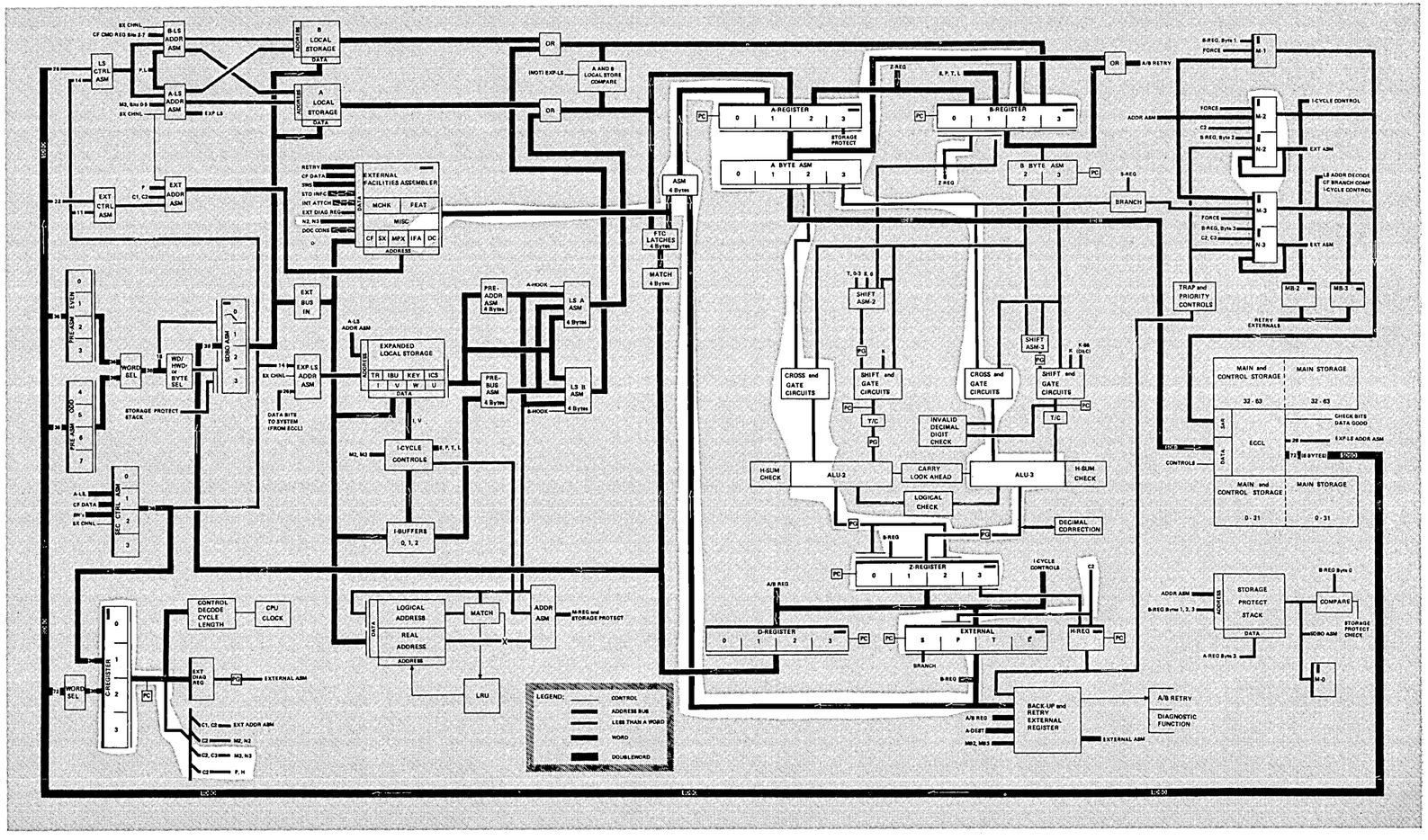
C2 (the K-value) is gated to the P-register.

The Z-register is gated to the D-register and then to the CX-register in the next cycle.

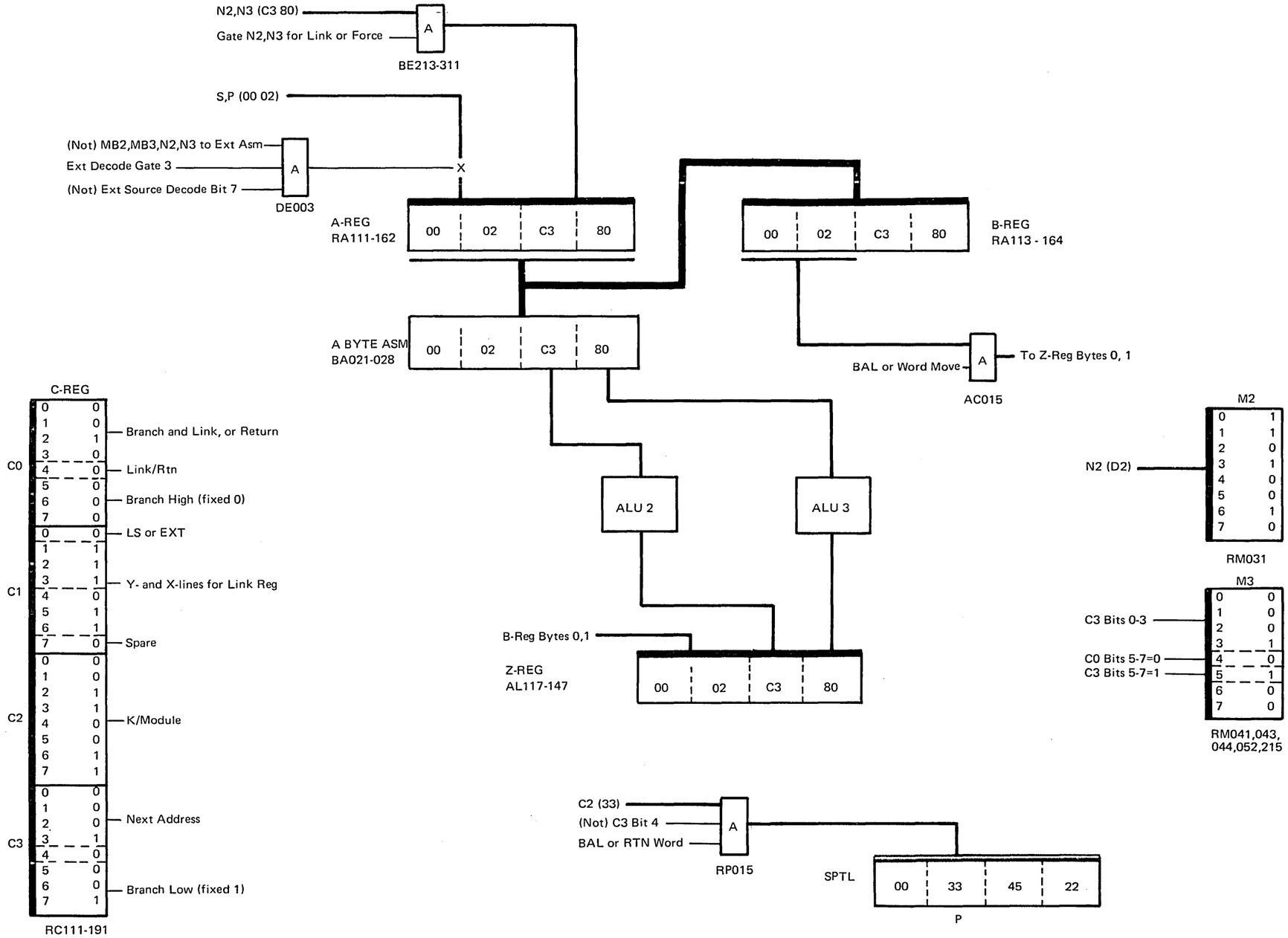
NOTE: The assumption for this example was that the BAL word was the first word of a trap microroutine. If the BAL word had not been the first word of a trap microroutine, the address stored into the link register would have been the address of the BAL word.



Cycle Time = 202.5 ns



Branch and Link Word Example (Part 2 of 2)



Return Word Example (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
LNK = 00 02 34 00		**RTN LNK**

Starting Values

LNK = 00 02 34 00

Objectives (No Interrupt)

Set the S, P, T, L registers with the I-cycles hardware.

S = Op Code

P = 02 (For Op Codes 00-1F, 40-5F, and 80-FF)

P = 62 (For Op Codes 20-3F and 60-7F)

T = 00

L = Immediate byte except L = Immediate byte +1 for Op Codes 20-2F and 35-37.

Set the M-registers from the I-cycles address assembler.

Objectives (Interrupt Pending)

Restore the S- and P-registers from the LNK register.

Set M2 from byte 2 of the LNK register.

Set M3 from C3 bits 0-3, the Branch High field, and the interrupt-pending condition.

Description

Read the LNK register from A-Local Storage.

Gate the LNK contents to the A-register, the A byte assembler, and the B-register.

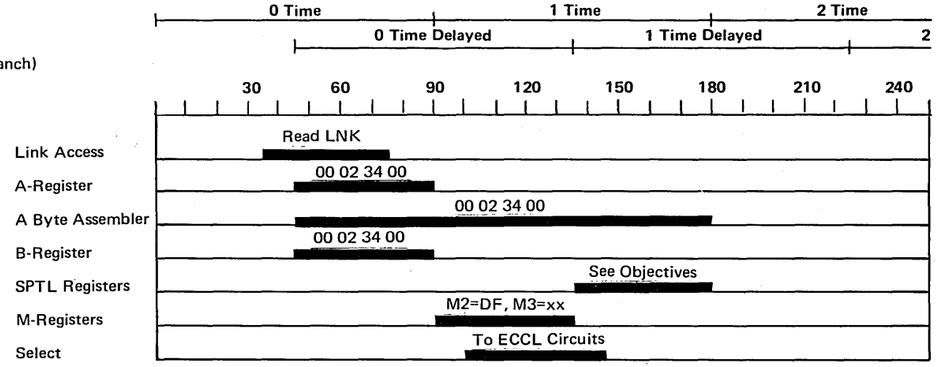
For the (No Interrupt) condition, set SPTL as specified in the Objectives.

Set the M-registers from the I-cycles address assembler.

For the (Interrupt Pending) condition, set the S- and P-registers from bytes 0 and 1 of the B-register. Set M2 from byte 2 of the B-register. Set M3 from the Next Address field, the Branch High field, and the test of the interrupt condition.

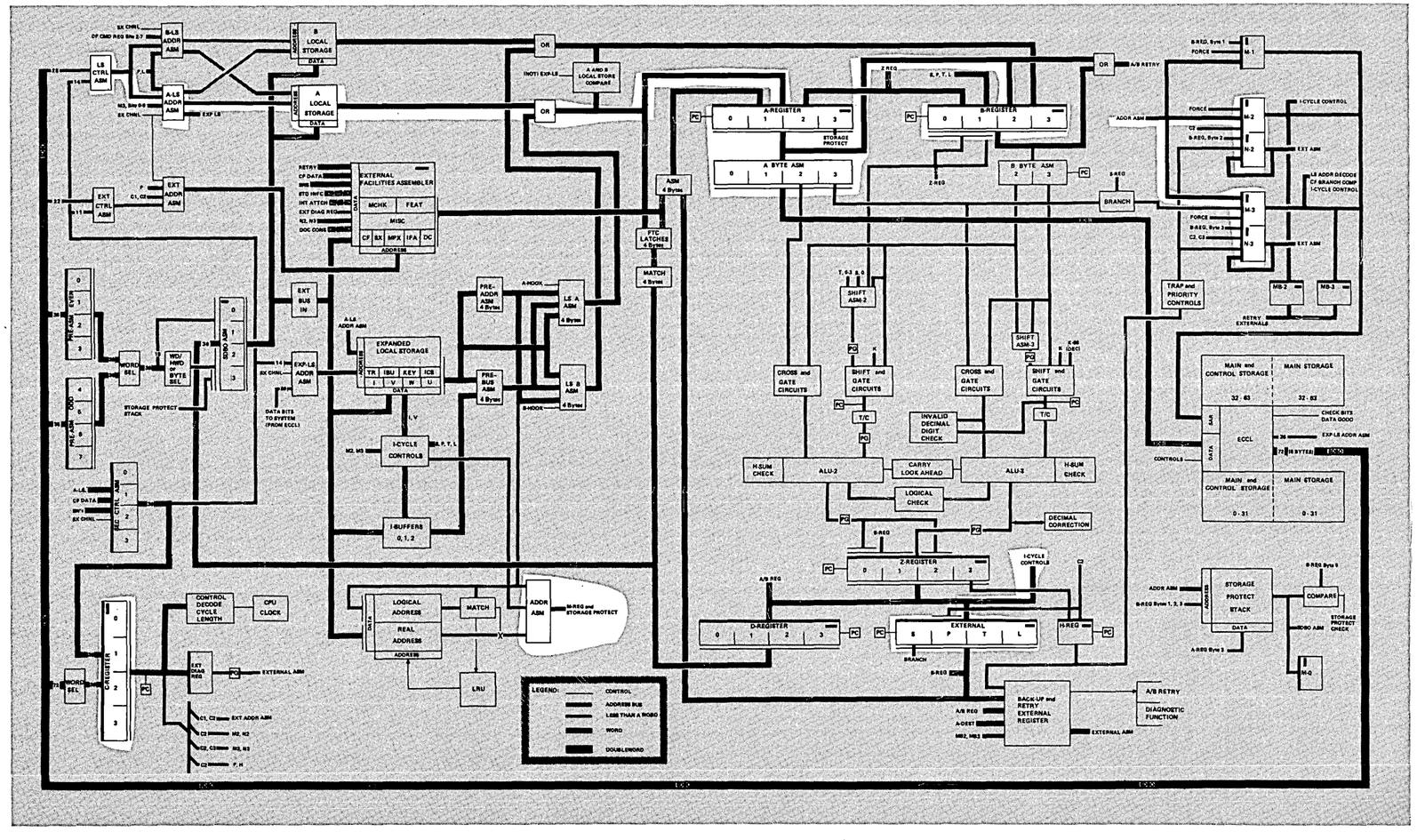
NOTE: If I0 were specified in the BL field, the example would be the same as I1 with no interrupt pending. No testing is done of the interrupt-pending condition.

NO INTERRUPT PENDING or (10 branch)

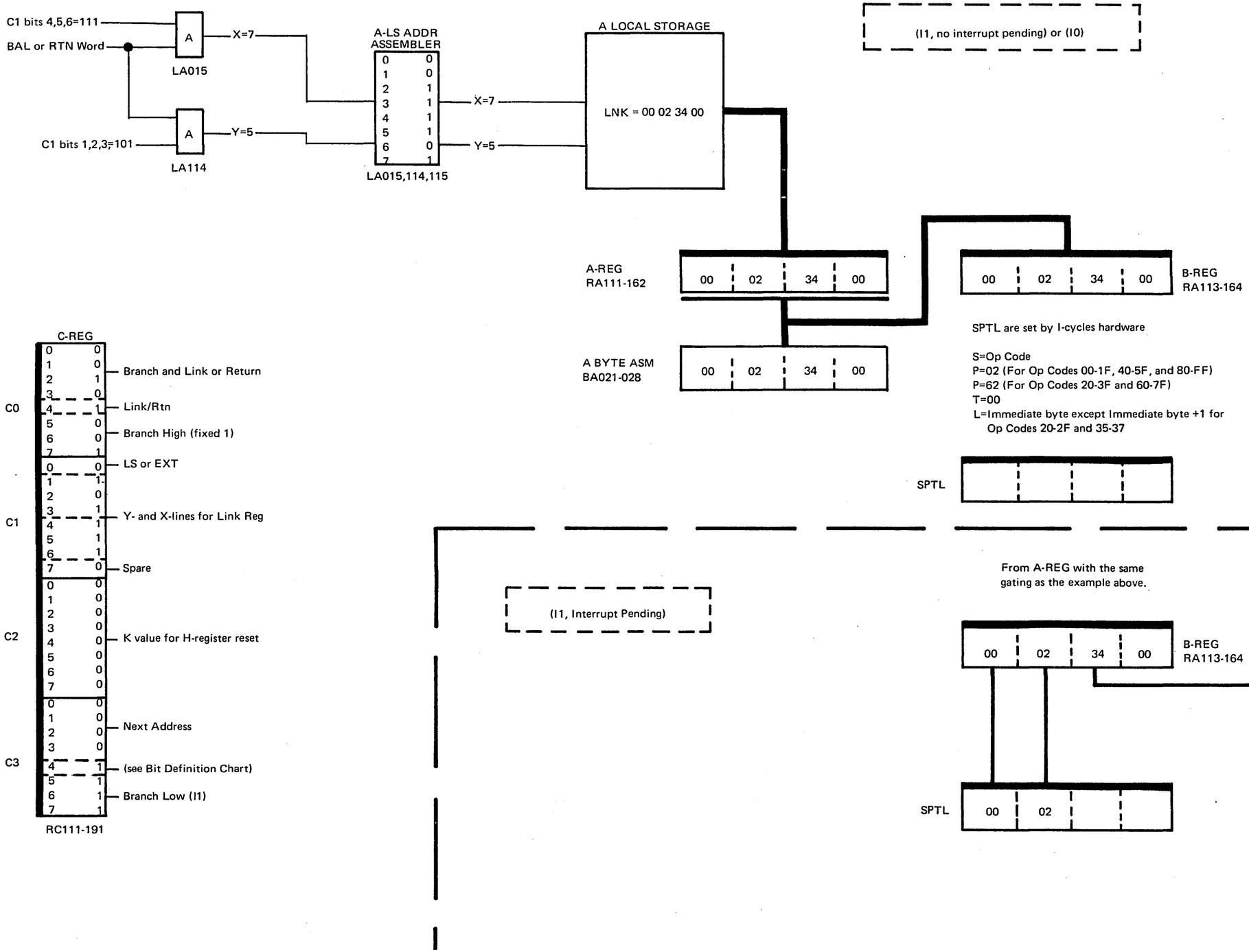


Cycle Time = 247.5 ns

The difference in the timing chart for an Interrupt Pending would be M2=34, M3=0C.



Return Word Example (Part 2 of 2)



Word-Move Word

- Can move a fullword or selected bytes of a word from one local-storage or external location to another.
- Can branch to a word in the current address module.
- Can designate the special STOP function.

The primary function of the word-move word is to move data from one local-storage or external word location to another. A fullword or any combination of selected bytes of the word source can be moved to the designated destination.

The selection of bytes is done through the use of a mask, the value of which is specified in the statement field by a hex digit.

The mask is defined as a four-bit binary value, designated by a hex digit (0-F) in the statement field. The mask bits are labeled 0, 1, 2, 3. The bit numbers correspond to the byte numbers of the word source to be moved. Any bit of the mask that is on specifies a move of the corresponding byte of the source. For example, a hex 9 in the statement is represented by bits 0 and 3 of the mask field on. The move would involve bytes 0 and 3 of the word source. Bytes 0 and 3 of the word source would be moved to bytes 0 and 3 of the destination word. Bytes 1 and 2 of the destination are unchanged.

The Word Move has the facility to branch to any control word in the current address module. Branch testing is limited to bits of the S-register.

The special STOP function causes the following:

- Registers M and N are not set with the next address.
- Access to control storage is not allowed.
- CPU clock continues to run.
- Execute the word-move function each clock cycle.

Word-Move Word Statements

The following types of statements may appear in the statement field of the microlistings:

```
WS=WS, Dh
WS=WS, Sh
WS=WS, Sh STOP
WS=WS, Dh STOP
```

Where WS is any valid word source, and h is the hex digit specifying the mask. The D and S designate which of the words is to be addressed by X and Y values contained in byte 1 of the control word. D specifies destination, S specifies the source. STOP specifies the special-stop function.

Next Label Field

The Next Label field determines the next address formation by the control word. This field is formed in the following manner:

```
YYYYYY BX, BH, BL
```

YYYYYY is the name of the word or branch set being branched to. BX, BH, BL are the Branch Test fields that specify the leg address of the control word in a branch set.

The following table indicates the types of symbols that may be found in the branch fields:

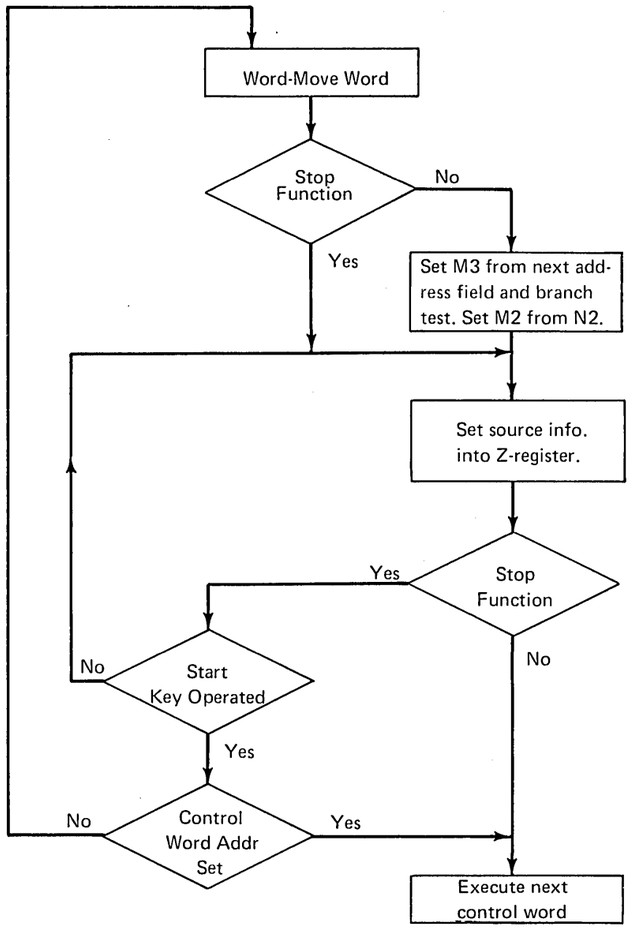
BX	BH	BL
0	0	0
1	1	1
2	S1	Z0
3	S0	S3
	S2	S5
	S4	S7
	S6	

BX — In the BX field, the hex numbers fix the status of bits 2 and 3 of M3 when the next address is formed. The hex value specified by this field is carried in bits 2 and 3 of C3 of the control word.

BH — In the BH field, the numbers 0 and 1 fix the status of bit 4 of M3 when the next address is formed. The S symbols refer to S-register bits. Bit 4 of M3 is set to the status of the S bit tested.

BL — In the BL field, the numbers 0 and 1 fix the status of bit 5 of M3 when the next address is formed. The symbol Z0 causes a test of S4 and S5 to be made. If S4, S5 = 11, bit 5 of M3 is set to 1 when the next address is formed. The S symbols refer to S-register bits. Bit 5 of M3 is set to the value of the S bit tested.

Word-Move Execution



Word-Move Examples

NEXT LABEL	STATEMENT
RSTRT S0, S3	R=Y, D6

The statement designates a move of the Y-register to the R-register under a mask of 6. The mask of 6 indicates that only bytes 1 and 2 of the Y-register are to be moved to R. Bytes 0 and 3 of the R-register are not changed.

The D in the statement specifies that the X- and Y-lines of the destination are to be carried in the control word.

The next address is formed by testing bits 0 and 3 of the S-register, and branching to the branch set labeled RSTRT, the leg of which is identified by the status of bits 0 and 3 of the S-register.

NEXT LABEL	STATEMENT
RTDTG	Q=MCKA, SF

This statement designates a move of the external register MCKA to the local-storage register Q.

A mask of F, indication a fullword move is carried in the mask field of the control word.

The S specifies that the source X- and Y-lines are to be carried in the control word.

The branch is made unconditionally to the control word named RTDTG.

Bit Definition of the Word-Move Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word Move				Version	Branch High			Destination								Source Note 1				Mask				Next Address					Branch Low		
								LS or EXT		Y		X			LS ELS																
0	0	1	1	0	000=0 001=1 010-S1 011=S0 100=S2 101=S4 110=S6			0=LS 1=EXT							0=LS 1=ELS											0=--- 1=Stop		000=0 001=1 010=Z0 011=--- 100=S3 101=S5 110=S7			

Bit Definition of the Word-Move Word (Version 0)

Note 1 The source cannot specify an external register. SPTL, however, can be specified by C2, B0-3.

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word Move				Version	Branch High			LS or EXT		Y		X		Spare		Destination				Mask				Next Address					Branch Low		
0	0	1	1	1	000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6			0=LS ~~1=EXT~~ 0=LS 1=EXT																			0=--- 1=Stop		000=0 001=1 010=Z0 011=--- 100=S3 101=S5 110=S7			

Bit Definition of the Word-Move Word (Version 1)

C0 BITS 0-3

This field is a value of 0011 to designate the word-move word.

C0 BIT 4

This bit specifies the version of the word to be used. Bit 4=0 specifies version 0, which contains the X- and Y-lines of the destination. Bit 4=1 specifies version 1, which contains the X- and Y-lines for the source.

C0 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 4 of M3 when the next address is formed. The values 0 and 1 designate directly the setting of bit 4 of M3. The S symbols refer to bits of the S-register to be tested. Bit 4 fo M3 is set to the value of the S bit tested.

C1 BITS 0-6

This field carries the X- and Y-lines of the source or destination word depending on the setting of bit 4 of C0. C1 bit 0=0 designates that the X- and Y-lines are for a local-storage word. C1 bit 0=1 designates that the X- and Y-lines are for an external word.

C1 BIT 7

In the 0 version of the word, bit 7=0 specifies that the X- and Y-lines are for a local-storage register. Bit 7=1 specifies that the X- and Y-lines are for an expanded local-storage register.

C2 BITS 0-3

If bit 4 of C0 is 0, this field specifies the local-storage word that is to be the source of data to be moved.

If bit 4 of C0 is 1, this field specifies the local-storage or external word that is to be the destination of the data.

The external SPTL is the only external that can be a source in version 0.

C2 BITS 4-7

This is the mask field. Bits in the mask field correspond to bytes of the source word to be moved. A mask bit ON specifies that the corresponding byte of the source is moved.

C2 bit 4 points to byte 0
C2 bit 5 points to byte 1
C2 bit 6 points to byte 2
C2 bit 7 points to byte 3

C3 BITS 0-3

This field is part of the next word address that is set up when this word is executed. Bits 0-3 of C3 are gated to M3 bits 0-3.

C3 BIT 4

When this bit is on, the special STOP function is specified. When this bit is off, normal word-move operation takes place.

C3 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 5 of M3 when the next address is formed. The values 0 and 1 designate directly the setting of bit 5 of M3. Z0 causes the test of S4 and S5. If S4, S5 = 11, bit 5 of M3 is set to 1. The S symbols refer to bits of the S-register. Bit 5 of M3 is set to the value of the S bit tested.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

Word-Move Word Example (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
LXMD		MX = MF, S7

Starting Values

MX = 00 57 83 61
MF = 09 FS CD BA
Address of LXMD = 512 C

Objectives

Move bytes 1, 2, 3 of the MF-register to bytes 1, 2, 3 of the MX-register.

Branch to LXMD.

Description

Read the MF register from local storage A. Gate MF to the A-register.

The A-register is gated to both the A byte assembler and the B-register.

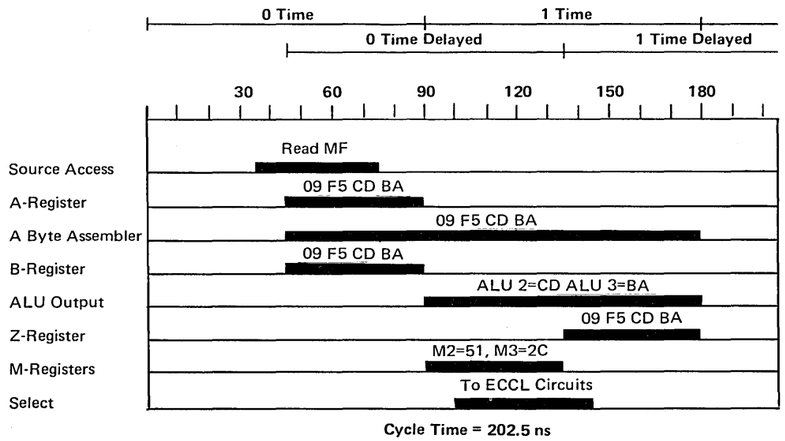
Bytes 2, 3 of the A byte assembler are gated through the ALUs to bytes 2, 3 of the Z-register.

Bytes 0, 1 of the B-register are gated to bytes 0, 1 of the Z-register

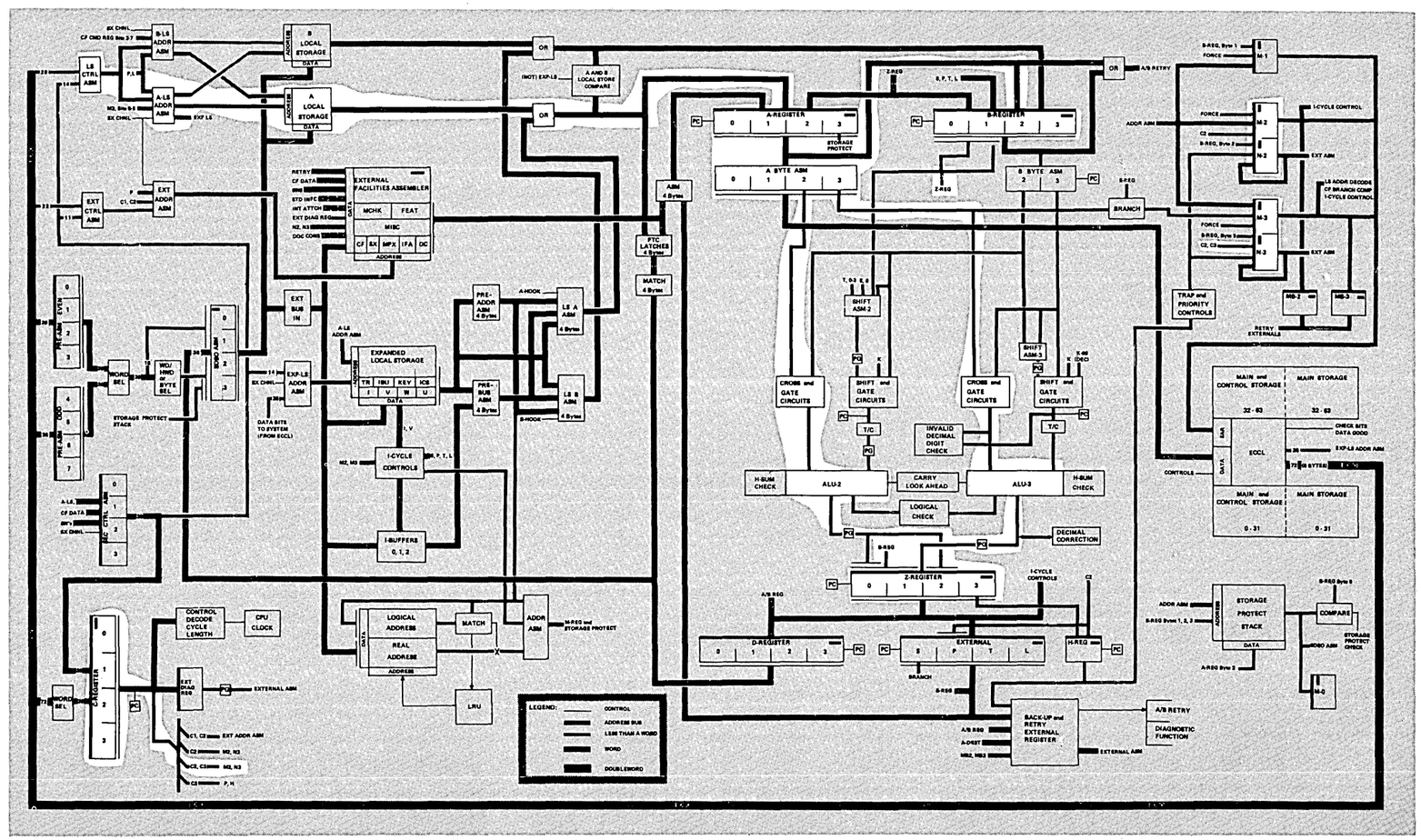
Set M3 with the address bits from the C-register.

Set M2 from N2.

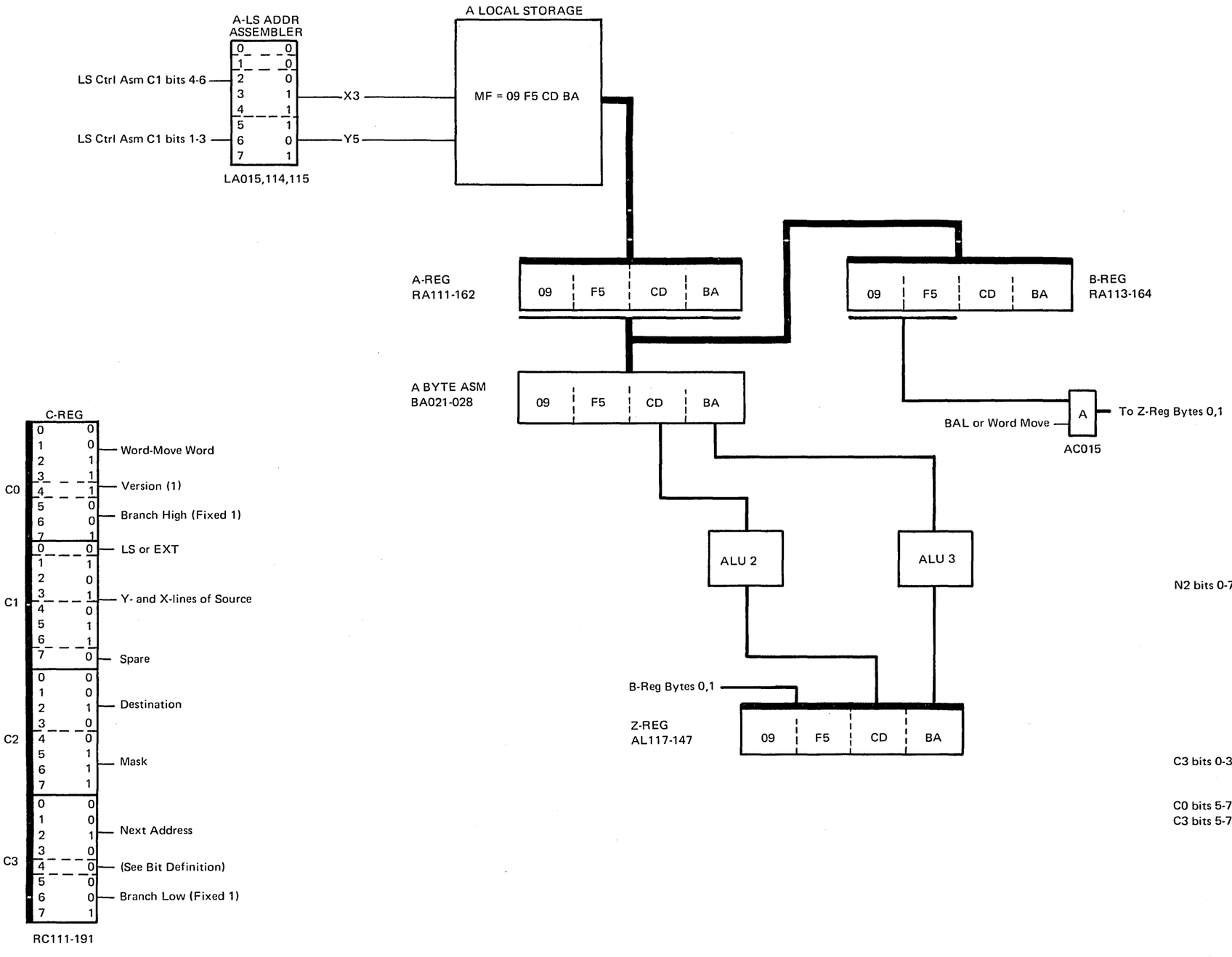
In the cycle following the word-move word, the D-register is set from the Z-register. The D-register is then gated to the MX-register.



Cycle Time = 202.5 ns



Word-Move Example (Part 2 of 2)



Storage Word

- Can read data from or store data into:
  - Main storage
  - Control storage
  - Local storage
  - External registers
  - Protect stack
- Can perform a Branch and Link.
- Capable of up to four-way branching.
- Two versions:
  - Non-K-Addressable
  - K-Addressable

The main function of the storage word is to move data between a storage location and some working area in the CPU.

In the non-K-addressable type, the data address is located in a local storage register. No external register may be used as an address source except the SPTL register. The address contained in the address-source register may be a control-storage address or a main-storage address. In the case of a control-storage address, only the low 16 bits of the address-source register are used in setting the M-register. For main-storage addressing, the low 20 bits are used to set the M-register.

The facility to update the address in the address-source register is available for the non-K-addressable version only. The address update may be an increment or a decrement operation. The value of the update is normally implied by the subform of the word. For example:

Word operation, update by 4.
Halfword operation, update by 2.
Byte operation, update by 1.

The update function applies to the address-source register. After the contents of the address-source register are used to set the M-register, an update of the address-source contents can occur.

A special case where the update value is not implied occurs when the special stat set TH is specified. The update value is indicated by the value of bits 0-3 of the T-register.

T Bits 0-3	Update Value
0	0
1,2,4,5,8,A	1
3,6,C	2
7,E	3
9,B,D,F	4

In the K-addressable type, the data address is formed by:

- Forcing parts of the M-Regs to a specified value.
- Setting the remainder of the M-Regs from the K field of the storage word.

Storage Word Statement

The form of the statement area for a non-K-addressable word is:

(Stat Set) (Subform) (Data Reg) (Mode, Addr Source, Update, Special)

The Stat Set field applies only to the non-K-addressable word, with the decrement-count function specified.

Valid Stat Set symbols are:

S2
S45
Z6

Refer to the bit-definition description for details of the Stat Set symbols.

The valid subforms for the non-K-addressable word are:

RDW	Read word
STW	Store word
RDH	Read halfword
STH	Store halfword
RDB	Read byte
STB	Store byte
SSK	Store protect key
ISK	Read protect key

The Data Register can be any local-storage or valid external-register location.

The mode of addressing control storage is indicated by the symbol CS written before the address-source symbol. This specifies that the low two bytes of the address-source register are to be used to set M2 and M3 for a control-storage access.

The non-K-addressable word normally operates in address-adjust mode. Storage protect may be suppressed by the symbol NPR written after the address-source symbol.

No address-adjust with storage protection is active when the symbol ADJ is not found in front of the address-source register symbol.

Example

```
STW Q ADJ, R
```

If local mode is active, address-adjustment circuits provide address to M-register instead of directly from B-register. See OS/DOS compatibility description for details of address-adjustment hardware.

```
STW Q R
```

Storage protect active, no address-adjustment.

The address-source register may be any local-storage location. In address-adjust mode, only the expanded local-storage registers I, U, V, W, and TR may be used for the address-source register. Externals cannot be used as an address source, with the exception of the SPTL register.

The update function is indicated by a plus or minus sign and a value written after the address-source symbol. The update value is actually implied by the subform:

Word operation —— update value 4.
Halfword operation —— update value 2.
Byte operation —— update value 1.

The form of the statement area for a K-addressable storage word is:

(Subform) (Data Reg) (Mode, K-value)

The valid subforms for the K-addressable word are:

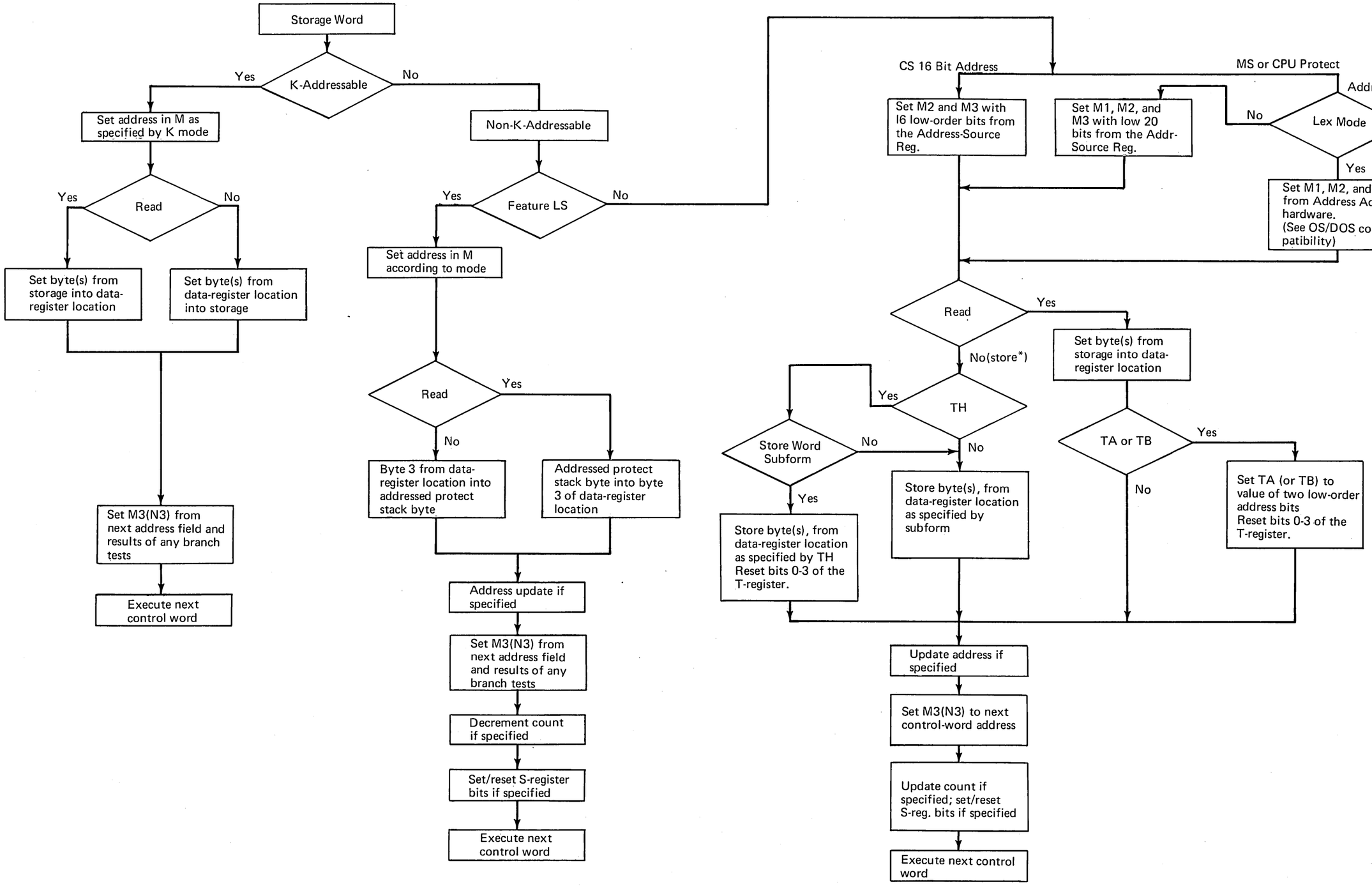
RDW	Read word
STW	Store word
RDH	Read halfword
STH	Store halfword
RDB	Read byte
STB	Store byte

The data register can be any local-storage or valid external-register location.

The modes that can be specified are:

DM,hh	Address main storage, M1 and M2 set to 0. M3 set with K field value.
CM,hh	Address control storage, using current module address in M2, set M3 with hh.
DC,hh	Address control storage, setting M2=FF and set M3 with hh.
Ch,WS	Address control storage, setting M2=Fh and set M3 with contents of byte 3 of the address-source register indicated by the WS symbol.

Storage-Word Execution



Non-K-Addressable Examples

RDW U V

Read a word from main storage, using the address found in the V-register to set the M-register. The word read from main storage is placed into the U-register

RDH I W+2

Read the halfword, addressed by the contents of the W-register, from main storage and place the halfword into bytes 2 and 3 of the I-register.

After the halfword has been accessed, increment the contents of the W-register by 2.

RDB R Y-1

Read the byte, addressed by the contents of the Y-register, from main storage and place the byte into the byte-3 location of the R-register.

After the byte has been accessed, decrement the contents of the Y-register by 1.

STW Q U+TH

This word is the store-under-mask version. The contents of the Q-register are stored at the location specified by the address found in U, under the mask designated by bits 0-3 of the T-register. Any of the mask bits ON causes the corresponding byte of the Q-register to be stored. If bits 0-3 of T were 0110, bytes 1 and 2 of Q would be stored. If bits 0-3 of T were 1011, bytes 0, 2, and 3 of the Q-register would be stored.

The update for this word is not done in the normal manner. The value of the update is specified by the bit setting of the T-register bits 0-3. Refer to the table showing the T-register setting and the update values.

If this statement had been written STW Q U, TH the operation would be identical, but no update would be performed.

RDW U CS, V+4

Read a word from control storage, addressed by the contents of the V-register, and place the word into the U-register. The low bytes of V, 2 and 3, set M2 and M3 to address control storage.

After the read is performed, update the contents of the V-register by 4.

RDW V R, TA

Read a word from main storage, addressed by the contents of the R-register, and place that word into the V-register.

The symbol TA means, set bits 4 and 5 of the T-register to the value of bits 6 and 7 of byte 3 of the address used to access storage. This setting occurs before any update to the address has been performed. In this example, no address update was called for.

If the statement were RDW V R, TB the operation would be the same except that bits 6 and 7 of the T-register would be set to the value of the low-order address bits.

When TA or TB are called for as shown, the high bits (0-3) of the T-register are reset.

Z6 STW Q W, DCNT

Store the word from the Q-register, at the location in main storage, addressed by the contents of the W-register.

The symbol DCNT specifies the decrement-count operation. The count is contained in bytes 2 and 3 of the odd register of the even/odd pair formed by the address-source and the count registers. In this example the address of the address-source register W is 12; the count register therefore, is at address 13.

The count is always decremented. The value of the decrement is the same as that used to update the address-source register. If an address update is not called for, the decrement value is still the same as if an address update were to be performed.

The stat set symbol Z6 causes the following action:

Set S4=1 if bits 0-5 of the low-order byte of the count register, after the update, are all zero.

Set S4=0 if bits 0-5 of the low-order byte of the count register, after the update, are not all zero.

Set S5=1 if bits 4-7 of the low-order byte of the count register, after the update, are all zero.

Set S5=0 if bits 4-7 of the low-order byte of the count register, after the update, are not all zero.

K-Addressable Examples

RDW Q DM, C4

Read the word from main storage, directly addressed in the following manner:

M1 forced to 0
M2 forced to 00
M3 set to C4

The symbol DM specifies a direct main-storage access. The K-mode field (C1 bits 6, 7) in the control word contains 00 to indicate this function.
The word read from 0 00 C4 is placed into the Q-register.

STB V3 CM, FE

Store byte 3 of the V-register into control storage, addressed in the following manner:

M2 retains current module address
M3 set to FE

The symbol CM specifies a control-storage access using the current module address. Byte 3 of the V-register is stored at address FE in the current address module. This is the module that contains the storage word being executed.

STB EXTDST DC, B3

Store the external byte (EXTDST) in control storage, addressed in the following manner:

M2 forced to FF
M3 set to B3

The symbol DC indicates a direct access of control storage. The module address FF is the area specified as the directly addressable control-storage area.

RDW R C7, V3

Read a word from control storage addressed in the following manner:

M2 high 4 bits forced to F low 4 bits set to 7
M3 set from byte 3 of the V-register

The symbol Ch (h=hex digit), specifies a control-storage access using an indirect word address and a direct means of specifying the module address.
The word read from control storage is placed into the R-register.

RDH DK DC, D8

Read a halfword from control storage addressed by:

Forcing M2 to FF
Setting the K field D8 into M3

Set the halfword read out of control storage into bytes 2, 3 of the expanded local-storage register DK, address (7C).

Bit Definition of the Storage Word (Non-K-Addressable)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Storage Word		Subform			Branch High			Data Register				K or Inc/Dec		Stat Set		Address Source				Modes		Special Stat Set		Next Address					Branch Low		
0	1	000=Read Word 001=Store Word 010=Read Half Wd 011=Store Half Wd 100=Read Byte 101=Store Byte 110=RDWRL/RDMP 100=Read Key 101=Store Key Subform for (special) stat set			000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6 111=M6							00=K-Addr 01=No Adr 00=K-Addr 01=No Addr update 10=+ 11=-		00=— 01=S2 10=S45 11=Z6						00=CS 16 bit address 01=MS 10=ADR ADJ 11=CPU prot		00=— 01=TA 10=TB 11=special 00=— 01=TH 11=special					0=— 1=Dec Cnt		000=0 001=1 010=Z0 011=SDC/VAL(0) 100=S3 101=S5 110=S7 111=M7		

Co BITS 0, 1

This field identifies the control word as a storage word.

C0 BITS 2-4

This field specifies the subform of the storage word. The designations for reading or storing, and the size of the data to be handled, are specified by this field.

Read Word — The fullword read from main or control storage is set into the data-register location.

Store Word — The contents of the entire data register are stored at the location in storage specified by the address-source register.

Read Halfword — The halfword read from main or control storage is set into bytes 2 and 3 of the data register.

Store Halfword — Bytes 2 and 3 of the data register are stored at the halfword location specified by the address-source register.

Read Byte — The byte read from main or control storage is set into byte 3 of the data register.

Store Byte — Byte 3 of the data register is stored at the location specified by the address-source register.

RDWLR/RDMP — Word operation for fetching matrix printer word 1 and word 2 for decode.

Read Key — The byte in the storage protect stack is set into byte 3 of the data-register location.

Store Key — Byte 3 of the data register is stored at the location in the stack that is specified by the address-source register.

C0 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 4 of M3 when the next address is formed. The fixed values, 0 and 1, designate directly the setting of bit 4 of M3. The S symbols refer to bits of the S-register. Bit 4 of M3 is set to the value of the tested S bit. M6 refers to bit 6 of the low-order byte of the address-source register after address updating has been performed. Bits 4 of M3 is set to the value of bit 6 of the low-order byte of the updated address.

C1 BITS 0-3

This field designates the local-storage or external register that is to be the source or destination of data. On read operations, this field specifies the destination of the data read from storage. On store operations, this field specifies the source of data to be stored.

C1 BITS 4, 5

This field specifies the address-update operation. The update constant is implied by the subform of the word:

Word operations, constant is 4
Halfword operations, constant is 2
Byte operations, constant is 1

Bits 4, 5=01 means no address update.
Bits 4, 5=10 means an increment update.
Bits 4, 5=11 means a decrement update.
Bits 4, 5=00 designates the K-addressable version of the storage word.

C1 BITS 6, 7

This is the Status Set field. The status-set facility applies to storage words using the decrement-count facility.

S2 — S2 is set to 1 if the count is not zero after update. S2 is set to 0 if the count is zero after update.

S45— S4=1 if bits 0-3 of count byte 3 are zero after update, set S4 to 0 otherwise.
S5=1 if bits 4-7 of count byte 3 are zero after update, set S5 to 0 otherwise.

Z6 — S4=1 if bits 0-5 of count byte 3 are zero after update, set S4 to 0 otherwise.
S5=1 if bits 4-7 of count byte 3 are zero after update, set S5 to 0 otherwise.

C2 BITS 0-3

This field designates the local-storage register that contains the address (main or control storage) that data is read from or stored into.

The external register SPTL can be addressed by this field. No other external can be used as an address-source register.

C2 BITS 4, 5

This field specifies the kind of addressing that is to be performed by the address from the address-source register.

Bits 4, 5=00 specifies that the control-storage area is to be addressed. Bytes 2 and 3 of the address-source register set M2 and M3.

Bits 4, 5=01 specifies a main-storage access with no storage protection in effect. Bytes 1, 2, and 3 of the address-source register are used to set M1, M2, and M3.

Bits 4, 5=10 specifies address-adjust mode. M1, M2, and M3 are set from the address-adjustment hardware. See OS/DOS compatibility for a description of address adjustment.

Bits 4, 5=11 specifies a main-storage access with storage protection. Bits 0-5 of byte 0 of the address-source source register contain the protect key. Byte 1, 2, and 3 of the address-source register are used to set M1, M2, and M3.

C2 BITS 6, 7

This field contains special status-set information.

For Read Operations:

- 6, 7=00 no special status set.
- 6, 7=01 set bits 4 and 5 of the T-register with the value of bits 6 and 7 of byte 3 of the address source, before address update. T-register bits 0-3 are reset.
- 6, 7=10 set bits 6 and 7 of the T-register with the value of bits 6 and 7 of byte 3 of the address source, before address update. T-register bits 0-3 are reset.
- 6, 7=11 allows the subform field to specify the read-key operation.

For Store Operations:

- 6, 7=00 no special status set.
- 6, 7=01 used with store-word operations only. The bytes that are stored depend on the setting of bits 0-3 of the T-register. Each bit of the T-register corresponds to a byte of the source. Any bit, 0-3 of T that is on, results in the corresponding byte of the source being stored.
- 6, 7=10 Reserved
- 6, 7=11 allows the subform field to specify the store-key operation.

C3 BITS 0-3

This field represents part of the next control-word address and is gated to bits 0-3 of the M3 register when the next address is formed.

C3 BITS 4

When bit 4=1, the decrement-count facility is in effect. The count is always decremented by the value that the address is updated by. The address and count must be in an even/odd pair of local-storage registers: the address in the even register and the count in the odd register. The count is in bytes 2 and 3 of the odd register. When only decrement count is specified, the count must still be in an odd local-storage register.

C3 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 5 of M3 when the next address is formed. The fixed values 0 and 1 designate directly the setting of M3 bit 5.

Z0 causes bits 4 and 5 of the S-register to be tested. If S4, S5=11, M3 bit 5 is set to 1. If either S4 or S5 is 0, set M3 bit 5 to 0.

S3, S5, and S7 refer to S-register bits. M3 bit 5 is set to the value of the tested S bit.

M7 refers to bit 7 of the low-order byte of the address-source register after updating has been performed. Bit 5 of M3 is set to the value of bit 7 of the low-order updated address.

SDC (SUPPRESS DATA CHECK)

When C3, B5-7 = 011, SDC (Suppress Data Check) is specified. By bringing up the validate line to memory, this function prevents setting an ECC single- and double-error indication for data stored into storage, and results in good ECC for both new data being stored and unchanged data in storage. Note, however, that other checks, such as a P-register parity check, will be indicated if they occur, even through SDC is specified. SDC is applicable to diagnostic microprogram operations. It is used mainly to disable the ECC circuitry so that storage can be validated.

If SDC is specified on a read operation, the setting of MCKB bit 5 (double error) is prevented and MCKB bit 1 is set in the case of a double error. This prevents a double error from causing a machine check. With MCKB bit 1 set, the next read operation with SDC specified allows the data with the double error to be read from storage by bringing up the maintain-data line to storage.

The branch-low bit is 0 when SDC is specified. That is, M3(N3), B5 is set to 0 when the next-control-word address is set into the M-register.

VAL (VALIDATE)

Validate is active only with a "store word" subform. When active, Validate forces good ECC bits on the doubleword in storage, specified by the storage address.

Bit Definition of the Storage Word (K-Addressable Type)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Storage Word		Subform			Branch High			Data Register				K Addr		K Modes		Address Source				K				Next Address					Branch Low		
0	1	000=Read Word 001=Store Word 010=Read Half Wd 011= 011=Store Half Wd 100=Read Byte 101=Store Byte			000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6 111=M6							00		00=MS (0 00 KK) 01=CS (CM KK) 10=CS (FF KK) 11=CS (FK 8 bit ADD Addr)												0=---			000=0 001=1 010=Z0 011=--- 100=S3 101=S5 110=S7 111=M7		

C0 BITS 0,1

This field identifies the control word as a storage word.

C0 BITS 2-4

This field specifies the subform of the storage word. The designations for reading or storing, and the size of the data to be handled, are specified by this field.

Read Word — The fullword read from main or control storage is set into the data-register location.

Store Word — The contents of the entire data register are stored at the location specified by the mode and K values indicated.

Read Halfword — The halfword read from main or control storage is set into bytes 2 and 3 of the data register.

Store Halfword — Bytes 2 and 3 of the data register are stored at the halfword location specified by the mode and K values indicated.

Read Byte — The byte read from main or control storage is set into byte 3 of the data register.

Store Byte — Byte 3 of the data register is stored at the location specified by the mode and K values indicated.

C0 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 4 of M3 when the next address is formed. The fixed values, 0 and 1, designate directly the setting of bit 4 of M3. The S symbols refer to bits of the S-register. Bit 4 of M3 is set to the value of the tested S bit.

M6 — Although no address update can be called for in the K-addressable version, an update to the K value does occur in the ALU. The symbol M6 causes bit 6 of Z-register 3 to be tested. Bit 4 of M3 is set to the value of bit 6 of Z3.

C1 BITS 0-3

This field designates the local-storage or external register that is to be the source or destination of data. On read operations, this field specifies the destination of the data read from storage. On store operations, this field specifies the source of the data to be stored.

C1 BITS 4, 5

This field is 00 to indicate the K-addressable version.

C1 BITS 6, 7

This field specifies the K-addressable mode and M-register settings.

MS 0 00 KK — This mode provides for accessing of the low 256 bytes of main storage. The M-register is set in the following manner:

M1=0
M2=00
M3=KK (C2 bits 0-7)

CS Current Module + KK — This mode specifies an access of control storage. The set of the M-register is made in the following manner.

M2=N2
M3=KK (C2 bits 0-7)

Any word in the current address module can be accessed.

CS FF KK — This mode specifies an access of control storage, setting M in the following manner:

M2=FF
M3=KK (C2 bits 0-7)

CS FK + 8-bit address — This mode specifies an access of control storage, setting M in the following manner:

M2=FK, K is C2 bits 4-7.
M3=low byte of address-source register.

C2 BITS 0-3

This field normally contains the high hex digit of the K value.

When the CS FK + 8-bit address mode is specified, this field designates the local-storage register that contains the byte of address that is used to set M3 for a control storage access.

C2 BITS 4-7

This field contains the low hex digit of the K value specified.

C3 BITS 0-3

This field represents part of the next control-word address and is gated to bits 0-3 of M3 when the next address is formed.

C3 BITS 5-7

This field specifies the branch test or fixed value that determines the setting of bit 5 of M3 when the next address is formed. The fixed values 0 and 1 designate directly the setting of M3 bit 5.

Z0 causes a test of bits 4 and 5 of the S-register to be made. If S4, S5=11, M3 bit 5 is set to 1. If either S4 or S5 is 0, set M3 bit 5 to 0.

The S symbols refer to S-register bits. M3 bit 5 is set to the value of the tested S bit.

M7—Although no address update can be called for in the K-addressable version, an update to the K value does occur in the ALU. The symbol M7 causes bit 7 of byte 3 of the Z-register to be tested. Bit 5 of M3 is set to the value of bit 7 of the Z-register.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

Storage-Word Example, Read Halfword (Part 1 of 4)

NEXT LABEL	STAT	STATEMENT
BOUND M7		**RDH R Q + 2**

Starting Values

Q - Reg = 00 01 C3 F0

Contents of 1 C3 F0 = 2E CF B7 9A

Objectives

Read the halfword from main storage as specified by the address in the Q-register.

Set the halfword read from main storage into bytes 2, 3 of the data register R.

Update the address in the Q-register by +2.

Branch to the leg of the branch set BOUND, as specified by the low bit of the updated address found in the Q-register.

Description FIRST CYCLE

Read the contents of the Q-Reg from B local storage and gate to the B-register. This is the main-storage address.

Gate the address from the B-register to the M-registers and send the select pulse to the storage circuits.

Read the contents of the R-Reg from A local storage and gate to the A-register and A byte assembler. This data is not used, but the access is just not prevented.

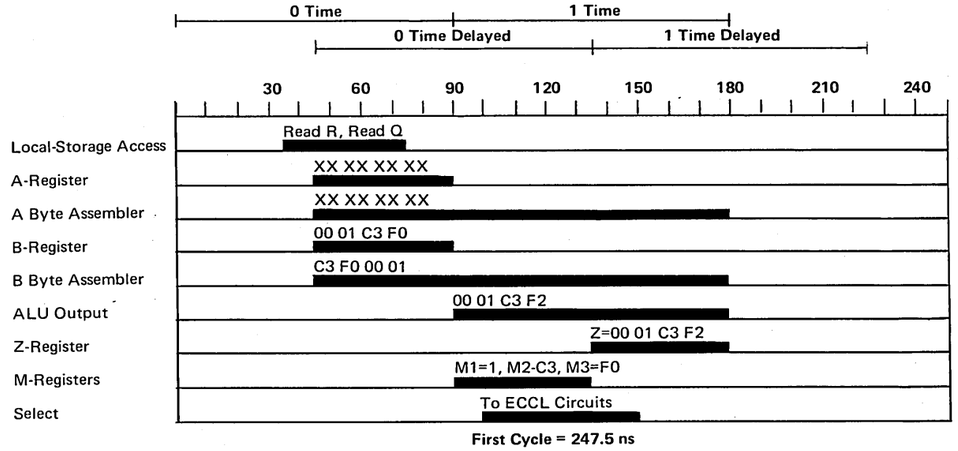
Gate bytes 2, 3 of the B-register to the B byte assembler and to the ALUs.

Perform the update (+2) on the low-order address bytes and gate the result to the Z-register.

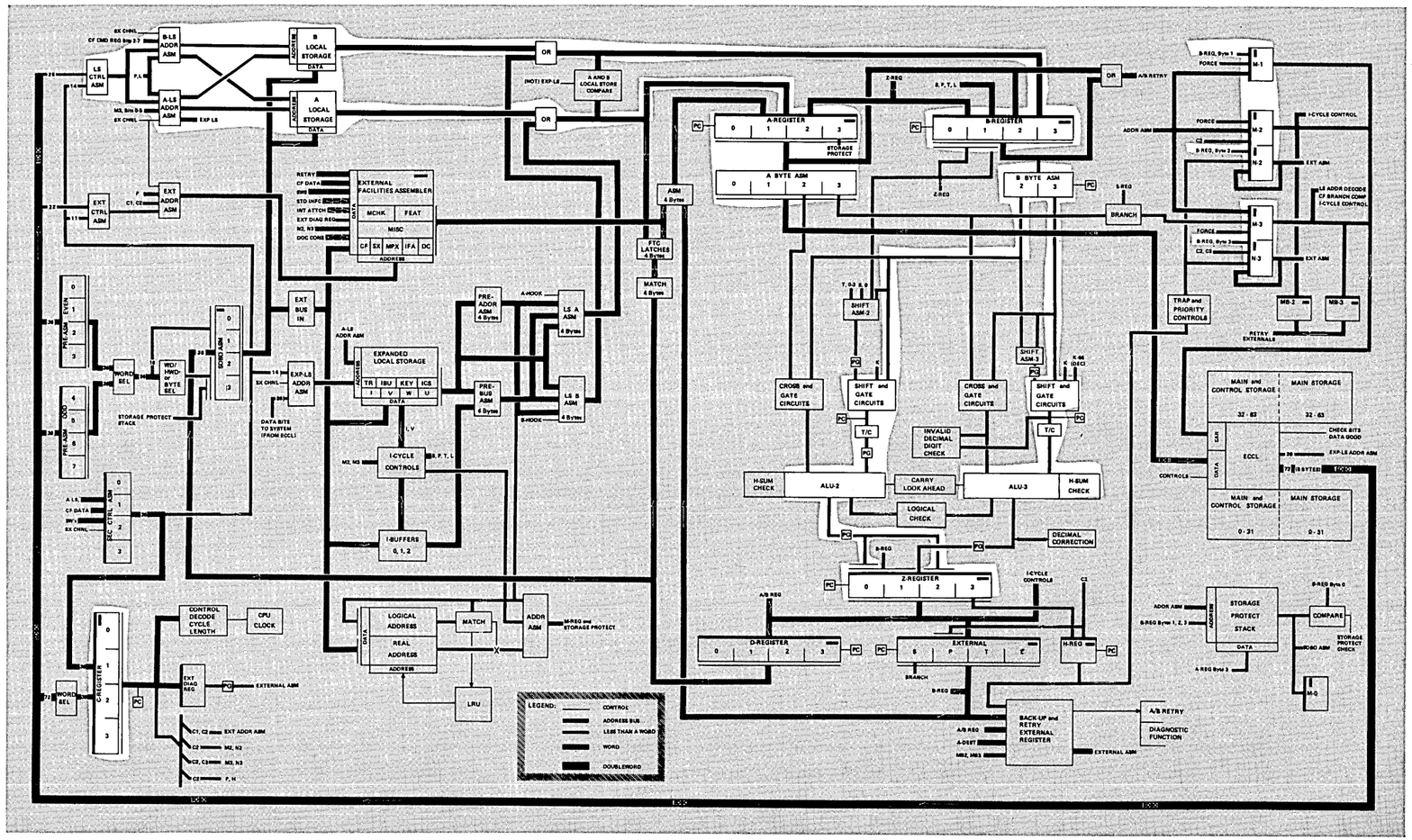
Gate bytes 0, 1 of the B-register to the B byte assembler and to the ALUs.

Perform an update to the high-order address bits if there was a carry from the update of the low-order address bytes.

Gate the result to bytes 0, 1 of the Z-register (done in the first part of the 2nd cycle).



First Cycle = 247.5 ns



Storage-Word Example, Read Halfword (Part 2 of 4)

Description SECOND CYCLE

Gate the Z-register contents (updated address) to the D-register.

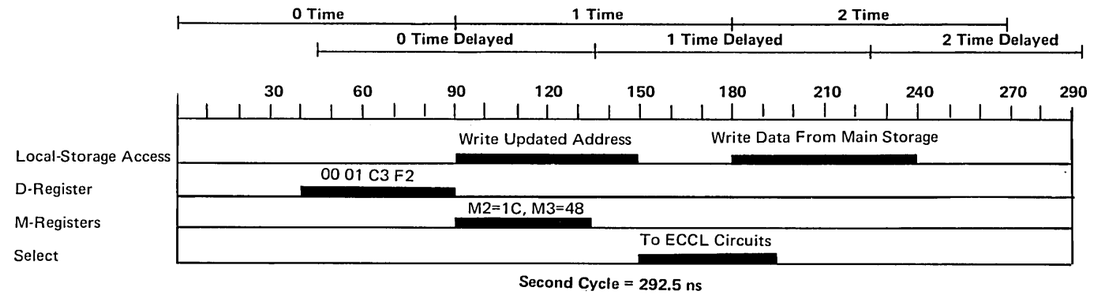
Gate the D-register contents through the SDBO assembler to the Q-register in both local storages.

Gate the data from main storage to the SDBO pre-assembler.

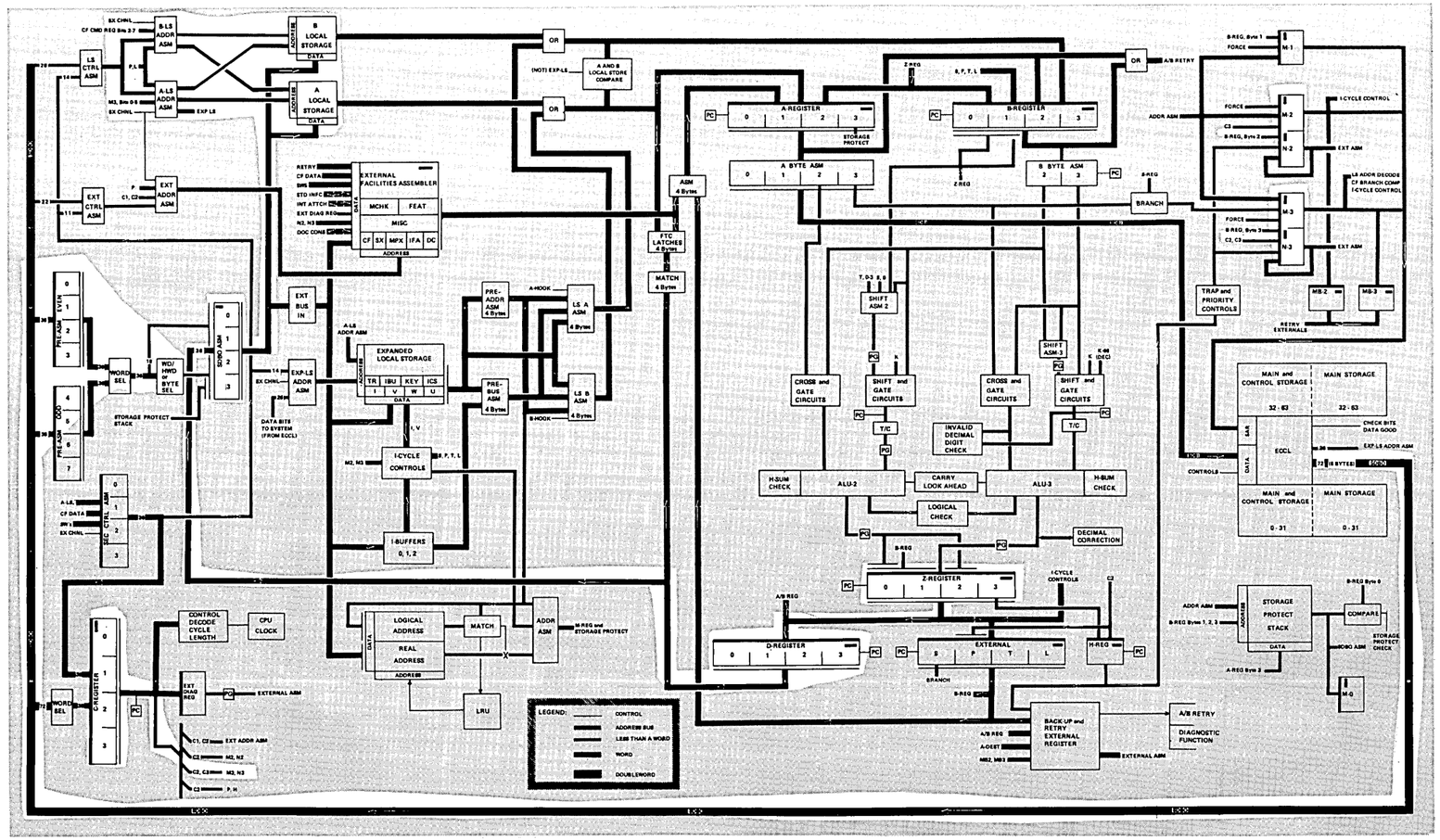
Perform the halfword selection and gate the halfword to bytes 2, 3 of the SDBO assembler.

Access both A and B local storage and store the halfword from the SDBO assembler into bytes 2, 3 of the R-register.

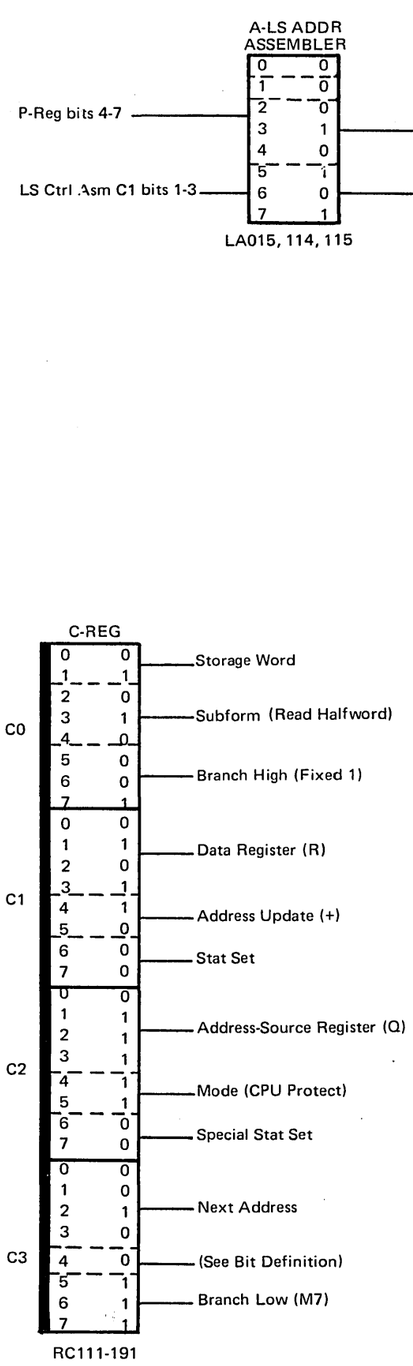
Set the M-registers with the next control-word address.

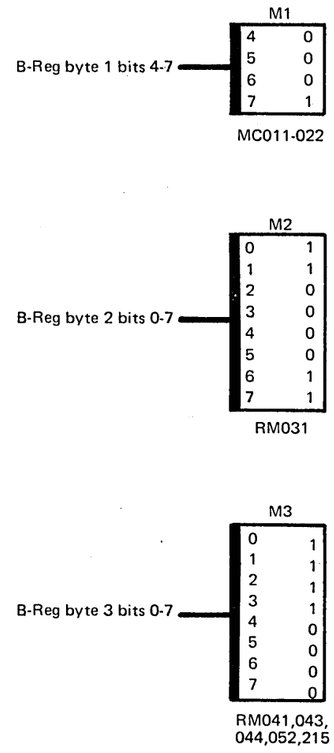


Second Cycle = 292.5 ns

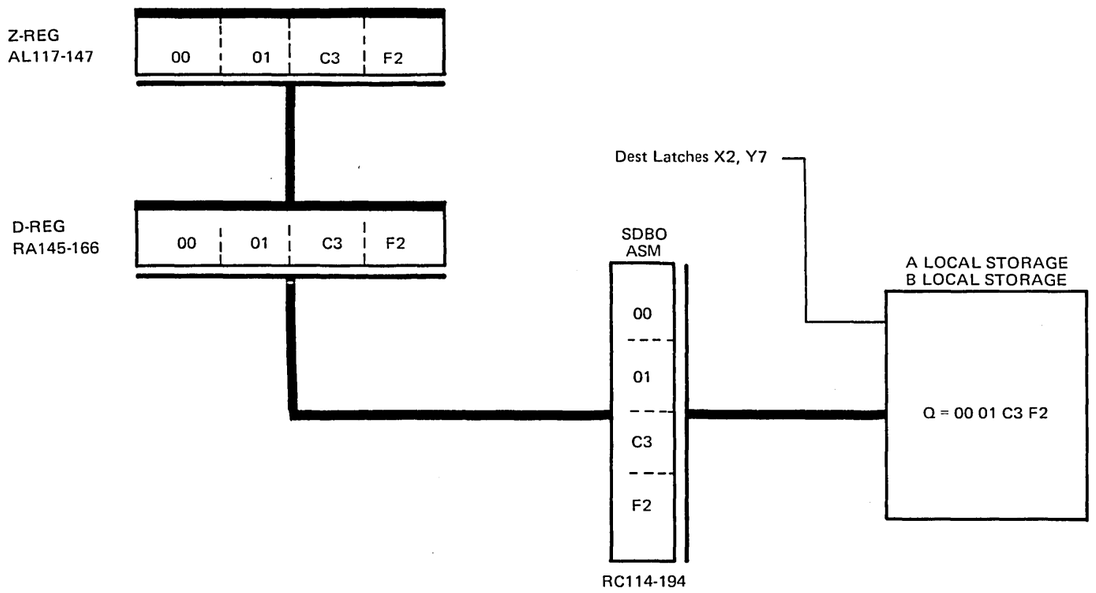


Storage-Word Example, Read Halfword (Part 3 of 4)

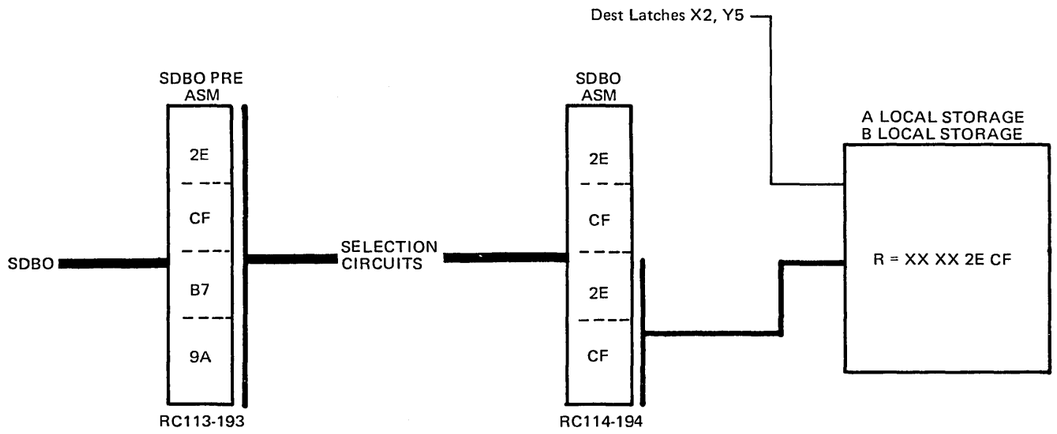


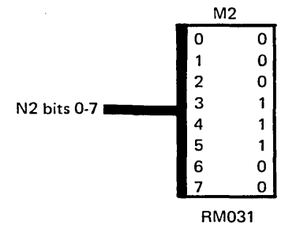


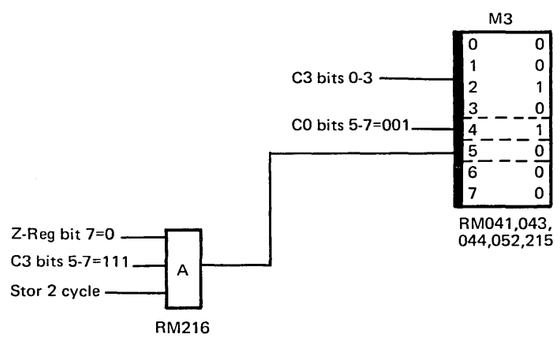
Storage-Word Example, Read Halfword (Part 4 of 4)



Second Half of Second Cycle







Storage-Word Example, TB Function (Part 1 of 4)

NEXT LABEL	STAT	STATEMENT
BOUND 2, M6, M7		**RDW R Q + 4, TB**

Starting Values

Q = 00 00 0C 92
P = 02
T = 83
R = XX XX XX XX
Address of BOUND 200 = DF20
Address of this word = 3E20
Contents of address 0 0C 90 = XX XX FC DE

Objectives

Read the word from main storage as specified by the address in the Q-register.

Set the word read from main storage into the R-register.

Update the address in the Q-register by +4.

Branch to the leg of the branch set BOUND, as specified by the low two bits of the address used to access main storage.

Set bits 6, 7 of the T-register with the status of the two low-order main-storage address bits.

Reset bits 0-3 of the T-register.

Description FIRST CYCLE

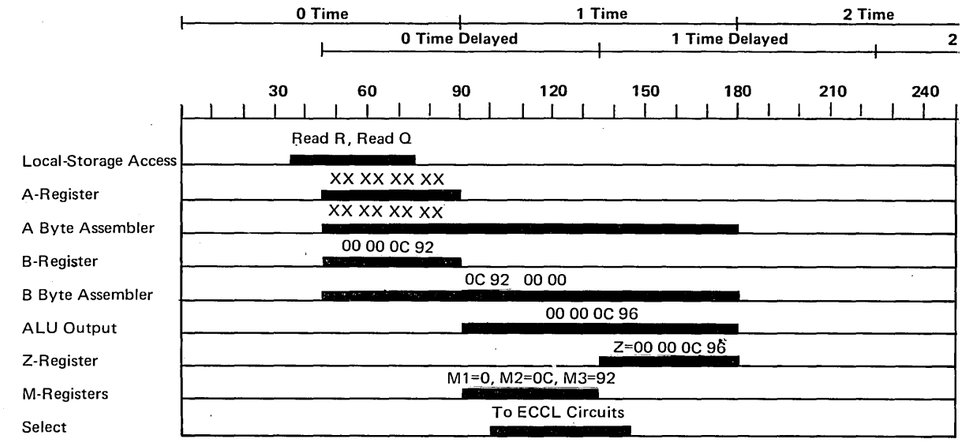
Read the contents of the Q-register from local storage B, and gate to the B-register.

Gate the low 20 bits of the B-register to the M-register. Send the select pulse to the storage circuits.

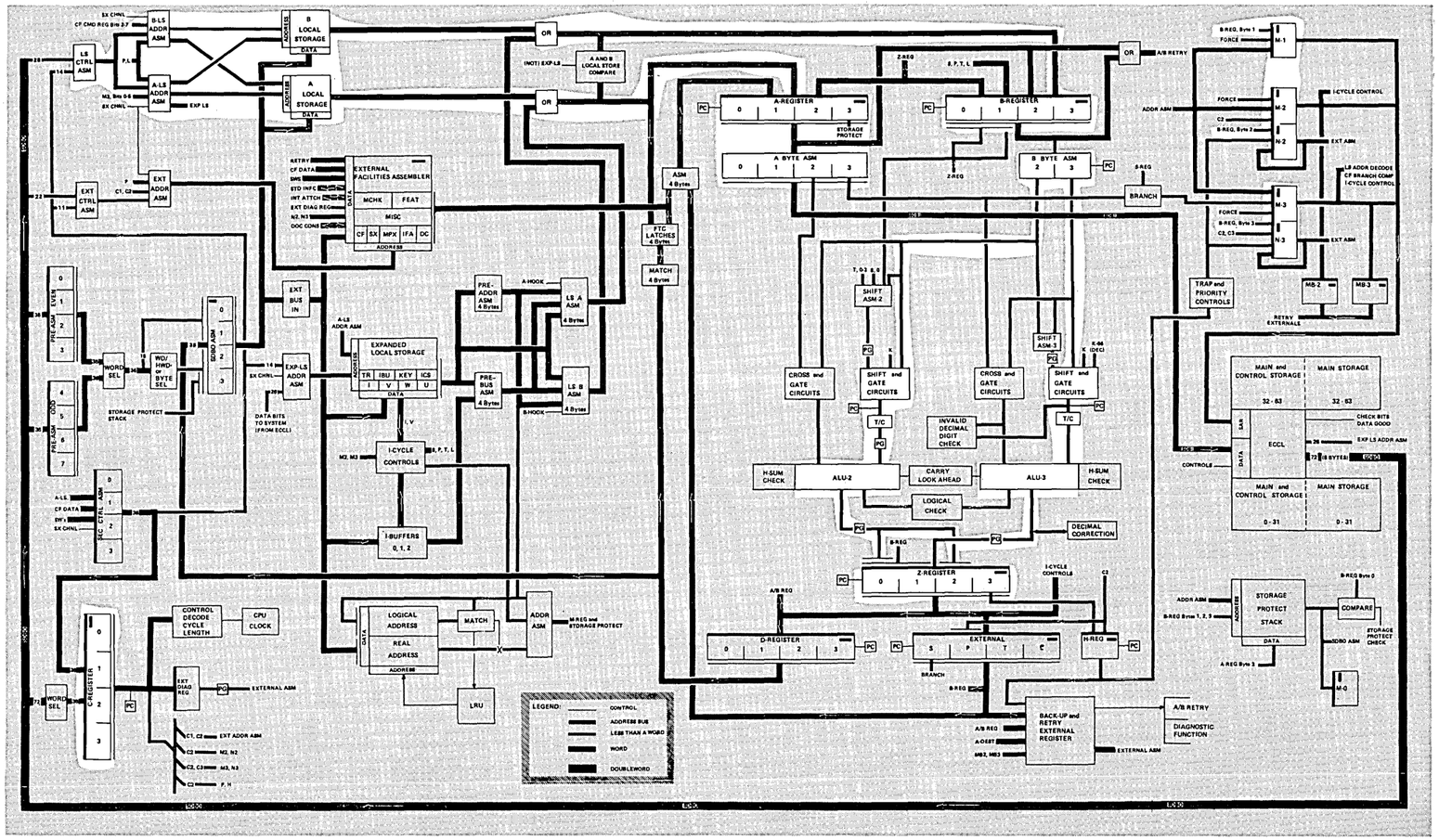
Read the contents of the R-register from local storage A, and gate to the A-register.

Perform the address update on the Q-register contents. Gate the result to the Z-register.

The high-order two bytes of the updated address are not gated to the Z-register until the early part of the second cycle.



First Cycle = 247.5 ns



Storage-Word Example, TB Function (Part 2 of 4)

Description SECOND CYCLE

Gate the Z-register contents (updated address) to the D-register.

Gate the D-register through the SDBO assembler to the Q-register in both local storages.

Gate the data from main storage through the SDBO pre-assembler to the SDBO assembler.

Gate the SDBO assembler contents to the R-register in both local storages.

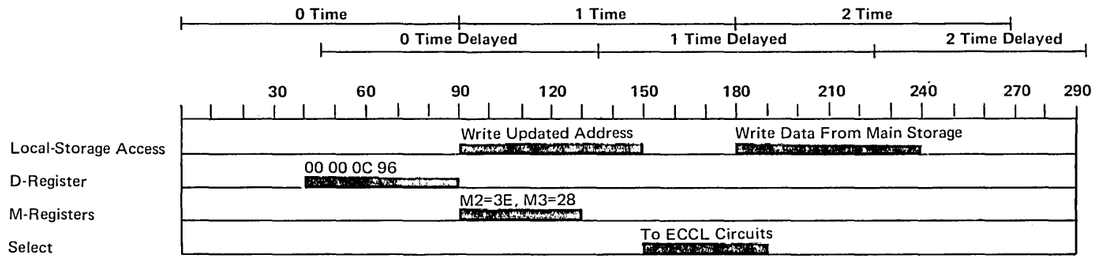
Set the M2-register from N2.

Set the M3-register with the next address bits and the result of the low address bits test.

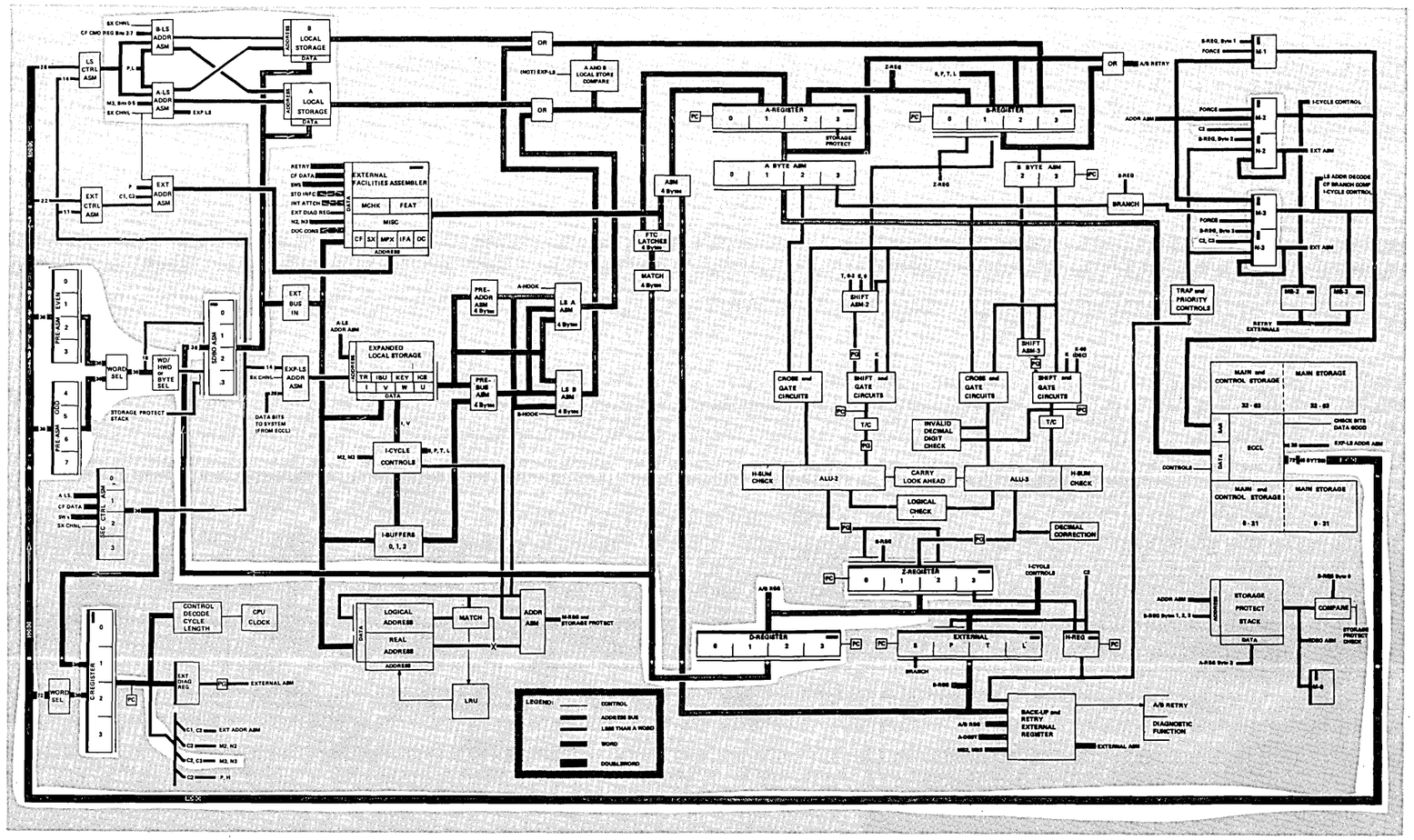
Set bits 6, 7 of the T-register with the two low-order address bits from MB3.

Reset bits 0-3 of the T-register.

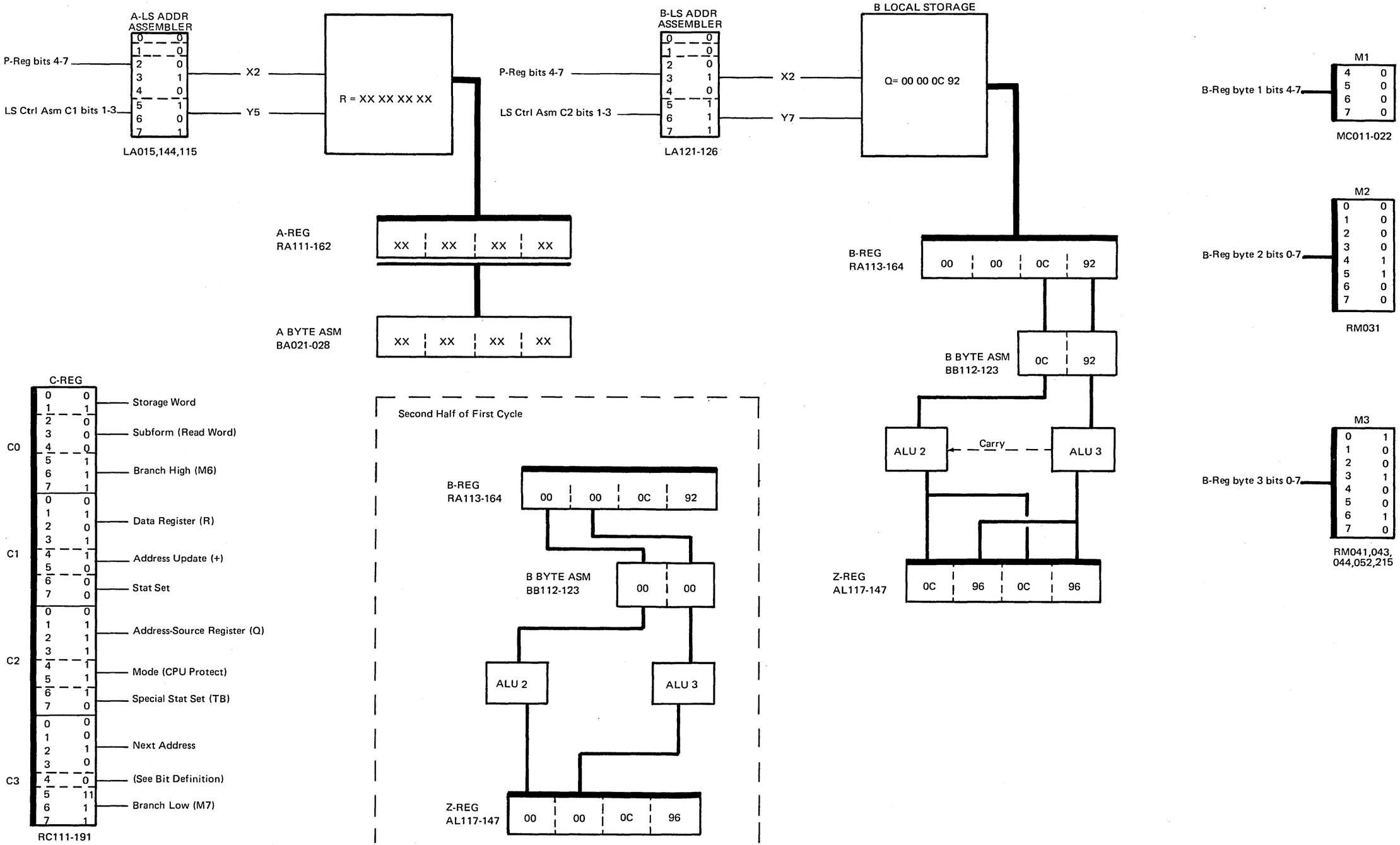
NOTE: The address that was used to access main storage was not on a word boundary. If the program wanted the fullword starting at the address 0 0C 92. another access of main storage will have to be made to get the 2 low order bytes of that word.



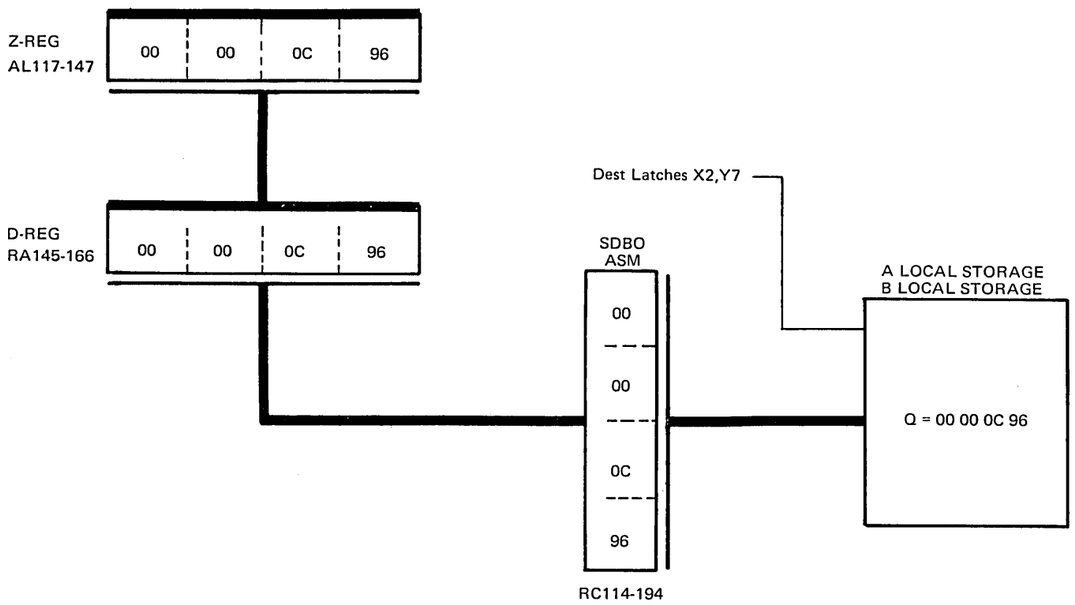
Second Cycle = 292.5 ns

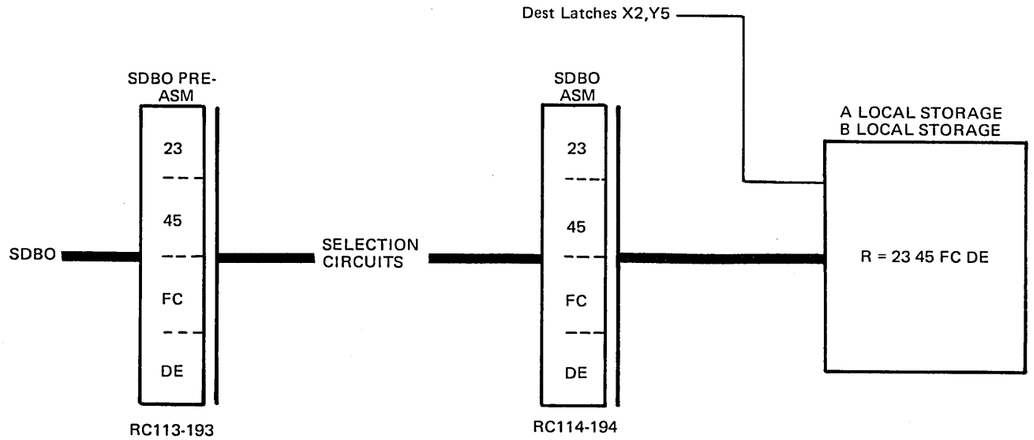


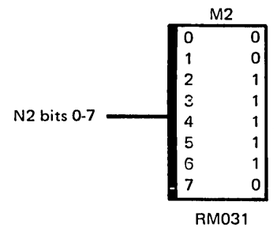
Storage-Word Example, TB Function (Part 3 of 4)

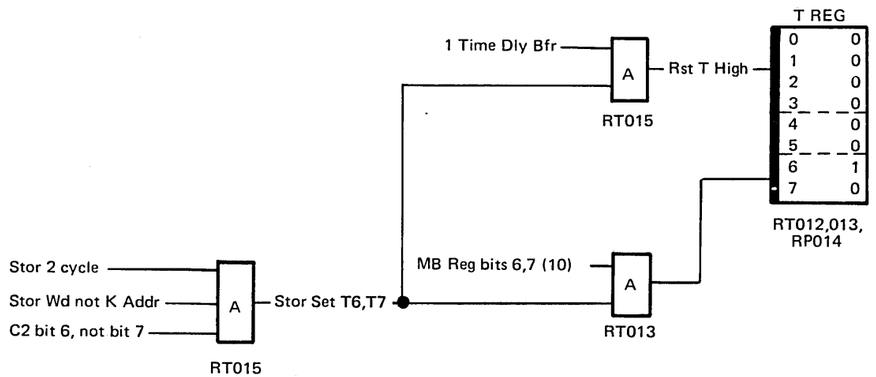


Storage-Word Example, TB Function (Part 4 of 4)









Storage-Word Example, Direct Control Storage (Part 1 of 4)

NEXT LABEL	STAT	STATEMENT
		RDW LH DC, 30

Starting Values

P = 02

L = 40

Contents of Control-Storage Address FF30 = 2C 9D F5 AB

Objectives

Read the word from control-storage address FF30.

Set the word read from control storage into the indirectly addressed local-storage register.

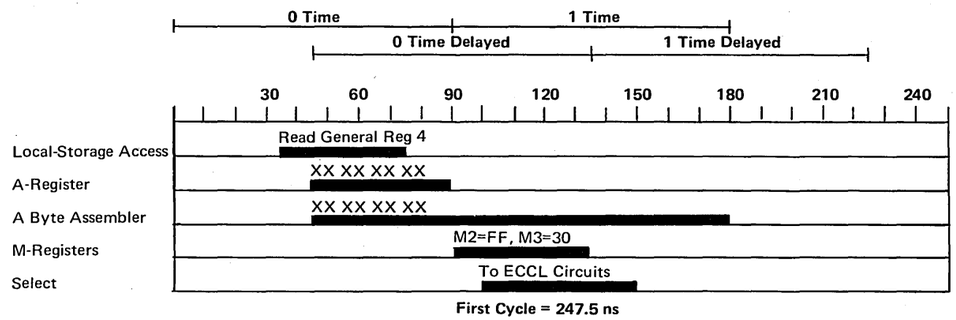
Set the next control-storage address into the M-register.

Description FIRST CYCLE

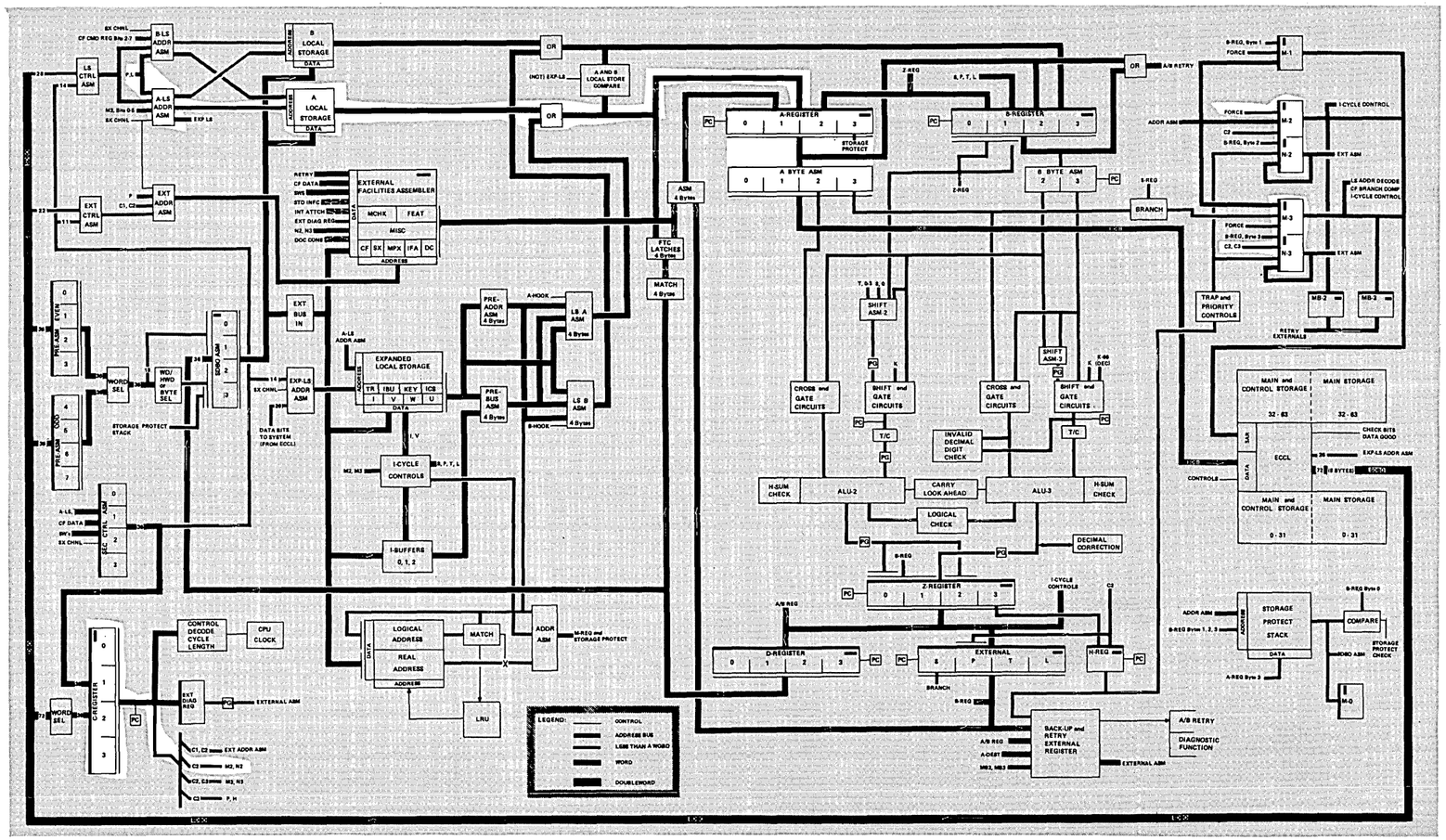
Access local storage A by setting up the indirect word address. Nothing is done with the data read out.

Set the M2-register with the force circuits.

Set M3 from C2 of the control word.



First Cycle = 247.5 ns



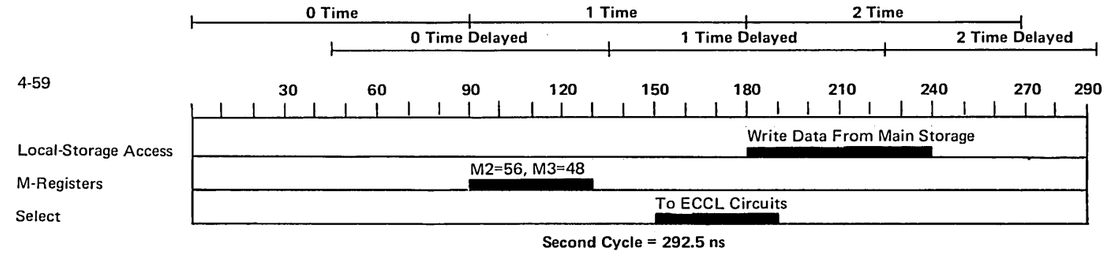
Storage-Word Example, Direct Control Storage (Part 2 of 4)

Description SECOND CYCLE

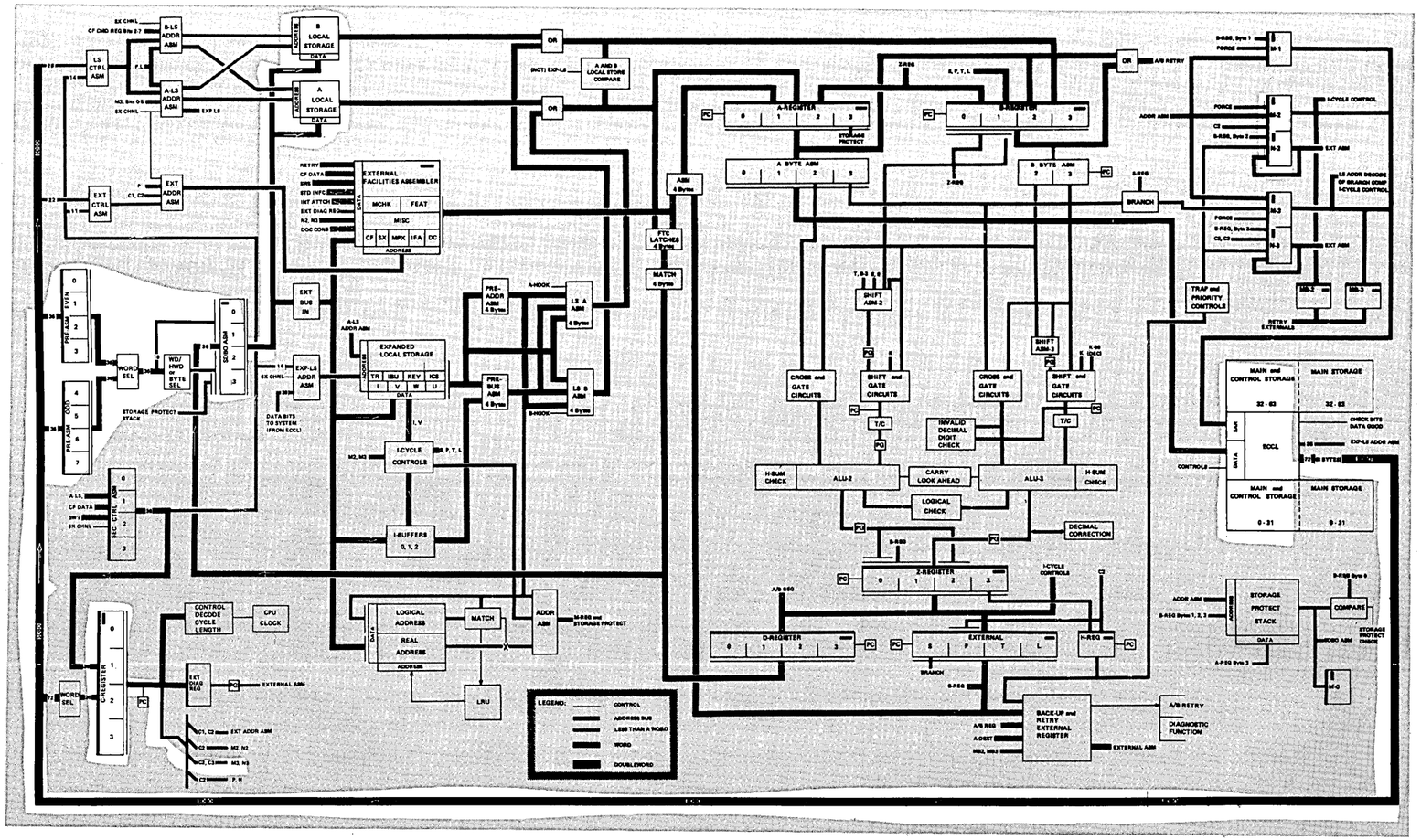
Gate the word read from control storage through the SDBO preassembler to the SDBO assembler.

Write the data word into the general-register location 4 of both local storages.

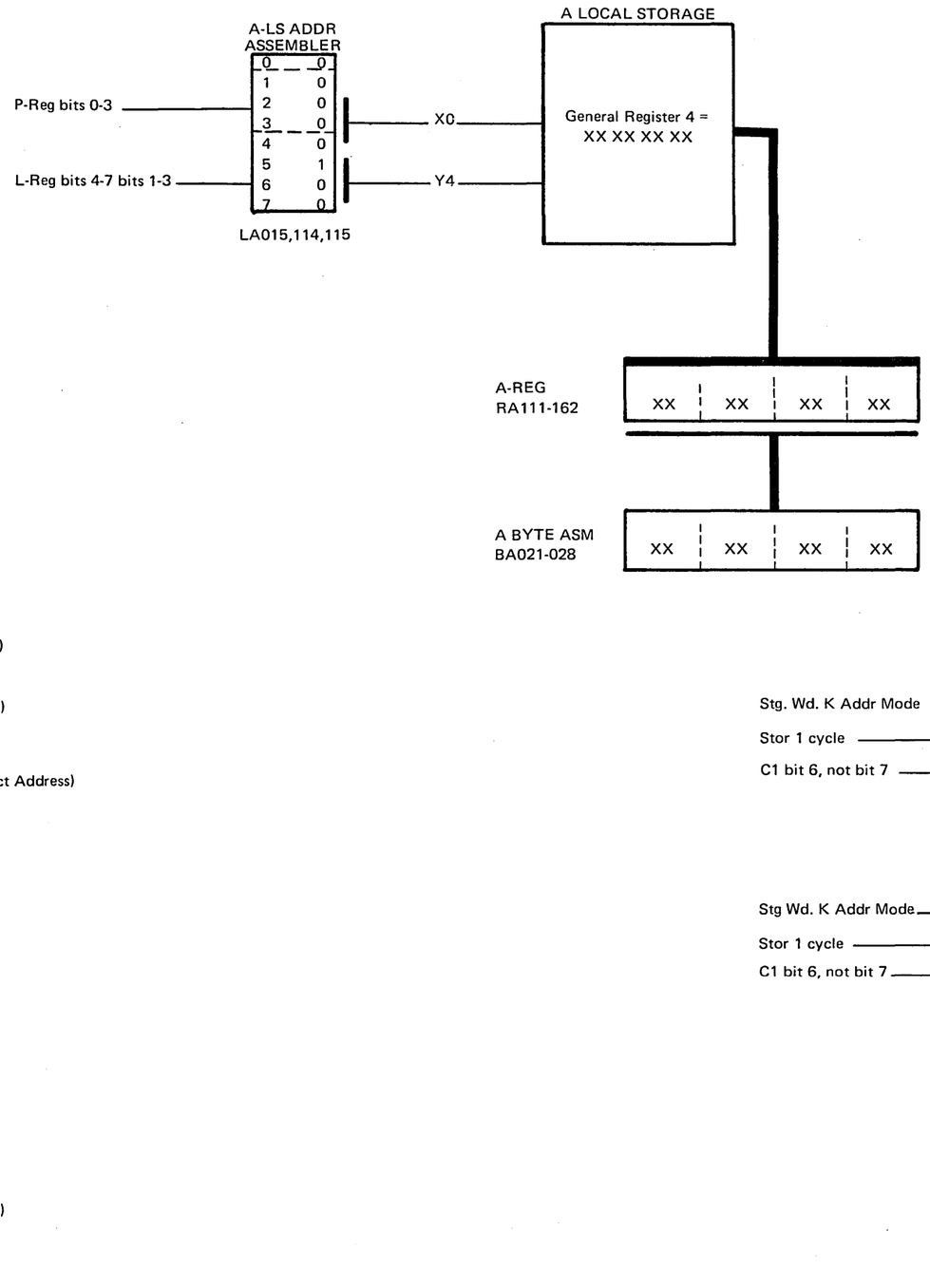
Set the M-register to the next control-word address.



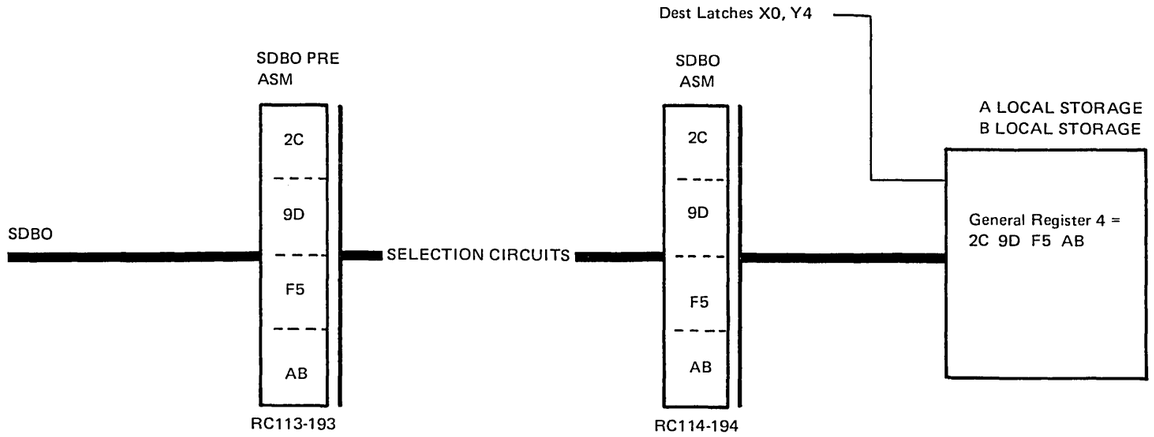
Second Cycle = 292.5 ns

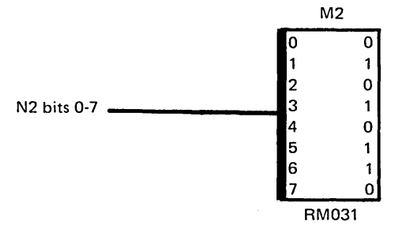


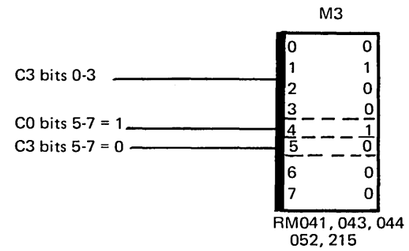
Storage-Word Example, Direct Control Storage (Part 3 of 4)



Storage-Word Example, Direct Control Storage (Part 4 of 4)







Storage-Word Example, Store Under Mask and Decrement Count

NEXT LABEL	STAT	STATEMENT
NUMZN S7	**Z6**	**STW R Y + TH,DCNT**

Starting Values

P = 02
S = 0C
T = 20
Y = 00 01 FD 000
Q = 3F CD 01 08
R = F7 F3 F2 C3
Address of NUMZN 0 = FCE0

Objectives

Store the contents of the R-register into main storage as addressed by the Y-register. Store the R-register under the mask bits provided by bits 0-3 of the T-register.

Update by 1 the address found in the Y-register (specified by the contents of bits 0-3 of the T-Register).

Decrement the count register, which is the next highest address in local storage (Q-register), by the same value as the address update.

Reset bits 0-3 of the T-register.

Set bit 4 of the S-register if bits 0-5 of the low-order byte of the updated count are 0. Set bit 5 of the S-register if bits 4-7 of the low-order byte of the updated count are 0.

Set the next control-word address in the M-register.

Description FIRST CYCLE

Read the data register (R) from the A local storage. Gate to the A-register, through the A byte assembler to the SDBI.

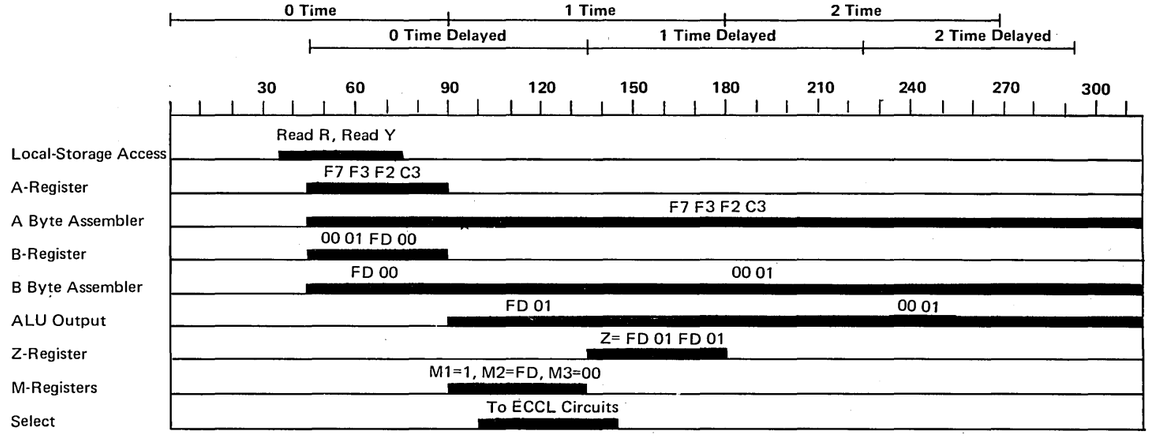
Read the address-source register (Y) from B local storage, and gate to the B-register.

Gate the low 20 bits of the B-register to the M-register, and gate the M-register to the SAR.

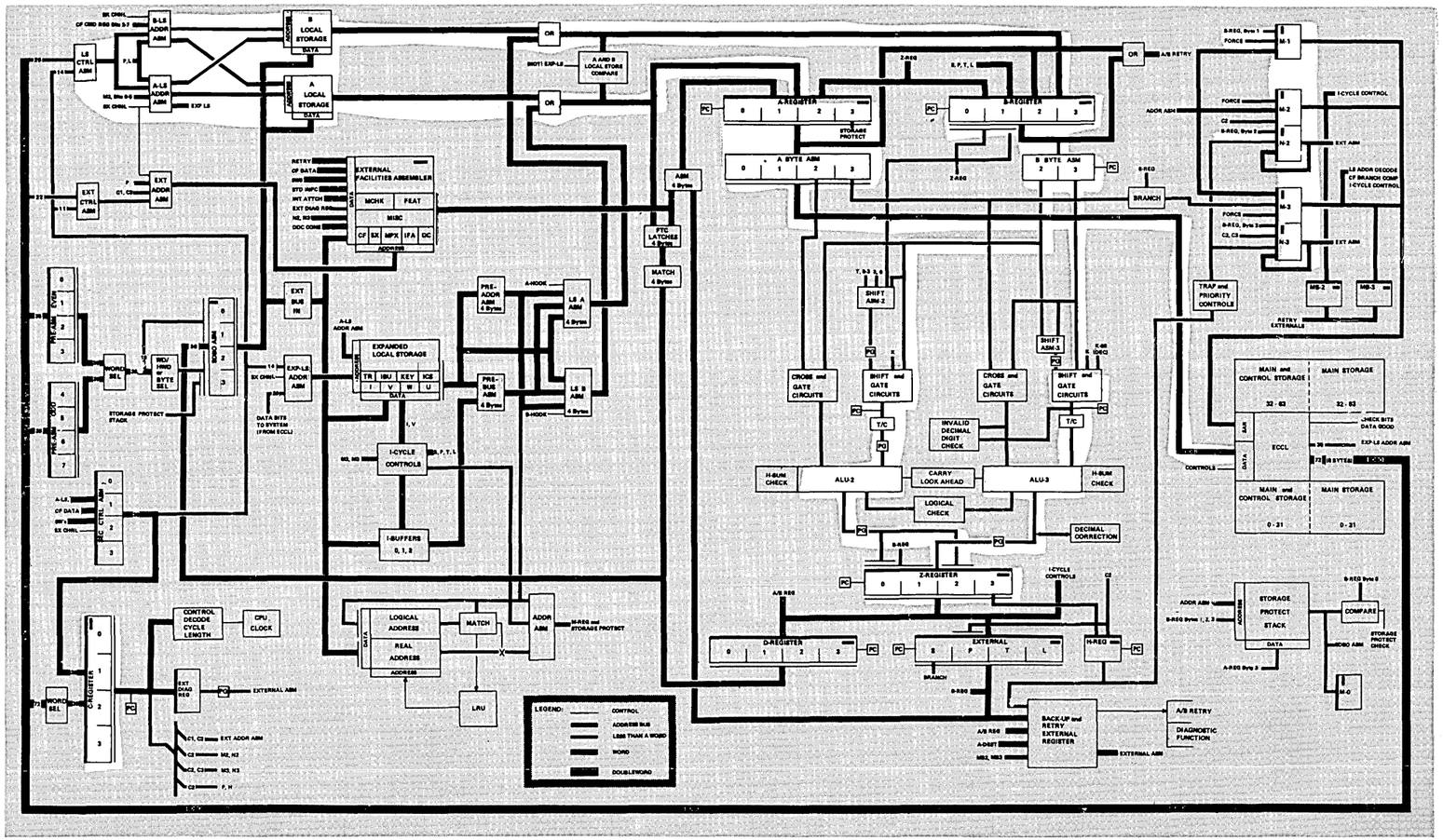
Gate bytes 2 and 3 of the B-register to its B byte assembler. Gate byte 2 to ALU2, and byte 3 to ALU3. Add in the K assembler input (1) to ALU3.

Gate the result to bytes 0 and 1, and 2 and 3, of the Z-register.

Gate bytes 0 and 1 of the B-register to the B byte assembler. Gate byte 0 to ALU2, and byte 1 to ALU3. Add in any carry from the previous ALU operation. Gate the result to bytes 0 and 1 of the Z-register.



First Cycle = 315 ns



Storage-Word Example, Store Under Mask and Decrement Count (Part 2 of 4)

Description SECOND CYCLE

Force the Y-line of the B destination latches odd. This addresses the count register (Q).

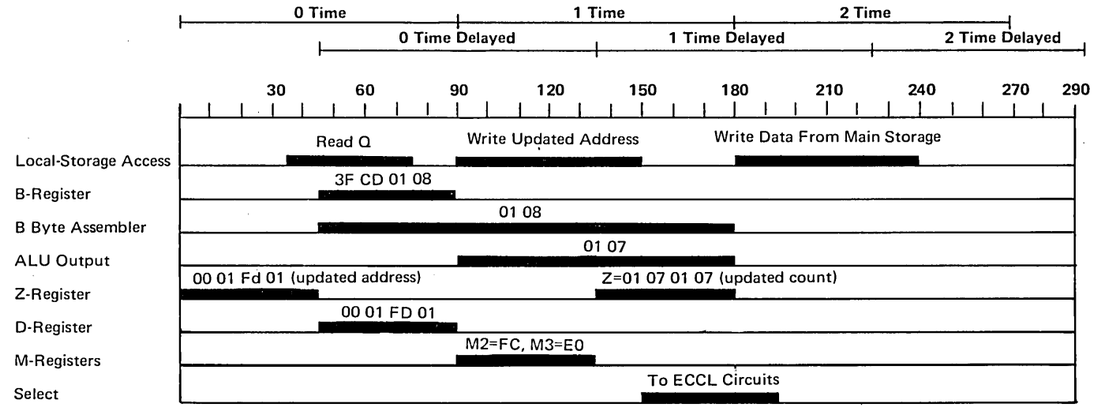
Read the Q-register from B local storage and gate to the B-register. Gate bytes 2 and 3 of the B-register to bytes 2 and 3 of the B byte assembler. Gate to the ALUs.

Gate the Z-register (updated address) to the D-register to the SDBO assembler.

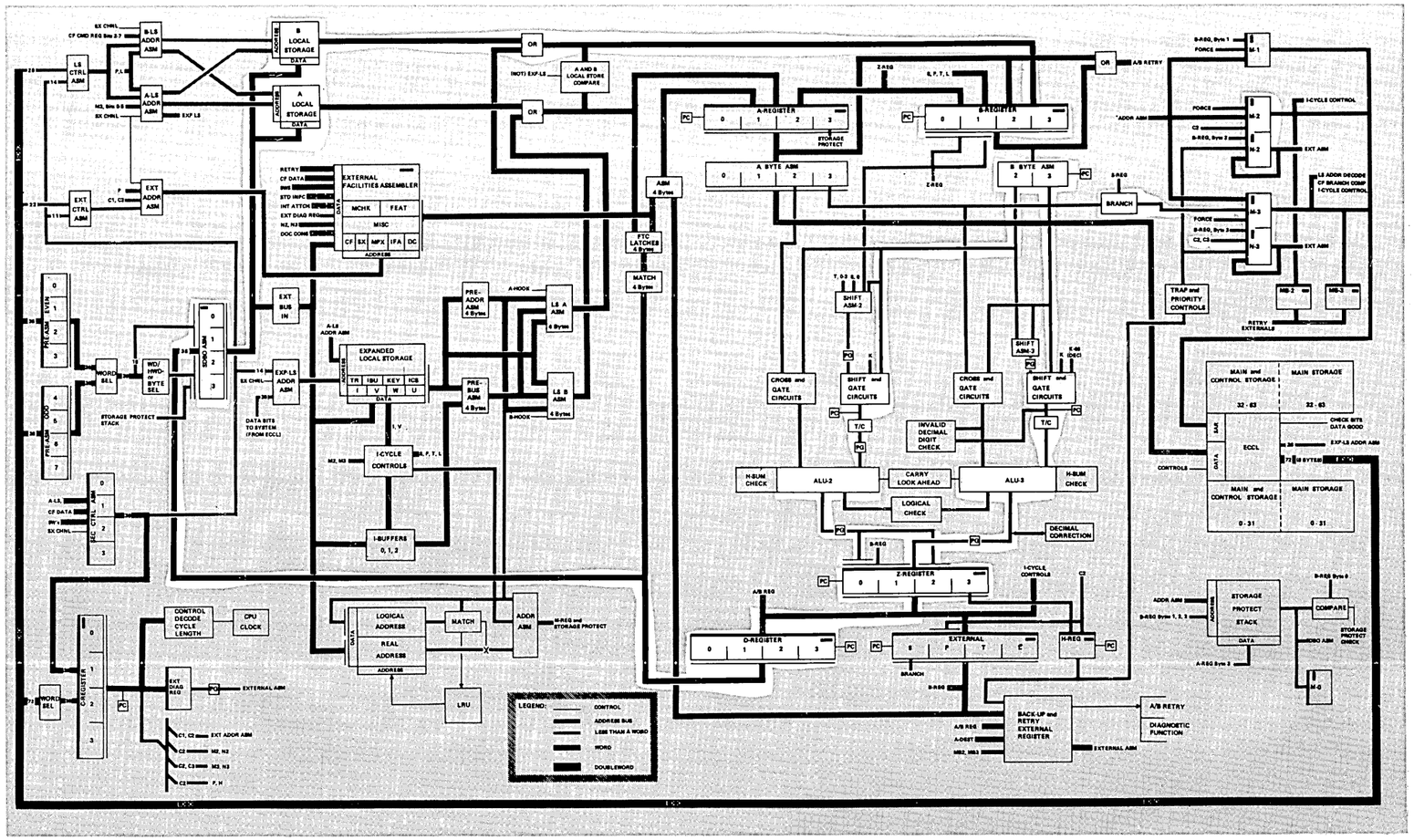
Gate the SDBO assembler to A and B local storage. Write the updated address into the R-register of both local storages.

Decrement the contents of the ALUs by the value of the address update (1). Gate the result to bytes 0 and 1, and 2 and 3, of the Z-register.

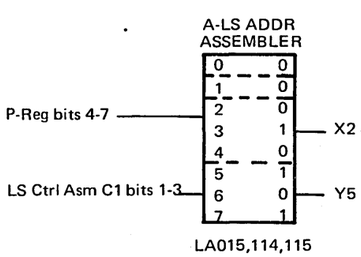
Set up the next control-word address in the M2 and M3 registers.

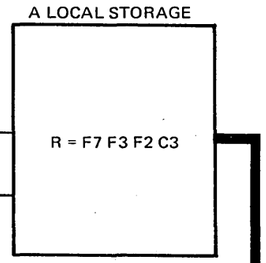


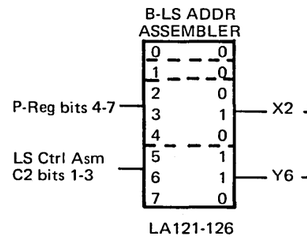
Second Cycle = 292.5 ns

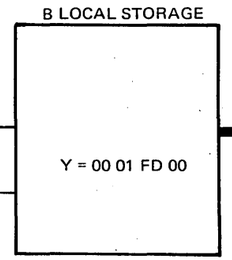


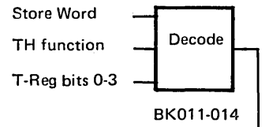
Storage-Word Example, Store Under Mask and Decrement Count (Part 3 of 4)

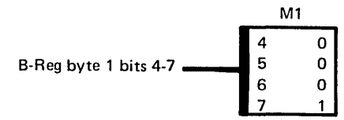


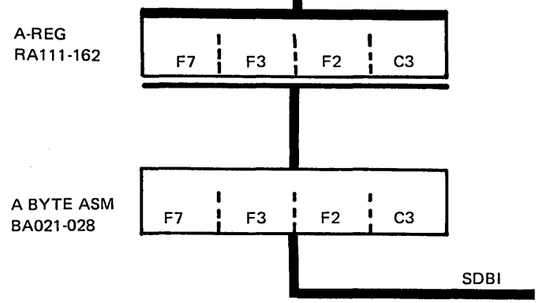


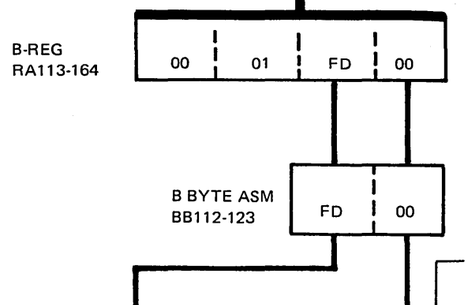


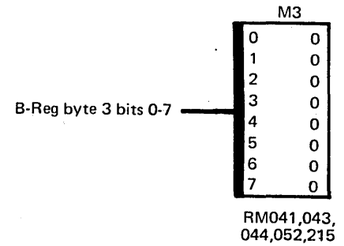


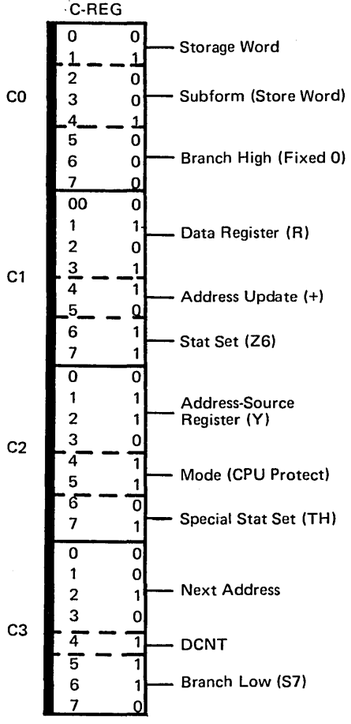


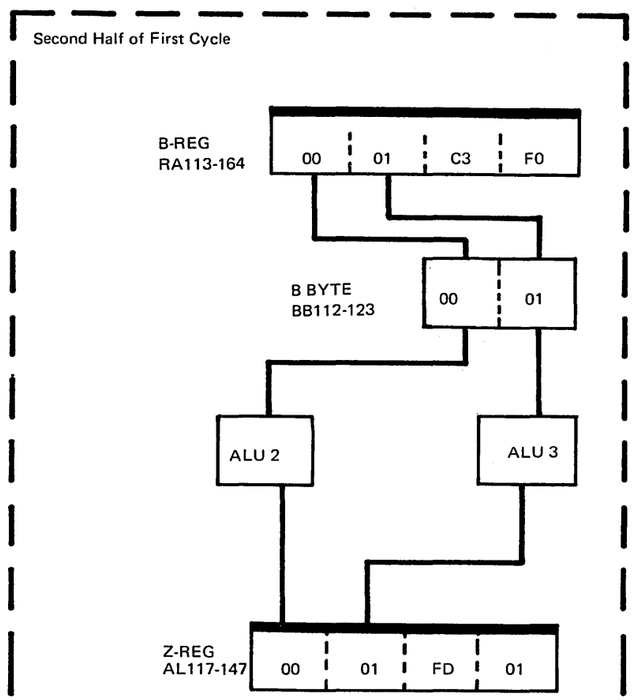


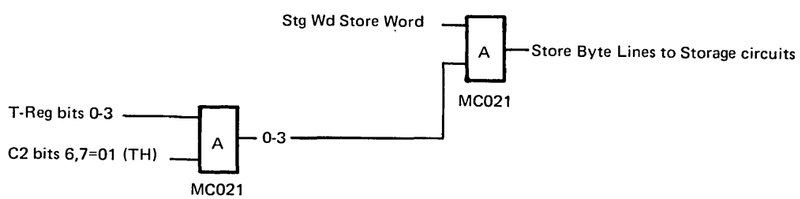




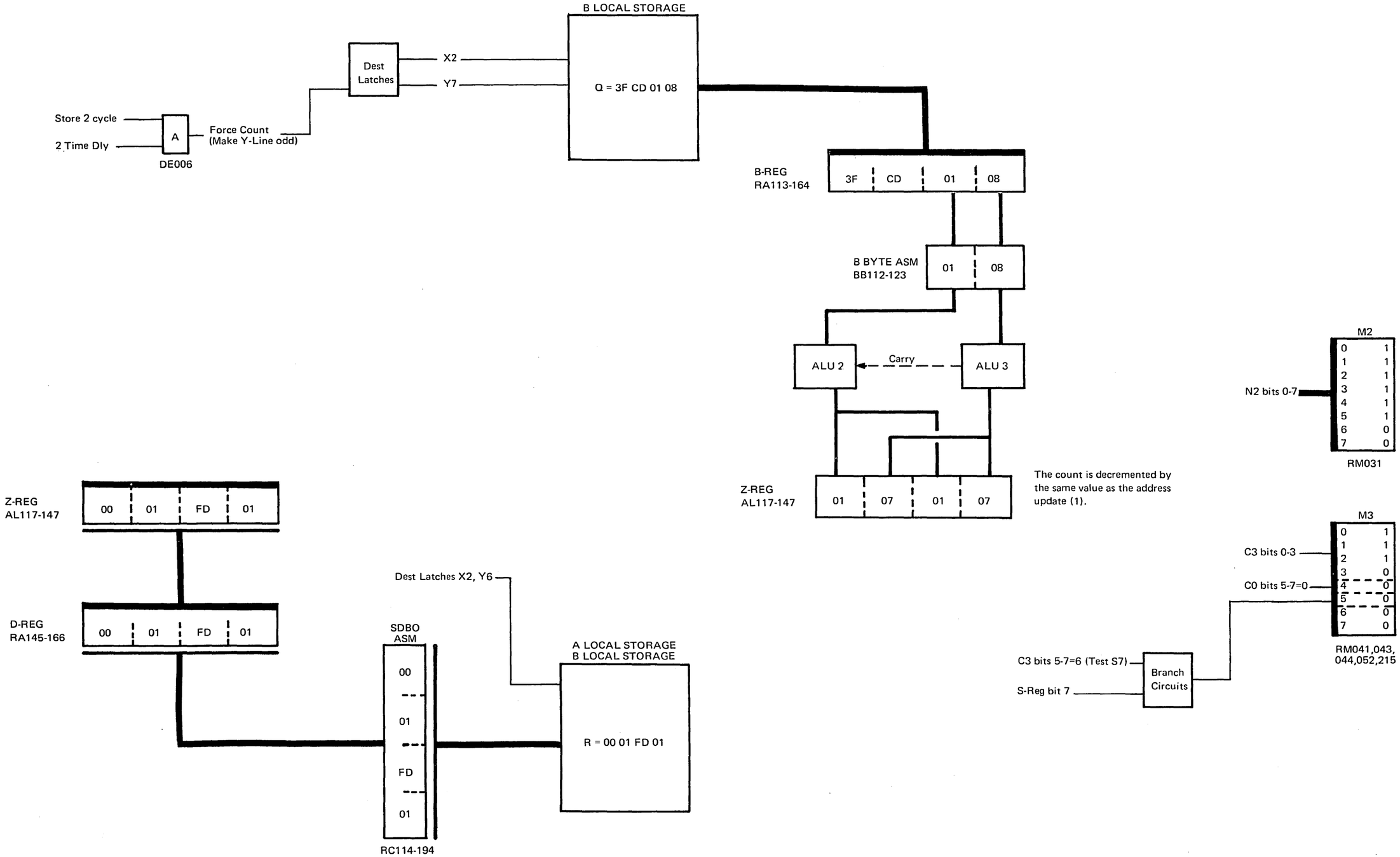








Storage-Word Example, Store Under Mask and Decrement Count (Part 4 of 4)



Arithmetic Word

- There are two types of arithmetic words:

  Type 10, recognized by a first hex digit of 8, 9, A, or B. Can specify a variety of arithmetic and logical functions, including fullword binary arithmetic and fullword shifting.

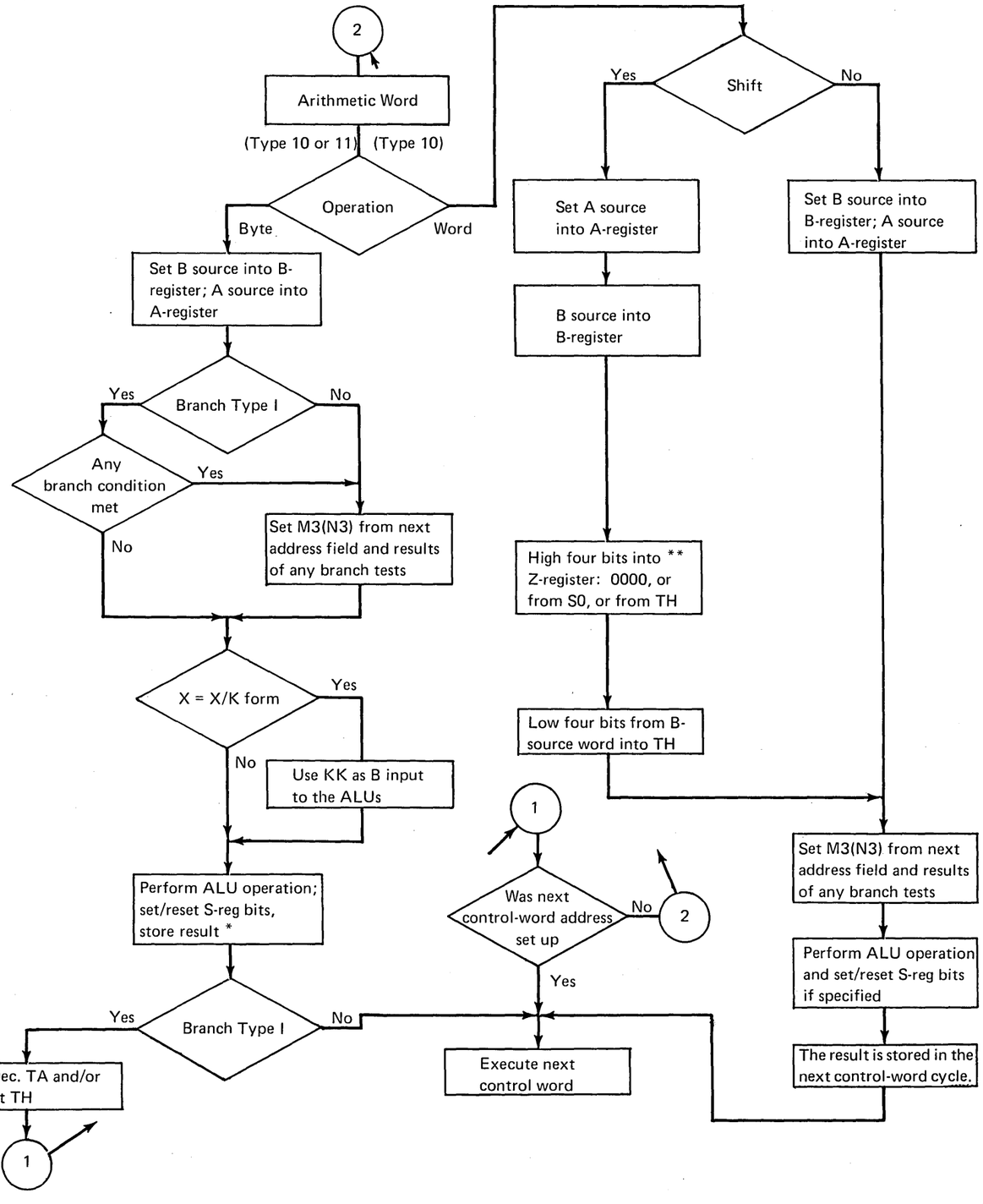
  Type 11, recognized by a first hex digit of C, D, E, or F. Can perform exclusive OR or true add functions on a byte basis only. Provides for crossing portions of the A-register inputs.

The arithmetic word operates on data from local storage or from certain externals. When an external is used, it is always the A source of the operation. The B source is either local-storage data or a hex value that is specified by the K-field of the word.

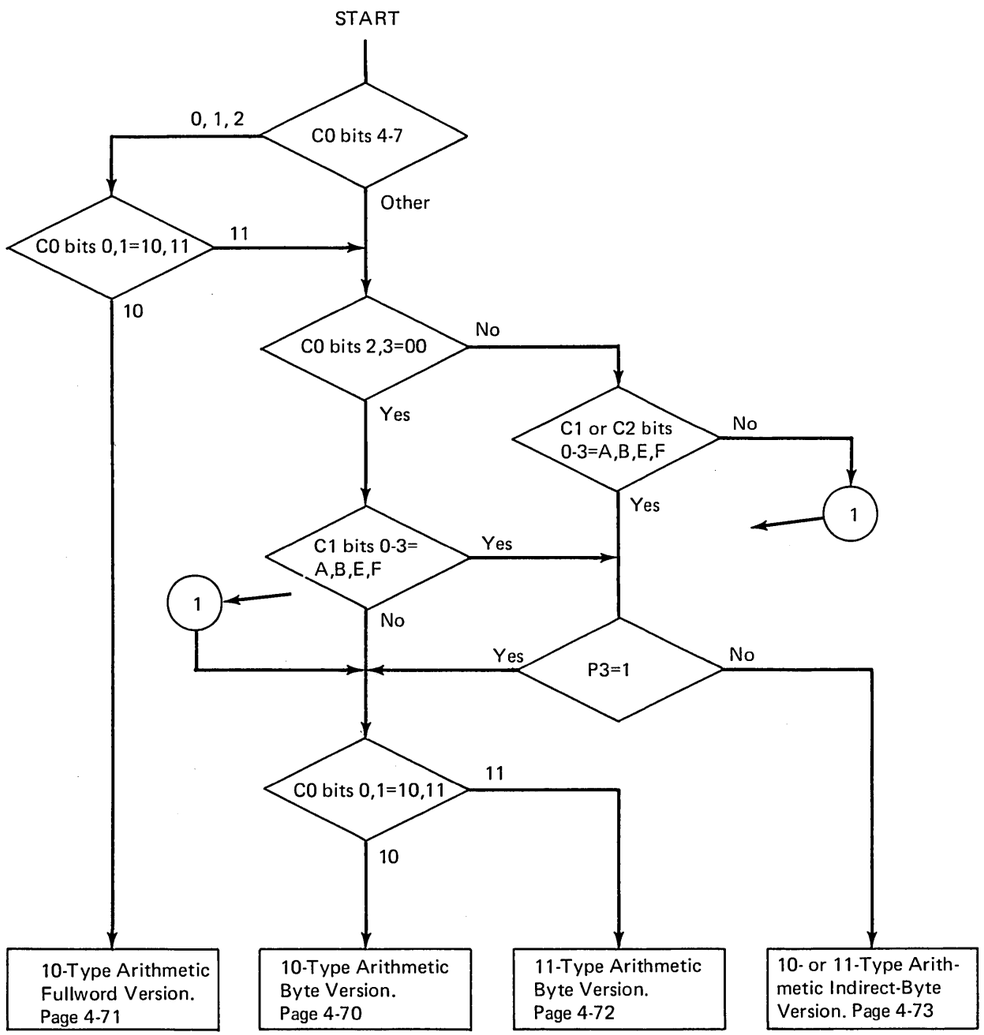
Both arithmetic-word types can utilize the indirect-byte addressing and branching facility. This facility is explained in the descriptions that follow.

There are four bit-breakdown charts used to describe the various functions of the two arithmetic-word types. The first chart indicates the fields and functions associated with the 10-type byte version with no indirect-byte function designated. The second chart represents the 10-type fullword capability with the optional-shift function. The third chart represents the 11-type byte version with no indirect-byte function designated. The fourth chart represents either the 10- or 11-type with indirect-byte addressing and branching used.

Arithmetic Word Execution



Arithmetic Word, Bit-Definition Diagram Selection



Bit Definition, Type-10 Byte Version

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		Form		Operation				A Source Word				A Source Byte		Stat Set		B Source Word				B Source Byte		A Hi-Lo		Next Address						Branch	
1	0	00=A=A/K 01=Z=A/B 10=A=A/B 11=B=A/B 00=Z=A/K		0011=,OE, 0100=+ 0101=+ (+1) 0110=C + (Rst S0) 0111=,A, 1000=,OR, 1001=C+ 1001=C + - (+C) 1010= C,D+-, (+C) 1011=B 1011=ABCK 1100= - 1101= - (+1) 1110= C - (+1) (Set S0) 1111=,A,				00=0 01=1 10=2 11=3		00=— 01=S12 10=S45 11=Z6				00=0 01=1 10=2 11=3		00=block 01=L 10=H 11=St						00=— 01=S2,S3 10=S4,S5 11=S6,S7	

C0 BITS 0, 1

This field specifies the arithmetic-word type 10.

C0 BITS 2, 3

The Form Field specifies what inputs are to be used and where the ALU result is to be destined. There are five forms provided.

C0 bits 2, 3=00 The form can be either A=A/K or Z=A/K depending on the operation field. The form A=A/K is used when the Operation Field is a value of 0000—1011. The form Z=A/K is used when the Operation Field is a value of 1100—1111.

C0 bits 2, 3=01 The Z=A/B form is designated.

C0 bits 2, 3=10 The A=A/B form is designated.

C0 bits 2, 3=11 The B=A/B form is designated.

In the A=A/K and A=A/B forms, the ALU result is set into byte specified by the A-Source Field (C1 bits 0-5).

In the B=A/B form, the ALU result is set into the byte specified by the B-Source Field (C2 bits 0-5).

In the Z=A/B and Z=A/K forms, the ALU result is not destined. The result is gated to the Z-register where the data can be tested to set certain S-register positions. The S-register can be set also by the other word forms.

C0 BITS 4-7

This is the Operation Field. Control of certain ALU inputs and ALU functions is determined by the value of this field. Setting or resetting of certain S bits is also controlled by this field.

C0 4-7=0011 Exclusive OR the A and B sources.

C0 4-7=0100 True binary add, no carry-in.

C0 4-7=0101 True binary add, carry-in of 1.

C0 4-7=0110 True binary add, set S bit 0 to 0, no carry-in, set S bit 3 to the value of the carry-out.

C0 4-7=0111 AND the A and B sources.

C0 4-7=1000 OR the A and B sources.

C0 4-7=1001 Binary add, true add if S0=0, complement add if S0=1, add in the status of S3, set S3 with the value of the carry-out.

C0 4-7=1010 Decimal add, true add if S0=0, complement add if S0=1, add in the status of S3, set S3 with the value of the carry-out.

C0 4-7=1011 Exclusive OR. If a parity error is detected on the A input to the ALUs, S4 is set to 1. S4 is not changed if no parity is detected.

C0 4-7=1100 Complement binary add, no carry-in.

C0 4-7=1101 Complement binary add, carry-in of 1.

C0 4-7=1110 Complement binary add, set S0 to 1, carry-in of 1, S3 set to the value of the carry-out.

C0 4-7=1111 Complement AND, the B input is complemented before ANDing with the A input.

C1 BITS 0-5

This field specifies the local storage or external byte that is to be the A input to the ALU.

C1 BITS 6, 7

C1 Bits 6, 7=00 No status set.

C1 Bits 6, 7=01 In binary operations, S1 is set to the value of the carry-out of bit 1 of the result. S2 is set to 1 if the ALU result is no zero. S2 is unchanged if the result is zero.

In decimal operations, S1 is set to 1 if an invalid invalid decimal digit is detected on either the A or B inputs. S1 is not changed if inputs are valid. S2 is set to 1 if the ALU result is not zero. S2 is not changed if the ALU result is zero.

C1 BITS 6, 7=10 S4 set to 1 if bits 0-3 of the ALU result are all zero.

S4 set to 0 if bits 0-3 of the ALU result are not all zero.

S5 set to 1 if bits 4-7 of the ALU result are all zero.

S5 set to 0 if bits 4-7 of the ALU result are not all zero.

C1 BITS 6, 7=11 S4 set to 1 if bits 0-5 of the ALU result are all zero.

S4 set to 0 if bits 0-5 of the ALU result are not all zero.

S5 set to 1 if bits 4-7 of the ALU result are all zero.

S5 set to 0 if bits 4-7 of the ALU result are not all zero.

Note: If S bit branching is specified by the control word, the branch testing is done before the S bits are modified by the operation.

C2 BITS 0-5

This field specifies the local-storage byte that is to be the B input to the ALU.

C2 BITS 6, 7

This field specifies the type of gating for the A input to ALU.

C2 bits 6, 7=00 Present 0 as the A input.

C2 bits 6, 7=01 Gate the low four bits of the A source only. Gate zeros as the high four bits.

C2 bits 6, 7=10 Gate the high four bits of the A source only. Gate zeros as the low four bits.

C2 bits 6, 7=11 Gate all eight bits of the A-source to the ALU.

C2 BITS 0-7

For the forms A=A/K or Z=A/K, this field represents the value to be used as the B input to the ALU.

C3 BITS 0-5

This field is part of the next control-word address that is gated to the M3-register when the next address formation takes place. C3 bits 0-5 are gated to bits 0-5 of M3. If branching is specified by bits 6, 7 of C3, the status of the bits tested is ORed with bits 4 and 5 of C3 as the next address is set up.

C3 BITS 6, 7

This field specifies the type of branch testing to be done to set up part of the next control-word address.

C3 bits 6, 7=00 No branch testing.

C3 bits 6, 7=01 OR the value of bits 2 and 3 of the S-register with bits 4 and 5 of C3 and set this result in bits 4 and 5 of M3.

C3 bits 6, 7=10 OR the value of bits 4 and 5 of the S-register with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

C3 bits 6, 7=11 OR the value of bits 6 and 7 of the S-register with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

Bit-Definiton, Type-10 Fullword Version

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		Form		Operation				A Source				Stat Set				B Source				Shift				Next Address						Branch	
								Word				A Input				Word				B Input											
1	0	00A=A/K 01=Z=A/B 10=A=A/B 11=B=A/B		0000= C + (Rst S0) 0001= C - (+1) (Set S0) 0010= C +- (+C)								00=block 10=16 bits 10=24 bits 11=32 bits				00=--- 01=S12 10=124 11=Z24								00=4 bits 01=8 bits 10=12 bits 11=32 bits				00=--- 01=(SR4,0) 10=(SR4,S0) 11=(SR4,TH)								00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

C0 BITS 0, 1

This field specifies the arithmetic-word type 10.

C0 BITS 2, 3

The Form Field specifies what inputs are to be used, and where the ALU result is to be gated. There are four forms provided for the fullword version.

C0 BITS 4-7

There are only three operations provided with the fullword version.

C0 bits 4-7=0000 True binary add, set S0 to 0, no carry-in, set S3 with the status of the carry-out of the high-order bit position.

C0 bits 4-7=0001 Complement binary add, set S0 to 1, carry-in of 1, set S3 with the status of the carry-out of the high-order bit position.

C0 bits 4-7=0010 Binary add, true if S0=0, complement if S0=1, add in value of S3, set S3 with the status of the carry-out of the high-order bit position.

C1 BITS 0-3

This field specifies the local-storage or external word that is to be the A input to the ALU.

C1 BITS 4, 5

This field specifies what portion of the A-source word is to be gated to the ALU.

C1 bits 4, 5=00 Gate all zeros to ALU.

C1 bits 4, 5=01 Gate the low 16 bits of the word addressed by the A-source field to the ALU.

C1 bits 4, 5=10 Gate the low 24 bits of the word addressed by the A-source field to the ALU.

C1 bits 4, 5=11 Gate the full A-source word to the ALU.

C1 BITS 6, 7

This field specifies the status-set functions.

C1 bits 6, 7=00 No status set.

C1 bits 6, 7=01 S1 is set to the value of the carry-out of bit position 1 of byte 0 of the result. S2 is set to 1 if the entire result is not zero. S2 is not changed if the result is zero. S3 is set to the value of the carry-out of bit 0 of byte 0 of the result.

C1 bits 6, 7=10 Only the low-order 24 bits of the result are stored. No S bits are changed, even if specified.

C1 bits 6, 7=11 S2 is set to 1 if the low 24 bits of the result are not all zero. S2 is not changed if the result is zero. S3 is set to the value of the carry-out of bit 0 of byte 1 of the ALU result. Only the low 24 bits of the result are stored. If the R4, S0 shift is specified, S0 is not altered even if specified by the Operation Field.

C2 BITS 0-3

This field specifies the local-storage word that is to be the B input to the ALU.

C2 BITS 4, 5

This field specifies what portion of the B-source word is to be gated to the ALU.

C2 bits 4, 5=00 Gate the low 4 bits of the B-source word to the ALU.

C2 bits 4, 5=01 Gate the low 8 bits of the B-source word to the ALU.

C2 bits 4, 5=10 Gate the low 12 bits of the B-source word to the ALU.

C2 bits 4, 5=11 Gate the entire B-source word to the ALU.

C2 BITS 6, 7

This field specifies the shifting function. Shifting is performed at the B inputs to the ALUs. The result can be a fullword or the low 24 bits of the shifted result. The low-order four bits of the original B-source word are always set into bits 0-3 of the T-register.

C2 bits 6, 7=00 No shifting.

C2 bits 6, 7=01 Shift right 4 bit positions, the low 4 bits are set into bits 0-3 of T, the high 4 bits of the result are set to zero.

C2 bits 6, 7=10 Shift right 4 bit positions, the low 4 bits are set into bits 0-3 of T, the high 4 bits of the result are set according to the status of S0; if S0=0, set the high 4 bits to 0; if S0=1, set the high 4 bits to 1s.

C2 bits 6, 7=11 Shift right 4 bit positions, the low 4 bits are set into bits 0-3 of T, the high 4 bits are set with the original value of bits 0-3 of the T-register.

Note: Shift-in bits from S0 or bits 0-3 of the T-register are not allowed if the status Set Field = 10 or 11. In this case the high 4 bits are set to zero.

C2 BITS 0-7

For the forms A=A/K or Z=A/K, this field represents the value to be used as the B input to the ALU.

C3 BITS 0-5

This field is part of the next control-word address that is gated to the M3-register when the next address formation takes place. C3 bits 0-5 are gated to bits 0-5 of M3. If branching is specified by bits 6, 7 of C3, the status of the bits tested is ORed with the bits 4 and 5 of C3 as the next address is set up.

C3 BITS 6, 7

This field specifies the kind of branch testing to be done to set up part of the next control-word address.

C3 bits 6, 7=00 No branch testing.

C3 bits 6, 7=01 OR the value of S bits 2 and 3 with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

C3 bits 6, 7=10 OR the value of S bits 4 and 5 with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

C3 bits 6, 7=11 OR the value of S bits 6 and 7 with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

Bit Definition, Type 11

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith Op Form		Form		Op	A Gating			A Source						Stat Set		B Source				B Hi — Lo				Next Address						Branch	
								Word				Byte				Word				Byte											
1	1	00=A=A/K 01=L=A/B 10=A=A/B 11=B=A/B		0=,OE, 1=C+ (Rst S0)	000=block 001=L 010=H011=St 100=X block 101=XL 110=XH 111=X							00=0 01=1 10=2 11=3		00=--- 01=S12 10=S45 11=Z6						00=0 01=1 10=2 11=3		00=block 01=L 10=H 11=St								00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

C0 BITS 0, 1

This field specifies the arithmetic-word Type 11.

C0 BITS 2, 3

The Form Field specifies what inputs are to be used, and where the ALU result is to be destined. There are four forms available with the Type 11 arithmetic word.

In the A=A/K and A=A/B forms, the ALU result is set into the byte specified by the A-source field (C1 bits 0-5).

In the B=A/B form, the ALU result is set into the byte specified by the B-source field (C2 bits 0-5).

In the L=A/B form, the ALU result is set into the L-register.

C0 BIT 4

C0 bit 4=0 Perform an exclusive OR on the A and B source inputs.

C0 bit 4=1 True binary add, set S bit 0 to 0, no carry-in, set S bit 3 to the value of the carry-out.

C0 BITS 5-7

This field specifies how the A input is to be presented to the ALUs.

C0 bits 5-7=001 Gate bits 4-7 of the A-source byte to the ALUs. Gate zeros as the high four bits of the A-source input.

C0 bits 5-7=010 Gate bits 0-3 of the A-source byte to the ALUs. Gate zeros as the low four bits of the A-source input.

C0 bits 5-7=011 Gate the eight bits for the A-source byte straight to the ALUs.

C0 bits 5-7=100 Same as 5-7=000

C0 bits 5-7=101 The A-source byte is crossed before gating. After crossing, gate A input bits 4-7 to the ALUs. Gate zeros as the high four bits of the A input.

C0 bits 5-7=110 The A source is crossed before gating. After crossing, gate the high four bits to the ALUs. Gate zeros as the low four bits of A input.

C0 bits 5-7=111 The A-source byte is crossed; then gated to the ALUs. Original bits 0-3 are gated in as bits 4-7, original bits 4-7 are gated in as bits 0-3.

C1 BITS 0-5

This field specifies the local-storage or external byte that is to be the A input to the ALU.

C1 BITS 6, 7

C1 Bits 6, 7=00 No status set.

C1 Bits 6, 7=01 In binary operations, S1 is set to the value of the carry-out of bit 1 of the result. S2 is set to 1 if the ALU result is not zero. S2 is unchanged if the result is zero.

In decimal operations, S1 is set to 1 if an invalid decimal digit is detected on either the A or B inputs. S1 is not changed if inputs are valid. S2 is set to 1 if the ALU result is not zero. S2 is not changed if the ALU result is zero.

C1 Bits 6, 7=10 S4 set to 1 if bits 0-3 of the ALU result are all zero.

S4 set to 0 if bits 0-3 of the ALU result are not all zero.

S5 set to 1 if bits 4-7 of the ALU result are all zero.

S5 set to 0 if bits 4-7 of the ALU result are not all zero.

C1 Bits 6, 7=11 S4 set to 1 if bits 0-5 of the ALU result are all zero.

S4 set to 0 if bits 0-5 of the ALU result are not all zero.

S5 set to 1 if bits 4-7 of the ALU result are all zero.

S5 set to 0 if bits 4-7 of the ALU result are not all zero.

Note: If S bit branching is specified by the control word, the branch testing is done before the S bits are modified by the operation.

C2 BITS 0-5

This field specifies the local-storage byte that is to be the B input to the ALU.

C2 BITS 6, 7

This field specifies the kind of gating for the B input to ALU.

C2 bits 6, 7=00 Present 0 as the B input.

C2 bits 6, 7=01 Gate the low four bits of the B-source only. Gate zeros as the high four bits.

C2 bits 6, 7=10 Gate the high four bits of the B source only. Gate zeros as the low four bits.

C2 bits 6, 7=11 Gate all eight bits of the B source to the ALU.

C2 BITS 0-7

For the forms A=A/K or Z=A/K, this field represents the value to be used as the B input to the ALU.

C3 BITS 0-5

This field is part of the next control-word address that is gated to the M3-register when the next address formation takes place. C3 bits 0-5 are gated to bits 0-5 of M3. If branching is specified by bits 6, 7 of C3, the status of the bits tested is ORed with bits 4 and 5 of C3 as the next address is set up.

C3 BITS 6, 7

This field specifies the kind of branch testing to be done to set up part of the next control-word address.

C3 bits 6, 7=00 No branch testing.

C3 bits 6, 7=01 OR the value of bits 2 and 3 of the S-register with bits 4 and 5 of C3 and set this result in bits 4 and 5 of M3.

C3 bits 6, 7=10 OR the value of bits 4 and 5 of the S-register with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

C3 bits 6, 7=11 OR the value of bits 6 and 7 of the S-register with bits 4 and 5 of C3 and set the result in bits 4 and 5 of M3.

Bit Definition, Indirect-Byte Type 10 or 11

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		**Form**		**Operation or Op/AXHL**				**A Source**						**Stat Set**		**B Source**						**A/B Hi-Lo**		**Next Address**						**Branch**	
								Word								**Word**															
10 or 11		Refer to bit definition for 10 or 11 word		Refer to bit definition for 10 or 11 word				Indirect byte addressing is specified by the value 1010 1011 1110 1111				00=--- 01=--- 10=+TA 11=-TA		00=--- 01=S12 10=S45 11=Z6		Indirect byte addressing is specified by the value 1010 1011 1110 1111				00=--- 01=--- 10=+TB 11=-TB		00=block 01=L 10=H 11=St						00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

Indirect-byte operations apply to both types of arithmetic words. *Indirect byte* means that the byte source A or B is addressed by T-register bits rather than bits in the control word itself.

The A-source byte is addressed by TA (bits 4 and 5 of the T-register). The B-source byte is addressed by TB (Bits 6 and 7 of the T-register).

T4, 5=00 address byte 0 of A-source
T4, 5=01 address byte 1 of A-source
T4, 5=10 address byte 2 of A-source
T4, 5=11 address byte 3 of A-source.

The B-source is addressed in the same way by bits 6 and 7 of the T-register.

TA and TB can be incremented or decremented during the operation of the indirect-byte function. This updating is specified by the value in C1 bits 4, 5 and C2 bits 4, 5.

The A-source or B-source field must be at a value of A, B, E, or F to specify the indirect-byte addressing facility. If the A- or B-source fields are at any other value, C1 or C2 bits 4, 5 specify the byte directly.

If indirect-byte addressing is specified for a source that is also the destination, the byte address specified by TA or TB causes a bit to be set in T-high in the following manner:

TA or TB	T 0 1 2 3
00	1 x x x
01	x 1 x x
10	x x 1 x
11	x x x 1

Another facility of the indirect-byte operation is the repeated execution of the arithmetic word until some branch test condition is met. The branch conditions are determined by the values in TA or TB and in some cases, by S-register bits.

If no branch condition is met, the address in the M- and N-registers remains unchanged, and the arithmetic word is re-read out of control storage, set into the C-register and re-executed. TA and TB are updated each time the word is executed, and the bit in T-high is set according to the destination byte.

Indirect-byte branching can be masked off by bits in the next address field. If C3 bits 2 and 3 are 11, then indirect branching is inoperative even if specified by C1 and C3 bits 4, 5. If C1 bits 4, 5 specify no action and C2 bits 4, 5 specify +TB or -TB, and C3 bit 3=1, indirect branching is inoperative and the next control-word address is set up during the first execution of the word. This would also be true for the case where a +TA or -TA is specified, C2 bits 4, 5 call for no action, and C3 bit 2=1.

If indirect-byte branching and S-register bit branching are both specified, S-register bit branching can be masked off by C3 bits 4 and 5. For example, if S2S3 (C3 bits 6, 7=01) is specified, and indirect branching is specified, C3 bit 4=1 prevents a branch due to the S2=1 condition. C3 bit 5=1 prevents a branch due to the S3=1 condition.

S-register bit branching operates differently if indirect branching is not specified. In this case, the S bits specified by C3 bits 6 and 7 are ORed with C3 bits 4 and 5 to form next address bits 4 and 5 regardless of the status of the S bits. That is, change of the next control-word address may be prevented only when indirect-byte branching is specified.

EXCEPTION: If indirect-byte branching is specified for both the A and B sources and S4, 5=11, C3 bits 0-5 are set into M3(N3) bits 0-5 and no ORing with any branch bits takes place.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

Examples of Indirect-Byte Arithmetic-Control Words

WORD	NEXT LABEL	STAT	STATEMENT
A3E8F80C	**ALIGN AB, 1, 1**		**YI=YI0, OE, QI, +TA+TB**

In this example the indirectly addressed byte of the Q-register is exclusively ORed with the indirectly addressed byte of the Y-register (which is blocked from entering ALU). The operation, in effect, is a move of bytes of the Q-register to bytes of the Y-register.

The operation calls for incrementing both TA and TB and branching on TA and TB specified in the Next Label field as AB. The branch test in this case is for either TA or TB at a value of 11.

Assume that the value in TA and TB is 00 at the start of the execution of this word. The operation would be as follows:

Test TA and TB; if not equal to 11, M3(N3) are not changed (unless S4S5=11).

Execute; move byte 0 of Q to byte 0 of Y, increment TA and TB by 1 (TA=01, TB=01). Set bit 0 of T to 1.

Reread this word from control storage, test TA and TB; if not equal to 11, M3(N3) are not changed.

Execute; move byte 1 of Q to byte 1 of Y, increment TA and TB by 1 (TA=10, TB=10). Set bit 1 of T to 1.

Reread this word from control storage, test TA and TB; if not equal to 11, M3(N3) are not changed.

Execute; move byte 2 of Q to byte 2 of Y, increment TA and TB by 1 (TA=11, TB=11). Set bit 2 of T to 1.

Reread this word from control storage, test TA and TB; both are new equal to 11. Set up the next control-storage address in the M(N) registers.

Execute; move byte 3 of Q to byte 3 of Y, increment TA and TB by 1 (TA=00, TB=00). Set bit 3 of T to 1. Read out next control word.

The T-register = 11110000 after this word has finished execution. The entire Q-register has been moved to the Y-register.

WORD	NEXT LABEL	STAT	STATEMENT
E1F0E264	**LAST**		**QI=QIL, OE, YIH**

This example shows an arithmetic word that specifies indirect-byte addressing, but no indirect-byte branching. Assuming that the T-register is 00000111 at the start of this word execution, the high four bits of byte 3 of the Y-register are combined with the low four bits of byte 1 of the Q-register and the result is gated to byte 1 of the Q-register. The word labeled LAST is then branched to.

WORD	NEXT LABEL	STAT	STATEMENT
8DAE9097	**UPL1 A1, S6, 1**	**S45**	**Z=LHI-K90+1, -TA**

This arithmetic word specifies indirect-word, indirect-byte addressing of the A source, and indirect branching on TA. The A1 in the Next Label field designates the test of TA and the fixing of next address bit 3=1.

Assume that TA=01 and S6=0 at the start of this word:

Read out the arithmetic word; test TA for a value of 00, TA not 00. M3(N3) are not changed.

Execute; add the complement of 90 plus 1 with the contents of byte 1 of the indirectly addressed word, and place the result in the Z-register. Set:

S4=0 if high 4 bits not zero.
S4=1 if high 4 bits equal zero.
S5=0 if low 4 bits not zero.
S5=1 if low 4 bits equal zero.

Decrement TA by 1 (TA=00).

Reread this word from control storage, TA now equal to 00. Set next control-word address into M(N) registers.

Execute; add the complement of 90 plus 1 with the contents of byte 0 of the indirectly addressed word, and place the result in the Z-register. Set S4 and S5 as explained. Decrement TA by 1 (TA=11). Read out the next control word.

Note: Because the result is destined only to the Z-register, the T-high bits are not set to indicate the destination bytes.

Arithmetic-Word Example, Byte Operation (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
DECOP S4, S5		**MW3C = MA3XL + MW3**

Starting Values

P-Reg = 03
S-Reg = FC
MA = 56 71 BF 57
MW = A6 D5 BD 28
Address of DECOP 00 = FC90

Objectives

Add the value of MA byte 3 (crossed and gated low) to the value of MW byte 3.

Gate the result to MW byte 3.

Set S3 with the value of the carry from the ALU operation.

Branch to the leg of DECOP as a result of the ORing of S4, S5 with the next address bits 4, 5.

Description

Read the MA-register from A local storage, gate to the A-register.

Read the MW-register from B local storage, gate to the B-register.

Gate byte 3 of the A-register to bytes 2, 3 of the A byte assembler.

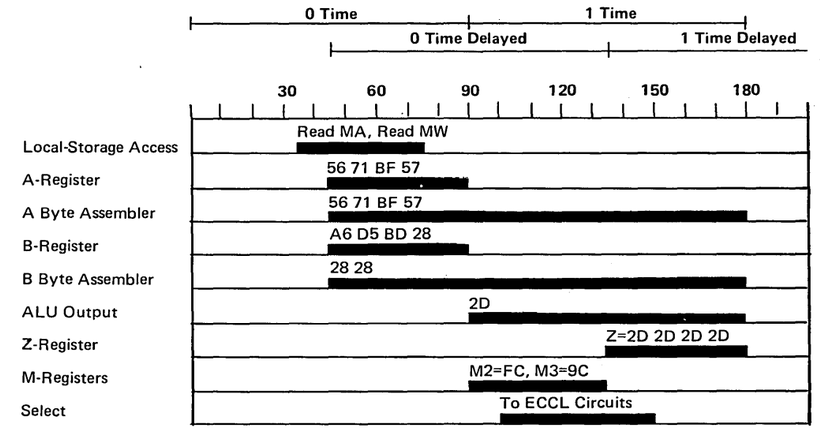
Gate byte 3 of the B-register to both bytes of the B byte assembler.

Perform the cross and low gate on byte 3 of the A input. Gate to both ALUs.

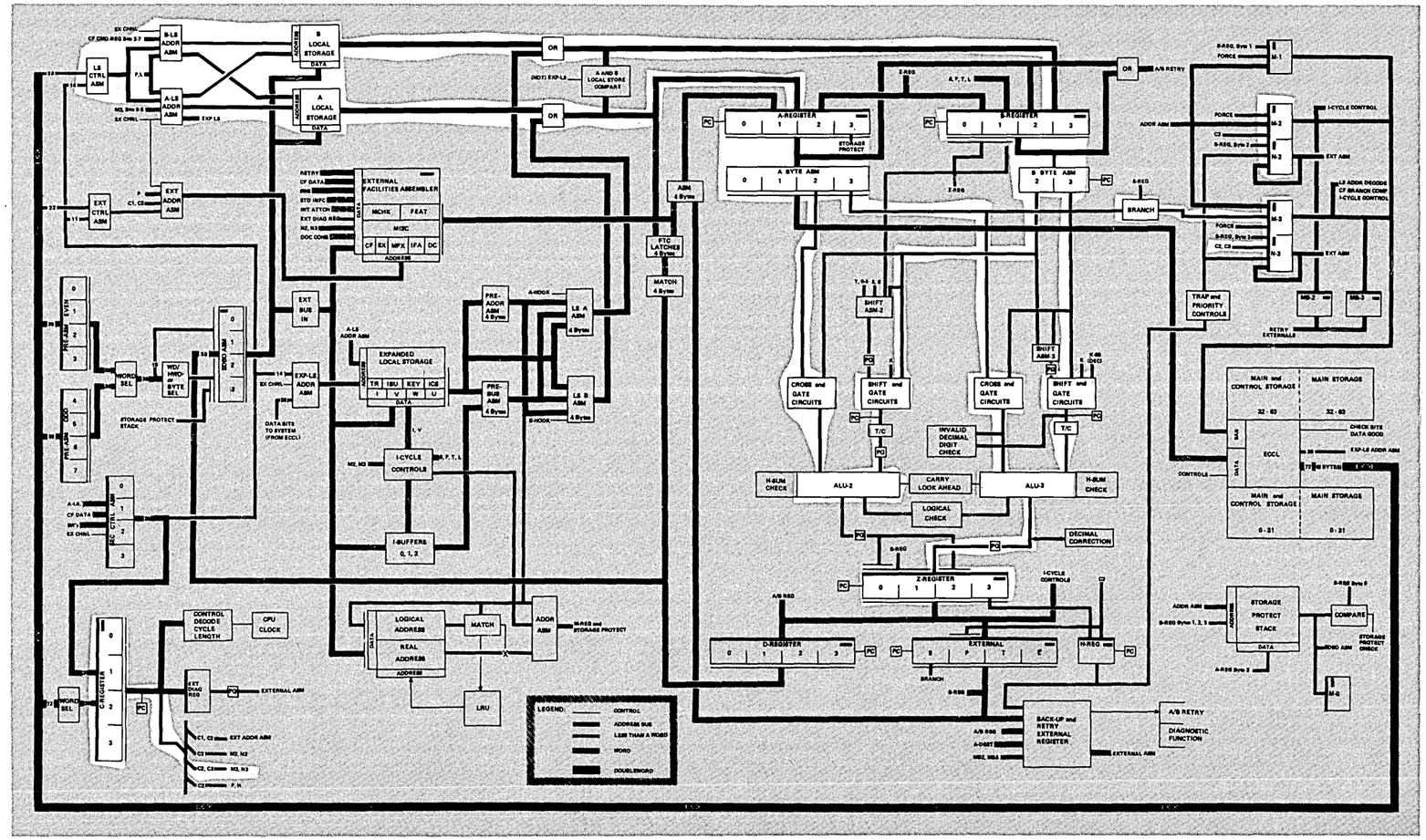
Gate the B input to both ALUs.

Perform the ADD and gate the result to all bytes of the Z-register.

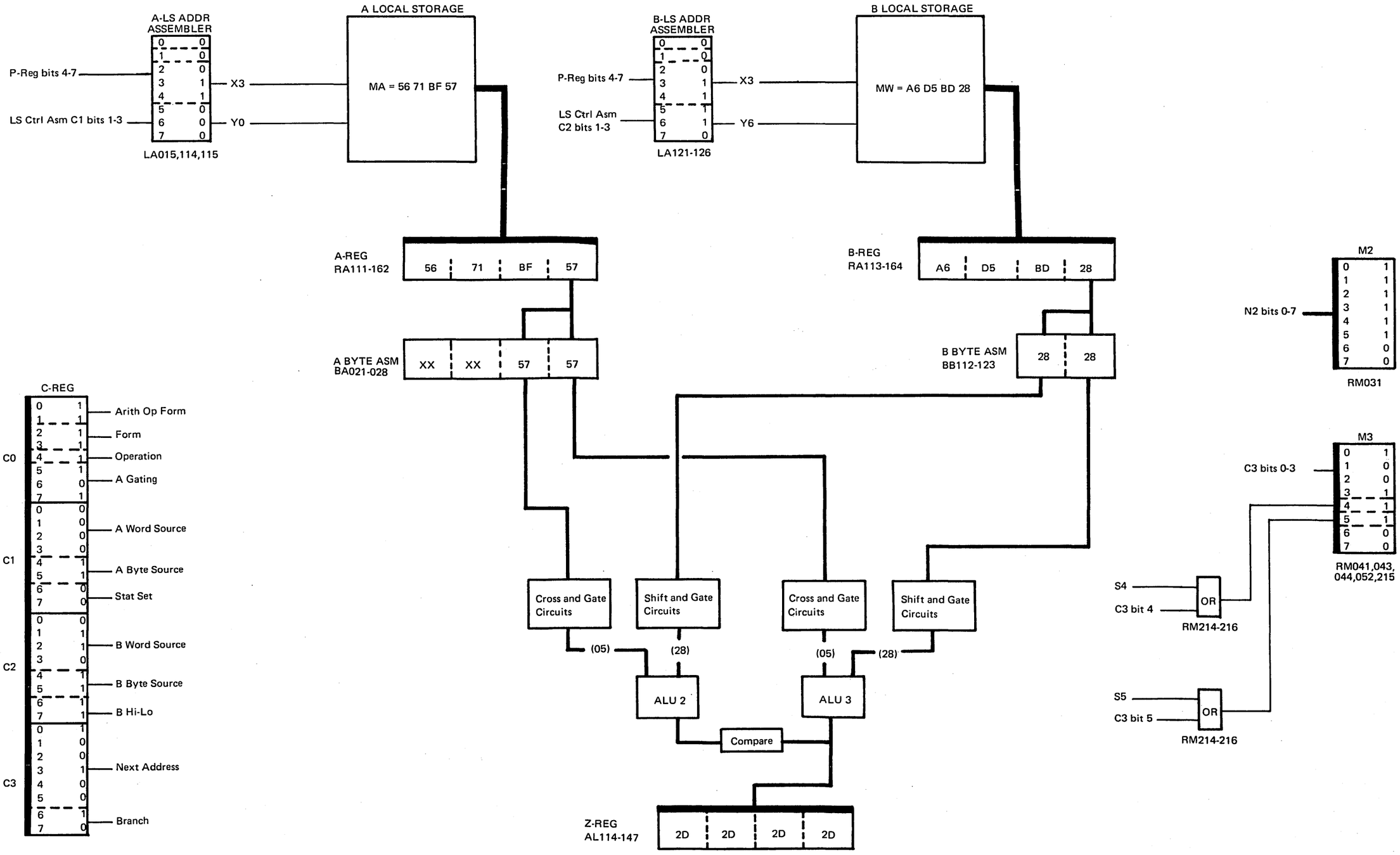
Set M2 from N2. Set M3 from the Next Address field and the result of the OR function with the S4, S5 bits.



Cycle Time = 202.5 ns



Arithmetic-Word Example, Byte Operation (Part 2 of 2)



Arithmetic-Word Example, Indirect-Byte Operation (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
LDPPI AB,X,X		**LHI=LHI0,OE,YI,+TA+TB**

Starting Values

S = 00
P = 02
L = 70
T = 0A
Address of this word = 1CF0

Objectives

Exclusive OR the indirectly addressed byte of the Y-register with the blocked A input to the ALUs.

Gate the result to the indirectly addressed byte of the indirectly addressed word in local storage, as specified by the symbol LHI.

Set a bit in T-high as specified by bits 4,5 of the T-register. This is done for both executions of this arithmetic word.

Increment TA and TB by 1 each time the word is executed.

Description

Test TA and TB for a value of 11. Condition is not met, the next control-word address is not set. N2,N3 are gated to M2,M3.

Read the indirectly addressed word (general register 7) from A local storage. Gate to the A-register. Gate byte 2 of the A-register to bytes 0-3 of the A byte assembler.

Read the Y-register from B local storage and gate to the B-register. Gate byte 2 of the B-register to bytes 2,3 of the B byte assembler.

Perform the exclusive OR of the blocked A input and byte 2 of the B source. Gate the result to bytes 0-3 of the Z-register.

Set bit 2 of the T-register.

Increment TA and TB by 1, TA and TB=11.

Read the arithmetic-control word from control storage and place it in the C-register.

Test TA and TB for a value of 11. Condition is met, set the next control-word address in the M-register.

Gate the first execution result to byte 2 of general register 7.

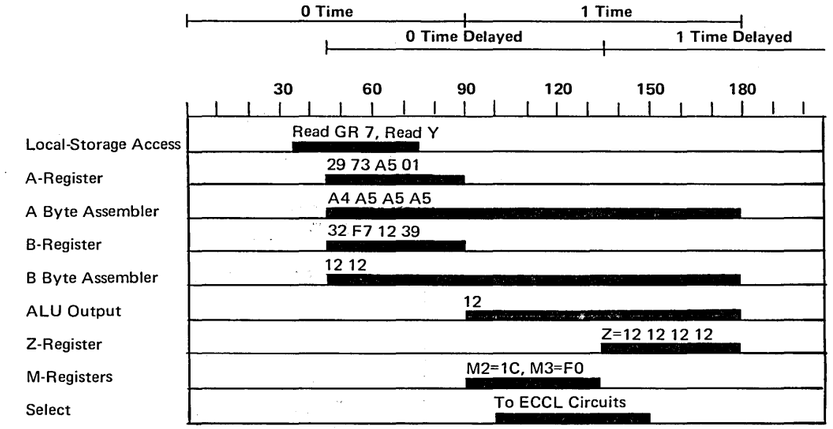
Perform the exclusive OR of the blocked A input and byte 3 of the B source.

Gate the result to bytes 0-3 of the Z-register.

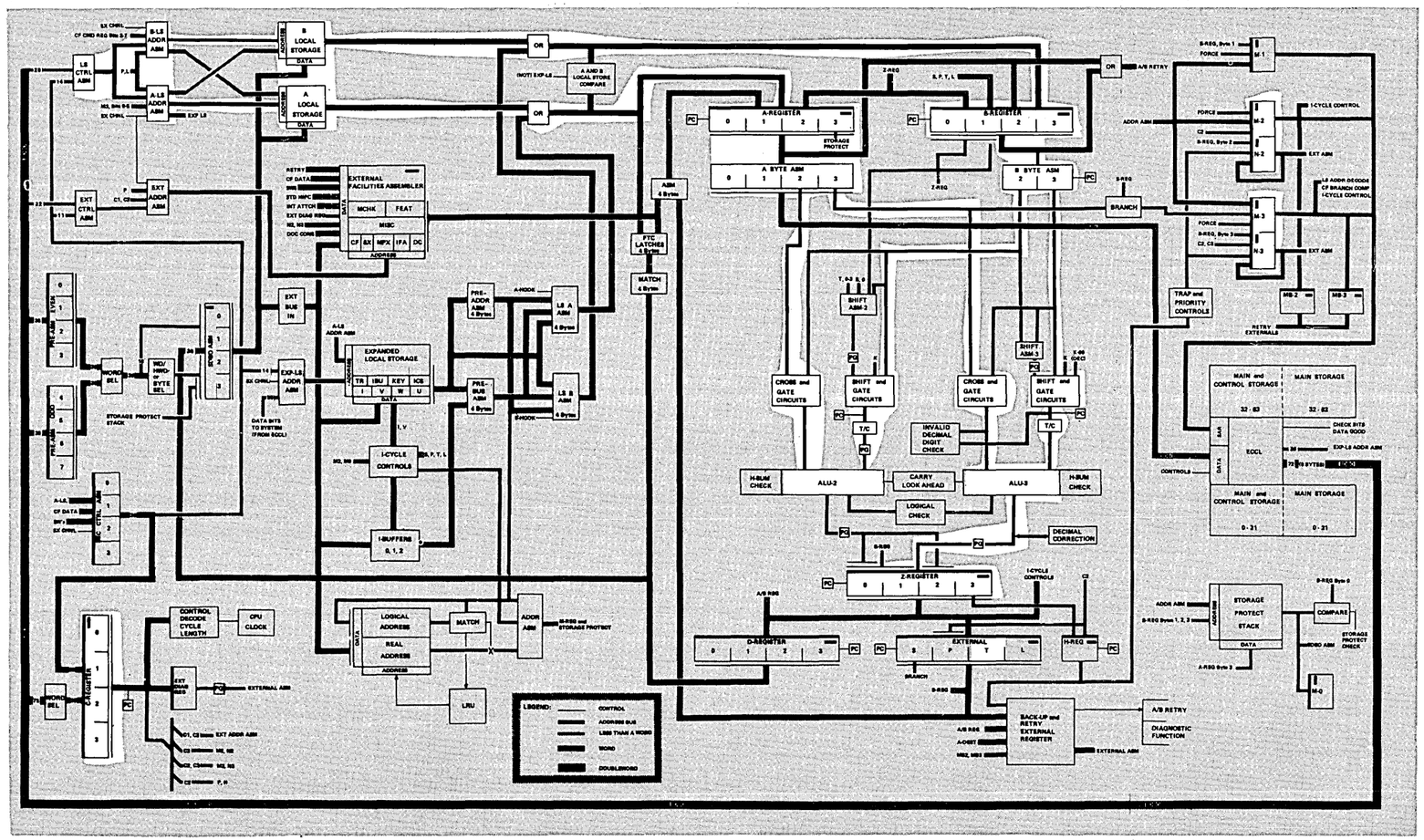
Set bit 3 of the T-register.

Increment TA and TB. After the second execution of this word, the T-reg = 0011 0000.

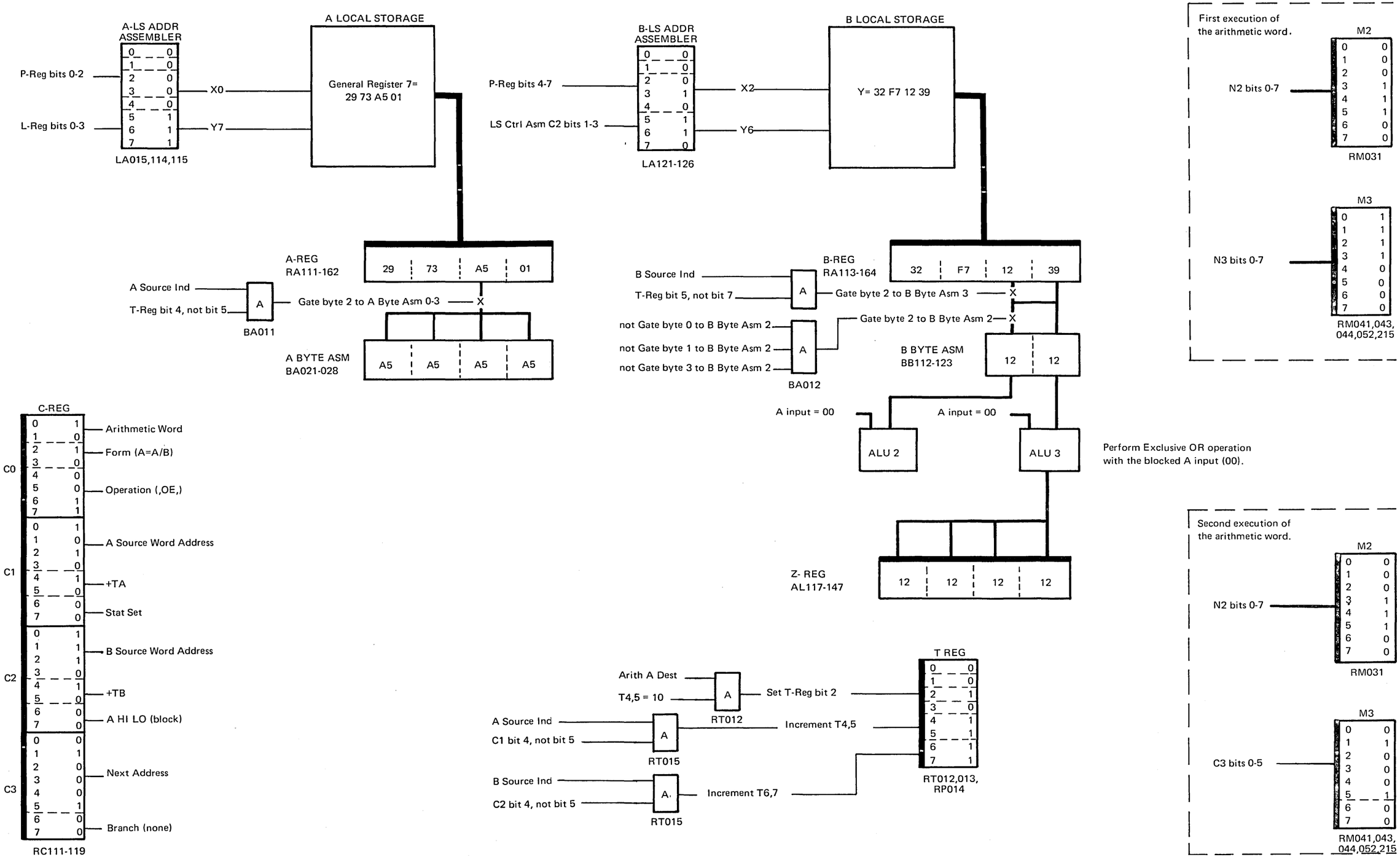
Branch to the next control word.



Cycle Time = 202.5 ns



Arithmetic-Word Example, Indirect -Byte Operation (Part 2 of 2)



Arithmetic-Word Example, Fullword Operation (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
	124	R = R+Q8

Starting Values

P = 02
R-Reg = 14 97 C5 A2
Q-Reg = 2D EC 59 72
Address of this word = 1C80

Objectives

Add the low 8 bits of the Q-register to the contents of the R-register.

Gate the low 24 bits of the result to the R-register.

Branch to the next control word.

Description

In the first half of the cycle, read out the R- and Q-registers and gate them to the A- and B-registers.

Gate the A-register to the A byte assembler.

Gate bytes 2, 3 of the B-register to bytes 2, 3 of the B byte assembler.

Gate byte 2 of the A byte assembler to ALU2. Gate byte 3 of the A byte assembler to ALU 3. Gate byte 3 of the B byte assembler to ALU3. The B input to ALU 2 is 00 because the B-source gating specifies low 8 bits only.

Perform the true ADD in both ALUs. Gate the ALU 3 result to bytes 1 and 3 of the Z-register. Gate the ALU 2 result to bytes 0 and 2 of the Z-register.

Set the next control-word address in the M-register.

In the second half of the cycle, gate bytes 0 and 1 of the B-register to bytes 2, 3 of the B byte assembler. Gate bytes 0 and 1 of the A-register to bytes 2, 3 of the A byte assembler.

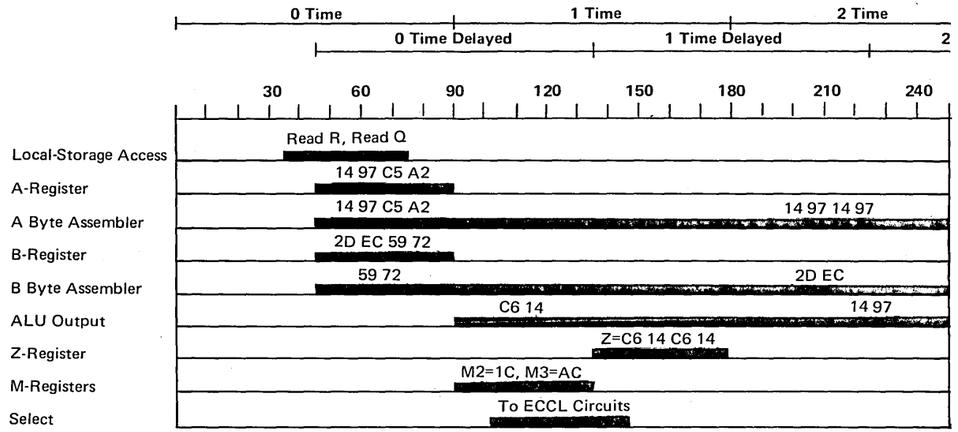
Gate byte 2 of the A byte assembler to ALU 2. Gate byte 3 of the A byte assembler to ALU 3.

The B inputs to both ALUs are 00 because of the B-source gating.

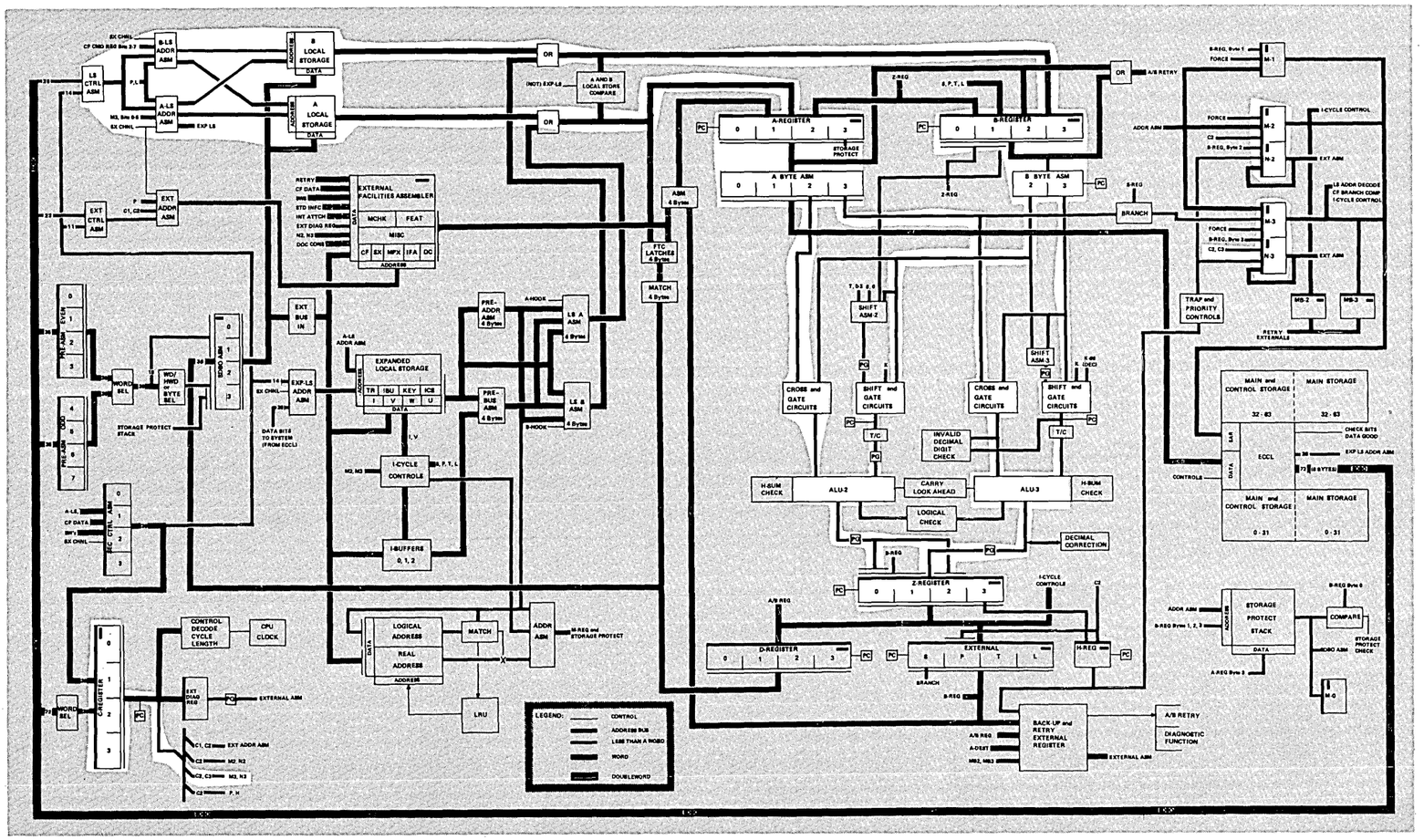
Perform the true ADD in both ALUs.

In the cycle following this control word, gate the ALU 3 result to byte 1 of the Z-register. Gate the ALU 2 result to byte 0 of the Z-register.

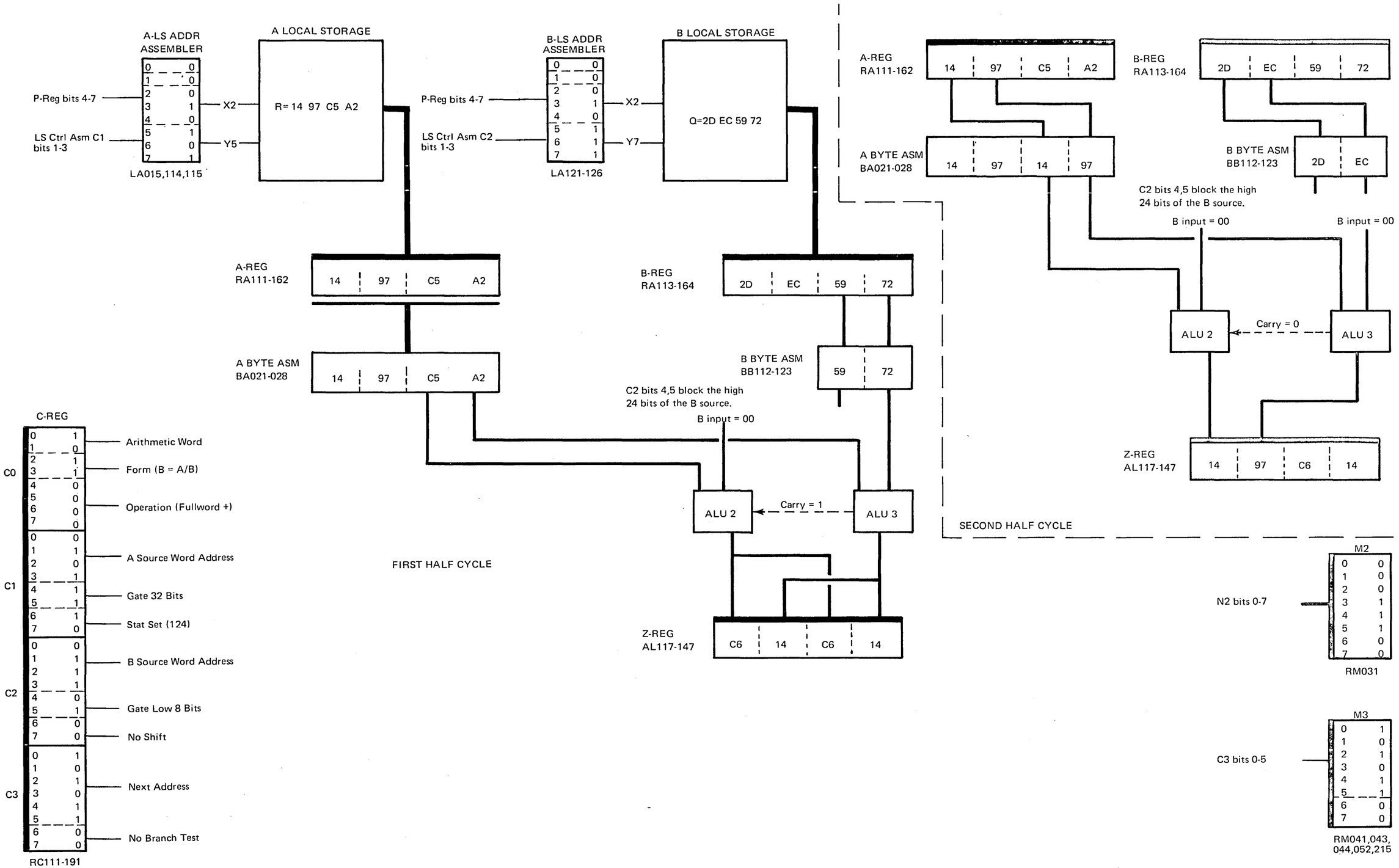
The Z-register is gated to the D-register through the SDBO assembler to both local storages. Only the low 24 bits of the result are stored in the R-register. Byte 0 of the R-register is not disturbed.



Cycle Time = 247.5 ns



Arithmetic-Word Example, Fullword Operation (Part 2 of 2)



Arithmetic-Word Example, Shift Operation (Part 1 of 2)

NEXT LABEL	STAT	STATEMENT
LENGTH 0, 1		**RC=0+R (SR4,TH)**

S tarting Values

P = 02
T = F0
R-Reg = 2D 3C EE 33

Objectives

Shift the R-register right four bits.

Save the low-order bits shifted out, in the high four bits of the T-register.

Shift the original high four bits of the T-register into the high four bits of the R-register.

Reset bit 3 of the S-register.

Description

Read the R-register from local-storage A and B into the A- and B-registers.

Gate the A-register to the A byte assembler. Gate the low four bits of byte 1 of the B-register, to the high four bits of ALU 2. Gate bytes 2, 3 of the B-register to bytes 2, 3 of the B byte assembler.

Gate the high four bits of byte 2 of the B byte assembler to the low four bits of ALU 2.

Gate the low four bits of byte 2, and the high four bits of byte 3 of the B byte assembler, to ALU 3.

Perform the ADD of the B inputs and the 00 A inputs in both ALUs.

Gate the ALU 2 result to bytes 0, 2 of the Z-register. Gate the ALU 3 result to bytes 1, 3 of the Z-register.

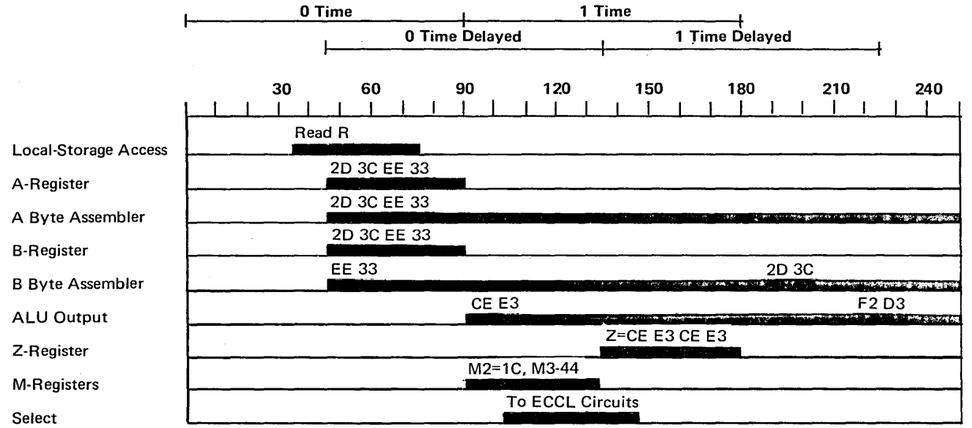
Gate bytes 0, 1 of the B-register to bytes 2, 3 of the B byte assembler.

Gate the high four bits of the T-register to the high four bits of ALU 2. Gate the high four bits of byte 2 of the B byte assembler to the low four bits of ALU 2.

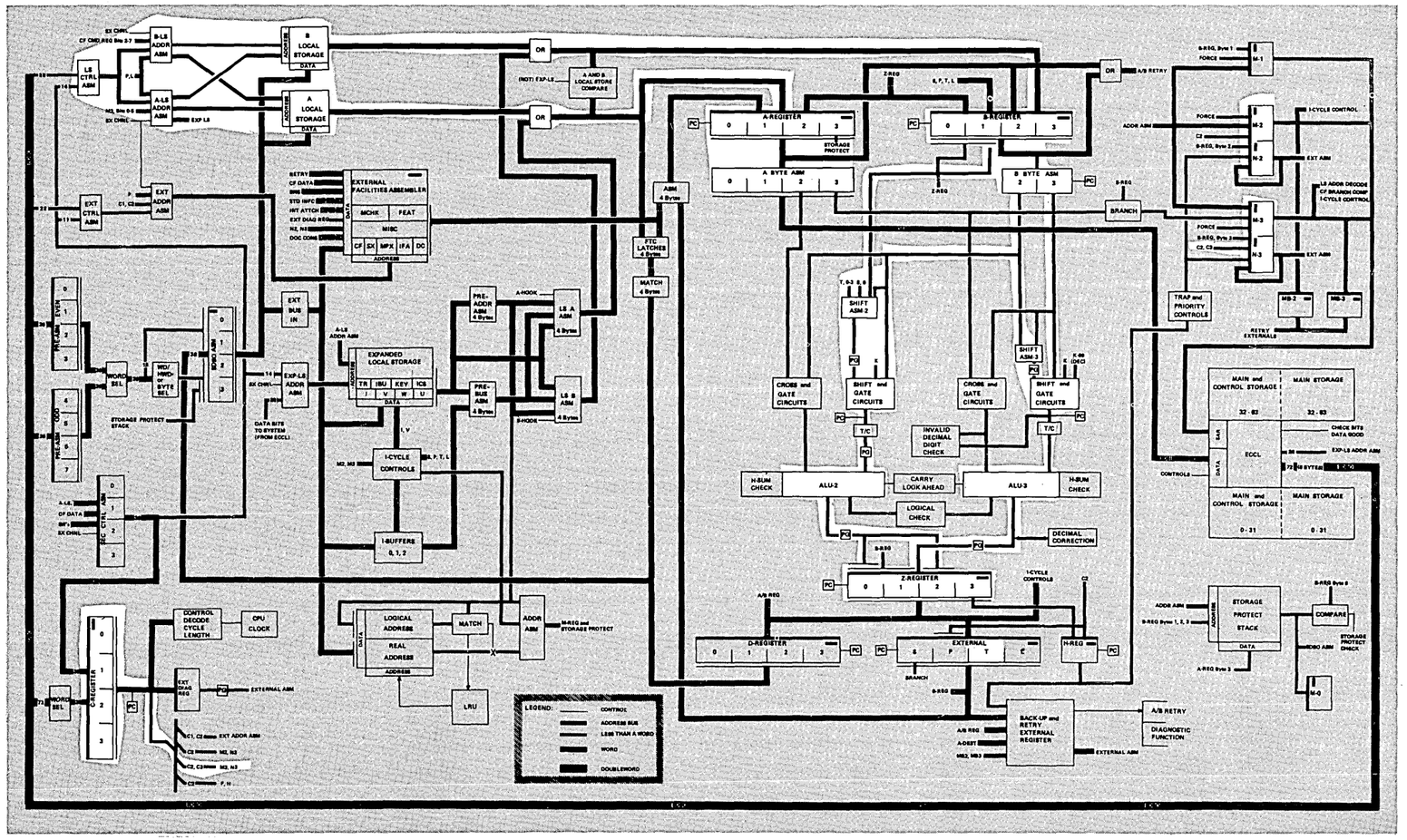
Gate the low four bits of byte 2, and the high four bits of byte 3 of the B byte assembler to ALU 3.

Perform the ADD of the B inputs and the 00 A inputs in both ALUs.

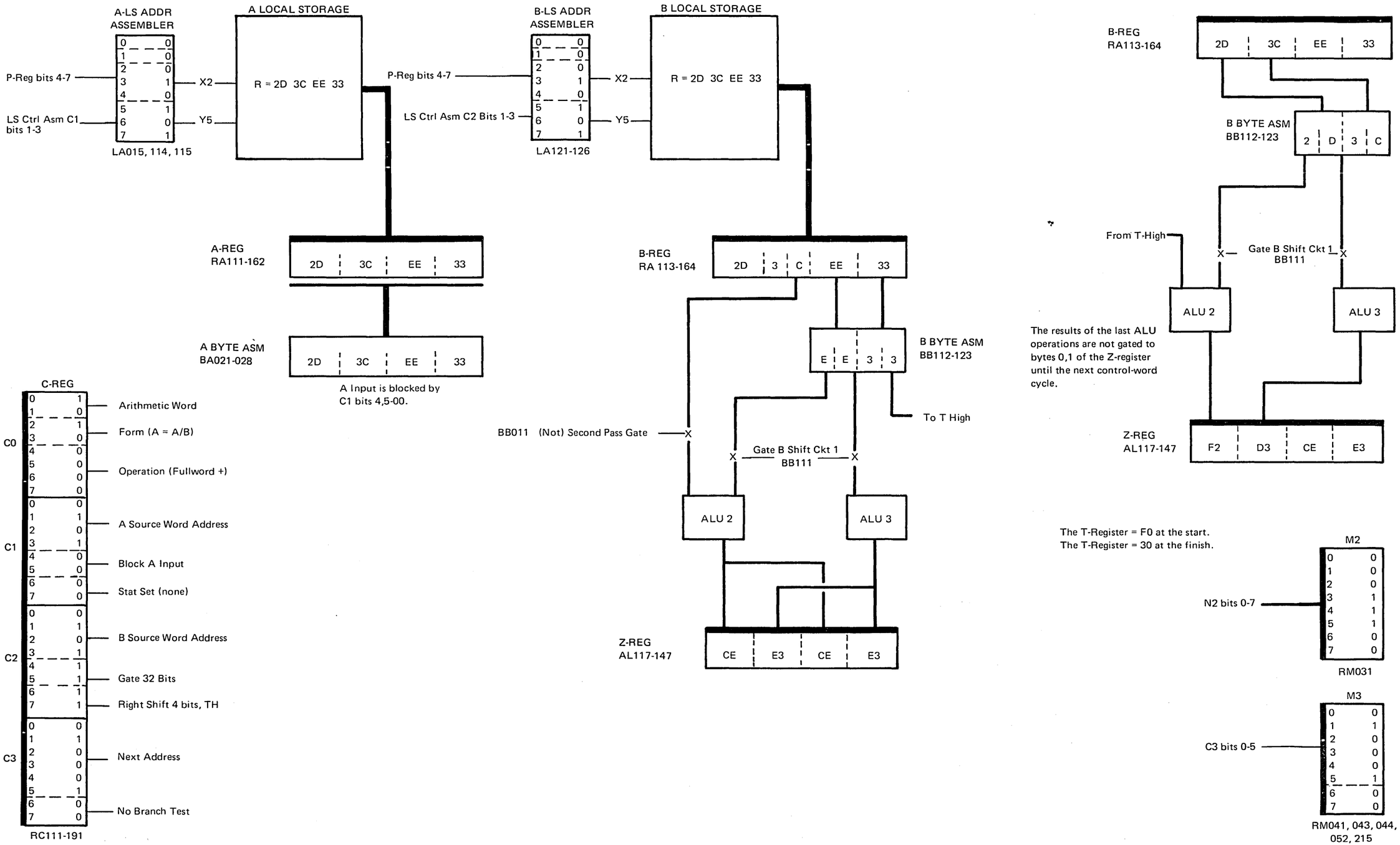
In the cycle that follows, the ALU 2 result is gated to byte 0 of the Z-register. The ALU 3 result is gated to byte 1 of the Z-register.



Cycle Time = 247.5 ns



Arithmetic-Word Example, Shift Operation (Part 2 of 2)



Microprogram Reference Material

Branch Symbols

SYMBOL	DEFINITION
0	Insert a 0 in bit/bits of M3
1	Insert a 1 in bit/bits of M3
B0-B7	Insert value of specified bit
BH	Insert a 1 if bits 0-3 of the branch source are not equal to zero
BL	Insert a 1 if bits 4-7 of the branch source are not equal to zero
I0	Force a return to I-Cycles
I1	Force a return to I-Cycles if no interrupt pending.
M6	Insert value of M6 after the address update
M7	Insert value of M7 after the address update
NZ	Insert a 1 if bits 0-7 of the branch source are not zero
S0-S7	Insert value of specified S-Register bit
Z0	Insert a 1 if S4 and S5 are both 1
TH	Special case— this symbol causes bits 0 and 1 of the T-Register to replace bits 2 and 3 in M3 and causes a module-switch function to occur
AB	Used in arithmetic words that are indirectly addressing byte facilities. If TA is decrementing, allow a branch to occur when TA=00. If TA is incrementing, allow a branch to occur when TA=11. If TB is decrementing, allow a branch to occur when TB=00. If TB is incrementing, allow a branch to occur when TB=11.

Stat Set Symbols

SYMBOL	DEFINITION
S2	In a storage-word operation, S2 is set to 1 if the count field is not zero. If the count field is zero, S2 is set to 0.
S12	In binary-byte operations, S1 is set to the value of the carry from the 1-bit position. S2 is set to 1 if the result is not zero. In binary fullword operations, S1 is set to the value of the carry from the 1-bit position of byte 0 of the result. S2 is set to 1 if the result is not zero. In decimal operations, S1 is set to 1 if an invalid decimal digit is detected in either the A or B source. S1 is not changed if digits are valid. S2 is set to 1 if the result is not zero. S2 is not changed if the result is zero.
S45	In arithmetic operations, S4 and S5 are set in the following manner: S4=1 if bits 0-3 of Z-Bus=0 S4=0 if bits 0-3 of Z-Bus not=0 S5=1 if bits 4-7 of Z-Bus=0 S5=0 if bits 4-7 of Z-Bus not=0 In the storage-word operation, S4 and S5 are set in the manner described, except that the test is made on the low byte of the count field after the count update.

SYMBOL	DEFINITION
Z6	This stat set causes S4 and S5 to be set in the same manner as the S45 stat set definition except that the setting of S4 depends on the status of bits 0-5 of the result instead of bits 0-3. In the storage word, S4 and S5 are set by testing bits 0-5 and 4-7 of the low byte of the count field.
Z24	In the fullword arithmetic operation, S2 is set to 1 if the low 24 bits of the result are not zero. If the low 24 bits are zero, S2 is not changed. S3 is set to the value of the carry-out of the bit 0 position of byte 1 of the result. Only the low 24 bits of the result are gated to the destination.
I24	Used in arithmetic operations to indicate that only the low 24 bits of the result will be gated to the destination. I24 inhibits the setting of any S-bits even if they are designated to be set. If I24 is specified with an arithmetic shift operation the high four bits of the result are 0's no matter what shift type is indicated.

Control-Word Bit Charts (Part 1)

Bit Definition of the Branch and Module-Switch Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch and Module Switch				Branch High				Branch Source						Branch Source Dest		Module Address								Next Address				Branch Low			
								Word				Byte																			
0	0	0	0	0000=0 0001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6 0111=BH 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7						00=0 01=1 10=2 11=3		00=--- 01=(S=Branch Source) 10=(T=Branch Source) 11=(L=Branch Source)												0000=- 0001=1 0010=Z0 0011=NZ 0100=S3 0101=S5 0110=S7 0111=BL 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7			

Bit Definition of the Branch and Link or Return Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch and Link or Return			Link Rtn	Branch High				Link Address								K/Module								Next Address					Branch Low		
								LS or EXT		Y		X		Spare																			
0	0	1	0	0=link 1=rtn			0000=0 0001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6		0=LS 1=EXT							BAL— If C3 bit 4=0, this field is set into the P-register. If C3 bit 4=1, this field contains the Module address for M3. RTN— Any bit in this field resets the corresponding bit in the H-register										BAL 0=P 1=MS RTN 0=--- 1=use NA				0000=0 0001=1 0010=Z0 0011=IO 0100=S3 0101=S5 0110=S7 0111=I1			

Control-Word Bit Charts (Part 2)

Bit Definition of the Branch Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch				Branch High				Branch Source						K Hi-Lo			S/R	S/R Source		K				Next Address				Branch Low			
								Word				Byte																			
0	0	0	1	0000=0 0001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6 0111=BH 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7								00=0 01=1 10=2 11=3		00=MS 01=L 10=H 11=St		0=--- 1=DK	0=,OR, 1=,A,	00=BS 01=S 10=P 11=GA		◄ - - - - - - - - Module Address - - - - - - - - ►								0000=0 0001=1 0010=Z0 0011=NZ 0100=S3 0101=S5 0110=S7 0111=BL 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7			

Bit Definition of the Word-Move (Version 0)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word Move				Version	Branch High			Destination								Source Note 1				Mask				Next Address					Branch Low		
								LS or EXT		Y		X		LS ELS																			
0	0	1	1	0	000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6			0=LS 1=EXT						0=LS 1=ELS												0=--- 1=Stop		000=0 011=1 010=Z0 011=--- 100=S3 101=S5 110=S7			

Note 1. The source cannot specify an external register. SPTL, however, can be specified by C2, B0-3.

Bit Definition of the Word-Move Word (Version 1)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word Move				Version	Branch High			Source								Destination				Mask				Next Address					Branch Low		
								LS or EXT		Y		X		Spare																			
0	0	1	1	1	000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6			0=LS 1=EXT																		0=--- 1=Stop		000=0 001=1 010=Z0 011=--- 100=S3 101=S5 110=S7			

Control-Word Bit Charts (Part 3)

Bit definition of the Storage Word (Non-K-Addressable)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Storage Word		**Subform**			**Branch High**			**Data Register**				**K or Inc/Dec**		**Stat Set**		**Address Source**				**Modes**		**Special Stat Set**		**Next Address**					**Branch Low**		
0	1	000=Read Word 001=Store Word 010=Read Half Wd 011=Store Half Wd 100=Read Byte 101=Store Byte 110=RDWRL/RDMP 100=Read Key 101=Store Key Subform for (special-stat set			000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6 111=M6							00=K-Addr 01=No Addr update 10=+ 11=-		00=— 01=S2 10=S45 11=Z6						00=CS 16 bit address 01=MS 10=ADR ADJ 11=CPU prot		00=— 01=TA 10=TB 11=special 00=— 01=TH 10=— 11= special		Read Store			0=— 1=Dec Cnt		000=0 001=1 010=Z0 011=SDC/VAL(0) 100=S3 101=S5 110=S7 111=M7		

Bit Definition of the Storage Word (K-Addressable)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Storage Word		**Subform**			**Branch High**			**Data Register**				**K Addr**		**K Modes**		**Address Source**				**K**				**Next Address**					**Branch Low**		
0	1	000=Read Word 001=Store Word 010=Read Half Wd 011=Store Half Wd 100=Read Byte 101=Store Byte			000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6 111=M6							00 00=MS (0 00 KK) 01=CS (CM KK) 10=CS (FF KK) 11=CS (FK 8 bit Addr)																0=—					000=0 001=1 010=Z0 011=— 100=S3 101=S5 110=S7 111=M7		

Bit Definition of the Arithmetic Word (Type 10 Byte Version)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		**Form**			**Operation**			**A Source**						**Stat Set**		**B Source**						**A Hi-Lo**		**Next Address**						**Branch**	
								Word				**Byte**				**Word**				**Byte**											
1	0	00=A=A/K 01=Z=A/B 10=A=A/B 11=B=A/B 00=Z=A/K			0011=,OE, 0100=+ 0101=+ (+1) 0110=C + (Rst S0) 0111=,A, 1000=,OR, 1001=C + - (+C) 1010=C ,D+-, (+C) 1011=ABCK 1100=- 1101=-(+1) 1110=C-(+1) (Set S0) 1111=,A-,							00=0 01=1 10=2 11=3		00=— 01=S12 10=S45 11=Z6						00=0 01=1 10=2 11=3		00=block 01=L 10=H 11=St						00=— 01=S2,S3 10=S4,S5 11=S6,S7		

Control-Word Bit Charts (Part 4)

Bit Definition of the Arithmetic Word (Type 10 Fullword Version)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		Form		Operation				A Source						Stat Set		B Source						Shift		Next Address						Branch	
								Word				A Input				Word				B Input											
1	0	00=A=A/K 01=Z=A/B 10=A=A/B 11=B=A/B		0000=C + (Rst S0) 0001=C - (+1) (Set S0) 0010=C +- (+C)								00=block 01=16 bits 10=24 bits 11=32 bits		00=--- 01=S12 10=124 11=Z24						00=4 bits 01=8 bits 10=12 bits 11=32 bits		00=--- 01=(SR4,0) 10=(SR4,S0) 11=(SR4,TH)						00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

Bit Definition of the Arithmetic Word (Type 11)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith Op Form		Form		Op	A Gating			A Source						Stat Set		B Source						B Hi-Lo		Next Address						Branch	
								Word				Byte				Word				Byte											
1	1	00=A=A/K 01=L=A/B 10=A=A/B 11=B=A/B		0=,OE, 1=C + (Rst S0)	000=block 001=L 010=H 011=St 100=X block 101=XL 110=XH 111=X							00=0 01=1 10=2 11=3		00=--- 01=S12 10=S45 11=Z6						00=0 01=1 10=2 11=3		00=block 01=L 10=H 11=St						00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

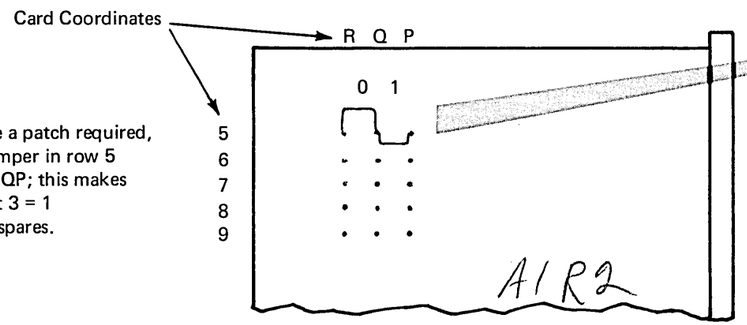
Bit Definition of the Arithmetic Word (Indirect Byte, Type 10 or 11)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith Op Form		Form		Operation or Op/AXHL				A Source						Stat Set		B Source						A/B Hi-Lo		Next Address						Branch	
								Word								Word															
10 or 11		Refer to bit definition for 10 or 11 word		Refer to bit definition for 10 or 11 word				Indirect byte addressing is specified by the value 1010 1011 1110 1111				00=--- 01=--- 10=+TA 11=-TA		00=--- 01=S12 10=S45 11=Z6		Indirect byte addressing is specified by the value 1010 1011 1110 1111				00=--- 01=--- 10=+TB 11=-TB		00=block 01=L 10=H 11=St						00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

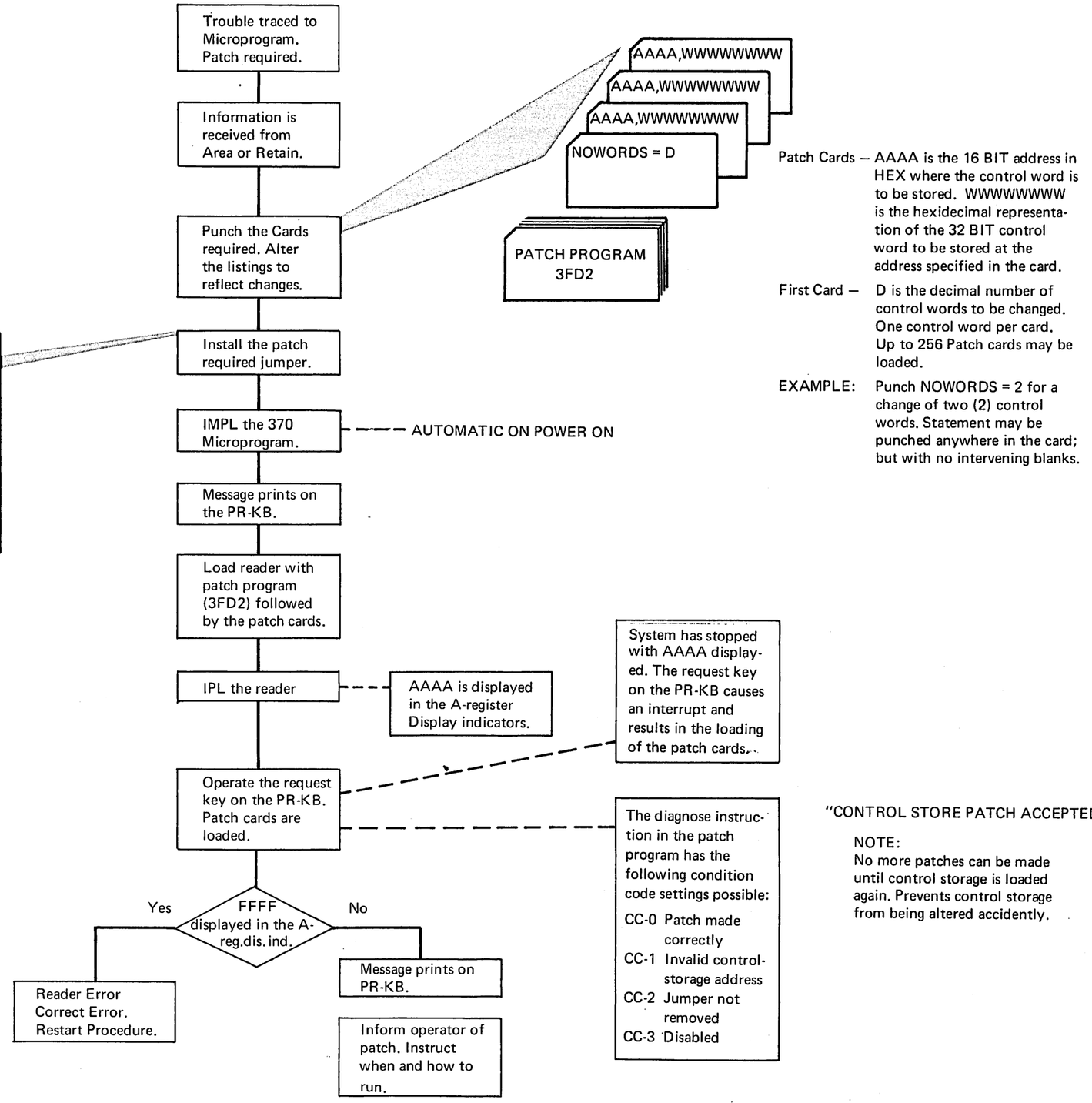
Microprogram Temporary Fix Procedure

PURPOSE:
Provide the capability for making temporary fixes to the 370 microprograms, stored in reloadable control storage. Macro program 3FD2 is used in conjunction with microprograms GGAD and GRST to accomplish this task. 3FD2-MPTF (card deck shipped with machine). GGAD* - contains diagnose instruction and patch routine. GRST* - system reset program checks to see if a patch is required.

* - on IMPL DISK



To designate a patch required, move the jumper in row 5 from RQ to QP; this makes TOD low bit 3 = 1
Bits 4-7 are spares.



Patch Cards — AAAA is the 16 BIT address in HEX where the control word is to be stored. WWWWWWWW is the hexidecimal representation of the 32 BIT control word to be stored at the address specified in the card.

First Card — D is the decimal number of control words to be changed. One control word per card. Up to 256 Patch cards may be loaded.

EXAMPLE: Punch NOWORDS = 2 for a change of two (2) control words. Statement may be punched anywhere in the card; but with no intervening blanks.

"CONTROL STORE PATCH ACCEPTED"

NOTE:
No more patches can be made until control storage is loaded again. Prevents control storage from being altered accidently.

The BC module in control storage is reserved for microprogram patches. All patches to the microprogram should be made in this area to prevent inadvertent overlay of existing control words.

Patch example
Control word at address C080 is to be replaced with patch.

ADDR	WORD	
C024	D66C7D80	Store control word 0000BC00 at address C080. This is a branch and module switch to the first word of the reserved patch area.
C080	F56C7C9C	
C028	C07600C0	
C02C	C07CF0C0	

Patch words have been loaded into the BC module by the patch program.

BC Module

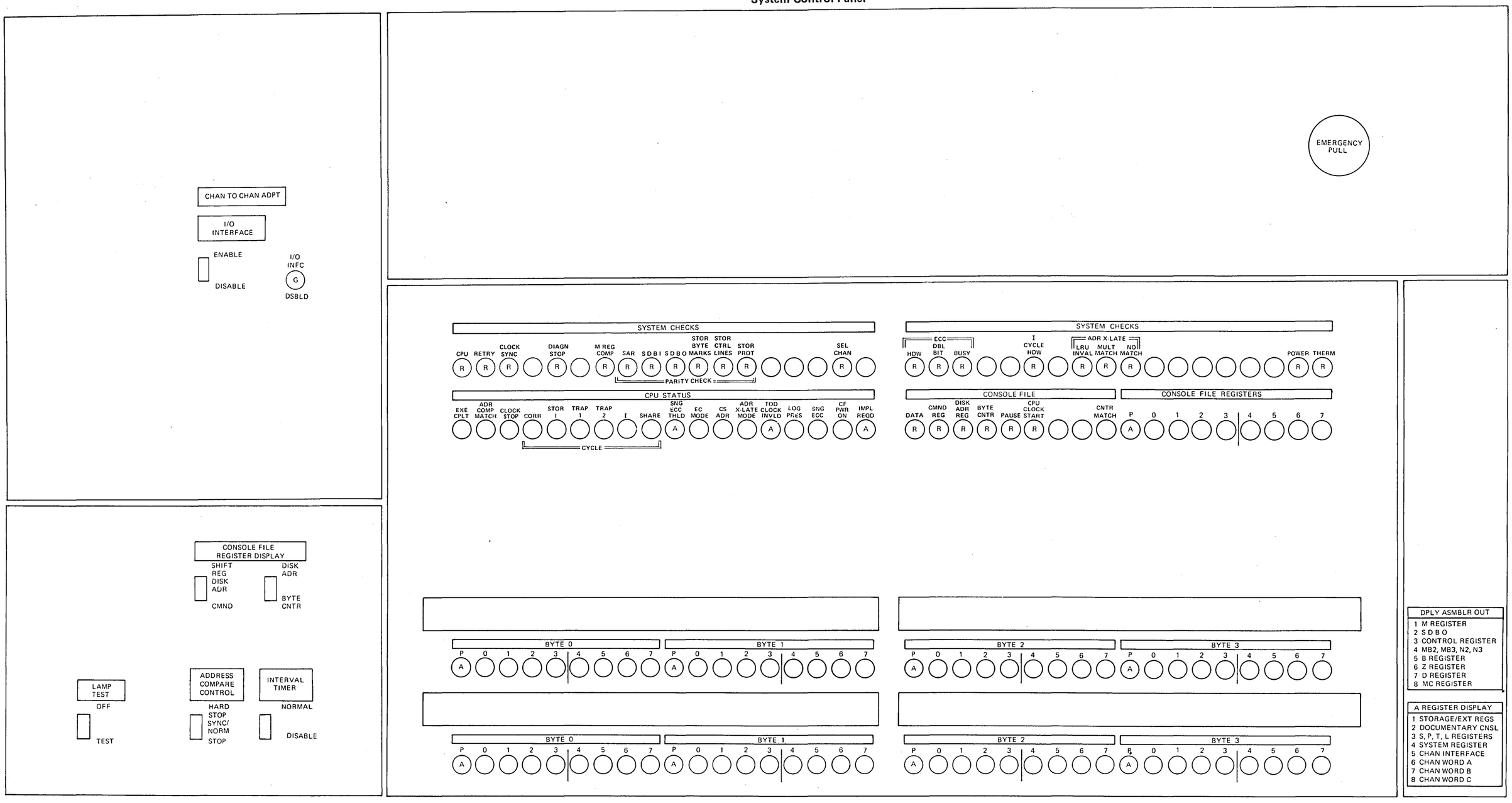
ADDR	WORD	
BC00	F56C7C04	
BC04	D6677D08	
BC08	843EFF0C	
BC0C	0000C028	Branch and module switch to C028 to continue processing.

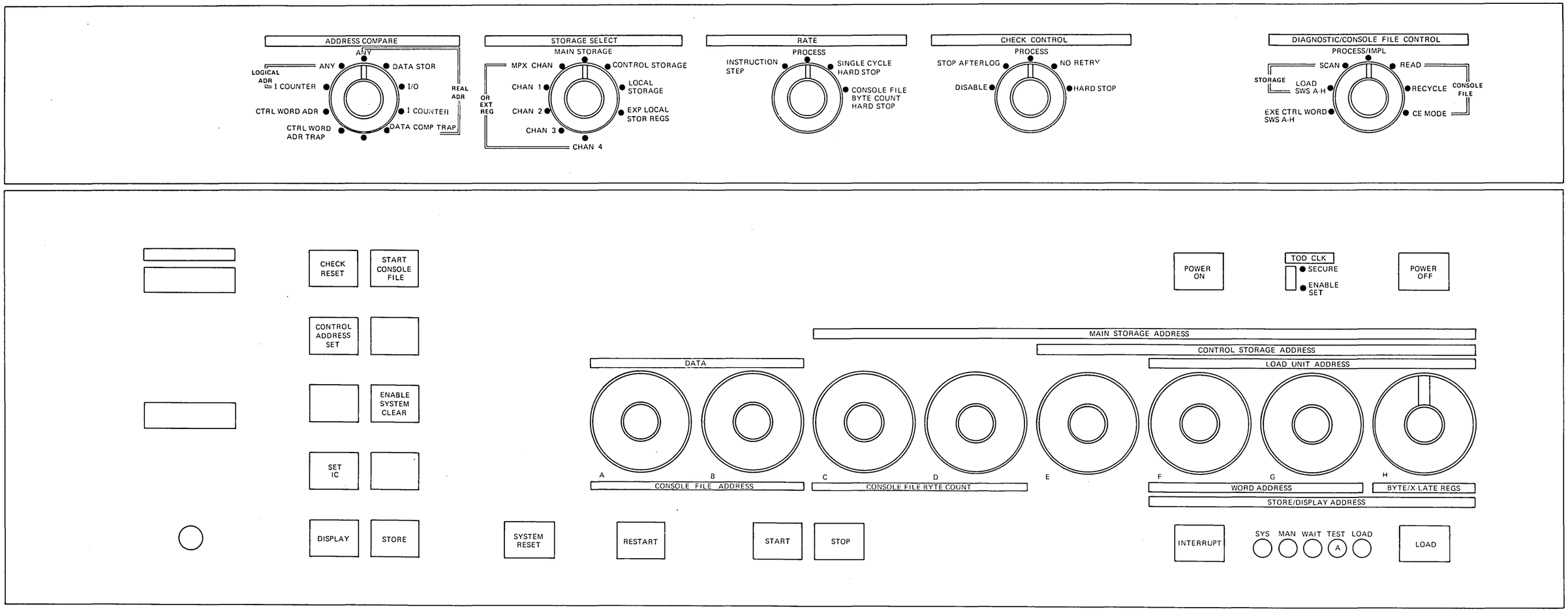
CHAPTER 5. SYSTEM CONTROL PANEL

INTRODUCTION

The System/370 Model 145 Central Processing Unit (CPU) has indicators and manual controls that permit operation of the system and observation of the results of any operation. These indicators and controls are assembled on a panel that serves as both an operator's system control console and a maintenance control panel.

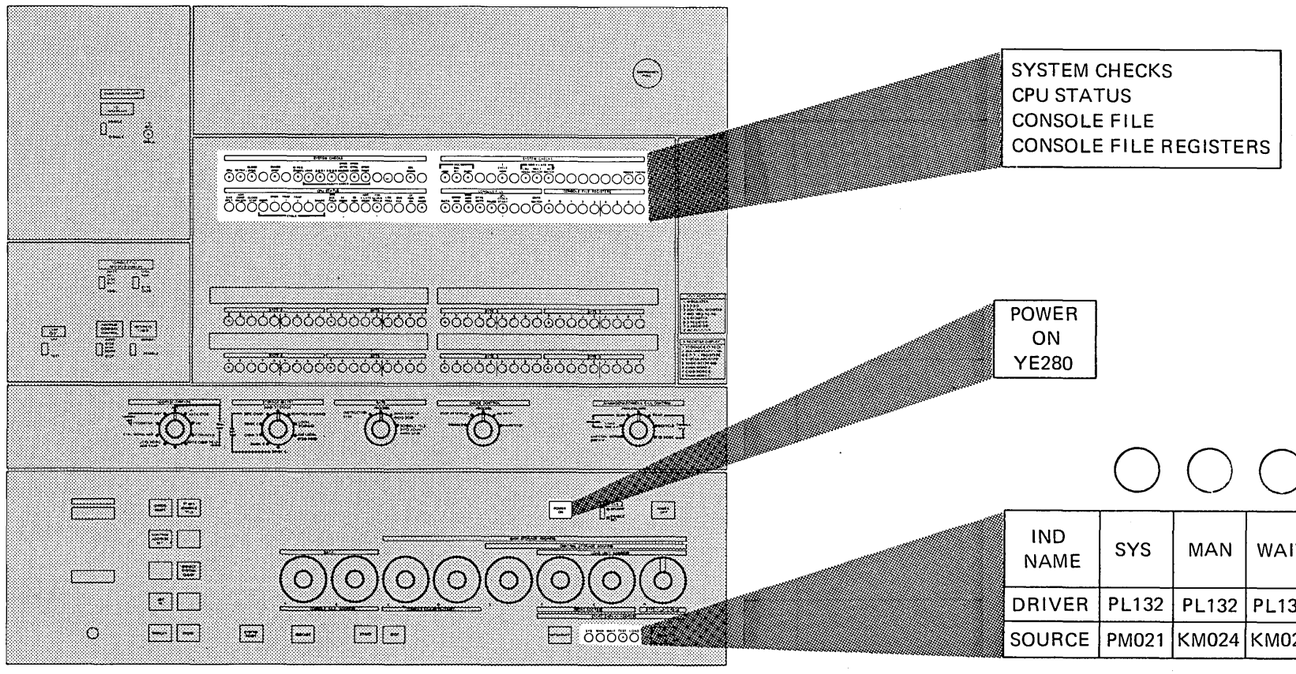
System Control Panel




ADDRESS COMPARE
ANY
ANY
DATA STOR
LOGICAL ADR
I COUNTER
I/O
REAL ADR
CTRL WORD ADR
I COUNTER
CTRL WORD ADR TRAP
DATA COMP TRAP
STORAGE SELECT
MAIN STORAGE
MPX CHAN
CONTROL STORAGE
CHAN 1
LOCAL STORAGE
OR EXT REG
CHAN 2
EXP LOCAL STOR REGS
CHAN 3
CHAN 4
RATE
PROCESS
INSTRUCTION STEP
SINGLE CYCLE HARD STOP
CONSOLE FILE BYTE COUNT HARD STOP
CHECK CONTROL
PROCESS
STOP AFTERLOG
NO RETRY
DISABLE
HARD STOP
DIAGNOSTIC/CONSOLE FILE CONTROL
PROCESS/IMPL
SCAN
READ
STORAGE
LOAD SWS A-H
RECYCLE
CONSOLE FILE
EXE CTRL WORD SWS A-H
CE MODE
CHECK RESET
START CONSOLE FILE
CONTROL ADDRESS SET
ENABLE SYSTEM CLEAR
SET IC
POWER ON
TOD CLK
SECURE
ENABLE SET
POWER OFF
MAIN STORAGE ADDRESS
CONTROL STORAGE ADDRESS
LOAD UNIT ADDRESS
DATA
A
B
C
D
E
F
G
H
CONSOLE FILE ADDRESS
CONSOLE FILE BYTE COUNT
WORD ADDRESS
BYTE/X-LATE REGS
STORE/DISPLAY ADDRESS
DISPLAY
STORE
SYSTEM RESET
RESTART
START
STOP
INTERRUPT
SYS
MAN
WAIT
TEST
LOAD
A
LOAD

FIXED INDICATORS

LOGIC PAGE REFERENCE CHART



IND NAME	SYS	MAN	WAIT	TEST	LOAD
DRIVER	PL132	PL132	PL132	PL132	PL132
SOURCE	PM021	KM024	KM024	PM021	RS013

SYSTEM CHECKS

IND NAME	CPU	RETRY	CLOCK SYNC		DIAGN STOP		M REG COMP	SAR	SDBI	SDBO	STOR BYTE MARKS	STOR CTRL LINES	STOR PROT				SEL CHAN	
	R	R	R		R		R	R	R	R	R	R	R				R	
DRIVER	PL142	PL152	PL142		PL 142		PL141	PL142	PL142	PL142	PL142	PL151	PL142				PL142	
SOURCE	RE011	RS011	RE045		KF025		RE046	RE041	RE041	RE041	RE041	RE051	RE043				GA111	

PARITY CHECK

SYSTEM CHECKS

IND NAME	HDW	ECC DBL BIT	BUSY			I CYCLE HDW		ADDR X-LATE LRU INVALID	MULTI MATCH	NO MATCH							POWER	THERM
	R	R	R			R		R	R	R	R							
DRIVER	PL151	PL151	PL151			PL151		PL152	PL151	PL151							YD612	YD612
SOURCE	RE051	RE051	RE051			RE051		RE044	RE044	RE044							YD721	YD721

CPU STATUS

IND NAME	EXE CPLT	ADR COMP MATCH	CLOCK STOP	CORR	STOR 1	TRAP 1	TRAP 2 CYCLE	I	SHARE	SNG ECC THLD	EC MODE	CS ADR	ADR X-LATE MODE	TOD CLOCK INVLD	LOG PRES	SNG ECC	CF PWR ON	IMPL REQD
DRIVER	PL141	PL141	PL141	PL141	PL141	PL152	PL161	PL152	PL152	PL161	PL141	PL152	PL152	PL151	PL141	PL151	PL141	PL141
SOURCE	KM024	KM024	PM021	PM022	KM024	KM024	RE053	RU054	JG032	RE062	RE011	RE053	RM074	CT112	RS011	RE051	KF042	RS013

CONSOLE FILE

IND NAME	DATA	CMND REG	DISK ADR REG	BYTE CNTR	PAUSE	CPU CLOCK START			CNTR MATCH
	R	R	R	R	R	R			
DRIVER	PL161	PL161	PL161	PL161	PL161	PL161			PL161
SOURCE	KF042	KF036	KF036	KF071	KF022	KF022			KF071

CONSOLE FILE REGISTERS

IND NAME	P	0	1	2	3	4	5	6	7
	A								
DRIVER	PL162	PL162	PL162	PL162	PL162	PL162	PL162	PL162	PL162
SOURCE	KF035	KF072	KF072	KF072	KF072	KF072	KF072	KF072	KF072

SYSTEM INDICATORS

Power On

The power-on key is backlighted by a two-color indicator. A red indicator lights five seconds after pressing of the key. When the power-up cycle is complete, the red indicator changes to white.

SYS (System)

Indicates that one of the two CPU use-meters is running.

MAN (Manual)

Indicates that the CPU clock is stopped or that a word-move STOP word is being executed. Several of the manual console controls are operative only when this indicator is on. (*Note:* If the CPU clock is stopped between the first and second cycles of a storage word or after a trap-1 cycle or a trap-2 cycle of a trapping sequence, the manual indicator is off. If the clock is stopped, the clock stop indicator will be on.

Wait

Indicates that the CPU clock is running but that program instruction processing is not occurring. The current PSW has the wait bit on. If an interrupt occurs (and is allowed, such as an I/O interrupt), processing is initiated as required by the program and WAIT turns off.

Test

Indicates that one or more of the following switches are not in the PROCESS or NORMAL position:

1. Address compare control toggle
2. Rate
3. Check control
4. Diagnostic/console file control
5. CE meter key

Load

This indicator is turned on when system register byte 2 bit 4 is set by the system-reset routine (GRST). The indicator and the system register are reset by the GMMS microroutine after successful completion of the IPL.

SYSTEM CHECKS

CPU
(MCKB byte 1, bit 7)

Indicates that a machine error has occurred and that a bit has been set ON in the machine-check registers (MCKA-MCKB). To determine the kind of error, display MCKA in the upper roller position 8, and MCKB in the system check indicators. (*Note:* MCKB byte 2, bits 1, 6, 7 and byte 3, bits 0-7 are not error conditions and are not displayed directly on the console. To investigate these bits manually, display external storage word 06.)

Retry

Indicates that the retry microroutine (GHRT) has determined that a machine-check error is either uncorrectable or unretryable. A branch to the hard machine-check routine (GHEC) takes place and RETRY is set ON along with system register byte 0 bit 1 and a word-move STOP is executed.

CLOCK SYNC
(MCKB byte 0 bits 6, 7)

This indicator turns on if any of the system clocks located on each of the logic boards is out of synchronism with the rest of the system clocks. The check is made at 135 to 180 to ensure that '1 Time Delay' is active and that '0 Time is not active. ('0 Time' is actually delayed 16 ns before being checked.)

DIAGN STOP (Diagnostic Stop)

This indicator is controlled by the diagnostic and system-reset microprograms. The indicator is turned on by a nonzero ALU output if the following conditions exist.

1. System is in diagnostic mode,
2. DIAG0, Bit 0-1, and
3. S12 or Z24 STAT SET is specified in the microword.

DIAGN STOP also turns on if a three-bit compare error is detected when the console file is executing a compare-type command. The console file ANDs each of the following checks with a specific file command to turn on the indicator:

1. C-register parity.
2. S-register Dup check.
3. M-register compare.
4. ALU logical check.

M-REG COMP
(MCKB byte 1 bits 0, 1, 2, 3)

Indicates a parity error in the control lines to the M- and N-Registers. See M-Register Dup Check in Chapter 13.

SAR Parity Check (Storage Address Register)
(MCKB byte 0 bit 0)

Indicates an error on the address lines from the M-register to storage. The check is performed on the output of the storage address register.

SDBI Parity Check (Storage Data Bus In)
(MCKB byte 0 bit 1)

Indicates a parity error detected on data issued to storage during a store operation. The data check is performed in the parity-out generator.

SDBO Parity Check (Storage Data Bus-Out)
(MCKB byte 0 bit 2)

Indicates a parity error detected on data fetched from storage (storage data bus-out).

Stor Byte Marks Parity Check (Storage Byte Marks)
(MCKB byte 0 bit 3)

Indicates an error detected (during a store operation) on the four byte-store lines to storage.

Stor Ctrl Lines Parity Check (Storage Control Lines)
(MCKB byte 2 bit 2)

Indicates that the generated parity bit does not compare with the instruction parity bit gated from the CPU.

Stor Prot Parity Check (Storage Protect)
(MCKB byte 0 bit 5)

Indicates that a storage-protect key with incorrect parity has been accessed from the storage-protect stack. The error can occur only on a storage 2 cycle to main storage.

Sel Chan (Selector Channel)

Indicates detection of one or more of the following checks in a selector channel:

1. Channel data check.
2. Channel control check.
3. Interface control check.

To determine which channel caused the error, display upper roller position 1 (byte 1 bit 0-3). To define which kind of error occurred, set STORAGE SELECT to the failing channel and set the lower roller to position 6 (byte 1 bit 4-6).

ECC Hdw (Hardware)
(MCKB byte 2, bit 4)

Indicates a logical error in either the read or write generator. The check is performed only during the read cycle of a storage operation.

ECC DBL Bit (Double Bit)
(MCKB byte 2 bit 5)

Indicates a double-bit data error detected by the ECC logic, a double-bit error in the check bit logic, or a double-bit error involving a data bit and a check bit.

ECC Busy
(MCKB byte 2 bit 3)

Indicates that the storage ECC logic received a select but did not cycle.

I-Cycle Hdw
(MCKB byte 2 bit 0)

Indicates that a parity-check error in I BFR 0, I BFR 1, Op Regs, IMM byte Regs, or that an ALU half sum error, has occurred. To determine which error has occurred, display EXP LS 56 bytes 2 and 3. (For bit definition, see EXP LS map in Chapter 2).

ADR X-Late LRU Inval (Address Translate LRU Invalid)
(MCKB byte 1 bit 6)

Indicates that the LRU array has degraded (because of hardware failures) to a point where the LRU does not function. It also turns on when the LRU address indicated by an invalid LRU position is accessed again.

Adr X-Late Mult Match (Address Translate Multiple Match)
(MCKB byte 1 bit 5)

Indicates that the array of real addresses had more than one match with the logical address used as its source. (The input matches more than one output.)

Adr X-Late No Match (Address Translate No Match)
(MCKB byte 1 bit 4)

Indicates that the array of real addresses does not match the logical address that was used as its source. (The input does not match the value in the address-array registers.)

Power

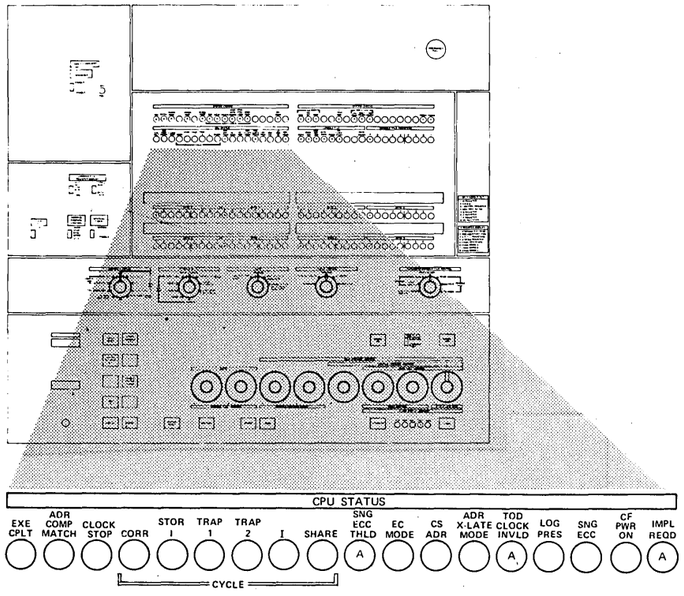
This indicator turns on whenever any of the following conditions exist in any frame.

1. Power incomplete to the system.
2. CB trip.
3. Open thermal.
4. MG check.
5. CE2 switch set to ERROR OVERRIDE.
6. CE5 switch set to either I/O HOLD or I/O OFF.

THERM (Thermal)

Indicates that an overtemperature switch has opened. A power-off sequence is initiated, and the power check indicator is turned on. The THERM indicator stays lit until the over-temperature condition no longer exists.

CPU STATUS



EXE CPLT (Execution Complete)

Indicates that a CPU stopped state (STOP word recycling) was reached as a result of:

1. Pressing STOP.
2. Instruction-step mode.
3. Obtaining a match with the address-compare control switch in stop-mode operation.

EXE CPLT turns off when START is pressed.

ADR COMP MATCH (Address Compare)

The signal that turns on this indicator also provides a sync pulse. The signal activates when a compare address from switches CDEFGH (or EFGH for control storage) matches the contents of the M-register for an address in control or main storage.

Clock Stop

Indicates that the CPU is in a hard stop.

Corr Cycle (Correction Cycle)

Indicates that the CPU is performing a dummy cycle because of an ECC check in control storage.

Stor-1 Cycle (Storage-1 Cycle)

Indicates that the CPU clock is stopped between the first and second cycles of a storage word. Manual operations using the store or display keys are not allowed when STOR 1 CYCLE is on.

Trap-1 Cycle

Indicates that the CPU clock is stopped between the first and second cycles of a trap sequence. Manual operations using the store or display keys are not allowed when TRAP 1 CYCLE is on. If a trap occurs during execution of a control word, the M-register is set to the trap address value. The N-register is not changed.

Trap-2 Cycle

Indicates that the first word of a trap routine has been executed.

I-Cycle

Indicates that hardware I-cycles are in use (DFXX).

Share Cycle

Indicates that the CPU is servicing a selector-channel share cycle. The indicator is on from the beginning of SHARE-1 through the end of SHARE-2. To determine which channel, display upper roller position 1 (byte 1, bit 0-3).

SNG ECC THLD (Single ECC Threshold)

Indicates that single ECC checks are occurring 3% of the time, while control storage is being accessed.

CS ADR (Control-Storage Address)

Indicates that control storage is being accessed. The actual address in M2 and M3 registers.

Addr X-Late Mode (Address Translate Mode)

Indicates that LEX (local execute) mode is set (EXP LS location 53 byte 1, bit 3) for OS/DOS compatibility.

TOD Clock Invld (Time-of-Day Clock Invalid)

Indicates that the TOD clock has stopped or missed a time increment. The indicator is turned off when a set clock instruction causes the clock to be set.

LOG Pres (Present)

Indicates that a log is present in the log area of main storage (location 232-511). Run SEREP to obtain edited printout.

SNG ECC (Single ECC)

Indicates that one of the 72 bits from storage has been corrected. This could have been one of eight check bits or one of 64 data bits (active only when in record or threshold mode).

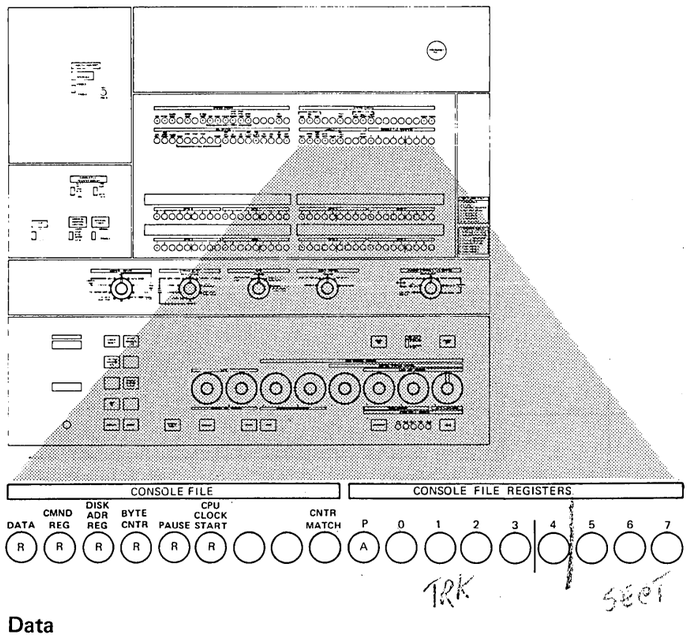
CF PWR On (Console-File Power On)

Indicates that power is applied to the console file. (Except when DIAGNOSTIC/CONSOLE FILE CONTROL is in CE mode.)

IMPL Reqd (IMPL Required)

Indicates that control storage may have to be reloaded because the system-reset routine did not go to successful completion. The indicator is turned on early in the system-reset routine, and turned off after successful completion of the internal diagnostic.

CONSOLE FILE



Data

The four conditions that can turn on this indicator are:

1. Failure to find both start bit and data latch on in a byte.
2. Odd/even byte-count check.
3. An even number of bits in the console-file shift register.
4. A data bit was read when not expected.

Cmnd Reg (Command Register)

Indicates that incorrect parity was detected in the console-file command register.

Disk Adr Reg (Disk Address Register)

Indicates incorrect parity in the console-file disk-address register.

Byte Cntr (Byte Counter)

Indicates that the console file tried to read beyond the end of a sector.

Pause

Indicates that a file-pause operation did not end before the next command byte was read.

CPU Clock Start

Indicates that the CPU clock failed to start when instructed by the console file.

Cntr Match (Counter Match)

Indicates a match between the value set into switches C and D and the byte counter. The sector address is set into switches A and B.

CONSOLE-FILE REGISTERS

Indicate the bit values contained in either the console-file disk-address register, the console-file command register, the console-file shift register, or the console-file byte counter. The display is controlled by the console-file register-display toggle switches. See "CONSOLE FILE REGISTER DISPLAY" toggle switch.

Note: The shift register is displayed through the command register. Always display the command register first or command data will be lost.

ROLLER INDICATORS

DPLY ASMBLR OUT (DISPLAY ASSEMBLER)-UPPER ROLLER

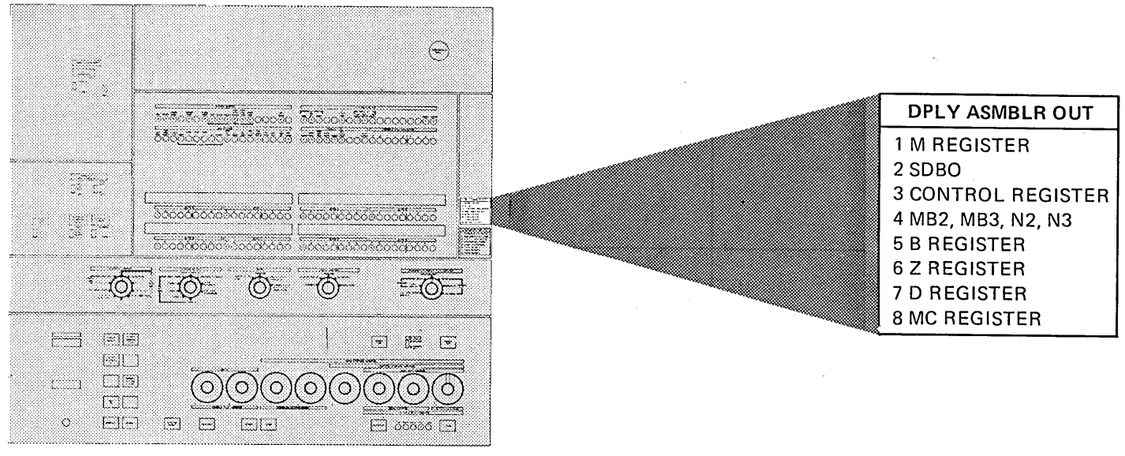
Displays major machine registers in thirty-six console indicators.

The register displayed is gated to the indicators by the roller switch setting.

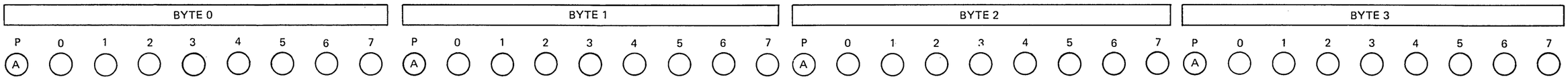
CPU clocks must be stopped for indicators to be valid.

The storage select switch must be in the main-storage position to display positions 4 and 8.

The storage-protect stack is displayed only if a storage word is executed from the switches, or the DK (displayed key) is initiated on the console writer.



Logic Page Reference Chart



Position 1. M Register

STOR PROT STK	STORAGE PROTECT STACK							SEL CHAN	SELECTOR CHANNEL				M REG	BYTE 1 (P & 4-7)				M REG	M-REGISTER BYTE 2								P	M-REGISTER BYTE 3								
	STORAGE KEY				FETCH PROT	REFER BIT	CHNG BIT		1	2	3	4	SAR					SAR	STOR 1 CYCLE OFF→M-REG = NEXT MICRO INSTR ADR; STOR 1 CYCLE ON→M-REG = DATA ADR								P	-5	4	3	2	1	U-L	BYTE SELECT	
	8	4	2	1									17	16	15	14		13	12	11	10	9	8	7	6									
MS201	MS201	MS201	MS201	MS201	MS201	MS201	MS201	MC017	GA111	GA111	GA111	GA111	MC017	MC017	MC017	MC017		RM033	RM031	RM031	RM031	RM031	RM032	RM032	RM032	RO032	RM041	RM043	RM044	RM043	RM044	RM215	RM215	RM052	RM052

Note: Parity bit is only associated with M-Reg bit 4-7.

Position 2. SDBO

SDBO	STORAGE DATA BUS OUT BYTE 0								P	STORAGE DATA BUS OUT BYTE 1								P	STORAGE DATA BUS OUT BYTE 2								P	STORAGE DATA BUS OUT BYTE 3							
	NOTE: IF CLK STOP-NORMAL MODE SDBO = LAST CTRL WD EXE OR DATA ACCESSED FROM STOR FOR WD TYPES 4-7 IF STOR/DISPLAY-MANUAL MODE SDBO DATA ACCESSED FROM MAIN OR CONTROL STORAGE																																		
RC113	RC123	RC123	RC143	RC153	RC163	RC173	RC183	RC193	RC113	RC123	RC133	RX143	RC153	RC163	RC173	RC183	RC193	RC113	RC123	RC133	RC143	RC153	RC163	RC173	RC183	RC193	RC113	RC123	RC133	RC143	RC153	RC163	RC173	RC183	RC193

Position 3. Control Register

C REG	MICROPROGRAM CONTROL REG BYTE 0				P	MICROPROGRAM CONTROL REG BYTE 1								P	MICROPROGRAM CONTROL REG BYTE 2								P	MICROPROGRAM CONTROL REG BYTE 3							
	WORD TYPE 0	1	2	3											NOTE: IF STOR 1 CYCLE OFF C REG = NEXT CONTROL WORD IF STOR 1 CYCLE ON C REG = CURRENT CONTROL WORD																	
RC111	RC121	RC131	RC141	RC151	RC161	RC171	RC181	RC191	RC111	RC131	RC131	RC141	RC151	RC161	RC171	RC181	RC191	RC111	RC121	RC131	RC141	RC151	RC161	RC171	RC181	RC191	RC111	RC121	RC131	RC141	RC151	RC161	RC171	RC181	RC191

Position 4. MB 2, MB 3, N2, N3

MB REG	STORAGE BACKUP ADR REG BYTE 2								P	STORAGE BACKUP ADR REG BYTE 3								N REG	NEXT CONTROL WORD ADR REG BYTE 2								P	NEXT CONTROL WORD ADR REG BYTE 3							
	NOTE: IF NOT IN STOR † CYCLE MB REG = ADDRESS OF LAST CONTROL WORD EXECUTED IF IN STOR 2 CYCLE (AFTER STOR 1 CYCLE ON) MB REG = BYTE 2 and 3 OF DATA ADDRESS																		NOTE: IF TRAP 1 CYCLE OFF N REG = NEXT CONTROL WORD ADDRESS IF TRAP 1 CYCLE ON N REG = LINK ADDRESS FOR TRAP																	
RM035	RM035	RM035	RM035	RM036	RM035	RM035	RM035	RM035	RM073	RM073	RM073	RM073	RM073	RM051	RM051	RM052	RM052	RM033	RM033	RM033	RM033	RM034	RM034	RM034	RM034	RM034	RM045	RM045	RM045	RM045	RM045	RM051	RM051	RM052	RM052

Position 5. B-Register

B REG	B REGISTER BYTE 0								P	B REGISTER BYTE 1								P	B REGISTER BYTE 2								P	B REGISTER BYTE 3							
																			NOTE: B-REGISTER BYTES 0-1 = INPUT TO Z REG BYTES 0-1 B-REGISTER BYTES 0-3 = B SOURCE INPUT TO ALU/OUTPUT FROM B LOCAL STORAGE																	
RA113	RA113	RA113	RA114	RA114	RA114	RA123	RA123	RA123	RA124	RA124	RA142	RA133	RA133	RA133	RA134	RA134	RA134	RA143	RA143	RA143	RA144	RA144	RA144	RA153	RA153	RA153	RA154	RA154	RA154	RA163	RA163	RA163	RA164	TA164	RA164

Position 6. Z-Register

Z REG	Z REGISTER BYTE 0								P	Z REGISTER BYTE 1								P	Z REGISTER BYTE 2								P	Z REGISTER BYTE 3							
																			NOTE: Z-REG BYTES 0-3 = OUTPUT OF ALU IN PRESENT CYCLE Z-REG BYTES 0-3 = INPUT TO SPTL REGS AND D REG																	
AL147	AL144	AL144	AL144	AL144	AL134	AL134	AL134	AL134	AL127	AL124	AL124	AL124	AL124	AL114	AL114	AL114	AL114	AL137	AL144	AL144	AL144	AL144	AL134	AL134	AL134	AL134	AL117	AL124	AL124	AL124	AL124	AL114	AL114	AL114	AL114

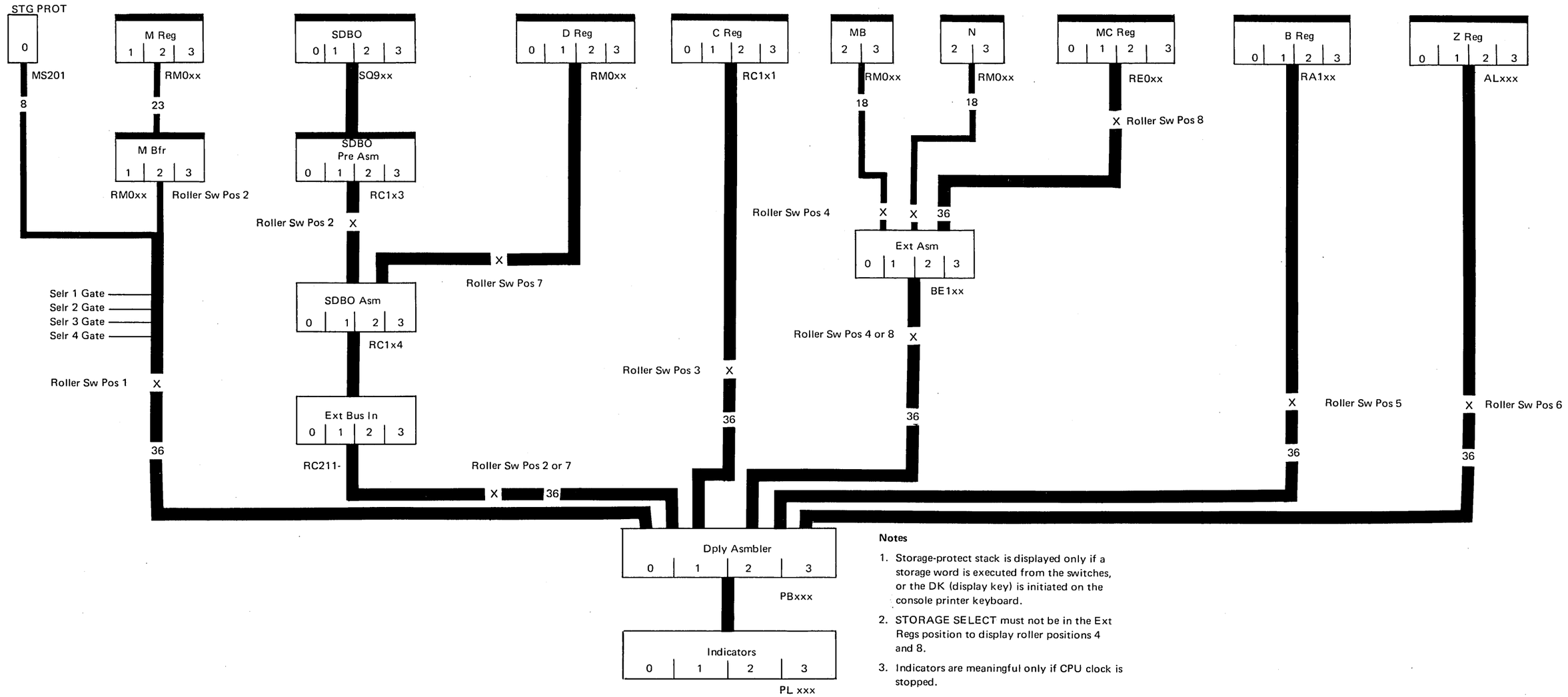
Position 7. D-Register

D REG	D REGISTER BYTE 0								P	D REGISTER BYTE 1								P	D REGISTER BYTE 2								P	D REGISTER BYTE 3							
																			NOTE: D REGISTER BYTES 0-3 = OUTPUT FROM Z REGISTER IN LSST CYCLE																	
RA115	RA115	RA115	RA115	RA116	RA116	RA125	RA125	RA125	RA125	RA126	RA126	RA135	RA135	RA135	RA135	RA136	RA136	RA145	RA145	RA145	RA145	RA146	RA146	RA155	RA155	RA155	RA155	RA156	RA156	RA165	RA165	RA165	RA165	RA166	RA166

Position 8. MC Register (MCKA External Address 06)

MCKA	LS A SORC ADR	LS B SORC ADR	LS A DEST ADR	LS B DEST ADR	DEST BYTE CTRL	LA A,B DEST ADR	LS CTRL ASM	C REG PTY	P	ACB REG PTY	COMP LS COMP	FLUSH THRU	H REG PTY		P REG PTY	T REG PTY	L REG PTY	P	ALU 2 HALF SUM	ALU 2 HALF SUM	ALU LOGL	B REG SHIFT	A REG SHIFT	B REG SHIFT	Z REG PTY	D REG PTY	P	EXT DEST X	EXT ESDT Y	TXT SORC Y	EXT CTRL AMS	INTV TIMER PTY	S REG DUP	TOD CLOCK	CS ADR
RE022	RE021	RE021	RE021	RE021	RE022	RE021	RE021	RE021	RE022	RE023	RE023	RE023	RE023		RE024	RE024	RE024	RE036	RE031	RE031	RE032	RE031	RE033	RE033	RE033	RE033	RE036	RE035	RE035	RE035	RE035	RE034	RE034	RE034	RE034

Display Assembler Out Data Flow



Notes

1. Storage-protect stack is displayed only if a storage word is executed from the switches, or the DK (display key) is initiated on the console printer keyboard.
2. STORAGE SELECT must not be in the Ext Regs position to display roller positions 4 and 8.
3. Indicators are meaningful only if CPU clock is stopped.

REMEMBER

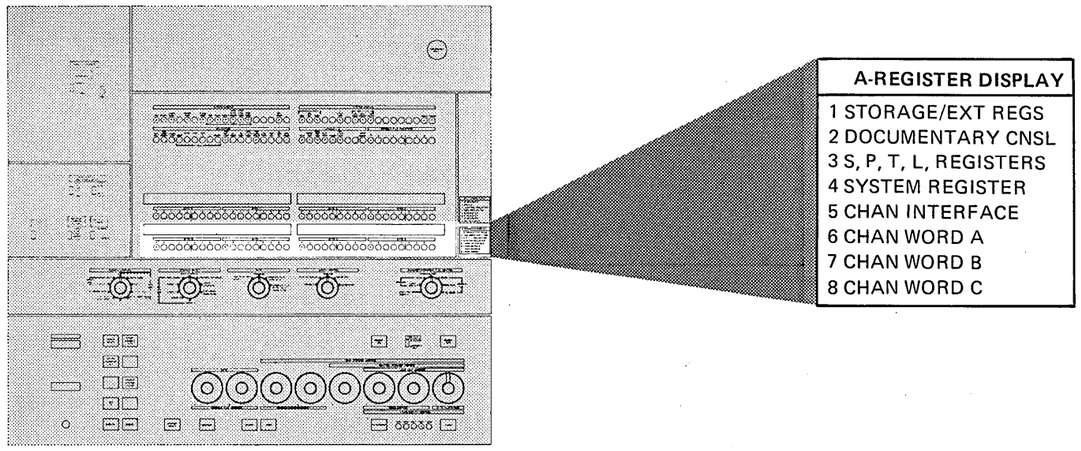
There is a Reader's Comment Form at the back of this publication.

A-REGISTER DISPLAY—LOWER ROLLER

Continually displays the contents of the A-Register.

When the CPU clock is stopped, data can be gated to the A-Register for display.

The data to be displayed in the A-Register is gated by the roller switch, the storage select switch, and the store/display address switches.



Logic Page Reference Chart

Notes:

1 Line 1 = A-Register
Line 2 = Local Storage
Line 3 = Expanded local storage
Line 4 = External assembler

2 Line 1 = IFA
Line 2 = Selector channel
Line 3 = Multiplexer channel

3 Line 1 = IFA
Line 2 = Selector channel

4 Selector Channel 1 is shown
Channel 2 use GF-GG logics
Channel 3 use GK-GL logics
Channel 4 use GP-GQ logics

BYTE 0									BYTE 1								BYTE 2								BYTE 3							
P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	7	P	0	1	2	3	4	5	6	◯

Position 1. Storage/Ext Regs (See Note 1)

A REG	A REGISTER BYTE 0									P	A REGISTER BYTE 1									P	A REGISTER BYTE 2									P	A REGISTER BYTE 3								
	INSTRUCTION COUNTER DISPLAY																				NOTE: IF CLOCK STOPPED NORMAL MODE A-REG = SOURCE INPUT TO ALU IF STORE/DISPLAY MANUAL MODE A-REG = EXT DATA/EXPANDED OR LOCAL STOR REGS																				
RA111	RA111	RA111	RA112	RA112	RA112	RA121	RA121	RA121	RA122	RA122	RA122	RA131	RA131	RA131	RA132	RA132	RA132	RA141	RA141	RA141	RA142	RA142	RA142	RA151	RA151	RA151	RA152	RA152	RA152	RA161	RA161	RA161	RA162	RA162	RA162					
LA312	LA312	LA313	LA313	LA314	LA314	LA315	LA315	LA316	LA312	LA312	LA313	LA313	LA314	LA314	LA315	LA315	LA316	LA322	LA322	LA323	LA323	LA324	LA324	LA325	LA325	LA326	LA322	LA322	LA323	LA323	LA324	LA324	LA325	LA325	LA326					
RV115	RV114	RV114	RV114	RV114	RV124	RV124	RV124	RV124	RV125	RV215	RV215	RV215	RV215	RV225	RV225	RV225	RV225	RV115	RV235	RV235	RV235	RV235	RV245	RV245	RV245	RV245	RV125	RV255	RV255	RV255	RV255	RV315	RV315	RV315	RV315					
BE112	BE112	BE113	BE113	BE114	BE114	BE122	BE122	BE123	BE132	BE132	BE133	BE123	BE124	BE124	BE133	BE134	BE134	BE142	BE142	BE143	BE143	BE144	BE144	BE152	BE152	BE153	BE162	BE162	BE163	BE153	BE154	BE154	BE163	BE164	BE164					

Position 2. Documentary CNSL

3210 TI / 3215	00:LC / 11:UC		B	A	8	4	2	1
	0	1	2	3	4	5	6	7
	PD025	PD025	PD025	PD025	PD025	PD025	PD025	PD025	PD025

3210 TA / 3215	RD LTH	WRITE LTH	STKD RESET	SHARE LTH	ATTN RESET	SENSE LTH	3215 DIAG RESET	ALARM
	PD041	PD041	PD041	PD041	PD041	PD041	PD041	PD041	PD041

3210 TT / 3215	ATTN	OPL	A/D MODE	TIME OUT	UPPER CASE	END 00:NO CK 10:TIME OUT	REQ-I	CNCL
	NOTE: IF 3215, BIT 3 & 4 = CKS →					01:DATA CK 11:SYNC CK		
	PD043	PD025	PD025	PD043	PD043	PD043	PD043	PD043	PD043

3210 TE / 3215	T1	T2	R1	R2	R2A	R5	CYC CLU	UC
	10:WIRE 11:GATE	FIRE LINES	PE TIME	PE ENV	NEW LINE	ZERO COUNT	FDBK EMIT	INHIB PRINT
	PD042	PD042	PD042	PD042	PD042	PD042	PD042	PD042	PD042

Position 3. S,P,T,L Registers

S REG	TRUE CPMT	BIT 1 CARRY	Z = 0	BIT 0 CARRY	Z HI ZERO	Z LO ZERO	GENERAL PURPOSE		P REG	LOCAL STORAGE OR EXTERNAL ADDRESSING: A EXP LS	P HIGH			B EXP LS	P LOW			T REG	T HIGH: 0	1	2	3	T A: 2	1	T B: 2	1	L REG	LOCAL STORAGE INDIRECT ADDRESSING: L HIGH				L LOW			
RS116	RS115	RS115	RS124	RS115	RS214	RS214	RS214	RS214	RP011	RP011	RP011	RP011	RP011	RP012	RP012	RP012	RP012	RP014	RT012	RT012	RT012	RT012	RT013	RT013	RT013	RT013	RL011	RL011	RL011	RL011	RL011	RL011	RL011	RL011	RL011

Position 4. System Register

SYS REG 0	MACH IRPT PEND	RETRY RTNE	MACH CHK MARK	3210 MOD 2	LOG PRES	SUB BLOCK MODE	SELR I/O OP	LS CS MODE	SYS REG 1	ADR CNTS	CPU IRPT FORCE	SAR IRPT FORCE	PSW RST		SYS CTRL IRPT	TIMER IRPT FORCE		SYS REG 2	SYS CLEAR	IMPL	CNSL FILE WAIT	CE MODE	01:PWR RST 10:SUB LOAD 11:SYS LOAD		INST PROC	H REG	MACH CHECK	RETRY	CPU HIGH	NO IFA: SX1 IFA	SX4 SX2,3	MPX	IFA	DOCM A/D
RS011	RS011	RS011	RS011	RS011	RS011	RS011	RS011	RS011	RS012	RS012	RS012	RS012	RS012	N/A	RS012	RS012	N/A	RS013	RS013	RS013	RS013	KM012	RS013	RS013	N/A	RS013	RH025	RH022	RH022	RH022	RH022	RH022	RH022	RH023	RH023

Position 5. Chan Interface, See Notes 2 and 4 (Page 5-12)

FTO	CYL	HEAD	DIFF	CTRL	CUA LD	SPF TAGS			FTI	128	64	32	16	8	4	2	1	FBO	FILE BUS-OUT								FDR	FILE DATA REGISTER							
GTO	OP-O	SEL-)	ADR-O	CMD-)	SERV-O	DATA-O	SUP-O	TAGS	GTI	OP-I	ADR-I	STAT-I	SERV-I	SEL-I	DATA-I	REQ-I	DISC-I	GO	SELECTOR CHANNEL BUS OUT								GR	SELECTOR CHANNEL DATA REGISTER							
MTO	OP-O	SEL-O	ADR-O	CMD-O	SERV-O	IRPT	SUP-O	OUT	MTI	OP-I	ADR-I	STAT-I	SERV-I	SEL-I	REQ-I	TRAP	DISC-I	MBI	MULTIPLEX CHANNEL BUS IN								MBO	MULTIPLEX CHANNEL BUS-OUT							
JK114	JK112	JK112	JK112	JK112	JK112	JK112	JK112	JK112	JK214	WF101	WF101	WF101	WF101	WF101	WF101	WF101	WF101	JK311	JK311	JK311	JK311	JK311	JK311	JK311	JK311	JK311	JK412	JK411	JK411	JK411	JK411	JK411	JK411	JK412	JK412
GB612	GB313	GB213	GB312	GB312	GB113	GB113	GB212	GB612	GB622	GB621	GB621	GB621	GB621	GB621	GB621	GB621	GB621	GB082	GB082	GB082	GB082	GB082	GB082	GB082	GB082	GB082	GB611	GB611	GB611	GB611	GB611	GB611	GB611	GB611	GB611
FA013	FA013	FA013	FA013	FA013	FA013	FA013	FA013	FA013	FA012	FA012	FA131	FA131	FA131	FA141	FA141	FA012	FA141	FA111	FA111	FA111	FA111	FA121	FA121	FA121	FA121	FA131	FA014	FA014	FA014	FA014	FA014	FA014	FA014	FA014	FA014

Position 6. Chan Word A, See Notes 3 and 4 (Page 5-12)

FFL GE	CD	CC	SLI	SKIP	ALLOW HALT	IN IN FWD	CS CT READY IN BWD	OUT	FCS GE	PCI	IL	PROG CHK	PROT CHK	DATA CHK	CTRL CHK	INTF CHK	CHAIN CHK	FST GS	CHAN BUSY	IRPT LTH	CUE POLL CTRL	CUB PRIME	DCC MODE	CMD RETRY	GATE A	GATE B	FGL GL	COUNT READY	COUNT ZERO	GR FULL	IRPT CND	SHARE ERROR		POSITION CODE	
JK114	JK111	JK111	JK111	JK111	N/A	JK512	JK512	JK512	JK214	JK111	JK513	JK212	JK212	JK212	JK212	JK212	N/A	JK314	JK313	JK313	JK313	JK313	N/A	N/A	JK313	JK313	JK414	JK512	JK512	JK511	JK416	JK513	N/A	JK612	JK612
GB611	GB712	GB712	GB712	GB712	GB313	GB412	GB412	GB412	GB621	GB721	GB411	GB411	GB411	GB411	GB412	GB114	GB513	GB631	GB213	GB213	GB313	GB211	GB213	GB513	GB633	GB643	GB641	GB413	GB413	GB311	GB413	GB412	N/A	GB711	GB711

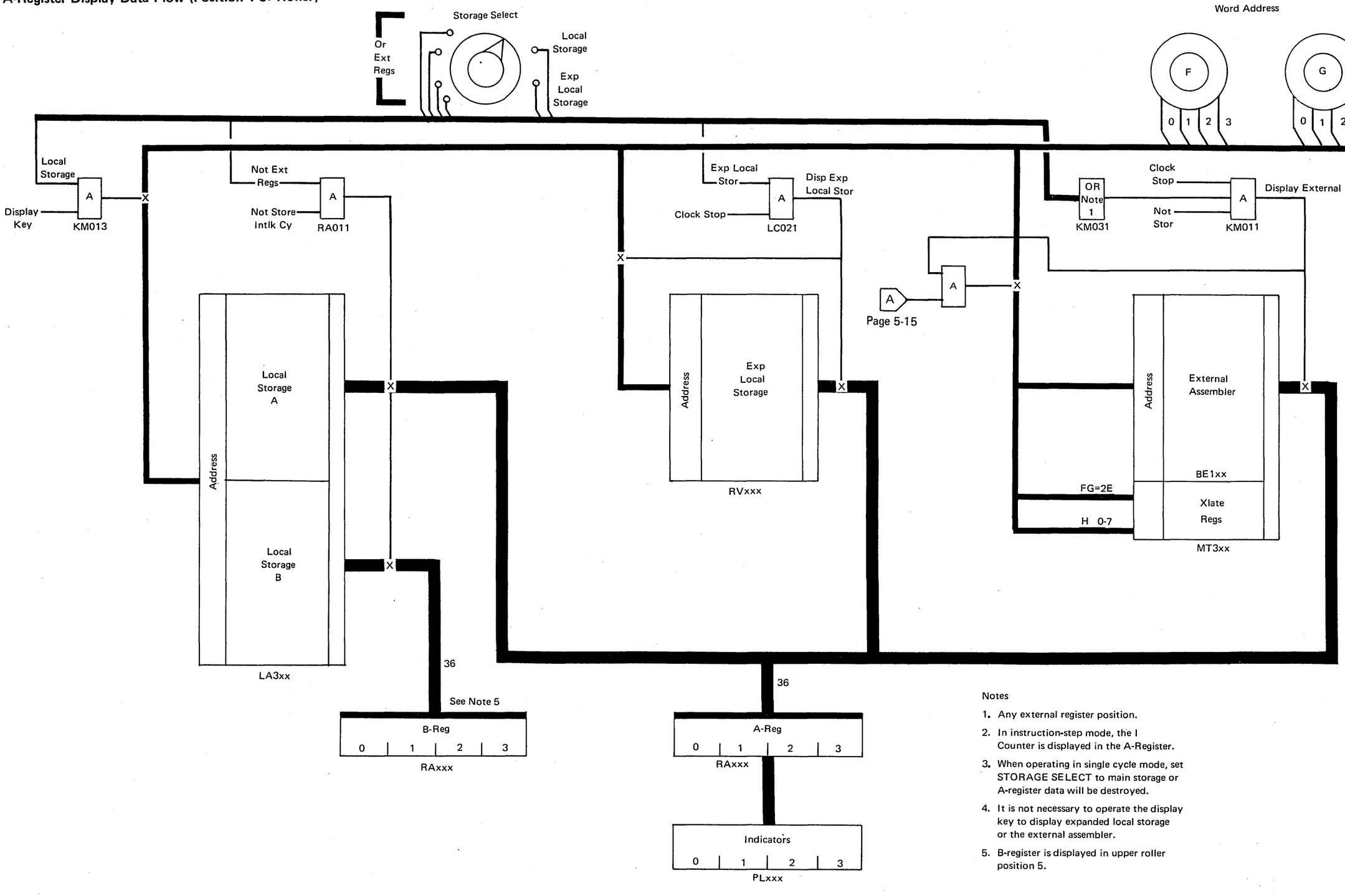
Position 7. Chan Word B, See Notes 3 and 4 (Page 5-12)

FDS		DISK STATUS: ON LINE	UNSAFE		PACK CHNG	EOC	MULTI MOD	SEEK ICPTL	FHC	HEAD CYLINDER SWITCHES: 128	64	32	16	8	4	2	1	FED	CE INLINE DISPLAY REGISTER									FMOD	MODULE SELECTED: 0	1	2	3	SELECTED MODULE: 4	5	6	7
	BUSY								GBF		BF-6	BF-5	BF-4	BF-3	BF-2	BF-1	BF-0	GCT	BFR CTL CK		CD REQ	BFR PTN	GCL: 0	1	2	3			GDL-0	GDL-1	GBP	0	1	2	3
JK114	WF101	WF101	WF101	N/A	WF101	WF101	JK511	WF101	JK214	WF102	WF102	WF102	WF102	WF102	WF102	WF102	WF102	JK314	JK312	JK312	JK312	JK312	JK312	JK312	JK312	JK312	JK414	JK611	JK611	JK611	JK611	JL511	JK511	JK511	JK511
N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	GC621	N/A	GC311	GC311	GC311	GC311	GC312	GC312	GC312	GC632	GC513	N/A	GC413	GC412	GC412	GC412	GC412	GC412	N/A	N/A	GC414	GC414	N/A	GC411	GC411	GC411	GC411

Position 8. Chan Word C, See Notes 3 and 4 (Page 5-12)

									FCH	COUNTER HIGH: 32K	16K	8K	4K	2K	1K	512	256	FCL	COUNTER LOW: 128	64	32	16	8	4	2	1	FOP	ADR: READ	WRITE	MARK	SRCH	INDEX: SCAN	START	FORMAT	SKIP
GB-0		SEL CHAN BUFFER BYTE 0							GB-1									GB-2	SEL CHAN BUFFER, BYTE 2								GB-3	SEL CHAN BUFFER, BYTE 3							
N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	JK211	JK211	JK211	JK211	JK211	JK211	JK211	JK211	JK211	JK311	JK311	JK311	JK311	JK311	JK311	JK311	JK311	JK311	JK412	JK411	JK411	JK411	JK411	JK411	JK411	JK412	JK412
GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC115	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114	GC114

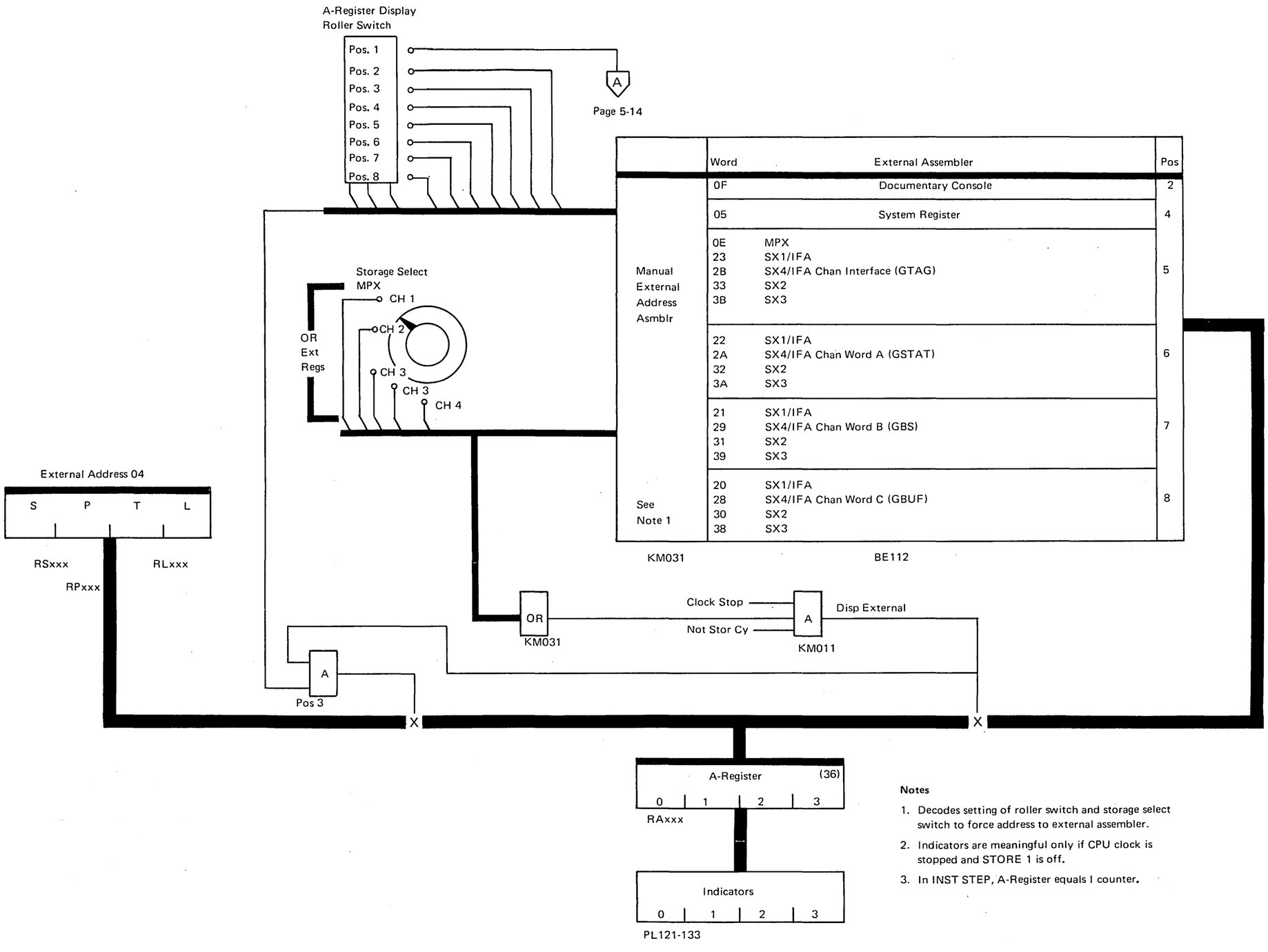
A-Register Display Data Flow (Position 1 of Roller)



Notes

1. Any external register position.
2. In instruction-step mode, the I Counter is displayed in the A-Register.
3. When operating in single cycle mode, set STORAGE SELECT to main storage or A-register data will be destroyed.
4. It is not necessary to operate the display key to display expanded local storage or the external assembler.
5. B-register is displayed in upper roller position 5.

A-Register Display Data Flow Positions 2-8 of Roller

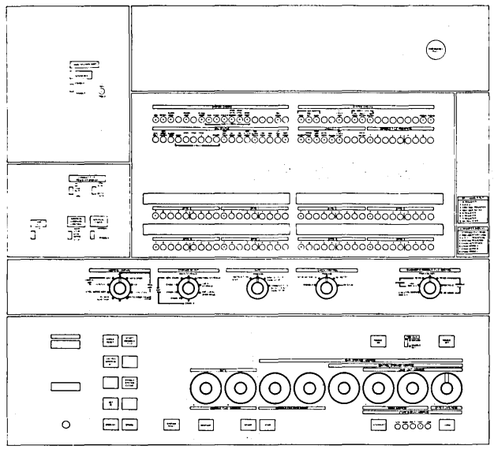


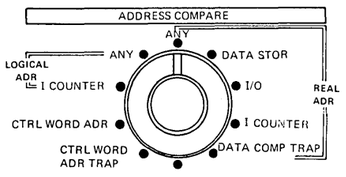
Notes

1. Decodes setting of roller switch and storage select switch to force address to external assembler.
2. Indicators are meaningful only if CPU clock is stopped and STORE 1 is off.
3. In INST STEP, A-Register equals I counter.

ROTARY CONTROL SWITCHES

ADDRESS COMPARE





- Works in conjunction with address compare control, storage select, and storage address switches C - H to provide control of CPU operations for diagnostic purposes.
- ADDRESS COMPARE selects the type of storage operation for which the match may occur and where the compare will be performed. (M-Register contents of Pre-Address Assembler.)
- ADDRESS COMPARE CONTROL determines how the operation will terminate after a match occurs. HARD STOP: Stops at the end of a micro-step.
  STOP: Stops at the end of E-phase of the current instruction.
  SYNC/NORM: Continues processing and provides a sync signal.
- STORAGE SELECT chooses either main or control storage to match against.
- STORAGE ADDRESS switches C-H (E-H for control storage) are set to the address to be matched.

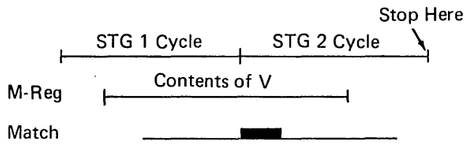
Real Adr Positions

Address match can occur only during a storage word (word types 4-7). The compare is made between the M-Reg contents and the address switches C-H (E-H for control storage).

EXAMPLE

Example: Main Storage Match

C090 RDW R V+4
Read a word into local storage
Location R using local storage
Location V to address main storage.



Note: These positions can not be used to match on a control-word address unless the control-storage address is fetched during the storage 1 cycle.

Any (Real Address)

A compare between the M-register contents and the value of address switches C-H is made on any storage-word operation to either main or control storage. This position does not guarantee a match on an instruction address. Use the I-counter position to match on an instruction address.

Data Stor (Real Address)

A compare between the M-register contents and the value of address switches C-H is made only when data is being stored into main or control storage.

I/O (Real Address)

A compare between the M-register contents and the value of address switches C-H is made during a share cycle or an I/O operation. I/O operations are determined by H-register bits.

I-Counter (Real Address)

A compare between the M-register contents and the value of address switches C-H is made when the I-register (Exp LS loc. 50) is used to access storage.

Note: In order for this position to be effective, I-CYCLES PREFETCH must be blocked. This is accomplished by having the address compare control switch set to either STOP or HARD STOP. BLOCKING PREFETCH causes the I-buffers to be reloaded each time the microprogram returns to I-Cycles (return to microword DF0C). If PREFETCH was not blocked (address compare control switch set on SYNC/NORM) an address match could occur only on a doubleword boundary.

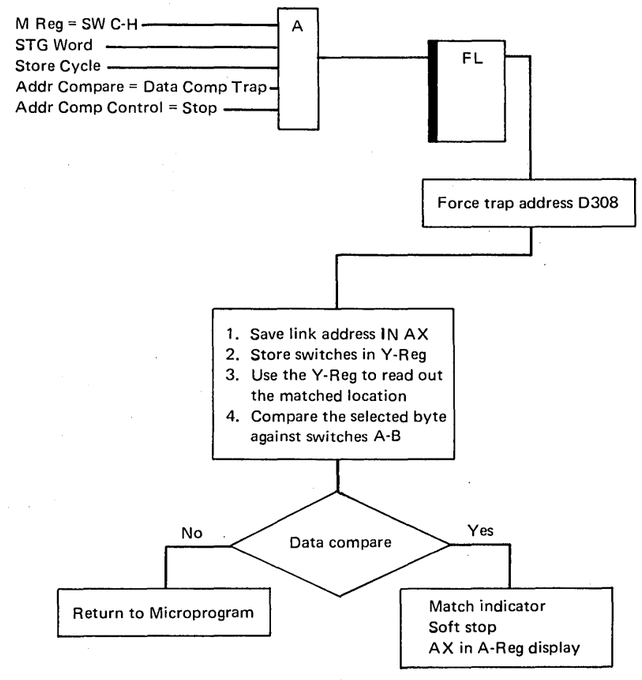
Service Aid:

1. If blocking of PREFETCH is desired as a troubleshooting aid, set ADDRESS COMPARE to I-COUNTER, ADDRESS COMPARE CONTROL to STOP or HARD STOP, and set an address that is not used by the program into address switches C-H. A stop will not occur, but PREFETCH will be disabled.
2. To block PREFETCH and obtain an address sync, add a jumper on B-gate B3G2D04 to B3V3B08 (PM013) and set ADDRESS COMPARE CONTROL to SYNC/NORM.

Data Comp Trap (Real Address)

Compares the M-register against address switches C-H during a store operation (main storage only). If compare is equal, a trap is taken to D308 and DATA switches A,B are compared against the data just stored. If the data compares, the address match indicator is turned on and the machine soft-stops. The address of the control word that would have been executed next had the data trap not occurred is displayed in the A-register display roller switch indicators.

To determine the address of the instruction that modified the storage byte, display the current PSW (refer to "Console PR-KB Manual Operations") and subtract the current instruction length code from the instruction address in the current PSW. The instruction found with this procedure may not have modified the data. An I/O data trap occurring during execution of this instruction could have modified the data. To determine which I/O data trap modified the data, lookup the address displayed in the A-register display roller switch indicators in the module charts in the back of the microlisting.



Service Aid:
By altering the microprogram in the GKCC routine, it is possible to stop when the data mismatches.

Example: Exchange the words for the Z0 branch in the following microinstruction.

XXXX, YYYY match Z0 RDW Y CM F8

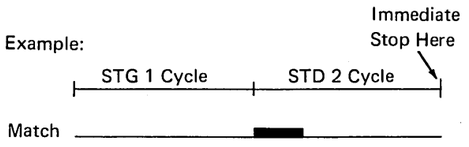
Using the microlisting, find the location of the statements xxxx (match Z0) and yyyy (match Z1). Alter the statement at yyyy to the word at xxxx. Then alter the statement at xxxx to the word at yyyy. Now, when the trap is executed and the compare is not equal, a stop will occur.

Logical Addr Positions

Address match can occur only during a storage word (word types 4-7). The compare is made between the address switches C-H (E-H for control storage), and the address gated to the pre-address assembler. This is the logical address before address adjustment occurs. Logical address positions apply only to instruction and operand addresses; not addresses used by channel operations.

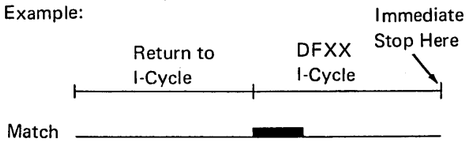
Any (Logical Address)

Compare between PAA and the value of address switches C-H is made when any register is gated to the PAA. This position does not guarantee a match on an instruction address. Use I-COUNTER position to match on an instruction address.



I-Counter (Logical Address)

Compare is made between the I-register and the value of address switches C-H. The match occurs when the microprogram returns to I-cycles to execute the matched I-counter value. The first word of I-cycles could be any DF address depending upon the kind of instruction and the prefetch conditions. The address does not have to be on a doubleword boundary, and therefore prefetch is not blocked in this position.



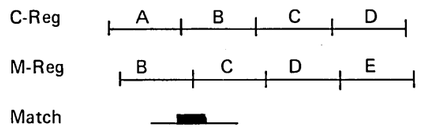
Ctrl Word Adr

Address match can occur only on a control-storage word. The compare is made between the contents of the M-register and address switches E-H. The position of the address compare control switch determines when the CPU cycle will be stopped.

Examples:

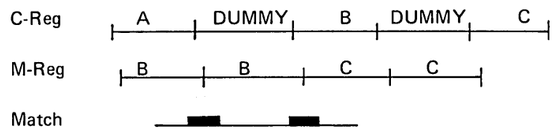
1. SYNC/NORM

Using microwords A, B, C, D, match on address of word B.



Sync pulse occurs at 0-45 time of word B.

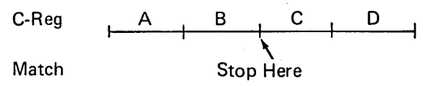
When in local storage control storage mode, two sync pulses occur because of the dummy cycle.



Use the second pulse for scope sync.

2. HARD STOP

Using microwords A, B, C, D, match on address of word B. Stop occurs after match word is executed.



If HARD STOP is used while in LSCS Mode, the stop occurs before the word is executed. The status of the roller indicators after a hard stop is:

Upper Roller
1 M-Reg = Address of word C
2 SDBO = Microword just executed (B)
3 C-Reg = Microword C
4 MB,N = MB: Addr of word B
N: Addr of word C
5 B-Reg = B: source of word B
6 Z-Reg = ALU: results of word B
7 D-Reg = ALU: results of word A

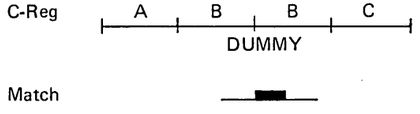
Lower Roller
1 A-Reg = A-source of word B

Display External Address 1A byte 2-3 (M-RTY) for address of word A.

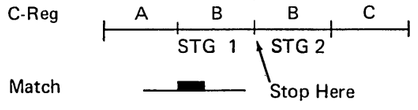
3. STOP

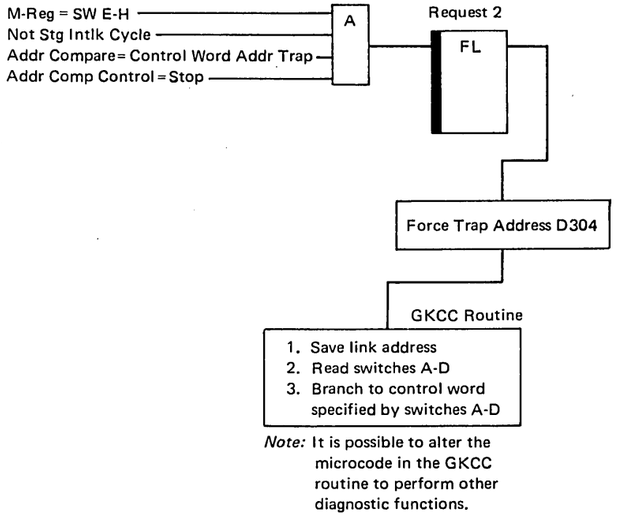
Performing an RX Add instruction. Machine soft-stops after the current instruction is executed.

If an ECC error occurs, the first B word becomes a dummy cycle.



If the microword is a storage word.



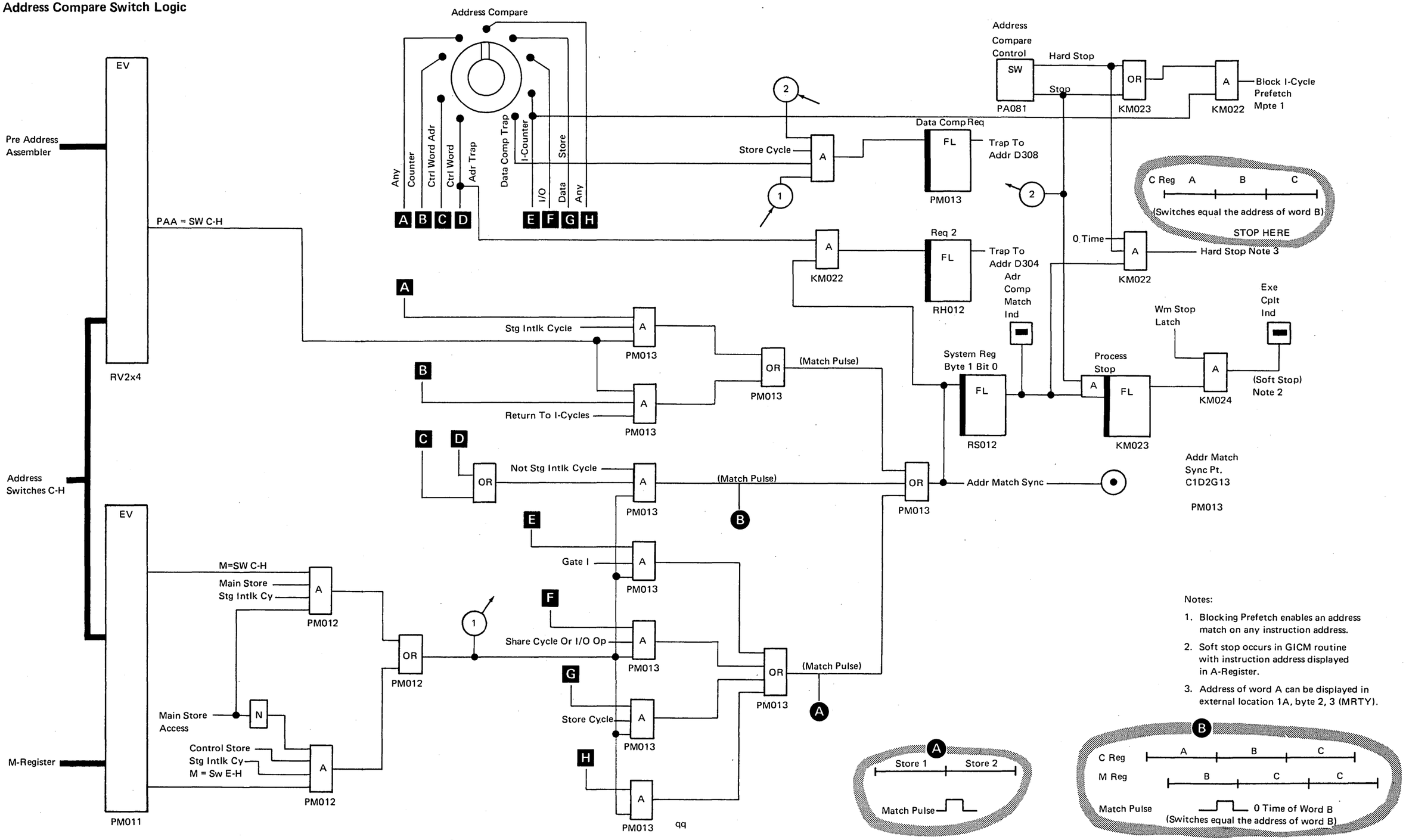


Note: It is possible to alter the microcode in the GKCC routine to perform other diagnostic functions.

Ctrl Word Adr Trap

Compares the control word address in the M-register against address switches E-H. If the compare is equal, a trap is taken to D304. The microroutine forces a branch to the address in the address switches A-D. ADDRESS COMPARE CONTROL must be in STOP. This position is used to set up microprogram loops for scoping.

Address Compare Switch Logic

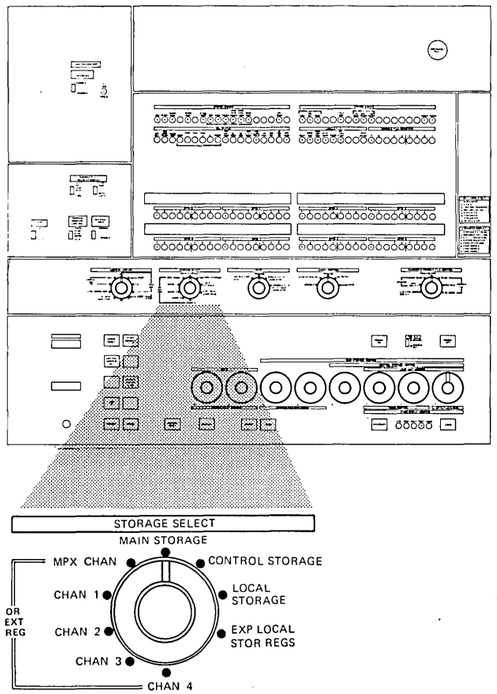


Notes:

1. Blocking Prefetch enables an address match on any instruction address.
2. Soft stop occurs in GICM routine with instruction address displayed in A-Register.
3. Address of word A can be displayed in external location 1A, byte 2, 3 (MRTY).

qq

STORAGE SELECT



- The storage select switch chooses the storage area to be used on manual store/display operations and address-compare operations.
- This switch is used in conjunction with: ADDRESS COMPARE, the upper and lower roller indicators, and the store and display keys.
- Moving this switch while the CPU clock is running has no effect on the system if ADDRESS COMPARE CONTROL is set to SYNC/NORMAL.

Note:

1. On a manual store operation, only one byte of data is stored.
2. When the System/370 microprogram is in control storage, it is normally much faster to store/display local, main- and control-storage locations from the console printer/keyboard (see the GKAD microroutine). The CE key must be used to store/display control storage (AS/DS) and local storage (AL/DL).
3. Store/display operations are inhibited during storage words and trap 1 and 2 cycles.
4. CPU clock must be stopped to store/display.

Main Storage

This position allows access to main storage for manual store/display operations, and gates main-storage addresses to the compare circuits for address-compare operations.

To perform a manual store operation:

1. Set the main-storage destination address into address switches C-H. (Switch H selects which byte will be altered.)
2. Set the data to be stored into switches A-B.
3. Press STORE.
4. The fullword containing the altered byte is displayed in SDBO (upper roller position 2).

To perform a manual display operation:

1. Set the main-storage source address into address switches C-H (byte selection bits of switch H are ignored).
2. Press DISPLAY
3. The fullword at the selected main-storage location is displayed in SDBO (upper roller position 2).

Control Storage

This position allows access to control storage for manual store/display operations and gates control-storage addresses to the compare circuits for address-compare operations.

To perform a manual store operation:

1. Set the control-storage destination address into address switches E-H. (Switch H selects which byte will be altered).
2. Set the data to be stored into switches A-B.
3. Press STORE.
4. The fullword containing the altered byte is displayed in SDBO (upper roller position 2).

To perform a manual display operation:

1. Set the control-storage source address into address switches E-H. (Byte selection bits of switch H are ignored.)
2. Press DISPLAY.
3. The fullword at the selected control-storage location is displayed in SDBO (upper roller position 2).

Local Storage

This position allows access to local storage for manual store/display operations. A-local storage is displayed in the A-register; B-local storage is displayed in the B-register. Both should be identical on manual store/display operations. The P-register setting has no effect.

To perform a manual store operation:

1. Set the local-storage destination address into address switches F-H (switch H selects the byte to be altered).
2. Set the data to be stored into switches A-B.
3. Press STORE.
4. The fullword containing the altered byte is displayed in the A-register (lower roller any position).

To perform a manual display operation:

1. Set the local-storage source address into address switches F-G. (Switch H is not used to display.)
2. Press DISPLAY.
3. The fullword at the selected local-storage location is displayed in the A-register (lower roller any position).

Note: Normal machine operation does not destine data to LS or an Ext Reg until the following cycle.

Example: C Reg LS000=0,OE,KFF
Data=FF

Assume that the machine was stopped after the microword LS000=0,OE,KFF was executed. Displaying LS00 would force the destination of LS00 with new data=FF. To see what the contents of LS00 were before updated, the machine would have to be stopped one cycle earlier. Another method would be to display the A- or B-register (depending upon the microword) to see the source data before LS was displayed.

Exp Local-Storage Registers

This position allows display of the expanded local-storage registers. These registers can not be stored manually (that is, using the store key, address switches, and the data switches). They are hardware registers--not monolithic array cards such as the local-storage registers.

To alter the expanded local storage registers:
Execute the AL mnemonic on the console printer/keyboard. The CE meter switch must be on.

To perform a manual display operation:

1. Set the expanded local-storage source address into address switches FG (switch H is not used).
2. It is not necessary to press DISPLAY.
3. The selected expanded local storage register is displayed in the A-register (lower roller any position).

External Registers Store/Display Operation

Setting STORAGE SELECT to any EXT REG position (MPX, CHAN1, CHAN2, etc.) allows access to the external registers for manual store/display operations.

Note: Many external registers are "source only" registers; therefore, data cannot be stored into all of them. See External Regs map in Chapter 2 for "source only" registers.

To perform a manual store operation:

1. Set the external register destination address into address switches FG. Switch H selects which byte will be affected. (Ensure that the selected register is not a "source only" type.)
2. Set the data to be stored into switches A-B.
3. Press STORE.
4. External register containing altered byte is displayed in the A-register (lower roller, position 1).

To perform a manual display operation:

1. Set external register source address into address switches FG (switch H is not used).
2. It is not necessary to press DISPLAY.
3. Selected external register is displayed in the A-register (lower roller position 1).

CAUTION

Upper roller positions 4 and 8 use the same display path as the external registers and are degated if STORAGE SELECT is in the EXT REGS position. Turn STORAGE SELECT back to MAIN STORAGE before displaying upper roller position 4 or 8.

Lower Roller Positions 2-8 Store/Display Operation

Documentary Cnsl; S, P, T, L Registers; System Register; Chan Interface; and Chan Words A, B, C are frequently used external registers that may be stored/displayed without using the store/display keys and address switches FG. They are displayed in lower roller positions 2-8 and are gated by setting the roller to the desired position when STORAGE SELECT is in EXT REGS positions.

Note: When roller positions 5-8 (Chan Words) are selected, STORAGE SELECT must be set to the channel for which store/display is desired.

To perform a manual store operation:

1. Set the roller to the desired external register.
2. Select the byte to be altered, using switch H. It is not necessary to set address switches F and G.
3. Set the data to be stored into switches AB.
4. Press STORE (first ensure that the selected register is not a "source only" type).
5. Selected register containing altered byte is displayed in lower roller.

To perform a manual display operation:

1. Set the roller to the desired external register.
2. It is not necessary to set address switches FG or press DISPLAY.
3. Selected external register is displayed in lower roller.

Address X/Late Regs Display Operation

These are special external registers that modify the logical address to a real address for OS/DOS compatability. These registers can only be displayed. STORAGE SELECT must be in EXT REGS and the lower roller in position 1.

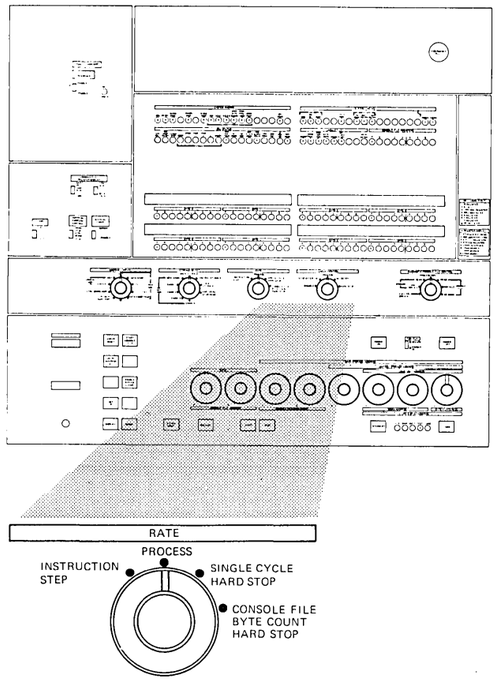
To perform a manual display operation:

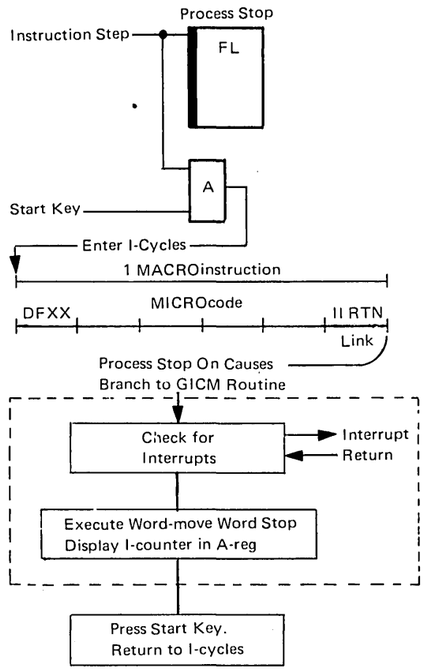
1. Set an address of 2E into switches FG.
2. Set switch H to the register (0-7) to be displayed.
3. It is not necessary to press DISPLAY.
4. The selected register is displayed in the A-register (lower roller position 1).

   CAUTION

   These registers may be out of parity. Bytes 0, 1 could be blank without parity. Byte 3 may be out of parity. This is normal operation. The reset state of bytes 0, 1 is all bits off including parity. If bytes 0, 1 are blank, ignore the parity of byte 3. If bytes 0, 1 have bits on, then the parity of byte 3 is only for bits 0-5 of byte 3.

RATE





Process

Allows normal CPU operation.

Instruction Step

In this position, one macroinstruction is executed each time START is pressed. If interruptions are pending, they are processed. EXE CPLT is on, and the next instruction address is displayed in the A-register (lower roller position 1).

The process stop latch is turned on by the instruction-step position. This latch is tested at the end of each instruction by the I1 RTN Lnk word in the microde. The I1 branch condition is met, and control is transferred to the GICM routine, where a stop word is executed until the next pressing of START.

Note: When the extended microdiagnostics are in control storage, INSTRUCTION STEP performs another function. The micromonitor tests the process stop latch and if on, causes the printout "cycle each test yes, no". More information on the microdiagnostics is located in the diagnostics user guide.

Single-Cycle Hard Stop

This position causes the CPU clock to stop at the end of the current microword in operation. If I/O devices are operating, overruns will occur.

This position is effective while stepping through a microroutine. Each pressing of START causes the CPU clock to run for one cycle. Because storage words require two cycles, two pressings of START are required to execute storage microwords.

Assume that the following microwords are in control storage, the RATE switch is in SINGLE CYCLE HARD STOP, and that the address in the M Reg-C008.

C000 P = 0, OE, K02
C004 V = V+W
C008 STW I R+4

The roller positions would contain the following information:

Upper Roller
(1) M-Reg = C008
(2) SDBO = V=V+W (word just executed)
(3) C-Reg = STW I R+4
(4) MB-Reg = C004
N-Reg = C004
(5) B-Reg = Contents of W-Reg
(6) Z-Reg = Results of V+W ·
(7) D-Reg = 02 02 02 02 (Previous operation P=0, OE, K02)

Lower Roller
(1) A-Reg = Contents of V-Reg

Note that the M-Reg, N-Reg and C-Reg are pointing to the next microword to be executed. MB contains the address of the microword just completed. The A-Reg, B-Reg and Z-Reg contain the arithmetic data of the microword just completed. The D-Reg has the arithmetic results of the previous microword. External address "1A" byte 2,3 (MRTY) contains the backup address C000. This address can be used to back trace through the microcode.

Notes:

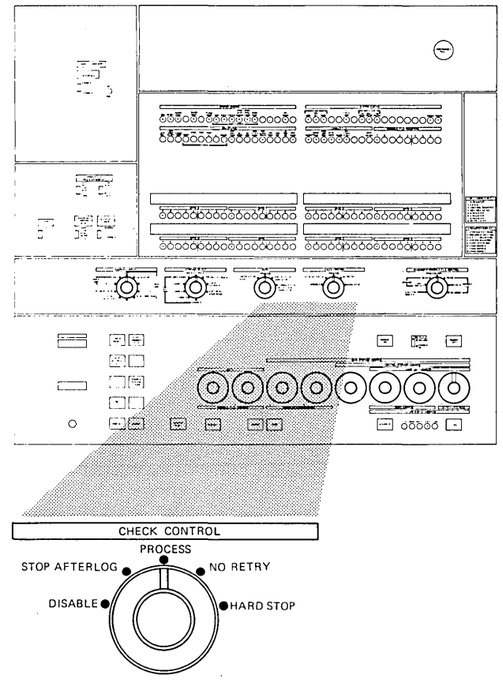
1. All store/display operations should be done while in single-cycle mode. See the write-up on the storage select switch.
2. The store and display keys are not functional during store 2, trap 1, trap 2 cycles.
3. In local storage control storage mode (LSCS), a dummy cycle is taken after every microword. To find where in local storage the microprogram is, use the microlisting, far left column.

M/LS
0F/3C -- The address in the M-Reg byte 3 is '0F', the actual local STG address is 3C (M-Reg bytes 1-2 are not used).

Console-File Byte Count Hard Stop

This position is used when troubleshooting the CPU with the basic microdiagnostics. In this position the console file stops reading data when the byte counter matches the value set in switches C,D. A complete write-up on how to use the byte counter follows DIAGNOSTIC/CONSOLE FILE CONTROL description.

CHECK CONTROL



Process

This is the normal processing position. Microinstruction retry traps are unconditionally allowed, and machine check traps are taken if allowed by PSW bit 13.

No Retry

Microinstruction retry traps are not allowed. When an error occurs, a machine-check trap is taken if allowed by PSW bit 13 and a machine-check interrupt is taken.

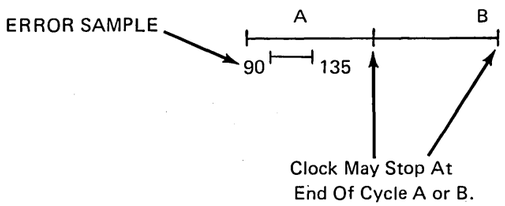
Hard Stop

In this position detection of a machine check causes the CPU clock to stop at the end of the cycle in which the error is detected. The error appears either in the upper row of lights (SYSTEM CHECKS) or position 8 of the upper roller (MCKA). Pressing START after the error stop causes the CPU to resume operation by conditioning with a retry trap. (To prevent retry, press CHECK RESET.) I/O operations may overrun when the stop occurs.

The machine stops in the cycle in which the error was detected, not necessarily the cycle in which the error occurred. To determine the failing microword, after the error occurs and the clock is stopped, set RATE to SINGLE CYCLE HARD STOP and then

press START once. Display external address '1A' (MRTY), bytes 2 and 3 contain the address of the failing microword for all types of errors (type 1, 2, or 3).

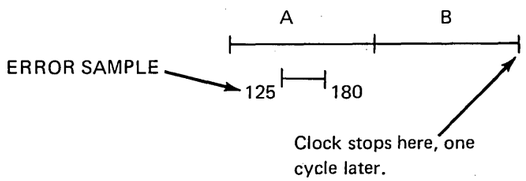
Type 1: Clock stops in the same cycle.



Note:

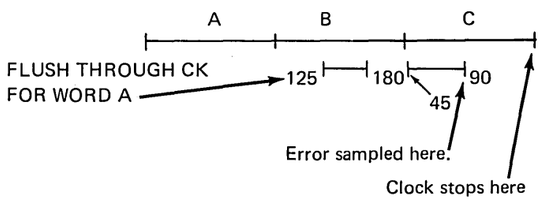
In this type of error, pressing START once loads MRTY with the failing microword address; however, destination gate that would have occurred is prevented.

Type 2: Clock stops in the following cycle.



MRTY has been loaded with the failing microword address.

FTC (Flush-Through Check), LS Comp (Local Store Compare) Clock stops two cycles later.



MRTY has been loaded with the failing microword address.

Type 3 errors: These are uncorrectable errrors. The retry hardware cannot attempt to reconstruct and repeat the failing microword.

Note:

1. Errors that occur while the machine is in I-Cycles (DF module) are handled differently. Instead of repeating the failing microword and continuing from that point, the entire I-Cycles operation is repeated from DFOC.
2. If diagnostic, register bit 3 (suppress all traps) is on, the hard-stop position is degated. The machine does not stop on errors. This function is used by the microdiagnostics.
3. Single ECC errors do not cause hard stop. To stop, jumper A-B2N2D07 to A-B2M2B07 (RE061). See Appendix A.

The following charts indicate the error type for each machine-check bit.

Bit	MCKA0		MCKA1		MCKA2		MCKA3	
0	LSA Source Address Check	1	ACB Register Parity Check	2	ALU2 Half-Sum Check	2	Ext Destine X Comp Check	3
1	LSB Source Address Check	1	LS Compare Check		ALU3 Half-Sum Check	2	Ext Destine Y Comp Check	3
2	LSA Destine Address Check	2	Flush-Through Check		ALU Logical Check	2	Ext Source Y Check	1
3	LSB Destine Address Check	2	H-Register Parity Check	2	B-Register Shift Check	2	Ext Control Asmbl Parity	1
4	Destine Byte Control Check	2			A-Register Parity Check	1	Interval Time Parity Check	3
5	LSA, B Destine Address Comp	2	P-Register Parity Check	2	B-Register Parity Check	1	S-Register Dup Check	2
6	LS Control Asmblr Check	1	T-Register Parity Check	2	Z-Register Parity Check	2	Time-Of-Day Clock Check	3
7	C-Register Parity Check	1	L-Register Parity Check	2	D-Register Parity Check	2	Control Stg Address Check	2

These indicators are shown in position 8 of the upper roller.

Bit	MCKB0		MCKB1		MCKB2		MCKB3	
0	Storage Address Check	1-2	M-Register Comp Check A	2	I-Cycle Hardware		CT Error Correction	
1	SDBI Parity Check	2	M-Register Comp Check B	2	Double ECC Error-SDC	2	C32 Data Bit Corrected	
2	SDBO Parity Check	2	M-Register Comp Check C	2	Control-Line Parity Check	2	C16 Data Bit Corrected	
3	Store Parity Check	2	M-Register Comp Check D	2	ECC Busy Check	2	C8 Data Bit Corrected	
4	Spare		Addr X-Late No Match	2	ECC Hardware Check	2	C4 Data Bit Corrected	
5	Store Protect Parity Check	2	Addr X-Late Multi Match	2	Double ECC Error	2	C2 Data Bit Corrected	
6	Clock Sync Check A	3	Addr X-Late LRU Invalid	2	Single ECC Error Correct		C1 Data Bit Corrected	
7	Clock Sync Check B	3	Any Machine Check		Single Data Bit Correct		C0 Data Bit Corrected	

These indicators are located in the SYSTEM CHECKS portion of the console. The indicators on the console are grouped into functional areas, not as they appear in MCKB. MCKB is an external register and can be displayed by setting address 06 in switches F, G, and STORAGE SELECT to EXT REGS.

Stop After Log

The CPU soft-stops (word move, stop word cycling) after the machine-check logout is placed in main storage. LOG PRESENT and RETRY is on to indicate that the log has occurred. After the logout occurs, either run SEREP to obtain the edited log-out or use the alter/display capability of the console printer/keyboard to dump the unedited logout. Refer to the error-handling section to determine what areas to display: (machine dependent or independent, or both). The machine-dependent logout pertains to the Model 145 only. The machine independent logout pertains to registers and data that is common to all System 370's.

Note:

1. The main purpose of this switch position is to enable you to get a handle on intermittent problems even though the OS or DOS error-handling facilities may not be available.
2. CAUTION

   Some System 360 programs may have data or instructions in lower storage. After the logout occurs, run SEREP and re-IPL. Do not attempt to continue processing.
3. The retry mechanism is fully functional and is not affected by the stop-after-log position.

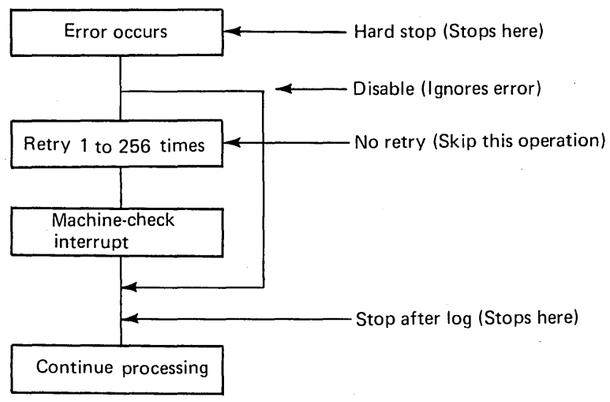
Disable

No microinstruction retry, machine-check traps, or machine-check interrupts are taken. The machine-check register is still set as a result of the error (logic pages RExxx).

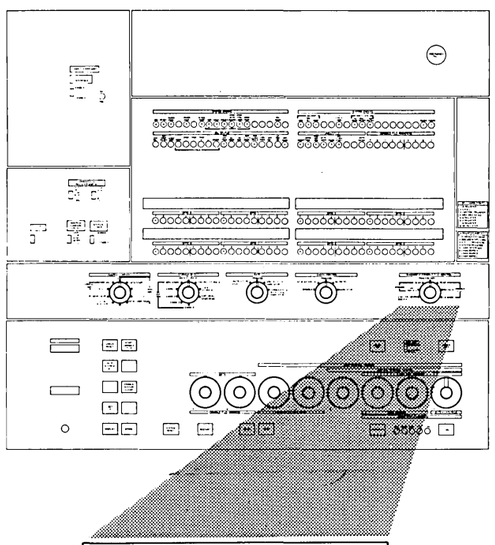
The main functions of disable are:

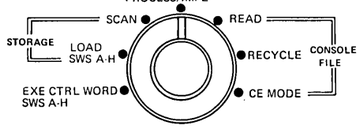
1. Allows the error latch(es) to be set (all errors are accumulated).
2. Allows normal operation such as destination gating lines to be active.
3. Prevents the CPU clock from stopping.
4. Prevents the machine-check trap request.
5. Allows writing into Phase 2I storage (blocks the cancel line from coming on during storage errors).
6. The function of restoring control storage when single ECC errors occur, is blocked. No attempt is made to correct the failing bit; the bad data remains in control storage.

A brief summary of the check control switch follows. A detailed operation can be found in the error-handling section.



DIAGNOSTIC/CONSOLE FILE CONTROL





Process /IMPL

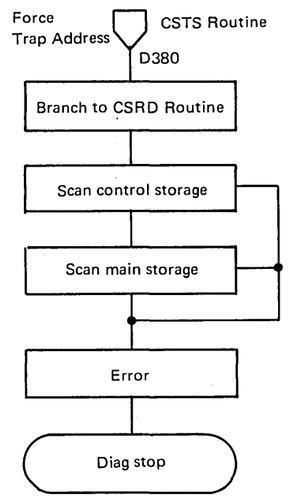
Allows normal CPU operation and console-file loading.

Scan (Storage)

Forces a trap to microroutine CSTS, where every word of main and control storage is read out and checked for errors. If any errors occur (except single ECC errors), the machine stops with DIAGNOSTIC STOP on. The microroutine is initiated when RATE is in PROCESS, and START is pressed. The following action occurs. This position is functional, only when the System 370 microprogram is in control storage.

Notes:

1. Comments in the CSRD routine provide helpful information such as: looping on a selected address and preventing address update; how to determine the failing address and data; how to loop the entire test.



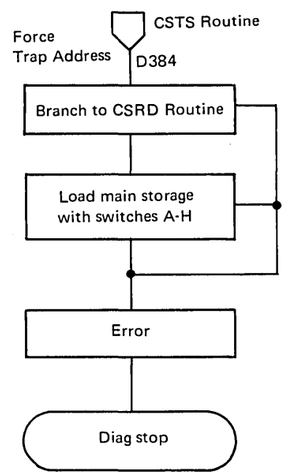
2. It is possible to hard-stop on single ECC errors by adding a jumper from A-B2N2D07 to A-B2M2B07 ('Clock Tie Up', logic RE061). With the jumper on, set the mode register bits for full recording mode (Ext Adr 08 byte 0-Bit 3, main-storage full recording; bit 4, CTRL STG-Full recording). The mode register bits being on allow the ECC check bits to be displayed in MCKB (Ext Addr 06 Byte 3) Register to determine the failing bit.
   See CPU service aids in Appendix A.

MCKB Byte 3

P	0	1	2	3	4	5	6	7
	CT	C32	C16	C8	C4	C2	C1	C0

Load Switches A-H (Storage)

Forces a trap to microroutine CSRD where the contents of switches A through H are loaded into main storage. Control storage is neither loaded nor scanned. The microroutine is initiated when RATE is in PROCESS and START is pressed. The following action occurs. This position is functional only when the System 370 microprogram is in control storage.

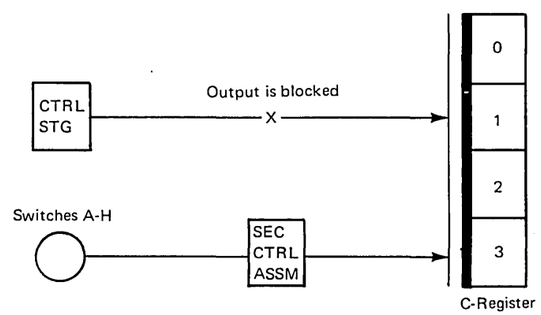


Notes

1. Comments in the CSRD routine provide information on how to loop this test, loop a single word, etc.
2. The storage-protect keys are not cleared by this routine.
3. Storing all zeros generates ECC bits of "3C"; storing all ones generates ECC bits of "FF".
4. DIAGNOSTIC STOP comes on for double ECC errors or any other machine checks. Single ECC errors do not stop the machine unless the jumper is on. (See Note 2 under SCAN.)

Exe Ctrl Word SWS A-H (Execute Control)

With the switch in this position, the C-register executes the control word in switches A through H. Instead of the C-register being loaded from control storage, word in the switches A-H is gated to the C-register. All other machine functions are normal. The M-register is loaded with the next address, and storage is cycled to fetch the next control word. However, the next control word is not gated into the C reg.



Notes

1. If a failure is caused by a single microword, set the failing microword in switches (A-H). Set RATE to PROCESS and press START. The word in the switches will be repeatedly executed. Sync the scope on 0 time.
2. If the word being executed from the switches calls for a local-storage address or external address be sure the P-register is set up properly.
3. Because storage is cycled, it is possible to have single ECC errors. Even though the data from storage is not used, a correction cycle still occurs.
4. The 'Exec Cntrl Word' position can be used to clear main or control storage as follows.
   a. Zero the P-Register.
   b. Zero LS 0 and LS 1.
   c. Set switches to 48081003
      "STW, LS00 CS LS01&4, VAL",
      this word clears control storage.
   d. Set switches to 48081403
      "STW, LS00 LS01&4, VAL"
      this word clears main storage.
   e. VAL in the two words above allow storage to loaded while ignoring ECC errors.

Read (Console File)

This switch position enables the console file to begin reading at the sector address determined by switches AB.

When a sector pulse arrives, the track/sector address is read from the disk. This address is compared with the value set in switches A, B. If a mismatch occurs and the switches (bits 0-4) are higher than the address read from the disk, the head is incremented to the next track to wait for a sector pulse. Conversely, if the switches are lower, the head decrements to the next lower track and waits for a sector pulse. This continues until an equal compare is generated. The head is now positioned properly, bits 5-7 are compared so that reading can begin at the correct sector. Data is read from the disk, and the normal reading operation continues.

Notes: This position can be used when it becomes necessary to bypass a sector or test because of a bad spot on a diagnostic disk or if an REA is installed and it invalidates a test. Use the micro-listing to determine the sector address of the next sector or test and set the A, B switches to this value. This should be a temporary measure. Order a new disk from the plant.

CAUTION

If a diagnostic test is bypassed, the fault-locating data may not be valid.

Recycle (Console File)

The console file recycles the entire sector address set in switches A-B. This position is used to loop a diagnostic test. The head is incremented/decremented as explained under READ. Reading from the sector begins with sector start and continues reading until an error occurs (diagnostic stop) or until sector end is reached. Reading resumes again on the next revolution.

CE Mode (Console File)

CE mode turns on the console-file power to allow working on the console-file offline (data from the file is *not* gated to the CPU). For removal and replacement procedures see Chapter 6. For adjustment procedures see the 23FD manual.

Console-File Byte-Counter Operation

The byte counter is a diagnostic aid that enables the console-file circuits to stop reading data anywhere in the sector. Without the byte counter, the only way to troubleshoot failures that occur in the basic microtests would be to sync the scope on the sector start pulse and use delayed sweep to locate the failure. By using the byte counter, the failing microword can be isolated and then placed in switches A-H and troubleshot using the EXE CTRL WORD SWS A-H position.

To stop or sync on a byte-counter match:

1. Set DIAGNOSTIC/CONSOLE FILE CONTROL to READ or RECYCLE.
2. Set switches A-B to the track/sector.
3. Set switches C-D to the byte value (other than zero) desired to stop or sync on.
4. To stop, set RATE to CONSOLE FILE BYTE COUNT HARD STOP.
5. Press START CONSOLE FILE.

Example: To stop after the microword LS01=SPTL,DF is executed, set switches C-D to "0F" and set RATE to CONSOLE FILE BYTE COUNT STOP.

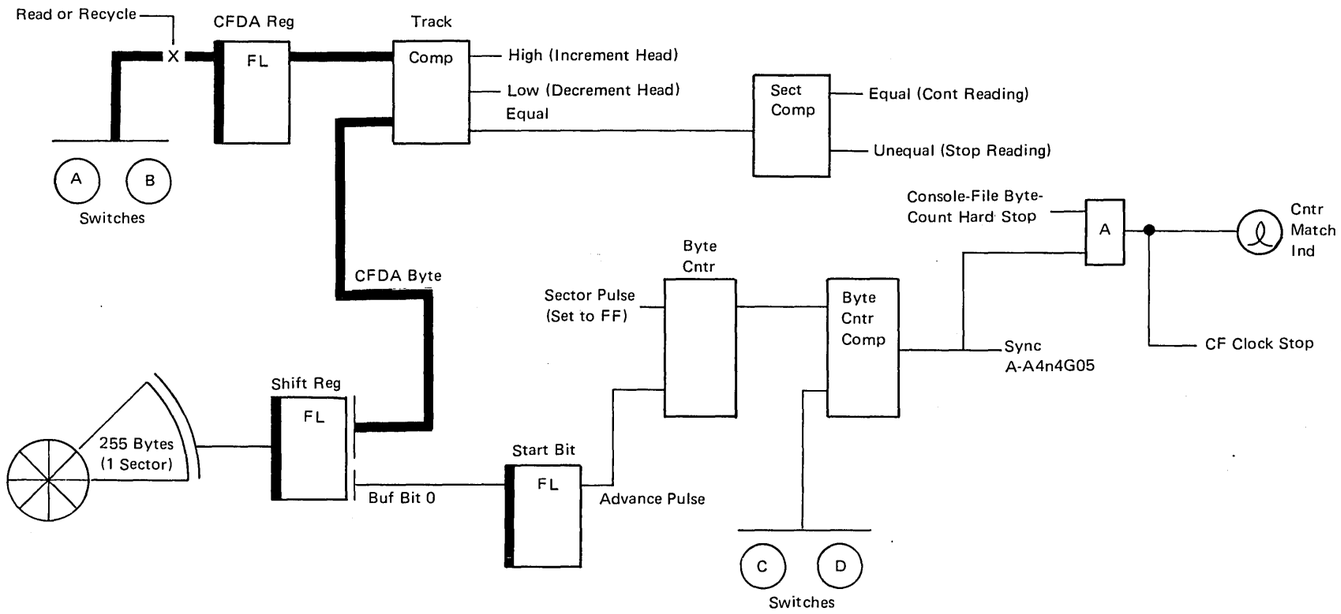
CNT	CMD	WORD	STATEMENT
05	70	0000AA00	SPTL=NOREG,SF
0A	70	3890CF00	LS01=SPTL,DF
0F	70	3010CF00	P=0, OE, K30

The console-file displays provide useful information when you troubleshoot console-file attachment circuits. Two spring-loaded toggle switches are used to display the disk address register, shift register, command register, and byte counter. Both switches must be in the normal position to display the disk address register. Moving either switch off the normal position displays the selected register.

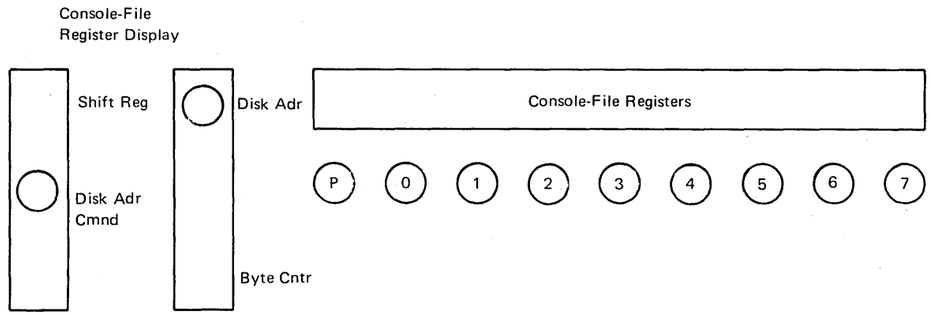
Note: The shift register is displayed through the command register. Always be sure to display the command register first, or command data will be lost.

Example 1, using the previous example of the byte counter.

Disk address = Sector address set into switches A, B.
Byte counter = '0F' (switches CD-0F).
Command Register = 70 (The 70 preceding 3010CF00).
Shift Register = 00 (3890CF00).



Console-File Displays

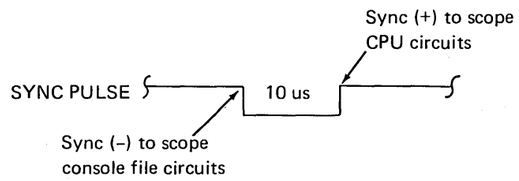


Example 2, assume a byte-counter stop at 0E (Switches CD=0E).

Byte counter = 0E (switches CD=0E).
Command Reg = 70 (The 70 preceding 3890CF00).
Shift Register = 00 (3890CF00).

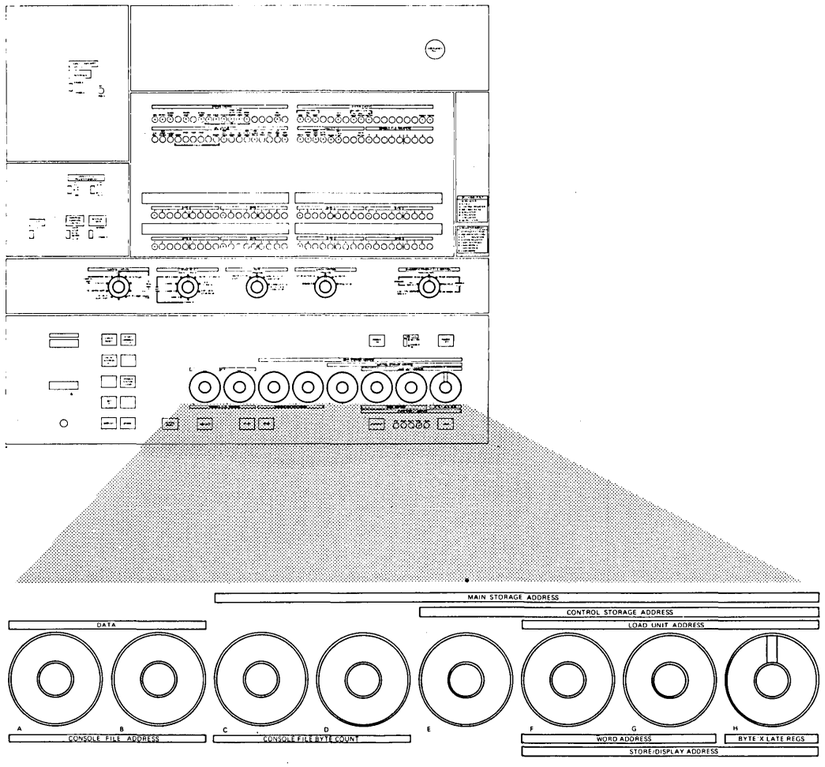
Notes:

1. Even though RATE is in SINGLE-CYCLE HARD STOP or CONSOLE-FILE BYTE-COUNT HARD STOP, the CPU clock is cycled and the microwords being gated to the C-register are executed. (The console-file attachment controls the CPU clock.) In the previous example (stop at 0E), the word LS01=SPTL,DF would be gated to the C register and executed.
2. To scope the CPU operation of the microword LS01=SPTL,DF while the console file is in recycle, set switches C,D to OE and sync the scope on A4N4G05. To observe the console-file circuits, sync minus; to observe the CPU operation, sync plus. (Set RATE to PROCESS).



3. A byte counter setting of 00 should not be used to sync or stop.

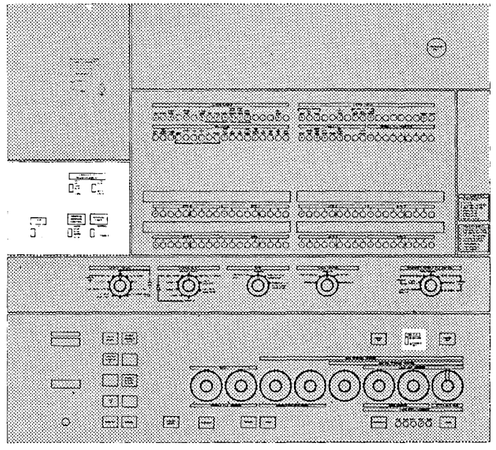
ROTARY SWITCHES A THROUGH H



Operating any of these eight 16-position switches during machine processing does not affect machine operation. Switch functions are:

Switches	Function
AB	These two switches are set to the hexadecimal value of data to be entered during manual store operations. These switches also specify the disk address of the console file during certain microprogram load operations. See writeup of DIAGNOSTIC/CONSOLE FILE CONTROL Read and Recycle positions.
CD	With the rate switch set to SINGLE-CYCLE HARD STOP, the byte number set into these two switches causes the console file to stop when that byte of a sector is read from the disk. See writeup of DIAGNOSTIC/CONSOLE FILE CONTROL Read and Recycle positions.
CDEFGH	A main-storage address (in hexadecimal) is set into those switches for: a. Manual store or display operations for main storage. (The two low-order bits from switch H specify the desired byte.) b. Functions specified by the address-compare switch.
EFGH	A control-storage address (in hexadecimal) is set into these switches for: a. Manual store or display operations for control storage. (The two low-order bits in switch H specify the byte for a store operation.) b. Functions specified by the address-compare switch.
FG	These switches are set to the word address for store or display operations (see "Store and Display Keys"). The two low-order bits from switch H specify the byte for store operations.
FGH	These switches are set to the load-unit address for an Initial Program Load (IPL) operation.

TOGGLE SWITCHES



TIME-OF-DAY CLOCK

Secure

The switch is spring-loaded in this position. If the set clock instruction is executed with the switch in this position, the value of the clock remains unchanged and the counting operation continues normally.

Enable Set

The switch must be momentarily held in this position to allow the set clock instruction to change the value of the clock.

CONSOLE-FILE REGISTER DISPLAY

These spring-loaded toggle switches control the data displayed in the console-file register indicators. Four console-file registers are selectable:

1. Disk-address
2. Shift
3. Command
4. Byte counter

When both switches are in the normal position, the disk-address register is displayed. Moving either switch off the normal position displays the selected register.

Note: The shift register is displayed through the command register. Always display the command register first or command data will be lost.

LAMP TEST

Console indicators should light when this toggle switch is operated. This function can be performed without affecting system operation.

ADDRESS COMPARE CONTROL

This three-position toggle switch is used in conjunction with the address-compare rotary switch. See write-up of ADDRESS COMPARE. The CPU should be in MANUAL state before this switch is operated so that contact bounce does not cause false indications.

Hard Stop

The CPU clock stops after completing the machine cycle in which the match occurs. The address of the word on which the match occurred is available in the MB2 and MB3 registers.

CAUTION

If an Alter/Display function is performed on the console printer keyboard after a hard stop, the CPU will return to Run mode.

Sync/Norm

This is the normal position for this switch. With the switch in this position, each address-compare match generates a sync pulse.

Stop

With the switch in this position, a soft stop (executing a STOP word) occurs whenever an address-compare match occurs. The action required is the same as if STOP had been pressed.

INTERVAL TIMER

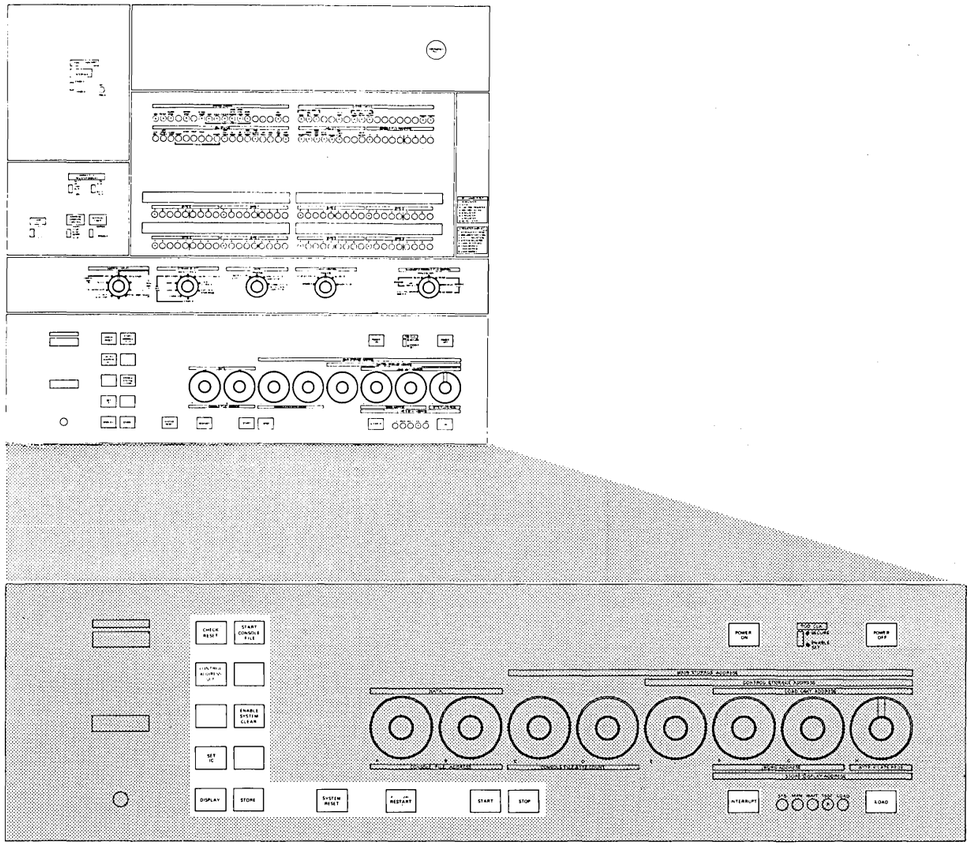
Normal

This position allows a value to be set into or read out of the interval timer (MS loc. 80). The value set is decremented, and an External interrupt is initiated (if allowed by PSW bit 7) when the count decrements through zero.

Disable

With the switch in this position, the timer continues to decrement. An interruption is not sent to the CPU if the count decrements through zero. The contents of the timer is not available to the data flow for interval-timer functions; but the timer location (hex address 50, decimal address 80) is available for program use.

KEYS



CHECK RESET

Operation of this key resets all machine-check circuitry, including the machine-check register, regardless of the mode of operation the machine is in.

START CONSOLE FILE

(See System/370 Microprogram Load flowchart in Appendix A) Operation of this key causes a circuit system reset. Console-file power turns on. Loading from the console file starts at track 0 sector 0 if the diagnostic/console-file control switch is set to PROCESS. If the diagnostic/console-file control switch is set to CONSOLE FILE READ, loading starts at the console-file address specified by console switches A and B. (The System/370 disk must be loaded starting at the track/sector 00.)

The start-console-file key turns red from the time it is pressed until the console file is ready to start reading. This time includes the five-second delay necessary to ensure that the disk is up to speed. The key stays red if no disk is mounted or if the disk does not turn.

When the file is ready to start reading, the key turns from red to white. At file-end time, the white indicator turns off if the machine-check latch is not set.

If a data check occurs that causes the file to stop or if the machine-check latch is on at file-end time, the key turns red. (Data checks are retried 15 times before a stop occurs.)

When loading (that is, IMPL or diagnostic operation) is completed, console-file power turns off. After a normal IMPL, the system-reset microcode executes, and the CPU enters the stopped state (Word-Move Word, Stop function executing).

CONTROL ADDRESS SET

Pressing this key causes no action unless:

1. The CPU is in a stopped state (that is, a Word Move STOP Word is being cycled), or
2. The CPU clock is stopped (that is, no CPU cycles are occurring).

If either condition exists (Manual indicator on), pressing SET CONTROL ADDRESS conditions the circuitry to load the values indicated by switches EFGH into both the M- and N-registers when START is pressed. The address set into M and N is for control storage only

ENABLE SYSTEM CLEAR

Operating this key while pressing LOAD or SYSTEM RESET, clears the following areas to zero with good parity or ECC bits.

1. General registers
2. Floating-point registers
3. Keys in storage
4. Main storage

SET IC (INSTRUCTION COUNTER)

Pressing this key causes no action unless:

1. The CPU is in a stopped state (Word Move Word, Stop function is being executed).
2. The CPU clock is stopped (that is, no CPU cycles are occurring).

If either condition exists (Manual indicator on), pressing SET IC causes a trap to the GKCC routine, which gates the contents of console switches CDEFGH into the instruction counter (Exp LS loc. 50). If the CPU clock is stopped, SET IC restarts the clock so that the trap can execute.

The CPU then enters the soft-stop state (Word Move Word, Stop function executing). If START is pressed, instruction processing starts at the instruction specified by the address in I (Exp LS loc. 50). The current PSW is not disturbed (except for the next-instruction address) by the Set-IC procedure.

Therefore, the current PSW at the time of the Set-IC procedure is the one used when instruction processing is started. At least one instruction is executed before any interrupts are handled when START is pressed after the Set-IC operation.

DISPLAY

This key is used in conjunction with STORAGE SELECT. It initiates two clock cycles (292.5 ns) to provide gating lines for displaying the contents of the storage location selected by STORAGE SELECT, and address switches C-H. One fullword (switch H byte select is ignored) is displayed in the A-register (lower roller position 1) for each pressing of the key. Procedures for manual display operations are discussed under STORAGE SELECT.

STORE

This key is used in conjunction with STORAGE SELECT. It initiates three clock cycles (292.5 ns) to provide gating lines for placing the value of data switches AB into the storage location selected by STORAGE SELECT, and Address switches C-H. One byte of data is stored (switch H selects which byte) for each pressing of the key while the fullword containing the newly stored byte is displayed in SDBO (upper roller position 2). Procedures for manual store operations are discussed under STORAGE SELECT.

SYSTEM RESET

Pressing this key turns on the IMPL REQD indicator, resets CPU clock registers, circuit registers, controls in the CPU, and online I/O units (including outstanding sense and status information). This hardware reset is independent of any microprogram. In addition, releasing SYSTEM RESET initiates a system-reset microprogram (GRST routine) by starting the CPU clock. This microprogram:

1. Resets word one of every multiplexer channel UCW in control storage. (Word one of a UCW contains the op's and flags, UCW/channel status, and high- and low-count bytes.)
2. Sets S, P, T, and L registers to 00.
3. Sets machine-check A-register, machine-check B-register, and diagnostic register to all 0's.
4. Sets ACB-register to a constant determined by the size of the control storage.
5. Restores LNK area of local storage to I-cycles address.
6. Resets MPX registers MT0 and MB0 and also DOC registers TA and TE.
7. Drops Operational Out to online I/O devices. Raises Operational Out to online I/O devices after at least six microseconds.
8. Sets to zero the UCWs used with the disconnected command chaining (DCC) selector channel (block-multiplexer channel).
9. Sets control registers to the initial state.
10. Sets current PSW to zeros.
11. When ENABLE SYSTEM CLEAR is operated at the same time as SYSTEM RESET, clears the following areas to zero with good parity or ECC bits.

    General registers
    Floating-point registers
    Keys in storage
    Main storage

12. Sets the contents of the TOD clock to zero and places the clock in the not-set state during a system reset initiated by power on.

Main-storage is not affected by the system-reset microprogram. The resident-diagnostic microprogram (CSRD routine) executes near the end of the system-reset microprogram. The resident-diagnostic microprogram:

1. Checks CPU hardware that could prevent system operation by the customer.
2. Scans local storage for correct parity.
3. Scans control storage for double errors detected by the ECC circuitry.

Any errors detected during the resident-diagnostic microprogram turn on the diagnostic-stop indicator.

After successful completion of the resident-diagnostic microprogram, control returns to the system-reset routine, which turns off IMPL REQD. IMPL REQD can serve, therefore, as an indication that the microprograms are not present in control storage.

After IMPL REQD turns off, the system-reset microprogram tests the IPL indicator. If the indicator is on, the system-reset microprogram branches to the IPL microprogram. If the indicator is off, the system-reset microprogram branches to a Word-Move STOP Word, which places the machine in the stopped state.

The hardware reset can be performed without execution of the system-reset microprogram with the following procedure:

1. Set the rate switch to SINGLE-CYCLE HARD STOP.
2. Press SYSTEM RESET.
3. Press SET CONTROL ADDRESS before the CPU clock starts.

The microprogram executed depends upon the control-storage location addressed by switches EFGH when SET CONTROL ADDRESS is pressed.

RESTART

This key is effective when the machine is running or when it is in the soft-stop mode. Pressing RESTART causes a trap to the GRST routine. The GRST routine branches to GICM, where the current PSW stores into main-storage locations 8 through 15. The new PSW loads from location 0 and the CPU operation start under control of the PSW.

START

Various CPU stop conditions reset when START is pressed. When START is released, at least one machine cycle occurs (that is, the CPU clock is started, but other conditions, such as the rate switch set to SINGLE-CYCLE HARD STOP, may cause it to stop after one cycle).

The CPU clock must be stopped or the machine must be in the soft-stop state for START to be effective.

STOP

This key is effective, only if the CPU is running (CPU machine cycles are occurring). When STOP is pressed:

1. PROCESS STOP is set. This causes a trap to the GICM routine.
2. Execution of the current instruction is completed.
3. Any outstanding interrupts are taken to the point of storing and loading the appropriate PSWs.
4. The CPU recycles a Word Move STOP Word.
5. The instruction counter (EXP LS loc. 50) displays in the lower roller switch indicators (that is, in the A-register) while the Word Move STOP Word recycles. (The lower roller switch can be set to any position in order for the instruction address to be displayed, but the clock-stop indicator must be off.)

Even though the CPU is in the soft-stop state, channel traps for chaining, data handling, or status handling execute if requested. Other traps (such as for a machine check) that occur because of errors detected during the I/O traps are also exeucted. (The use meter runs and the SYS indicator is on for any of these operations for at least 400 ms.

However, after the word-move STOP word is first executed, no further interrupts (I/O, machine-check, etc) can be taken until the CPU is restarted (such as by pressing START). If START is pressed, one instruction processes before any interrupts are handled, even if the interrupt(s) is/are pending before START is pressed.

Because I/O operations can occur during the CPU stop condition just described, the operator should not initiate a single-cycle hard-stop operation unless it is certain that either:

1. All I/O operations are completed, or
2. Loss of I/O information is not important.

INTERRUPT

Pressing this key turns on bit 1 of the external-interrupt (EXTINT) register. If PSW bit 7 is on, the interrupt latch is set. At the completion of instruction execution, the RETURN microword interrogates the interrupt latch. If the latch is on, a branch to the GICM routine is taken, where the interrupt is executed.

LOAD

When this key is pressed, a circuit system-reset is performed; when it is released, the system-reset, the resident-diagnostic, and the IPL (Initial Program Load) microprograms execute. LOAD is on during the IPL operation. This indicator turns off after successful completion of the IPL operation.

The load unit is specified as follows:

1. Rotary switches GH specify the unit address.
2. Rotary switch F specifies the channel (MPX, SX1, SX2, SX3, SX4, depending upon the system configuration and the channel desired).

POWER ON

This key turns red when operated. It turns white at the end of a successful power-on sequence. The power-on sequence for the CPU and all online I/O units starts when this key is pressed. A circuit system reset occurs. The information in storage is not valid after a power-on sequence. An IMPL operation is performed after the power-on operation to initialize main and control storage, if all switches are in PROCESS position and the proper disk is mounted.

Main storage clears to zeros and validates as part of the system-reset microprogram after power-on. A solid double-bit check detected when attempting to store zeros in bytes 4, 5, 6, and 7 of a doubleword causes a machine check.

I/O units are power-on sequenced, one by one. If power cannot be brought up for a unit, power-on sequencing stops until corrective action is completed for that unit. All online I/O units reset. The power-on sequence bypasses offline I/O units. The time required for a power-on sequence depends on the number of online I/O units that are powered on.

POWER OFF

Pressing this key removes power to the CPU and to online I/O units. Main-storage and control-storage information is lost.

I/O interrupt conditions reset when POWER OFF is pressed. Therefore, any pending interrupts are lost during a power-off operation.

Power to I/O units drops without regard to sequencing.

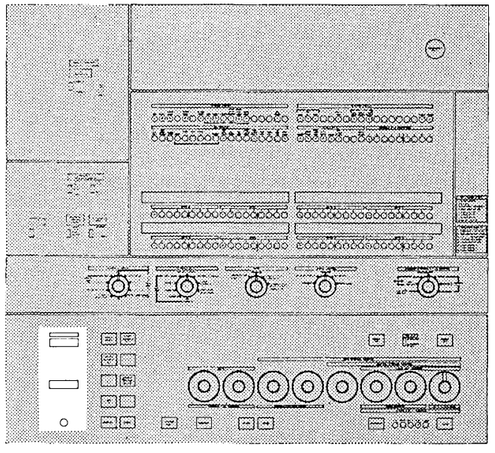
The power-off key takes precedence over the power-on key.

The recommended power-off procedure is:

1. Press STOP.
2. Press POWER OFF

A power-supply failure or an overtemperature condition causes a power-off sequence and turns on the power-check indicator on the console. Pressing CHECK RESET turns off the indicator and allows a retry of power-on. If the condition that caused the power-off sequence is temporary, such as a temporary loss of input power or a temporary overtemperature condition, pressing POWER OFF resets the power check and allows the power-on sequence to operate. If the check was caused by some condition that would damage the machine, such as a tripped CB, the power-check indicator turns on again.

USE METERS



The console has two direct-reading (usage) meters that measure operating time: a customer meter and a CE (customer engineer) meter. The position of a key switch (below the meters on the console) determines which of the two meters records operating time. The customer engineer has the key for this switch. Whenever he is performing either scheduled or unscheduled maintenance in the CPU, he will set the switch to cause the CE meter to operate. One of these meters (determined by the key-switch setting) operates whenever the CPU clock is running and:

1. A word-move STOP word is not being recycled,
2. The CPU is not performing a soft stop that results from a single-cycle operation, and
3. The CPU is not in the wait state.

The usage meter runs for a manual store or display operation because the CPU clock runs for such an operation.

Either usage meter runs for a minimum of about 400 ms each time it is started.

CHAPTER 6. CONSOLE FILE ADAPTER

INTRODUCTION

The console file (IBM 23FD Disk Drive) is the microprogram loading device for the 3145. The console file adapter is the communication and control link between the CPU and the console file (CF).

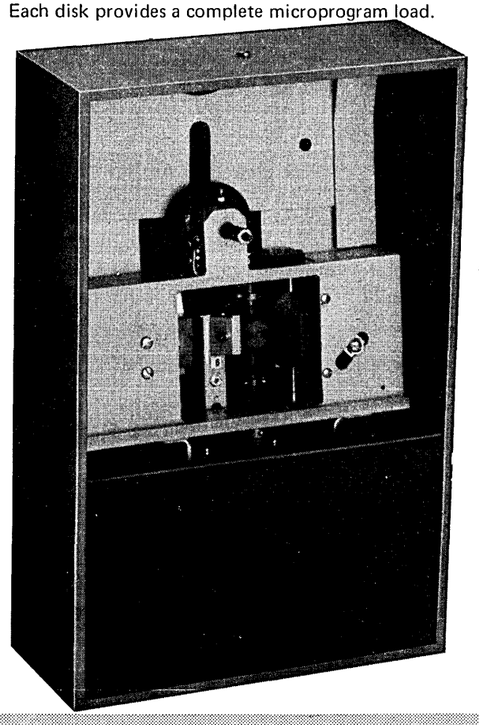
The adapter has its own set of commands for controlling the console file, for internal diagnostics, and for CPU diagnostics.

An initial microprogram load (IMPL), or a diagnostic operation initiated from the CPU console, turns console-file power on. Console-file power turns off at the completion of the load or diagnostic operation.

Note: This chapter does not contain theory or maintenance information for the IBM 23FD or 23FD II Disk Drives. For this information, refer to the Theory-Maintenance Manual, IBM 23FD Disk Drive, SY26-4154, or the Theory-Maintenance Manual, IBM 23FD II Disk Drive, SY26-4175.

23FD DISK FILE

- Initial microprogram loading (IMPL) device for the system.
- Single read-only head.
- Track-to-track accessing.
- Prerecorded interchangeable disks.
- Each disk provides a complete microprogram load.



Controls: Read-head loading. Power. Accessing.

File data. Sector pulses.

CONSOLE FILE ADAPTER

- Controls 23FD
  - Power on/off
  - Reading
  - Accessing
- Automatic IMPL when system is powered on.
- Loads data from the 23FD into the CPU.
- Loads control words directly into the C-register and initiates execution.
- Comparison checks certain next-control-word address bits.
- Loads starting at any console-file disk address.
- Recycles specific sectors for scoping.
- Controls the CPU clock.
- Automatic retries on data checks (read errors).
- Stop mode on errors.
- Parity checking.
- Odd/even byte check.
- Sector byte-counter match stop.
- Sector byte-counter sync.
- The console-file adapter circuitry is checked with a diagnostic disk.

Error indications. Compare data.

Data and control lines.

CPU

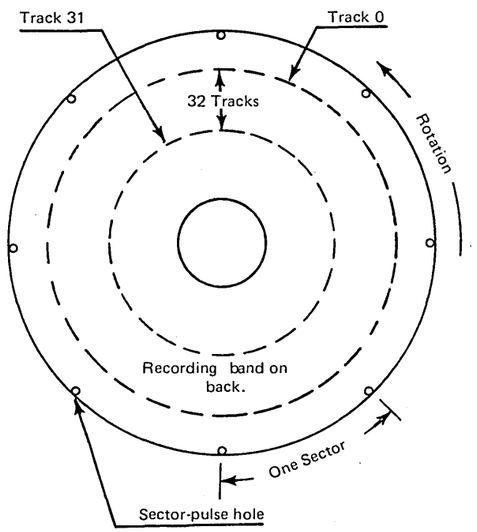
Control lines and data from the console-file adapter go to CPU:

- Secondary control assembler.
- C-register.
- Expanded local-storage address assembler.
- External control assembler.
- Local-storage control assembler.
- B-local-storage address assembler.
- External facilities assembler.

Disk Format

- 32 tracks (0 to 31) per disk.
- 8 sectors (0 to 7) per track.
- 256 sectors (00 to FF) per disk.
- Rotation time of 666.7 ms.
- Sector pulse every 83.3 ms.
- Sector-pulse duration:

  1.2 ms. - metal base 23FD.
  80 µs. - 23FD Model II (plastic base).

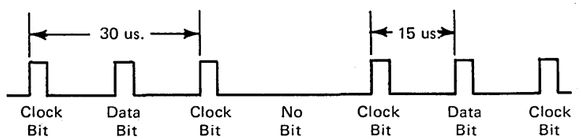


Front View

Data- and Clock-Bit Timing

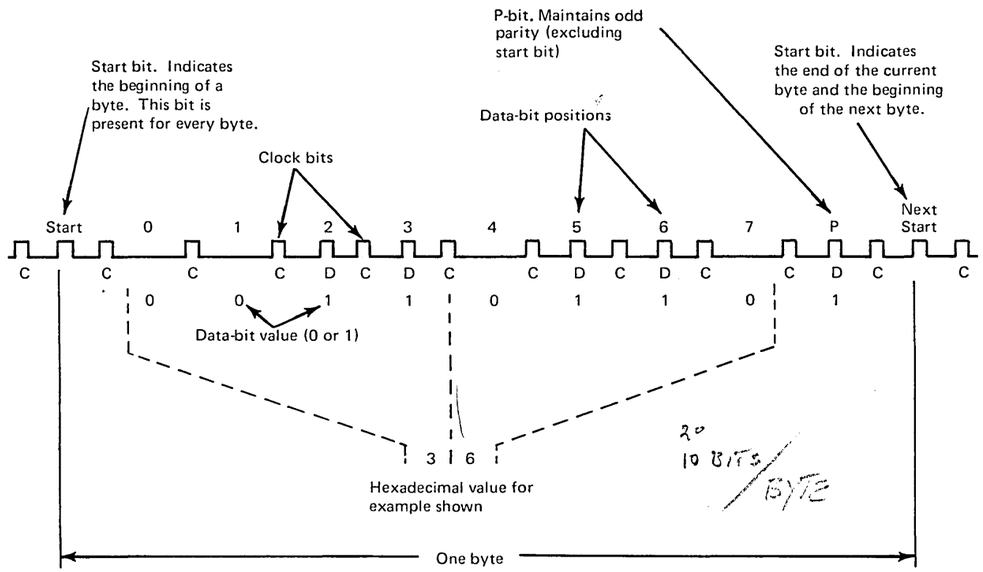
Clock bits are recorded over the entire track, including the area that passes the read head during sector-pulse time.

Data bits, when present, are recorded between clock bits.



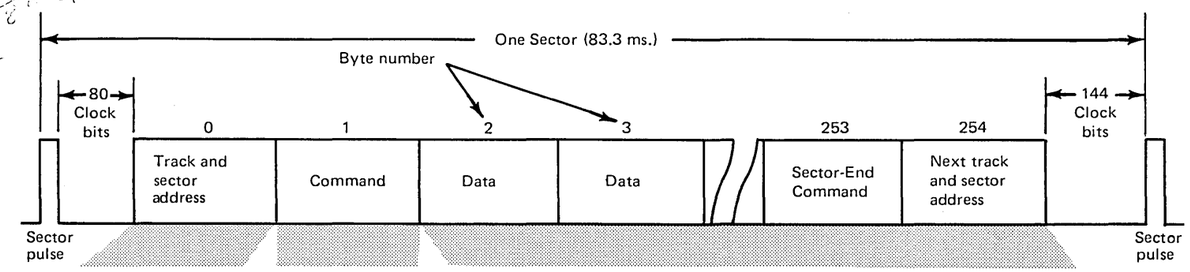
Byte Format

- Ten bit positions for each byte.
- Bytes are read serially, by bit.



Sector Format

- Maximum of 255 bytes per sector.



Always a console-file disk address (CFDA) byte.

Always a command byte.

Arrangement of byte types varies. The adapter recognizes byte types in this area by their sequence. The three byte types are:

1. Console-file disk address (CFDA).
2. Data.
3. Command.

Data Byte

Data bytes are recorded in groups of four. Each group is preceded by either a command byte or, when in compact data mode, by another group of four data bytes. For further information about compact data mode operation see 'CF Commands'.

Command Byte

Command bytes specify adapter circuitry action. A command byte has the same bit pattern as any other byte. The second byte read after a sector pulse is always recognized as a command byte. After the second byte, recognition of command bytes depends upon the sector format and the command being executed.

Console-File Disk Address (CFDA) Byte

- First byte read in any sector.
- Contains sector address (track and sector).
- Address range is 00 to FF (256 sectors).

		CFDA BYTE FORMAT						
		TRACK				SECTOR		
BIT	0	1	2	3	4	5	6	7
BIT VALUE	16	8	4	2	1	4	2	1

Track and Sector Determination Examples

Using assigned bit values:
CFDA byte = 1010 1100 (bin); AC (hex)
1. Track = 21 (total value of bits 01234).
2. Sector = 4 (total value of bits 567).

Using CFDA Track/Sector table:
CFDA byte = 1010 1100 (bin); AC (hex)
1. Locate "AC" in the table, or find bit configuration in left column and top row.
2. Get corresponding number from "TRACK" column on right (track = 21).
3. Get corresponding number from "SECTOR" row on the bottom (sector = 4).

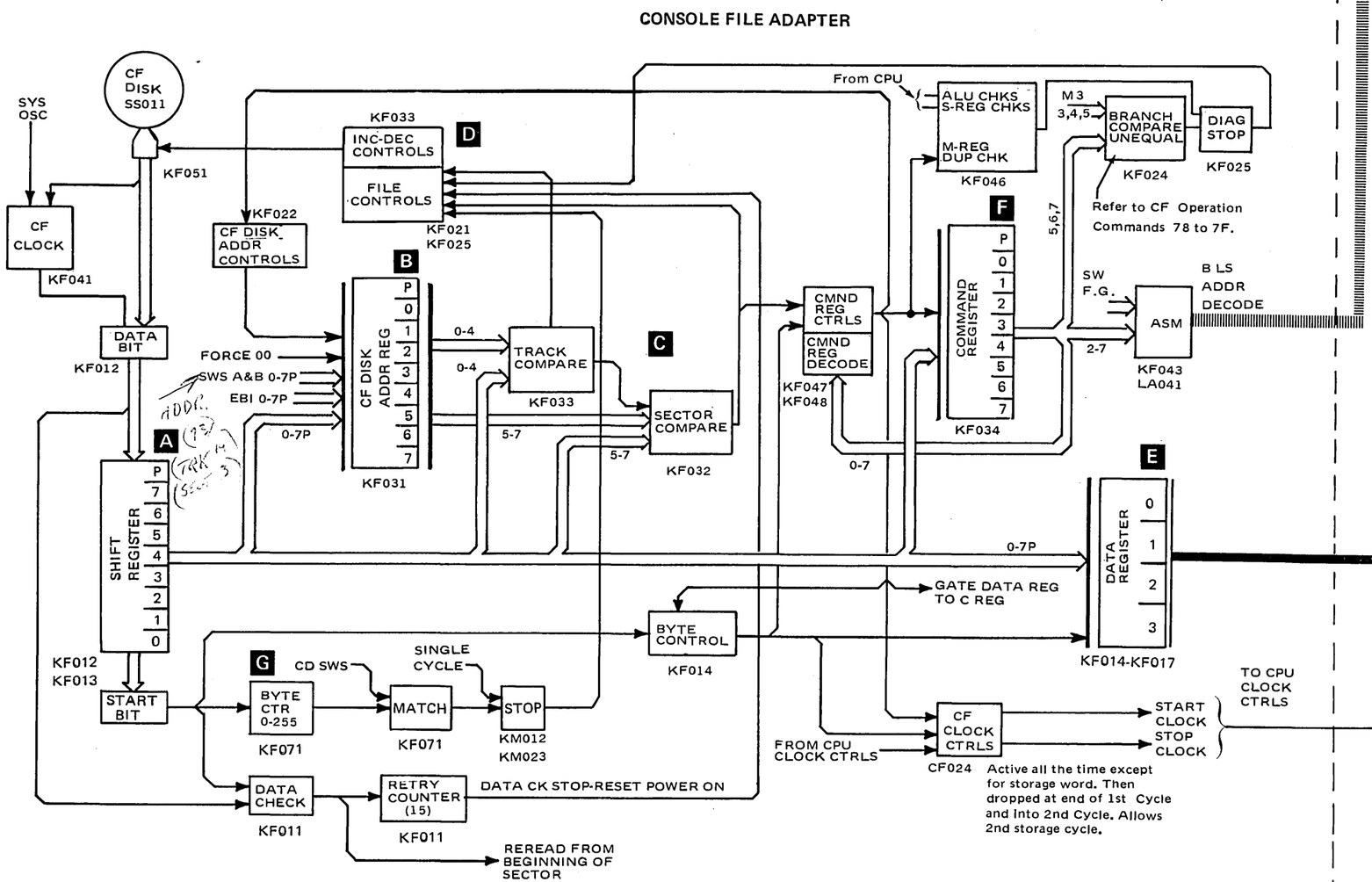
CFDA TRACK/SECTOR TABLE

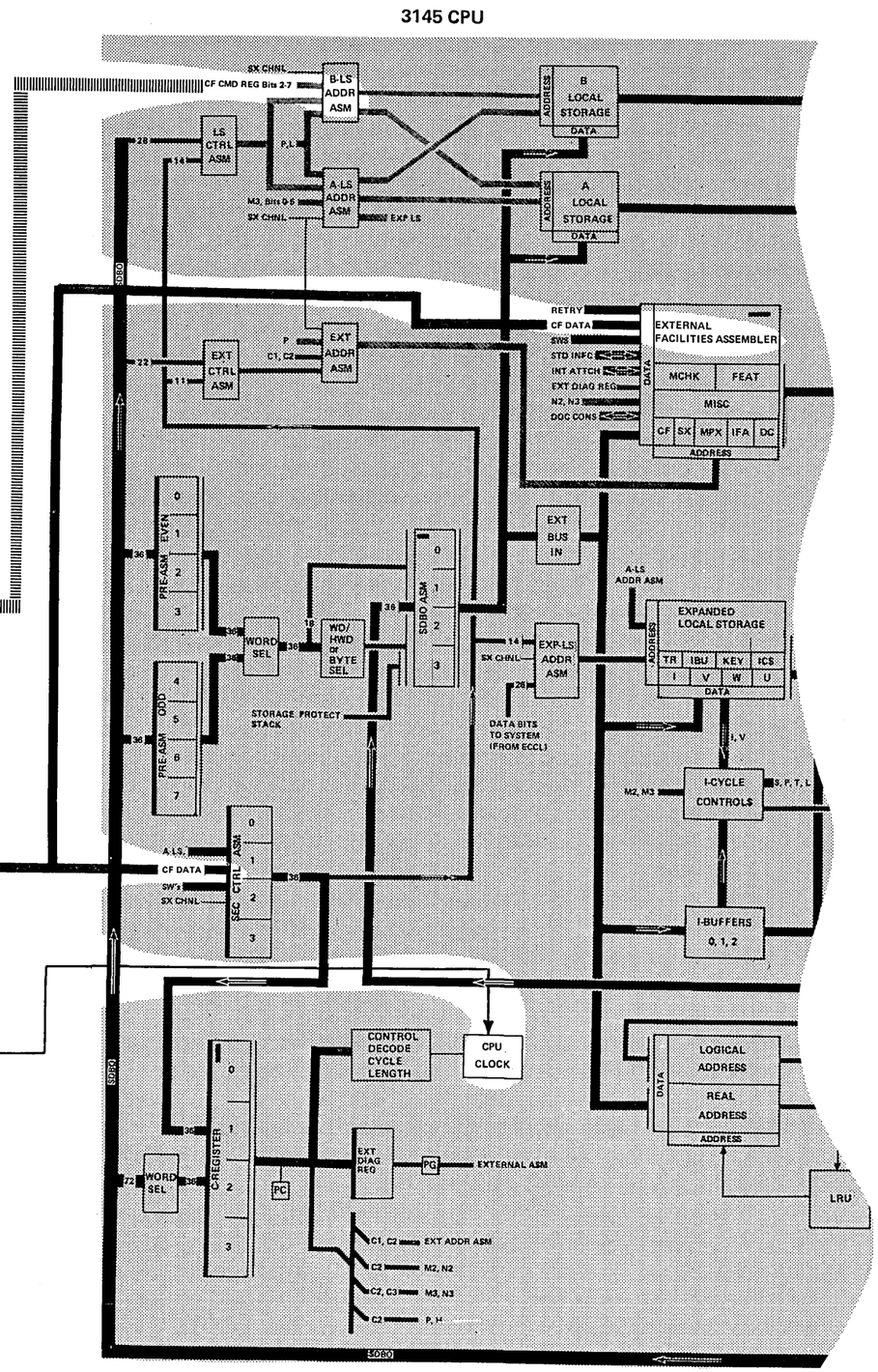
Bits 01234	Bits 5 6 7 000	001	010	011	100	101	110	111	TRACK
00000	00	01	02	03	04	05	06	07	00
00001	08	09	0A	0B	0C	0D	0E	0F	01
00010	10	11	12	13	14	15	16	17	02
00011	18	19	1A	1B	1C	1D	1E	1F	03
00100	20	21	22	23	24	25	26	27	04
00101	28	29	2A	2B	2C	2D	2E	2F	05
00110	30	31	32	33	34	35	36	37	06
00111	38	39	3A	3B	3C	3D	3E	3F	07
01000	40	41	42	43	44	45	46	47	08
01001	48	49	4A	4B	4C	4D	4E	4F	09
01010	50	51	52	53	54	55	56	57	10
01011	58	59	5A	5B	5C	5D	5E	5F	11
01100	60	61	62	63	64	65	66	67	12
01101	68	69	6A	6B	6C	6D	6E	6F	13
01110	70	71	72	73	74	75	76	77	14
01111	78	79	7A	7B	7C	7D	7E	7F	15
10000	80	81	82	83	84	85	86	87	16
10001	88	89	8A	8B	8C	8D	8E	8F	17
10010	90	91	92	93	94	95	96	97	18
10011	98	99	9A	9B	9C	9D	9E	9F	19
10100	A0	A1	A2	A3	A4	A5	A6	A7	20
10101	A8	A9	AA	AB	AC	AD	AE	AF	21
10110	B0	B1	B2	B3	B4	B5	B6	B7	22
10111	B8	B9	BA	BB	BC	BD	BE	BF	23
11000	C0	C1	C2	C3	C4	C5	C6	C7	24
11001	C8	C9	CA	CB	CC	CD	CE	CF	25
11010	D0	D1	D2	D3	D4	D5	D6	D7	26
11011	D8	D9	DA	DB	DC	DD	DE	DF	27
11100	E0	E1	E2	E3	E4	E5	E6	E7	28
11101	E8	E9	EA	EB	EC	ED	EE	EF	29
11110	F0	F1	F2	F3	F4	F5	F6	F7	30
11111	F8	F9	FA	FB	FC	FD	FE	FF	31
	0	1	2	3	4	5	6	7	
	SECTOR								

DATA FLOW

For detailed diagrams of console-file functional units and controls, refer to Diagrams 1-40 and 1-41 in IBM Maintenance Library, *3145 Processing Unit Maintenance Diagrams*, SY24-3580.

For power sequencing, refer to "Power On Flow Chart YE024.





SHIFT REGISTER A

Serial-read data from the file shifts through the single-byte shift register until the register is full (the start bit turns on).

When the byte in the shift register is a command byte, it is transferred to the command register. At the proper time, the command is decoded and executed. Bits 5, 6, and 7 are also used in the branch compare circuits. See "CF Commands," commands 78 to 7F.

When the byte in the shift register is a data byte, it is transferred to the four-byte console file data register (CFDR). The byte position entered in the CFDR depends upon the byte position within the console-file five-byte word. The word stored in the data register is normally a microprogram control word that is entered into the C-register or stored in control storage, local storage, or if running ASCP, main storage.

When the byte in the shift register is a console file disk address (CFDA) byte, it may be entered into the CFDA register or made available to the disk-address compare circuits to determine head movement.

CF DISK ADDRESS (CFDA) REGISTER B

The CFDA register consists of nine polarity holds and contains the address of the track and sector being read. A disk address may be entered into the register from the shift register, CPU switches A and B, microprogram (external bus-in), or forced to 00 by the adapter.

Initial entry is determined by the setting of the diagnostic/console file control switch. With the switch set to PROCESS/IMPL, the CFDA is forced to 00 when power is applied to the system or when the start console file key is operated. With the switch set to READ or RECYCLE, the CFDA register is set to the contents of CPU switches A and B.

When an IMPL is performed, the CFDA register is forced to 00 and the first sector is read. The address of the next sector to be read is set into the register by a control command that may be at any position in the sector or, by sector end at the end of each sector. After the address enters the register, it is available to the disk-address compare circuits at the next sector-address compare time.

Signals entering the disk-address compare circuits are available from the CFDA register and the shift register. Signals leaving the compare circuits are available to the head-control circuits and the command register.

COMPARE CIRCUITS C

The first CFDA byte read from the file enters the shift register and is compared to the CFDA register contents. If the track portion of the addresses do not compare equally, the 'sector ready' latch (which was set by a sector pulse to gate reading) resets, and reading stops. The read head moves one track before reading resumes. This operation continues until a track equal compare occurs.

When a track equal compare occurs, a sector search starts. After each sector pulse, the track and sector address byte is compared to the CFDA register contents. If the sector portions of the addresses are not equal, the 'sector latch' resets and reading stops until the next sector pulse.

When a sector equal compare occurs, reading continues and the next byte (always a command byte) enters the command register. Further operation is then determined by the command in the command register.

HEAD CONTROL AND TRACK ACCESSING D

No command is provided to move the head assembly from track to track. Track accessing occurs when the CFDA byte read from the file (into the shift register) and the address in the CFDA register do not compare equally.

When the track address read is higher than the address in the CFDA register, the head assembly moves one track toward the outside of the disk. A high compare sets the decrement latch, which energizes the OUT solenoid to start head movement.

When the comparison is low, the increment latch sets. This energizes the IN solenoid, and the head assembly moves one track toward the center of the disk.

The head lifts from the disk surface when either the 'ready' latch resets or the 'head' latch resets. Either of these conditions de-energizes the head magnet, which lifts the head from the disk surface.

CF DATA REGISTER (CFDR) E

The four-byte CFDR consists of 36 polarity holds (nine per byte). An operation command with bit 1 on must be in the command register to gate data into the CFDR. The first data byte read enters byte 0 of the CFDR. The second byte enters byte 1, etc.

Byte Control

Two flip flops control the gating of bytes from the shift register to the CFDR. When a command specifies that four bytes of data are following, both flip flops are set on (11). When the first data byte is in the shift register, both flip flops are set off (00). This gates the first data byte (byte 0) into the CFDR.

When the second byte of data is in the shift register, the first flip flop is set on (10) and gates byte 1. The flip flop sequence is.

Flip Flops	*Condition*
11	Reset
00	Gate byte 0
10	Gate byte 1
01	Gate byte 2
11	Gate byte 3

After the CFDR is full, its contents may be sent to the CPU.

COMMAND REGISTER F

The command register consists of nine polarity holds. Setting a byte into the command register is determined by three latches: 'command byte time', 'command register active', and 'command finished'.

After the CFDA byte is read and compared with the contents of the CFDA register, the next byte (always a command byte) enters the command register.

When the first command byte is in the shift register, the 'command byte time' latch sets. At CF clock-2 time, the command register and the 'command register active' latch set. The command being executed determines when the 'command finished' latch will be set. When 'command finished' sets, 'command active' resets. The next byte coming from the file is a command byte, and it is always set into the command register.

If the command being executed is a control command 1-byte op (next byte coming from the file is a command byte), the 'command finished' latch sets with 'CF control command finished'.

If the command being executed indicates that the next four bytes are data bytes, the 'command finished' latch sets after the last byte of data enters byte 3 of the CFDR.

If the command being executed is a control command 2-byte op (next byte coming from the file is a CFDA byte), the 'command finished' latch sets after the track and sector address enters the CFDA register.

When operating with the 'compact data mode' latch set, the 'command finished' latch may set when, or sometime after, the byte counter contains hex F9.

BYTE COUNTER G

The byte counter is a binarily coupled ring (1 resetting sets 2, 2 resetting sets 4, etc.). The counter is reset to FF before each sector is read. When a data byte, CFDA byte, or command byte enters the shift register, the counter advances by 1. The counter reaching FF with the file still reading sets the 'counter check' latch.

During some diagnostic tests, the byte counter is advanced by 17. This function is controlled by a CF command.

When the 'start bit' latch is on, the byte counter is compared with console file byte count switches C and D. A sync pulse is available as a maintenance aid when they compare equal. If the rate switch on the system control panel is set to CONSOLE FILE BYTE COUNT HARD STOP, the console file reading stops on an equal compare. For use of the byte counter as a maintenance aid, refer to "Chapter 5, System Control Panel."

CONSOLE FILE CLOCK

- The console-file clock supplies basic timings for controlling adapter functions. Clock timings are:

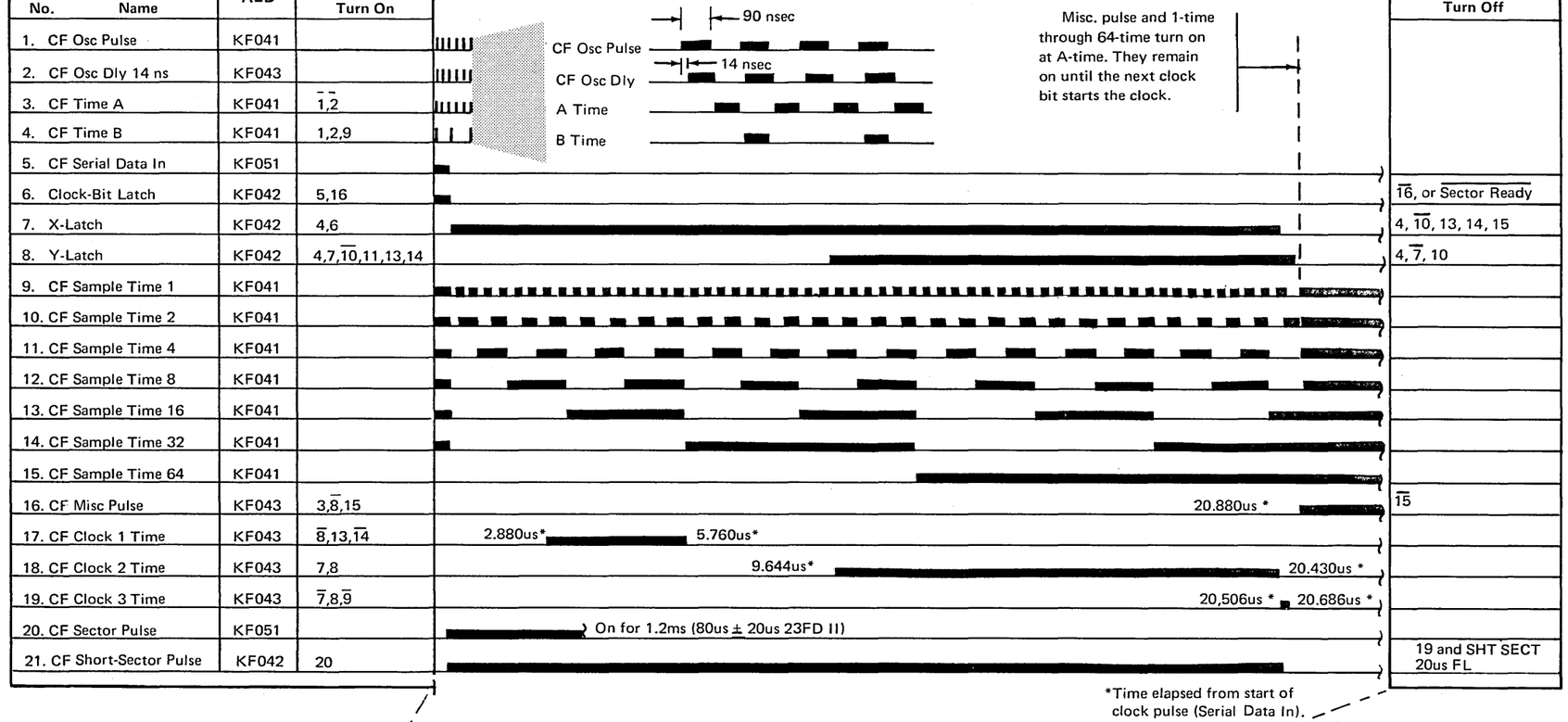
  CF clock 1
  CF clock 2
  CF clock 3
  CF miscellaneous pulse

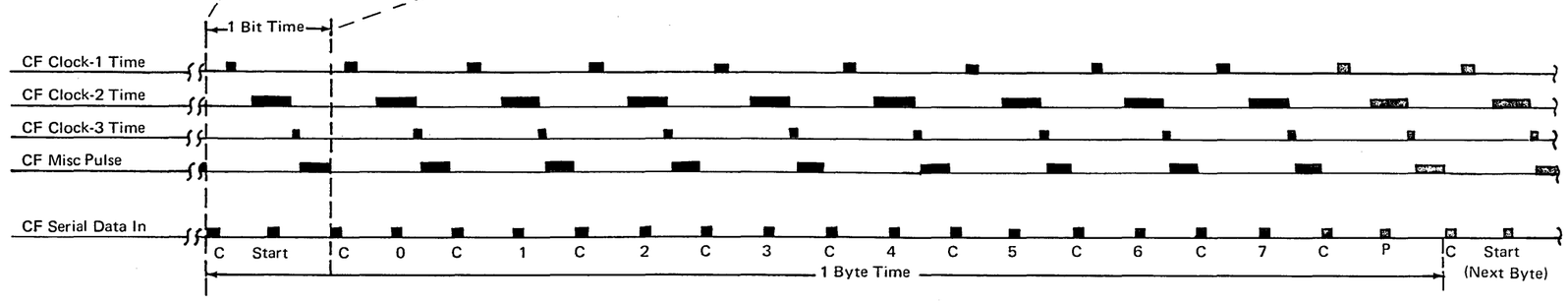
- Console-file clock timing is based on system-oscillator timing.

The clock is started either by the 'CF short sector pulse' line (see "Sector Ready") or by a clock pulse read from the file. Each time the clock starts, it runs for one cycle (one bit time) and then stops. The next clock pulse from the file restarts the clock for another cycle. Restarting the clock with each clock pulse synchronizes the attachment circuitry with the data reading from the file.

Using system oscillator pulses, console-file attachment circuitry develops 'CF Osc Dly 14 ns', 'CF time A', and 'CF time B'. The first clock bit read sets the 'clock bit' latch and starts the clock. The X- and Y-latches, and seven flip flops (CF sample-times 1 through 64) develop 'CF clock 1 time', 'CF clock 2 time', 'CF clock 3 time', and 'CF misc pulse'. The CF sample time flip flops are binarily coupled, and their advance is controlled by the 'CF time A' line.

No. Name	ALD	Turn On	Turn Off
1. CF Osc Pulse	KF041		
2. CF Osc Dly 14 ns	KF043		
3. CF Time A	KF041	$\overline{1,2}$	
4. CF Time B	KF041	1,2,9	
5. CF Serial Data In	KF051		
6. Clock-Bit Latch	KF042	5,16	$\overline{16}$, or Sector Ready
7. X-Latch	KF042	4,6	4, $\overline{10}$, 13, 14, 15
8. Y-Latch	KF042	4,7,$\overline{10}$,11,13,14	4, $\overline{7}$, 10
9. CF Sample Time 1	KF041		
10. CF Sample Time 2	KF041		
11. CF Sample Time 4	KF041		
12. CF Sample Time 8	KF041		
13. CF Sample Time 16	KF041		
14. CF Sample Time 32	KF041		
15. CF Sample Time 64	KF041		
16. CF Misc Pulse	KF043	3,$\overline{8}$,15	$\overline{15}$
17. CF Clock 1 Time	KF043	$\overline{8}$,13,$\overline{14}$	
18. CF Clock 2 Time	KF043	7,8	
19. CF Clock 3 Time	KF043	$\overline{7}$,8,$\overline{9}$	
20. CF Sector Pulse	KF051		
21. CF Short-Sector Pulse	KF042	20	19 and SHT SECT 20us FL





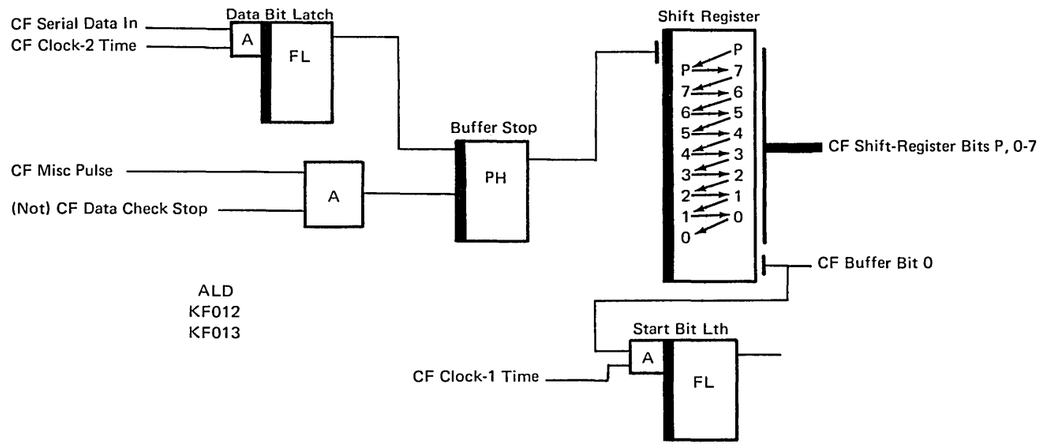
SHIFT REGISTER

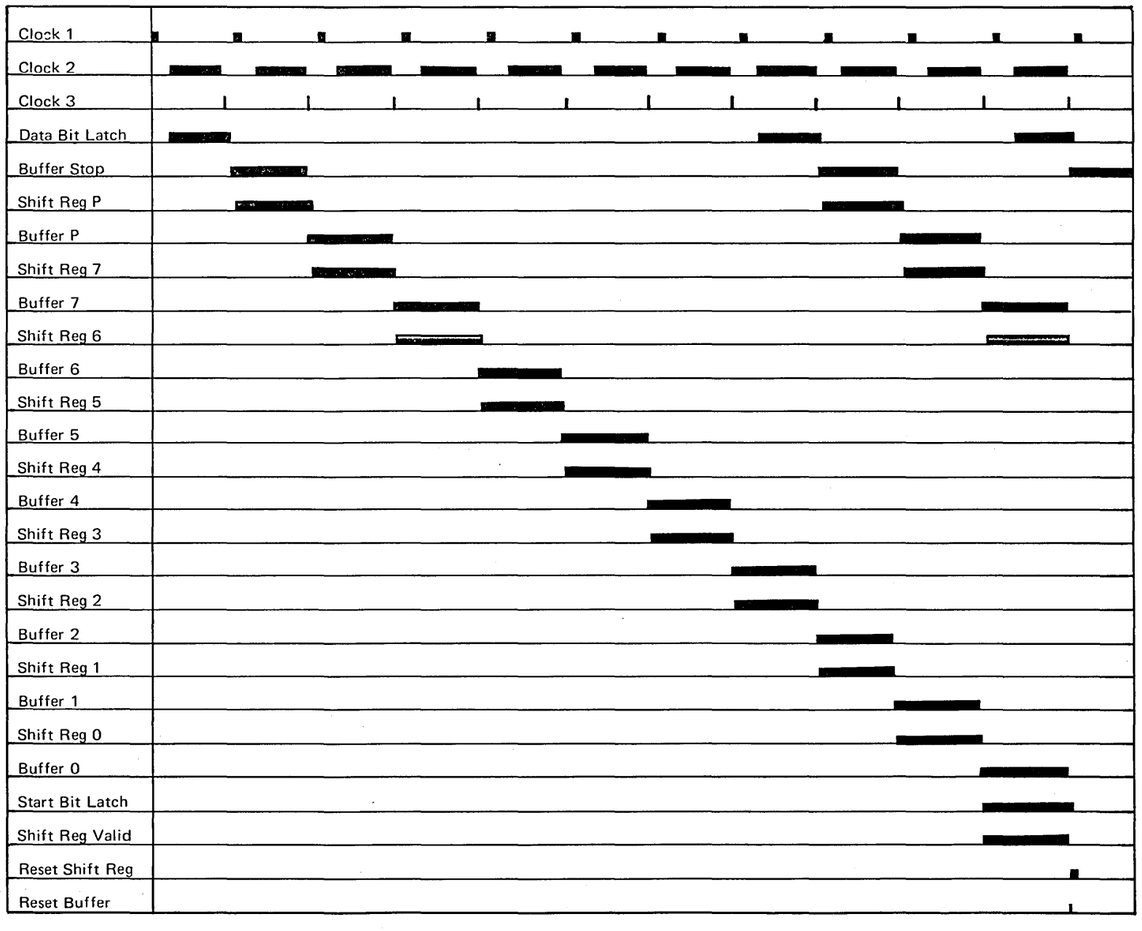
The shift register consists of ten buffer polarity holds, nine shift-register polarity holds (P and 0-7), the 'start bit' latch, and the 'data bit' latch. Serial-read data coming from the file enters the shift register via the 'data bit' latch. Only data bits enter the shift register (clock-2 time).

The first bit read is the start bit. The start bit is followed in order by bits 0 through 7 and the P-bit. The shift register is full when the 'start bit latch' is set.

Ten clock cycles are needed to fill the shift register. The shift register is gated to the data register, command register, disk-address register, or the compare circuits between clock-1 and clock-3 time of the eleventh clock cycle.

When data is being shifted through the shift register, the buffer polarity holds are set and reset a clock-3 time. When the shift register is full, the buffer polarity holds are zeroed by 'CF S-R valid clock 3'. During shifting, the shift-register polarity holds are set and reset by clock 1. When the shift register is full, it is zeroed by the first clock-1 time after 'CF S-R valid clock 3'.





This chart can be used with the diagram and timing chart to trace various byte configurations through the shift register.

Clock Cycle	1		2		3		4		5		6		7		8		9		10		11	
Time	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M	1,2	3,M
S=Shift B=Buffer	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S	B	S	B
Data-Bit Latch																						
Buffer Stop																						
PS																						
PB																						
7S																						
7B																						
6S																						
6B																						
5S																						
5B																						
4S																						
4B																						
3S																						
3B																						
2S																						
2B																						
1S																						
1B																						
0S																						
0B																						
Start Bit Latch																						

SECTOR READY

1. When power is applied to the console-file attachment circuitry, the 'sector latch' is set and the 'head latch' is reset by '(not) CF ready'. A
2. The first sector pulse activates 'CF short sector pulse'. B
3. 'CF short sector pulse' starts the clock and resets the 'sector latch'. C See note.
4. Resetting the 'sector latch' sets the 'head latch'. D
5. At 'CF clock-3 time', the 'Sht Sct 20 us' latch is set deactivating 'CF short sector pulse'. E
6. When 'CF sector pulse' drops, the 'Sht Sct 20 us' latch is reset. F

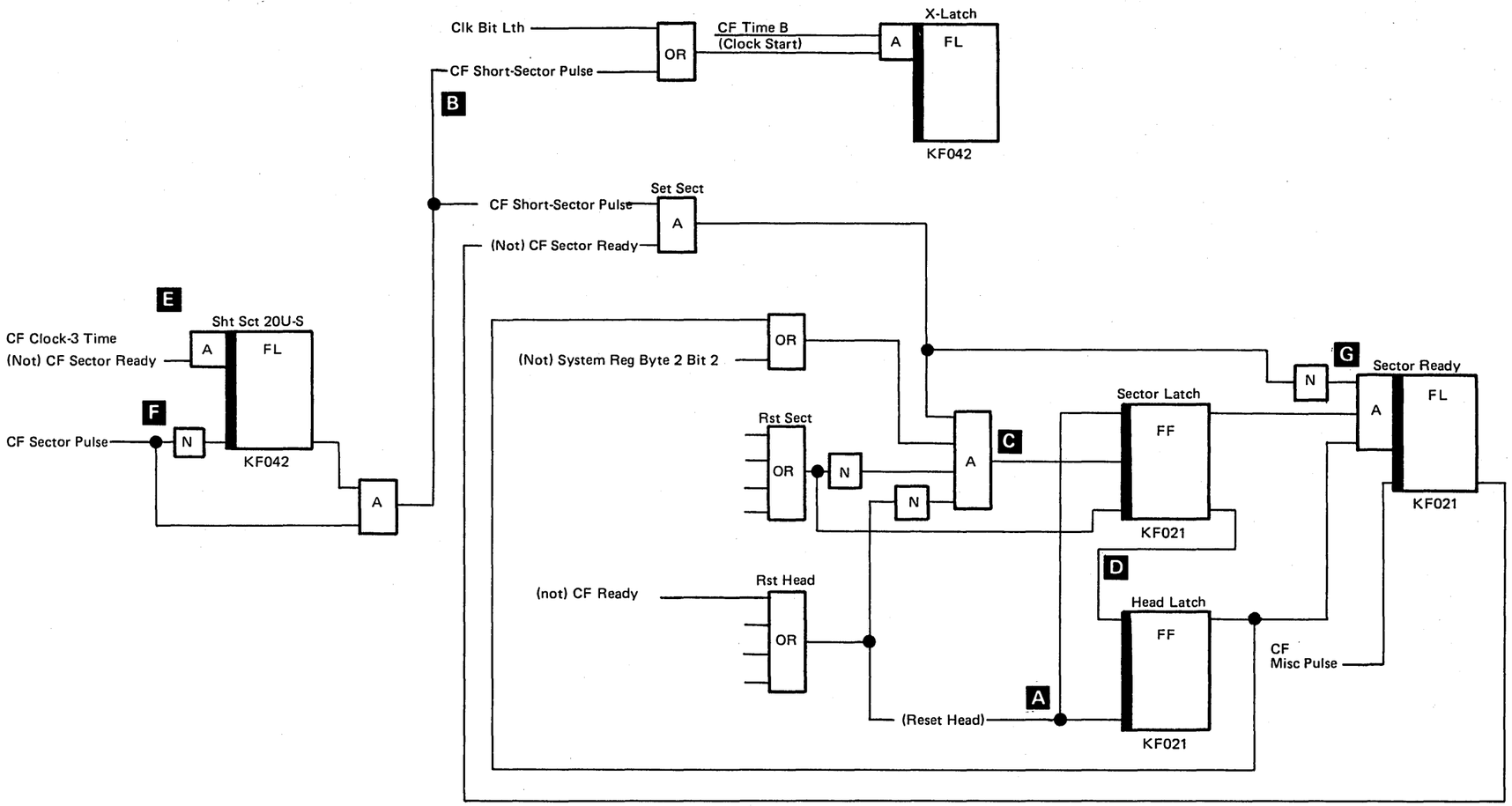
> The clock cycles only once for the first sector pulse. When the clock is started by 'CF short sector pulse', the clock bit coming from the file may be seen on the 'CF serial data in' line after the clock has started. This is normal operation when 'CF short sector pulse' starts the clock.

7. The second sector pulse activates'CF short sector pulse'. B
8. 'CF short sector pulse' sets the 'sector latch' and starts the clock. C
9. At 'CF clock-3 time' the 'Sht Sct 20 us' latch is set and the 'CF short sector pulse' line drops. The conditions needed to turn on the 'sector ready' latch are satisfied when 'CF short sector pulse' drops. E G

After the 'sector ready' latch is set, the clock is started by the clock bit coming from the file ('CF serial data in' line).

'Sector ready' remains on until'CF misc pulse' time of the cycle that either the 'sector Latch' or the 'head latch' resets.

Note: The presence of a disk on the file is indicated when the 'sector' and 'head' latches count two sector pulses.



CF COMMANDS

Console-file commands are of two kinds: operation commands and control commands. Operation commands perform operations with data that has been read from the file (not necessarily data immediately following the command byte). Control commands initiate direct action within the console-file adapter or perform diagnostic functions.

OPERATION COMMANDS

Command	Mnemonic	Hex	Function
CFDR to C-Register	C=R	20	Move CFDR to C-Register
CFDR to C-Register and Execute	C=R,X	30	Move CFDR to C-Register and execute C-register
Disk to C-Register	C=LR	60	Move disk data to CFDR to C-Register
Disk to C-Register and Execute	C=LR,X	70	Move disk data to CFDR to C-Register and execute C-Register
Disk to C-Register and Execute C-Register in compare mode	C=LR,mmm	78 to 7F	Move disk data to CFDR to C-Register and execute C-Register in compare mode
Execute C-Register with Direct Local Store Addressing, CFDR data	X, LS	80	Execute C-Register, use CFDR data, supply LS address from CF command bits 2-7
Execute C-Register with Direct Local Store Addressing, Disk Data	LR,X,LS	C0	Execute C-Register, use disk data, supply LS address from CF command bits 2-7

20 CFDR to C-Register
C=R

This command moves the four bytes in the CFDR into the C-register. This byte is followed by another command byte.

30 CFDR to C-Register and Execute
C=R,X

This command moves the four bytes in the CFDR to the C-register and executes them as a control word. The C-register is not changed after execution of the control word. This byte is followed by another command byte.

60 Disk to C-Register
C=LR

This command causes the four bytes following this command to read into the CFDR and then transfer into the C-register.

70 Disk to C-Register and Execute
C-Register
C=LR,X

This command causes the four bytes following the command byte on the disk to read into the CFDR, transfer into the C-register, and then execute as a control word. The C-register is not changed after execution of the control word.

78 to 7F Disk to C-Register and Execute
C-Register in Compare Mode
C=LR,mmm

These commands cause the four bytes following the command byte on the disk to read into the CFDR, transfer into the C-register, and then execute, in compare mode, as a control word.

Bit 4 equal to 1 indicates to the adapter circuitry that this function is to be performed in compare mode.

Execution of a word in the C-register results in setting up the next control-word address in the M (N) register even though that address is not used to access the next control word.

M3 bits 3, 4, and 5 are frequently set according to the results of branch testing. Bits 5, 6, and 7 of commands 78 to 7F are compared with M3 bits 3, 4, and 5 so that the setting of M3 bits 3, 4, and 5 can be checked. To use comparison checking, the CF command byte bits 5, 6, and 7 are set up so that the next control-word address is checked. Bits 5, 6, and 7 of the CF command byte correspond to M3 bits 3, 4, and 5 as follows:

Command Byte Bits	*M3 Bits*
5 6 7	3 4 5
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 0
0 1 1	0 1 1
1 0 0	1 0 0
1 0 1	1 0 1
1 1 0	1 1 0
1 1 1	1 1 1

If a mismatch occurs in compare-mode checking, the 'diagnostic stop' latch sets, CF and CPU operations stop (if diagnostic mode 3 is off), and the system diagnostic-stop check indicator turns on. If the command byte bits 5, 6, 7 match M3 bits 3, 4, 5, the console-file operation continues.

80 Execute C-Register with Direct Local Store Addressing, Console-File Register Data
X,LS

This command causes the CPU to execute the control word in the C-register (the C-register must contain a word-move word. The local-storage address for storage of CFDR data is decoded directly from bits 2 through 7 of the CF command byte. The four bytes in the CFDR are used as data. The C-register is not changed after the command is executed. This byte is followed by another command byte.

C0 Execute C-Register with Direct Local Store Addressing, Disk Data
LR,X,LS

This command causes the CPU to execute the control word in the C-register. Bits 2 through 7 of this command byte are the direct address of a local-storage register. This register contains the control- or main-storage address into which the CFDR data is to be stored. The four bytes following the command byte on the disk are read into the CFDR and used as data to be stored in this operation. The C-register is not changed after the command is executed. The four data bytes are followed by another command byte.

CONTROL COMMANDS

00/10 Compact Data Mode
MODE=DATA

This command sets the 'compact data' latch to allow data to load into the CPU without a command every four bytes (one control word). The byte following this command is another command byte. It is retained in the command register (Command Finished is inhibited) until the 'compact data' latch is reset. During reading, data-register byte-control flip flops control the operation. The 'compact data' latch resets when the byte counter equals F9 (250 bytes read), and normal operation resumes. The C-register must contain a storage word for this operation.

01/11 File Pause
FILEPAUSE

This command is normally used in diagnostic operations to execute a few control words from A local storage while waiting for the file to read the next byte. This command sets the 'file wait' latch (Sys 2, bit 2). The CPU clock starts, and the word currently in the C-register determines CPU operation. The address contained in the disk-address register is not changed by the microprogram routine.

The byte following a file-pause control command is always a command byte. If the microprogram does not turn off the 'file wait' latch before the next command byte is read, (300 us. nominal), the 'pause check' latch sets. (See "CF Error Checks.")

02/12 Byte Check
NOP

This command performs the odd/even byte-count check. (See "CF Error Checks.") It is always followed by another command byte.

03/13 Diagnostic Mode 3
MODE=D3

This command inhibits CF power off with diagnostic stop on to allow testing of the 'diagnostic stop' latch. Sys 2 bit 2 must be off.

Diagnostic mode 3 followed by File Pause forces Special File Pause; that is, same as File Pause except that file power may not be turned off on a diagnostic-stop condition.

Diagnostic mode 3 followed by File Wait forces Special File Wait; that is, same as File Wait except that Diagnostic Stop is inhibited.

If Diagnostic Stop is not reset when the storage microprogram resets file wait, the CF powers off.

Command	Mnemonic	Hex*	Function
Compact Data Mode	MODE=DATA	00/10	Sets 'compact data' latch.
File Pause	FILEPAUSE	01/11	Sets file-wait bit (Sys 2 bit 2) and continues reading. (300 us. max).
Byte Check	NOP	02/12	Performs odd-even byte check.
Diagnostic Mode 3	MODE=D3	03/13	Inhibits power off with Diag stop on. Followed by File Pause forces Special File Pause. Followed by File Wait forces Special File Wait.
M-Register Duplicate Check	STP=M-DUP	04/14	Stop if M-Reg duplicate check active.
S-Register Duplicate Check	STP=S-DUP	05/15	Stop if S-Reg duplicate check active.
ALU Check	STP=ALU	06/16	Stop if ALU logical check.
C-Register Parity Check	STP=C-PTY	07/17	Stop if C-Reg parity check (not Diag mode -1), or no C-Reg parity check (Diag mode-1).
Advance Byte Counter By +17	BYTCTR + 17	08/18	Advance the byte counter an extra 16 counts.
Diagnostic Stop	DIAG=STOP	09/19	Branch on Diagnostic Stop. If on: no action. If off: CF power off.
Force Parity CFDA/CMMD Regs.	CMD&ADR-P	0A/1A	Force P-bits in command and address Regs.
Error Check	STP=NO-CK	0B/1B	Stop if a console-file check not on.
Diagnostic Mode 1	MODE=D1	0C/1C	Set diagnostic mode 1, force P bits in CFDR, and inhibit CF checks except data check.
Diagnostic Mode 2	MODE=D2	0D/1D	Set diagnostic mode 2, inhibit CPU clock start, and inhibit CF checks except data check.
Diagnostic Mode Normal	MODE=NORM	0E/1E	Reset all CF diagnostic modes.
Reset CF checks	RST-CHKS	0F/1F	Reset all CF checks except data check.
File End	FILE-END	40/50	Turn off CF power and start CPU clock.
File Wait	FILEWAIT	41/51	Gate next byte into CFDA Reg. Set wait bit (sys 2 bit 2) and stop reading.
Sector End	SEND	42/52	Gate next byte into CFDA reg. Stop reading and seek to new address.
Conditional Sector End	SEND-IFMM	44/54	If bits 6 & 7 match M-Reg byte 3, bits 4 & 5, perform sector end; if not, continue reading.
Set CFDA No Sector End	SET-CFDA	4A/5A	Set CFDA Reg with next data byte and inhibit sector end.
Extra Bit Check	EXTRA-BIT	4C/5C	Force extra-bit check: data check.
Shift-Register Parity	SHIFT-PTY	4D/5D	Allow only start and stop bits to enter shift register. Force data check.
Block Stop Bit	NO-STPBIT	4E/5E	Block start bit of next byte: data check.
Odd-Even Byte Check	INVRTBIT3	4F/5F	Force odd-even byte check: data check.

*Bit 3 of each control command byte is formatted to permit an odd-even-count check of the bytes written in the sector. All bytes (CFDA, command, and data) are included in the count. The status of bit 3 is checked each time a control-command byte is read (see "CF Error Checks").

04/14 M-Register Duplicate Check
STP=M-DUP

If an M-register duplicate check is present when this command is executed, the 'diagnostic stop' latch sets and the CF powers off.

05/15 S-Register Duplicate Check
STP=S-DUP

If an S-register duplicate check is present when this command is executed, the 'diagnostic stop' latch sets and the CF powers off.

06/16 ALU Check
STP=ALU

If an ALU check is present when this command is executed, the 'diagnostic stop' latch sets and the CF powers off.

07/17 C-Register Parity Check
STP=C-PTY

This command sets the 'diagnostic stop' latch if either of the following conditions exists.

1. Correct C-register parity and diagnostic mode 1 is set.
2. Incorrect C-register parity and diagnostic mode 1 is not set.

08/18 Advance Byte Counter by +17
BYTCTR+17

This command is used to check the operation of the byte counter and the 'counter check' latch. This command advances the byte counter by 17. The counter is advanced by only +1 if bit 4 of the byte counter (value of 8 hex) is equal to a 1.

09/19 Diagnostic Stop
DIAG=STOP

This command branches on Diagnostic Stop: if on, no action; if off (diagnostic mode 3 must be off), CF powers off.

0A/1A Force Parity CFDA and Command Registers
CMD&ADR-P

This command forces parity bits on the outputs of the CFDA and command registers. If the parity bit is not present in either or both registers, a command register and/or CFDA register check should occur. If diagnostic mode 1 is not set, the CF powers off.

0B/1B Error Check
STP=NO-CK

This command tests for a CF error condition (address, command, byte counter, CPV clock start, or pause check). If no error condition exists, and if diagnostic mode 3 is reset, the 'diagnostic stop' latch sets and the CF powers off.

0C/1C Diagnostic Mode 1
MODE=D1

This command forces the parity bit ON in all bytes of the CFDR. If a CFDR to C-register operation is executed, a C-register parity check should occur. All error-stop conditions (except Diagnostic Stop if diagnostic mode 3 is off) are inhibited from removing power from the file. Data checks are recycled.

0D/1D Diagnostic Mode 2
MODE=D2

Allows all commands to be performed within the CF adapter circuitry and inhibits the following:

1. All error conditions (except Diagnostic Stop if diagnostic mode 3 is off) from turning off power.
2. CF 'operational' latch.
3. CF 'clock start' latch.
4. CPU 'clock start' line.

Data checks are recycled.

0E/1E Diagnostic Mode Normal
MODE=NORM

This command resets all three of the diagnostic modes.

0F/1F Reset CF Checks
RST-CHKS

This command resets File Wait (Sys 2 bit 2), Diagnostic Stop, and all CF error-check latches except data check.

40/50 File End
FILE-END

The file operation terminates, and console-file power turns off. The next byte is an address byte, and it enters the CFDA register. If no machine checks have occurred, the 'CF operational' latch resets and the CPU clock starts. The word in the C-register determines CPU operation.

41/51 File Wait
FILEWAIT

This command sets the file-wait latch (Sys 2, bit 2). This byte is followed by an address byte. After the address byte is stored in the CFDA register, the sector latch turns off to end reading, the CPU clock starts, and the word in the C-register determines CPU operation. Console-file power remains on. If the wait latch is not turned off before the next sector pulse is read, the head retracts from the disk surface.

This type of command byte is normally used in a diagnostic operation after A local storage or control storage is loaded with a set of control words. The control-word set is executed (from the A local storage) after the 'sector' latch is turned off. (Refer to "Operating Control Storage From Local Storage.") The address in the CFDA may be changed (only if CF power is on) during execution of the control words. At the end of the execution of the control words, the last control word executed turns off the 'file wait' latch so that reading from the file can resume.

42/52 Sector End
SEND

This command, normally, the last command byte in a sector, enters the next byte into the CFDA register. This indicates that the next sector is to be read. After the sector-end command byte is read, the 'sector' latch turns off until the next sector.

44/54 Conditional Sector End
SEND-IFMM

This command compares M3 bits 4 and 5 to bits 6 and 7 of the command byte. If the bits are the same, this command performs the same function as the sector-end command. If the comparison is unequal, the CFDA byte is ignored and the next command is taken from the byte following the CFDA byte.

4A/5A Set CFDA No Sector End
SET-CFDA

This command gates the next byte of data into the CFDA register and inhibits sector end. This command must be executed one byte before a force data-check command is performed.

4C/5C Extra-Bit Check
EXTRA-BIT

This command forces an extra bit at 1-time when the next byte is read. A data check is conditioned at start-bit and 3-time of this byte, and a recycle is performed to the address contained in the CFDA register. If the 'extra bit check' latch does not set, the 'diagnostic stop' latch sets and the CF powers off.

4D/5D Shift Reg Parity
SHIFT-PTY

This command allows only start and stop bits to enter the shift register. A shift-register parity check is generated, which forces a data check. Data check causes a recycle to the address contained in the CFDA register. If no parity check occurs, the 'diagnostic stop' latch sets and the CF powers off.

4E/5E Block Stop Bit
NO-STPBIT

This command blocks the start bit of the next byte from entering the shift register. This causes a data check and a recycle to the address contained in the CFDA register. If no data check occurs, the 'diagnostic stop' latch sets and the CF powers off.

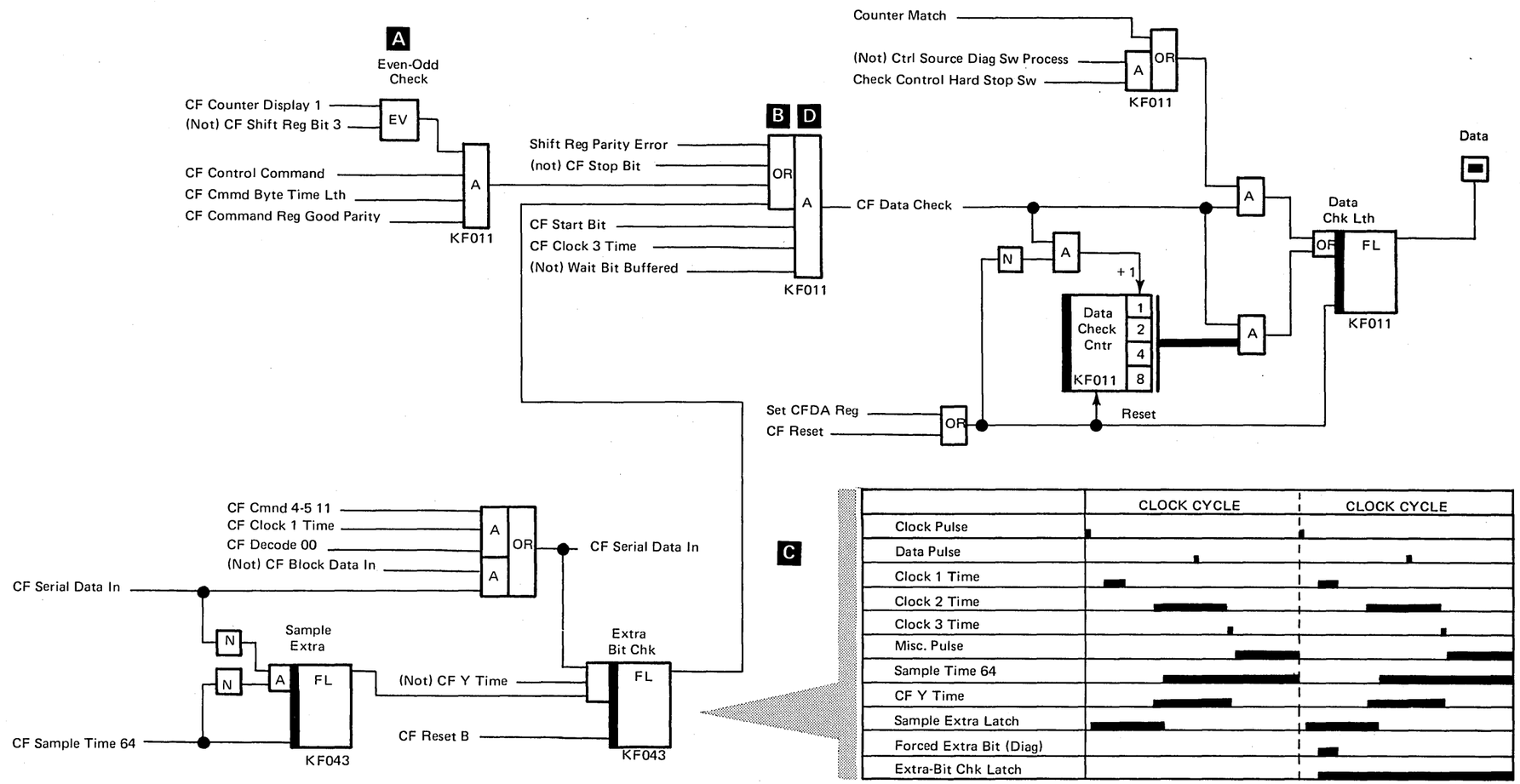
4F/5F Odd/Even Byte Check
INVRTBIT3

The microprogram assembler inverts the true value of bit 3 of this command byte before it is written on the file. When it is read from the file, it forces an odd/even byte check. This causes a data check and a recycle to the address contained in the CFDA register. If no check occurs, the 'diagnostic stop' latch sets and the CF powers off.

CF ERROR CHECKS

The action that occurs when a CF error is detected depends upon the settings of the diagnostic/console file control and check control switches. Results for each combination of switch settings are shown in the following chart.

SWITCH	CF ERROR CONDITIONS AND RESULTS				
	DIAG. CF/CONTROL SW	READ OR Process		RECYCLE	
	CHECK CONTROL SW	PROCESS	HARD STOP	PROCESS	HARD STOP
CHECK CONDITION	BYTE CNTR CMND REG DISK ADR REG PAUSE	STOP	STOP	RESTART	STOP
	CPU CLOCK START	STOP	STOP	RESTART	STOP
	DATA CHECKS Even-Odd Check Out of Sync/Missing Bit Extra-Bit Check Shift-Reg Parity Error	16 retries	STOP (Read) 16 retries (Process)	16 retries	STOP
	DIAGNOSTIC STOP (See 3145 MDM Diag. 1-41)	STOP	STOP	RESTART	STOP



DATA CHECKS

Even-Odd Check A

- Indicates incorrect number of bytes read from a sector.
- Checked each time a control-command byte is read.
- Can be tested by the force data-check command: INVRTBIT3 (hex 4F/5F).

Bit 3 of each control-command byte indicates whether the number of bytes from the beginning of the sector to the control-command byte is even (bit-3 = 0) or odd (bit-3 = 1). The byte-counter trigger-1 setting alternates as each byte (CFDA, command, or data) reads from a sector: 0 for byte 0, 1 for byte 1, 0 for byte 2, 1 for byte 3, etc. An unequal compare (trigger 1 ON and bit 3 OFF, or trigger 1 OFF and bit 3 ON) activates the 'CF Data Check' line.

Out-of-Sync or Missing-Bit Check B

- Indicates a failure to detect a start bit for the following byte.
- Checked when the start bit of the byte being read shifts from the shift-register buffer-0 position to the 'start bit' latch.
- Can be tested by the force data-check command: NO-STPBIT (hex 4E/5E).

Extra-Bit Check C

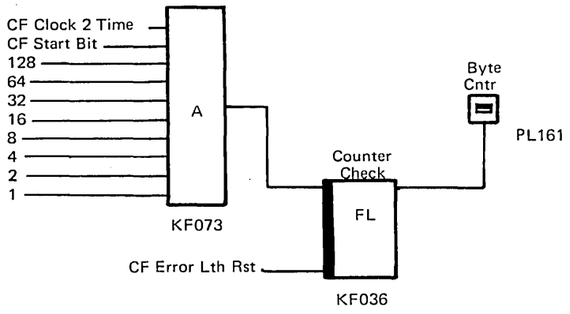
- Indicates detection of a bit on the 'CF Serial Data' In line during sample-extra time (between clock-bit time and data-bit sample time).
- Can be tested by the force data-check command: EXTRA-BIT (hex 4C/5C).

Shift-Register Parity Error D

- Indicates incorrect (even) parity in the shift register.
- Can be tested by the force data-check command: SHIFT-PTY (hex 4D/5D).

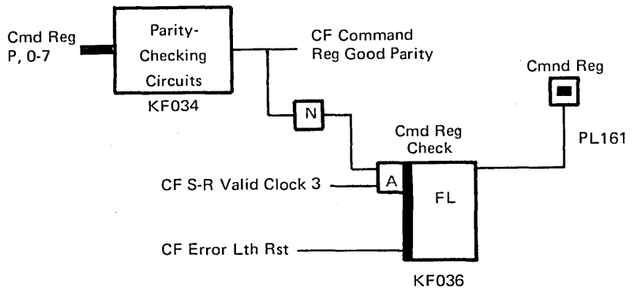
BYTE CNTR

- Indicates that the count in the byte counter has reached 255 and that the file is still reading.
- Can be tested by the control command: BYTCTR+17 (hex 08/18).



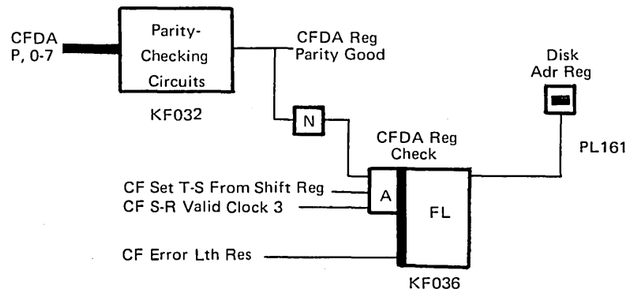
CMND REG

- Indicates command register incorrect parity.
- Checked each 'S-R Valid and Clock 3' time.
- Can be tested by the control command: CMD&ADDR-P (hex 0A/1A).



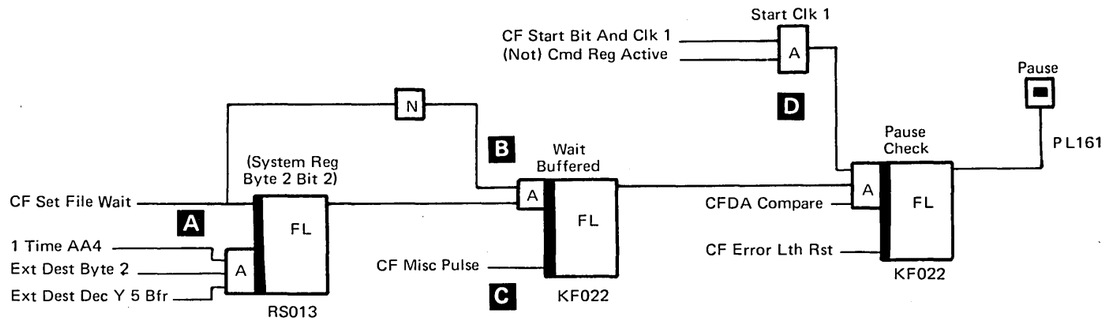
DISK ADR REG

- Indicates CF data address register incorrect parity.
- Checked each time CFDA register is set.
- Can be tested by the control command: CMD&ADR-P (hex 08/18).



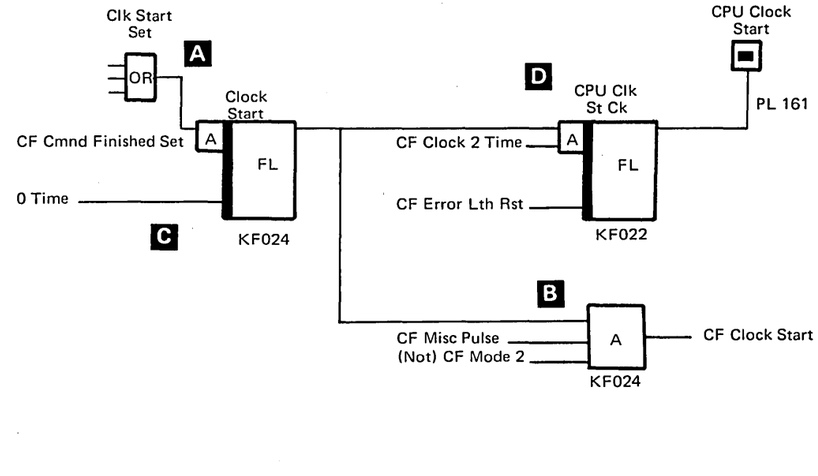
PAUSE

- Indicates that a file-pause operation did not end before the next command byte was read from the file.
  1. 'CF Set File Wait' (decoded from the file command) sets 'System Reg Byte 2 Bit 2' latch. A
  2. 'Wait Buffered' sets when 'CF Set File Wait' becomes inactive. B
  3. Reset to 'Wait Buffered' is not effective because the set remains active. C
  4. If microprogram does not reset 'System Reg Byte 2 Bit 2' to allow the reset of 'Wait Buffered', 'CF Start Bit And Clk 1' ANDed with 'CFDA Compare' sets 'Pause Check'. D



CPU CLOCK START

- Indicates that the CPU clock failed to start as requested by the console-file adapter.
  1. Console-file request for a CPU clock cycle sets 'Clock Start'. A
  2. The next 'CF Misc Pulse' brings up 'CPU Clock Start' to the CPU. B
  3. If the CPU clock fails to start, CPU clock '0 Time' does not reset 'Clock Start'. C
  4. Clock Start' remaining on allows the next 'CF Clock 2 Time' to set 'CPU Clk St Ck'. D

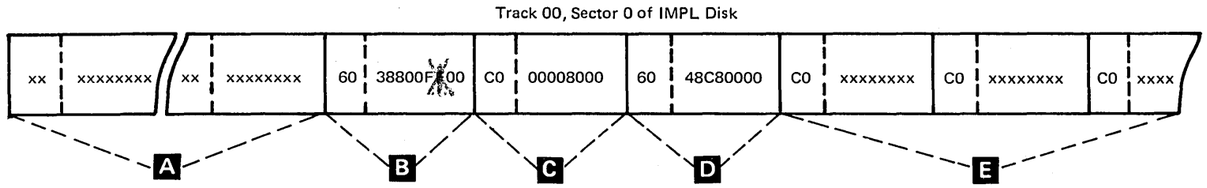


INITIAL MICROPROGRAM PROGRAM LOAD (IMPL) EXAMPLE

Objectives:

- Initialize CPU.
- Set up beginning control-storage address.
- Set storage word into C-register.
- Load control words into control storage.
- Update control-storage address for each control word loaded.

Note: This example IMPL is used to illustrate the data flow, control, and interaction between the console file and the CPU. IMPL routines may vary depending upon the EC level of the console file IMPL disk.



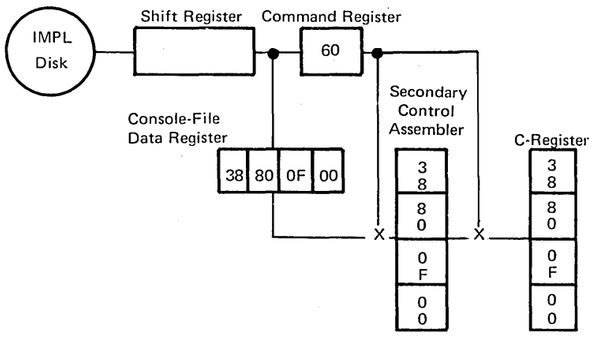
A Commands and data (CPU control words) in this area initialize:

LS01	H-register
SPTL registers	ACB register

These control words load directly into the C-register. They do not enter into local or control storage.

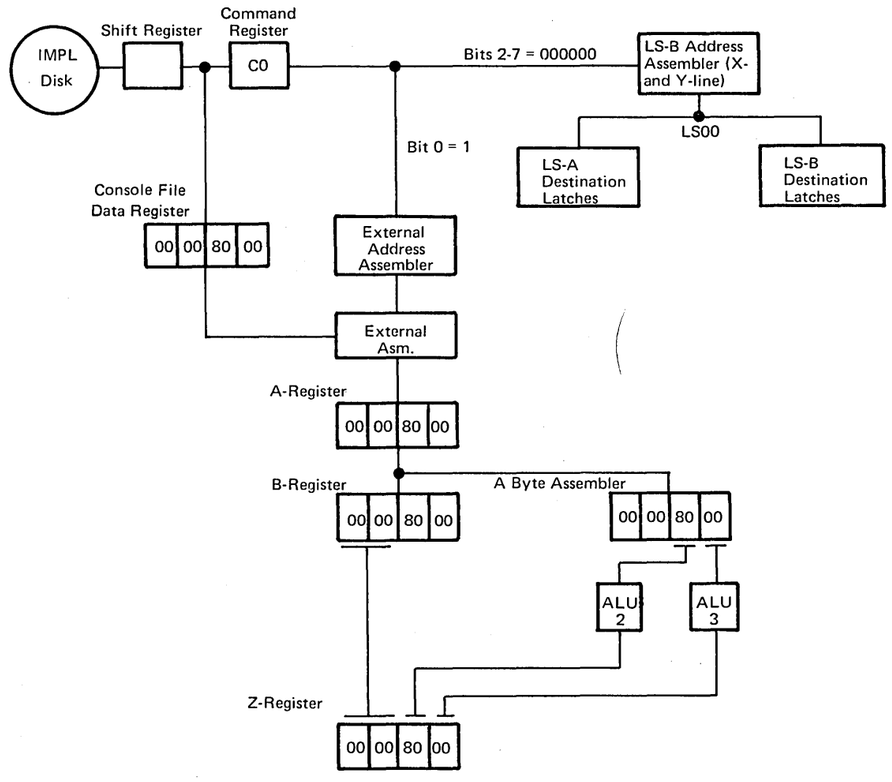
B Command 60 (Disk to C-Register).

1. Gate the next four bytes (38 80 0F 00) from the disk through the secondary control assembler to the C-register.
2. The CPU clock is not started.



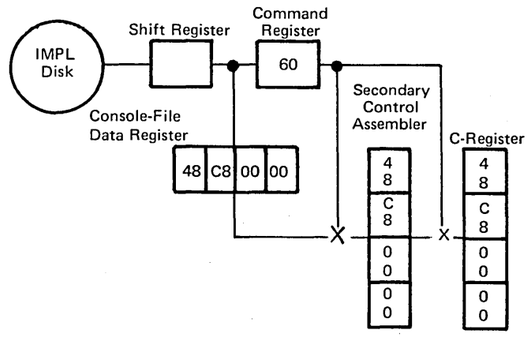
C Command C0 (Execute C-Register with Direct Local Store Addressing, Disk Data).

1. Read next four bytes (00 00 80 00) into CFDR.
2. Start CPU clock.
3. Word-move Word 38800F00 (previously set in C-register) executes, destining the CFDR contents to LS00.
   a. External source of CFDR is selected by CF command register bit 0 = 1.
   b. Destination of LS00 is selected by CF command register bits 2 through 7 = 000000.



D Command 60 (Disk to C-Register).

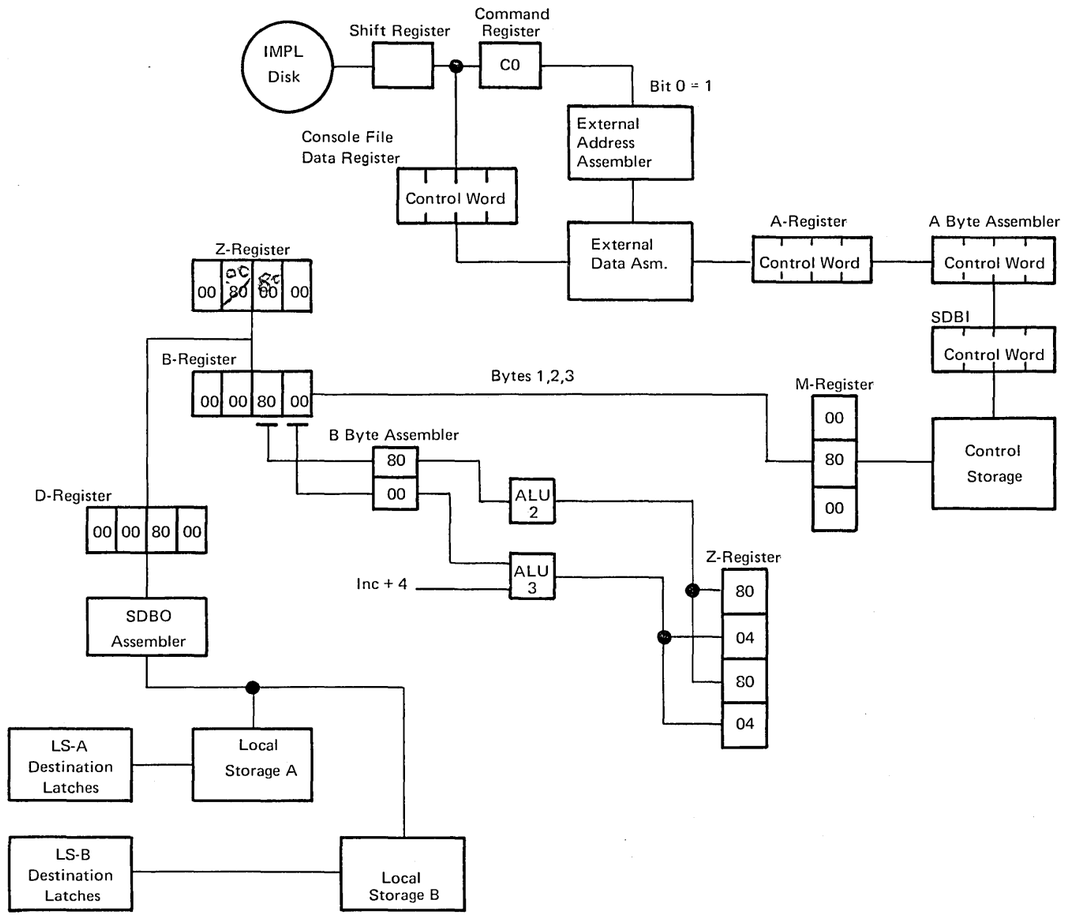
1. Gate the next four bytes (48 C8 00 00) from the disk through the secondary control assembler to the C-register.
2. The CPU clock is not started.



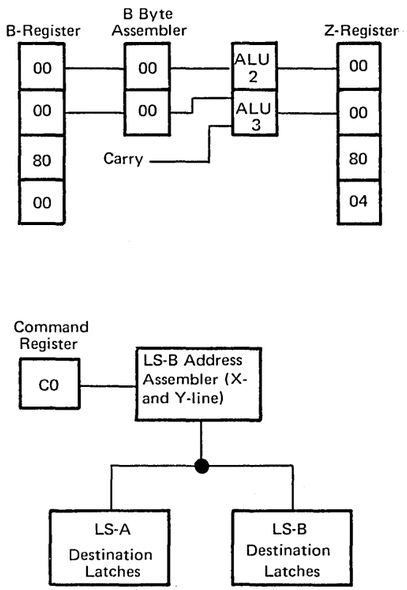
E Command C0 (Execute C-Register with Direct Local-Store Addressing, Disk Data).

1. Read the next four bytes (first control word to be loaded) into CFDR.
2. Start CPU clock.
3. Storage Word 48C80000 (previously set in C-register) executes, destining the CFDR contents to control storage.
   a. External source of CFDR is slected by CF command register bit 0 = 1.
   b. CF command register bits 2 through 7 = 000000 indicate that the contents of LS00 is the control storage destination address. Because the Z-reg contains the address destined to LS00 the control storage address is gated directly from the Z-register. (Refer to "Chapter 2. Functional Units, Local Storage", under the heading "Destination Look Ahead".)

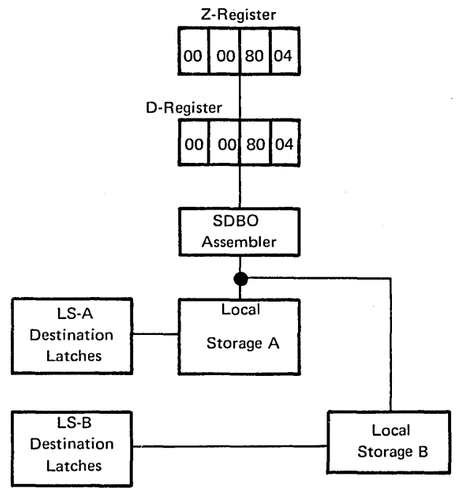
Store 1 Cycle



First Half Store 2-Cycle



Second Half Store 2-Cycle



CF REMOVAL and REPLACEMENT

Note: These procedures are for the removal and replacement of the console file only. For removal, replacement, and adjustment procedures for components of the console file, refer to the *Theory-Maintenance Manual, IBM 23FD Disk Drive,* SY26-4154, or the Theory-Maintenance Manual, *IBM 23FDII Disk Drive,* SY26-4175.

REMOVAL FOR SERVICE

1. Swing out the file assembly gate.
2. Raise the file assembly until the hinge pins disengage.
3. Set the file assembly on a working surface.

REMOVAL FOR REPLACEMENT

1. Swing out the file assembly.
2. Raise the file assembly until the hinge pins disengage.
3. Set the file assembly on the floor or console reading board.
4. Disconnect ac and dc cables from the file assembly.
5. Unplug the signal cable from the rear of the file.

File-Base Removal

1. Remove the left and right tilt arm stop studs. A
2. Remove the four lower mounting nuts on the rubber shock mounts securing the file to the mounting frame. B
3. Support the file base and slide it to the rear of the mounting frame.
4. Remove the tension from the left and right bracket springs. C
5. Place the file base on a working surface and remove the left and right tilt-out lever and bracket assemblies. D

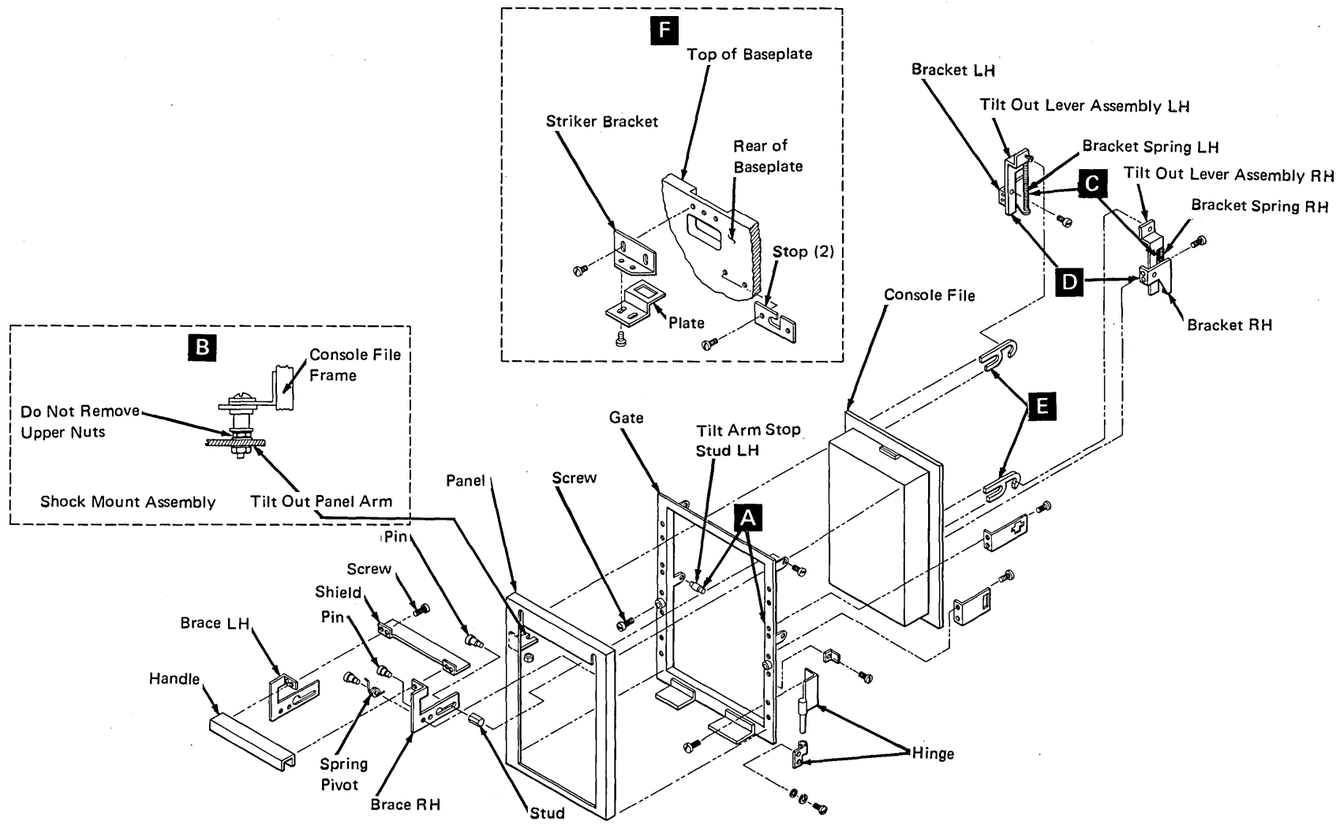
File-Base Replacement

1. Attach right and left tilt-out lever assemblies to file baseplate. D
2. Attach bracket spring to each tilt-out lever assembly. C
3. Hook right and left cover arms over tilt-out lever studs. E
4. Remove striker bracket, plate, and two stops from top rear of baseplate. F

   Note: These parts are used as cover retainers during shipment only and must be removed to allow proper operation of the file tilt-out feature.
5. Attach the file to the rear of the tilt-out panel by inserting the exposed portion of the long screws that run through each shock mount into the four slotted tilt-out panel arms. B
6. Support the file in place and add an additional nut to each shock-mount screw. Tighten nuts to secure file to tilt-out panel.
7. Attach tilt arm stop studs to file gate. A
8. Lubricate all pivoting points of the tilt-out frame and gate assembly with IBM 10 oil or its equivalent and all sliding parts with IBM 23 grease or its equivalent.

CONSOLE-FILE REPLACEMENT

1. Insert file gate brackets in the frame hinges.
2. Connect ac, dc, and signal cables.
3. Swing in the file assembly.
4. Run basic diagnostics.



CHAPTER 7. CONSOLE PRINTER-KEYBOARDS

REMEMBER

There is a Reader's Comment Form at the back of this publication.

INTRODUCTION

- The console printer-keyboard (PR-KB) is an auxiliary input-output device for manually altering or displaying:
  - Main storage
  - Local storage
  - Control storage
  - Storage protection key
  - Control registers
  - Floating-point registers
  - General registers
  - Current PSW
- Depending on its configuration, the system uses either a 3210 Model 1, or a 3215, as the attached printer-keyboard.
- A 3210 Model 2 is also available as a remote printer-keyboard, if desired.

  The printers are identifiable by bit 3 of byte 0 displayed in the A-register roller switch set to position 4 (System Register) as follows:
  - 0 = Local PR-KB--3210 Model 1 or 3215
  - 1 = Remote PR-KB--3210 Model 2

- With either PR-KB, the input function of the keyboard is independent of the output function of the printer, although these two units may be physically connected.
- In addition to printing, the PR-KB has only two other functions: space and new line.

  Space: advancing the print element one character-position to the right along the print line without printing.

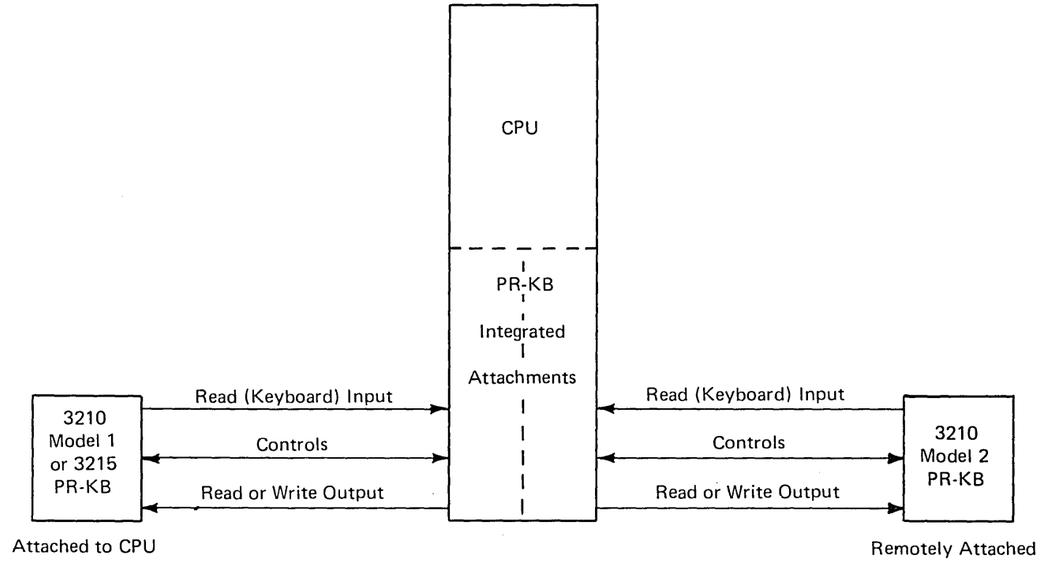
  New Line: returning the print element to the left margin accompanied by a line feed (indexing operation).
- The PR-KB is controlled by microprograms working through the integrated attachment circuitry in the processing unit.
- Data entered from the keyboard returns through the attachment circuitry to the printer and is printed.
- Regardless of the model used, all of the PR-KBs respond to the same PR-KB commands, use the same pin-feed platens, and have the same operator's console.

  This console consists of an 88-character keyboard and a set of control keys and indicators for use with the System/370. These keys and indicators are the same for the 3210 Model 1 and the 3215, but differ somewhat for the 3210 Model 2. They are described separately in each section.

This chapter describes the PR-KB and the integrated attachments for both the 3210 and the 3215. The chapter is divided into the following main categories:

- 3210 Console Printer-Keyboards
- 3210 PR-KB Integrated Attachment
- 3215 Console Printer-Keyboard
- 3215 PR-KB Integrated Attachment
- Alter/Display Operations
- Microprogram Operations
- Programming Information

PR-KB CONFIGURATIONS



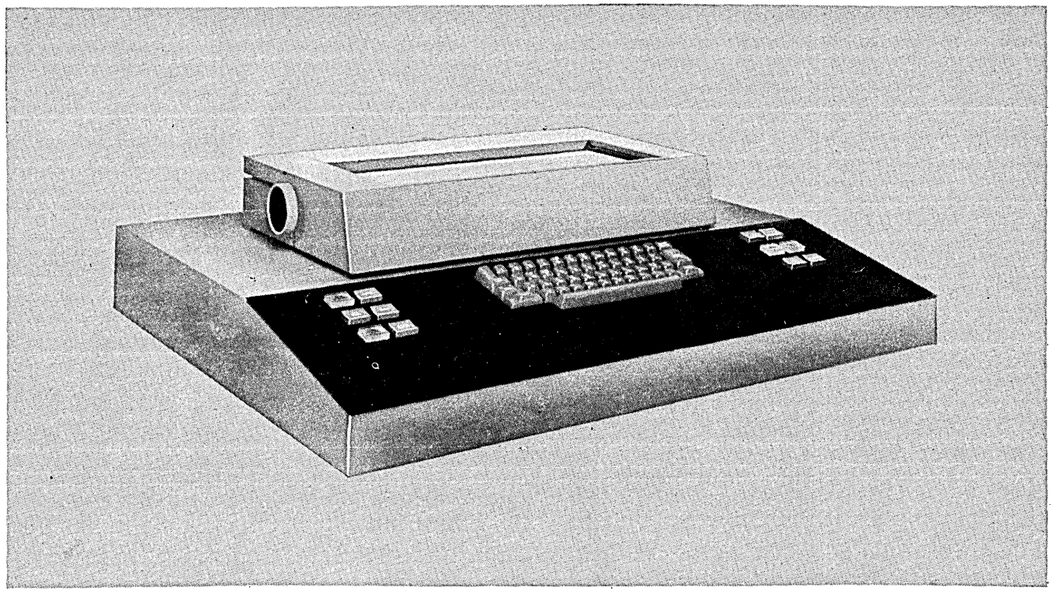
3210 Console Printer-Keyboards

- Both models of the 3210 use the Selectric® I/O-II printer with integrated keyboard.
- Both models have an operator's console consisting of the keyboard and a set of control keys and indicators for use with System/370.
- Diagnostic information is presented under "PR-KB Troubleshooting Hints" and "Console-Printer Maintenance Aids" in the Appendix.
- Maintenance information about the covers, interface connections, control keys and indicators may be found in:

  Theory-Maintenance Manual for the 3210 Console Printer-Keyboards, SY24-3559.
- Comprehensive second-level supplemental drawings for 3210 Controls are contained in 3145 Processing Unit Maintenance Diagrams manual, SY24-3580.

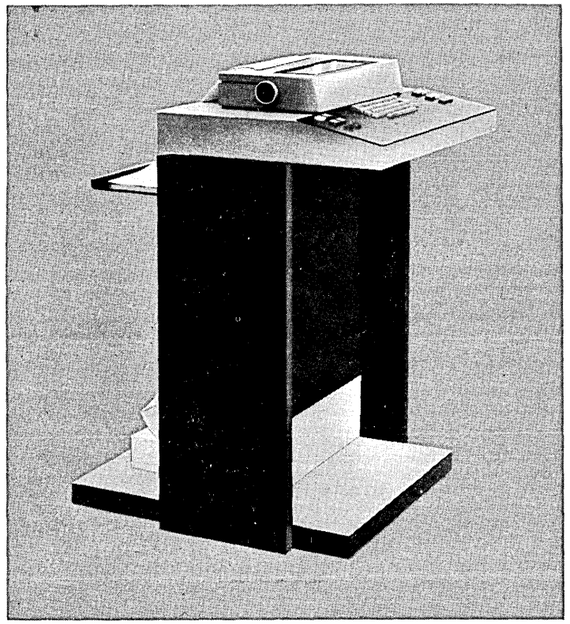
3210 Console Printer-Keyboard Model 1

- The 3210 Model 1 is mounted on the operator's table at the CPU.



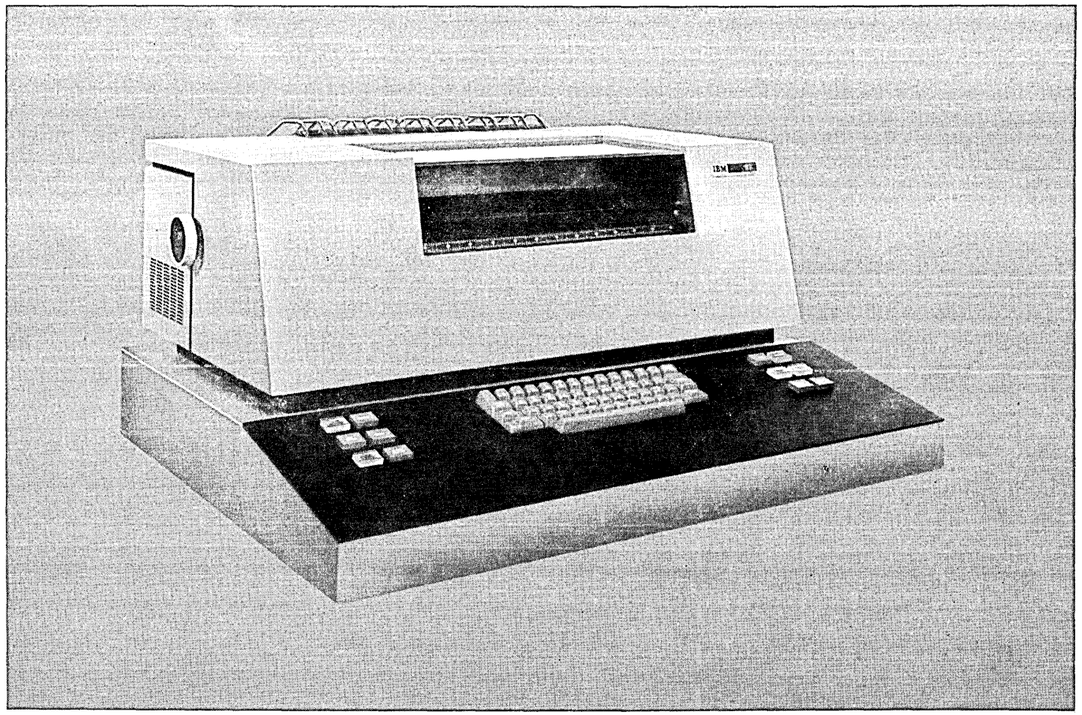
3210 Console Printer-Keyboard Model 2

- The Model 2 is pedestal-mounted and is remotely located. It is attached to the CPU by a connecting cable.
- The Model 2 has no alter/display capabilities; therefore, the associated operator's control key and indicator light for this function are not provided with this model. In the place of this key and light, the Model 2 has its own power-on and power-off keys. In all other respects it is the same as the Model 1.
- Maintenance information references for this model are the same as those given for the Model 1.



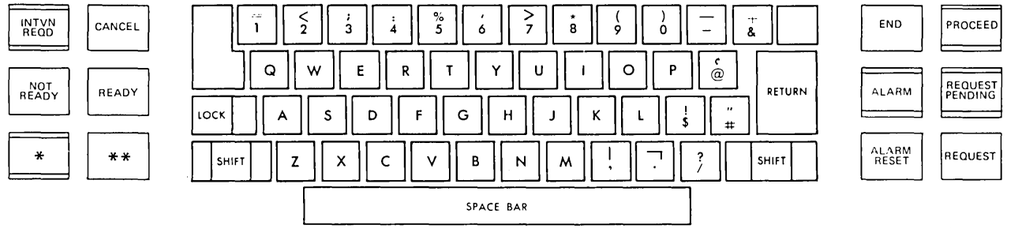
3215 Console Printer-Keyboard

- The IBM 3215 Console Printer-Keyboard is a wire-matrix printer having a separate keyboard.
- Diagnostic information is presented under "PR-KB Troubleshooting Hints" and "Console-Printer Maintenance Aids" in the Appendix.
- Maintenance information on this PR-KB may be found in Theory-Maintenance Manual for the IBM 3215 Console Printer-Keyboard, SY24-3560.
- Comprehensive second-level supplemental drawings for 3215 Data and Motor Controls are contained in 3145 Processing Unit Maintenance Diagrams manual, SY24-3580.



Operator's Console

The operator's console contains the keyboard and associated control keys and indicator lights. The control keys and indicator lights are on both sides of the keyboard. The keyboard is similar in appearance to that of the standard Selectric I/O printer except that the Tab and Backspace keys are blocked and not labeled. The character set represented by the 44 keys is clearly shown and requires no further explanation. Operation of the keys, however, differs in the way that the 3210 and the 3215 operate and encode the characters. The differences are described in separate sections of this chapter for these two PR-KBs.



INDICATORS KEYS

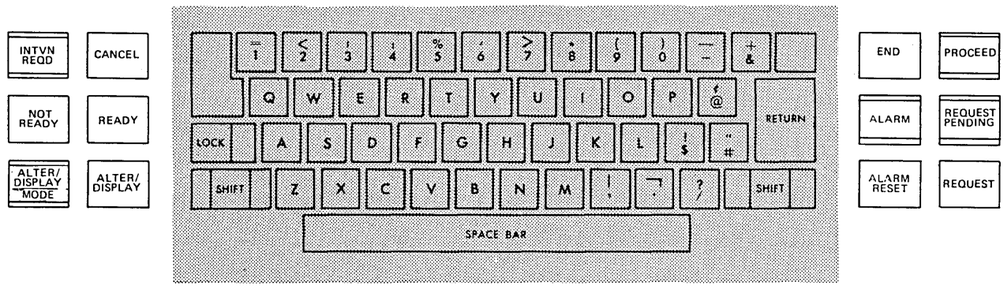
Legend

*ALTER/DISPLAY MODE (Indicator), 3210 Model 1 and 3215 only
POWER ON (Backlighted Switch), 3210 Model 2 only.
**ALTER/DISPLAY (Switch), 3210 Model 1 and 3215 only.
POWER OFF (Switch), 3210 Model 2 only.

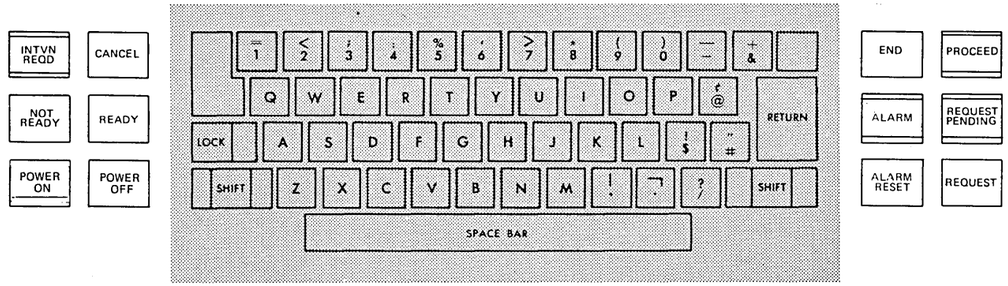
Control Keys and Indicators

- The control keys and indicators are tied-in directly with the TE-register. Their purposes are described here; their operation is described under "TE Register."

3210 Model 1 and 3215



3210 Model 2



Legend

INDICATORS KEYS

Legend

INDICATORS KEYS

CONTROL KEYS (SWITCHES)

The control keys are labeled according to the functions they perform. Except as indicated, these keys operate in the same way for all the printer-keyboard options and are described here in alphabetical order.

Alarm Reset: This key turns out the alarm indicator light if it is on.

Alter/Display: This key causes a request for an alter/display operation if the CPU is stopped. The alter/display mode indicator light turns on when the request is honored. The proceed light must also be on before any data may be entered from the keyboard.

This key also halts any alter/display operation that may be in process without taking the PR-KB out of the A/D mode. The proceed light will remain on, and the operator can begin to key another A/D operation.

On the 3210 Model 2, this key is replaced by the ac power-off switch.

Cancel: This key terminates a read operation and signals the system to disregard the data being transmitted.

End: This key ends either a read, write, store, or display operation.

Not Ready: This key makes the printer-keyboard "not ready" and puts it in an offline condition.

Power Off: This key (3210 Model 2 only) turns off ac power to the printer drive motor.

Power On: This key (3210 Model 2 only) is a backlighted switch that turns on ac power to the printer drive motor.

Ready: This key places the printer in a "ready" state such as after forms have been correctly positioned.

Request: This key causes the attachment circuitry in the attached system to attempt to present attention status to the CPU. The request-pending indicator turns on until the attention status is cleared. The program is normally written so that a read command is issued to the printer-keyboard attachment as a result of recognition of the attention status by the program.

Alarm: This light indicates that an alarm command was issued by the attached system. Note that the alarm itself is part of the attachment in the system; it is not found in the printer-keyboard.

Alter/Display Mode: This light indicates that a request for an alter/display operation has been honored.

On the 3210 Model 2 this light is replaced by the ac power-on switch.

Intvn Reqd (Intervention Required): This indicator lights when the printer is out of forms or the not ready switch has been operated. The intervention required indicator turns off when forms are properly loaded, and the ready switch is operated.

To turn off the light,

1. Load new forms.
2. Operate the ready switch.

Power On: This light (3210 Model 2 only) is part of the power-on switch (key). It lights whenever ac power to the printer drive motor has been turned on by operation of this key.

Proceed: When on, this light indicates that the operator can type input at the keyboard. It is turned on as a result of any of the following.

1. When an alter or display operation has been requested and the attachment is at a point at which typed input is required for the store or display operation.
2. As a result of a request-key operation. The program, in this case, has issued a read command to the printer-keyboard as a result of accepting the attention status caused by the request-key operation.
3. When the attachment has accepted a read command that has nothing to do with a request-key operation.

If the request key is pressed, and the adapter accepts an unrelated read command (one that is not the result of pressing the request key), both the request pending and the proceed lights will be on.

Request Pending: This indicator, when on, indicates that a request-key operation has been initiated, but that attention status has not yet been accepted by the CPU.

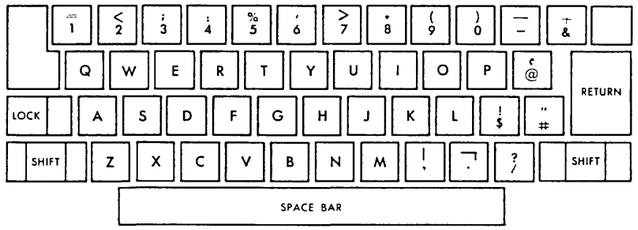
3210 CONSOLE PRINTER-KEYBOARDS

3210 KEYBOARD

- The printer and keyboard are mechanically connected but are electrically independent in operation.
- Contacts in the keyboard send coded signals to the adapter for each graphic and function character that is keyed.

The keyboard of the console printer is similar to that of the standard Selectric I/O printer except that the Tab and Backspace keys are blocked and unlabled.

The printer operational shaft restores the keyboard mechanically each print cycle.



STROBE

Strobe is a line brought up by the strobe contact in the keyboard, after the data contacts have made, to signal the control unit that data is available at the keyboard.

3210 KEYBOARD CODES

The keyboard transmits data to the system, using an eight-bit keyboard code. This code consists of the standard six BCD bits generated by the keyboard contacts, plus two bits to indicate case (shift). The BCD code consists of six bits plus parity: B, A, 8, 4, 2, 1, and C.

Each of the 44 printable (graphic) character keys, as well as the space bar, shift, and return keys, generates its own BCD code.

Because the single BCD code generated by each of the keys can represent either of its two associated graphic characters, the shift key contact determines which character will be transmitted.

The shift key is unique in that it generates two BCD codes.

Pressing either shift key during a read opeation generates the uppercase code (842). When the shift key is released, the lowercase code (BA842) is generated. Pressing the lock key generates the uppercase code and locks the shift. The shift is unlocked by pressing either shift key. As before, the lowercase code is generated when the shift key is released.

This code, like the keyboard code for any other character, is transmitted to the system, where the microprogram identifies it as a shift code rather than a character code; then does nothing further with it.

In addition to generating the BCD code, the shift key has a contact that signals the system directly regarding its uppercase (shift) or lowercase (no-shift) status. This contact adds two bits (0 and 1) to the BCD code to further indicate uppercase or lowercase characters.

Note: A single bit suffices to do this, but adding two bits (0 and 1) rounds out the BCD coding to an eight-bit configuration acceptable to the system, and at the same time maintains parity.

For uppercase characters, the two bits are both ones; for lowercase characters, they are both zeros.

For CPU use, the 3210 keyboard code must be translated to EBCDIC as described under "3210 PR-KB Code Translation." The keys and lights on either side of the typewriter keys are described under "Control Keys and Indicators."

3210 PRINTER

- The 3210 console printer is a version of the IBM Selectric ® I/O printer known as the Selectric I/O-II.
- The print element has a preferred character arrangement that was custom-designed for the System/370.

This section describes the optimized arrangement of characters on the print element, and the tilt/rotate codes used to position them for printing. These codes, supplied as output from the TE (Write Data) register in the attachment, are shown in the chart at the right.

Each half of the print element (side 0-front, side 1-back) has 44 graphics.

In write (print) operations, the EBCDIC representation of the character to be printed is modified (as described under "PR-KB Code Translation") and is used to address the corresponding tilt/rotate code in the translation table. The tilt/rotate code then enters the TE (write data) register. The 7-bit from the TE register selects the appropriate side of the print element, depending on whether the bit is 0 or 1.

This selection has nothing to do with uppercase or lowercase, but is governed by the preferred location of the graphic on the print element. The keyboard and tilt/rotate codes are entirely unrelated and independent.

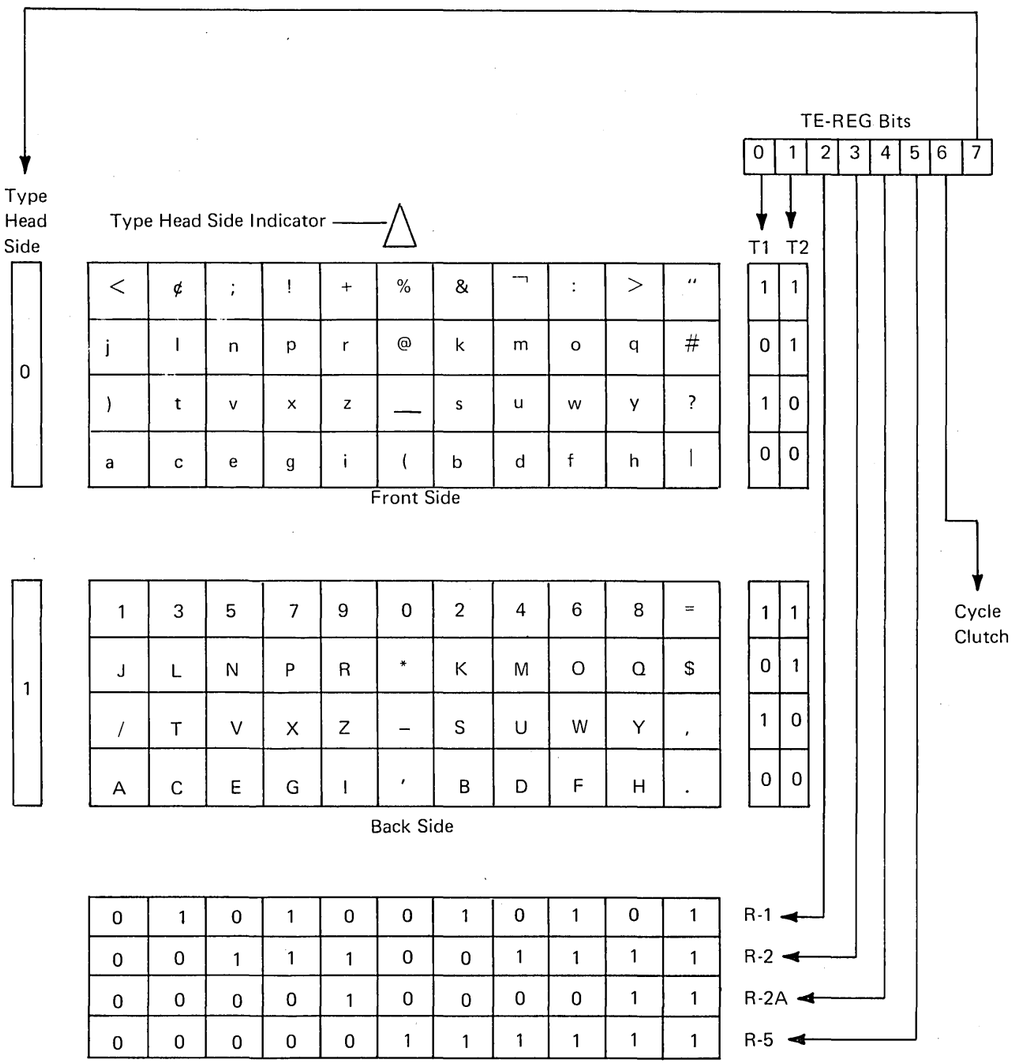
Example: The characters 1 and = are both on the *same side* of the print element.

A mode contact in the printer associated with the print element indicates to the attachment which side (front or back) of the element is aligned to print.

If this indication does not agree with that called for by the 7-bit in the TE register, the print element is rotated 180 degrees to present the opposite side for graphic selection.

The printer also has interlock contacts for indicating End of Form, End of Line, and Long Function, as well as three additional contacts ORed together to indicate Shift, Print, and Operational Feedback conditions.

Printer Tilt/Rotate Codes



3210 TRANSMISSION CODES

This section shows the keyboard codes generated by each of the 3210 character and function keys. Also shown are the tilt/rotate codes the system must furnish to print the various graphic characters.

HOW TO USE THIS TABLE

The table is divided into two sections, each having five columns. The wide center column in each section is the original six-bit keyboard code to which two positions are added in the adapter circuitry by the shift key contact. If the shift key is down, these two positions will contain 1s. If the shift key is up (in the no-shift position), these two positions will contain 0s. The resulting eight-bit code is acceptable to the adapter circuitry in the CPU.

On either side of the center column is a narrow column containing the character (graphic or function) represented by this new eight-bit code.

Note: The HEX value next to each graphic character in the narrow columns is an expression of the EBCDIC representation of that character and is not related to either the keyboard or T/R codes. It is given only for the reader's convenient reference. Characters in the No-Shift column have zeros in bit positions 0 and 1 of the Keyboard code; those in the Shift Key column have ones.

The Tilt/Rotate code for each graphic in the narrow columns is indicated in the adjacent wide columns by the connecting double-headed arrows.

Thus, for the graphic digit 5, the keyboard code is: 00 11 01 01, and the Tilt/Rotate code is: 11 01 00 11.

For the graphic symbol %, the keyboard code is: 11 11 01 01, and the Tilt/Rotate code is: 11 00 01 10. Parity bits are disregarded in this example.

Section 1

T/R Code T1	T2	R1	R2	R2A	R5	CY CL	SH	Keyboard Code 0 0	B	A	8	4	2	1	No Shift Char	Hex	Keyboard Code 1 1	P	0	1	2	3	4	5	6	7	Shift Key Char	Hex	T/R Code T1	T2	R1	R2	R2A	R5	CY CL	SH
0	1	2	3	4	5	6	7																						0	1	2	3	4	5	6	7
1	1	0	0	0	0	1	1								1	F1		0	x	x	1	1	0	0	0	1	=	7E	1	1	1	1	1	1	1	1
1	1	1	0	0	1	1	1								2	F2		0			1	1	0	0	1	0	<	4C	1	1	0	0	0	0	1	0
1	1	1	0	0	0	1	1								3	F3		1			1	1	0	0	1	1	;	5E	1	1	0	1	0	0	1	0
1	1	0	1	0	1	1	1								4	F4		0			1	1	0	1	0	0	:	7A	1	1	1	1	0	1	1	0
1	1	0	1	0	0	1	1								5	F5		1			1	1	0	1	0	1	%	6C	1	1	0	0	0	1	1	0
1	1	1	1	0	1	1	1								6	F6		1			1	1	0	1	1	0	'	7D	0	0	0	0	0	1	1	1
1	1	1	1	0	0	1	1								7	F7		0			1	1	0	1	1	1	>	6E	1	1	0	1	1	1	1	0
1	1	0	1	1	1	1	1								8	F8		0			1	1	1	0	0	0	*	5C	0	1	0	0	0	1	1	1
1	1	0	1	1	0	1	1								9	F9		1			1	1	1	0	0	1	(	4D	0	0	0	0	0	1	1	0
1	1	0	0	0	1	1	1								0	F0		1			1	1	0	0	0	0	)	5D	1	0	0	0	0	0	1	0
0	1	1	1	1	1	1	0								#	7B		0			1	1	1	0	1	1	"	7F	1	1	1	1	1	1	1	0
0	1	0	0	0	1	1	0								@	7C		1			1	1	1	1	0	0	¢	4A	1	1	1	0	0	0	1	0
															LC	06		0			1	1	1	1	1	0	LC									
1	1	0	0	0	1	0	0								SP	40		1			0	0	0	0	0	0	SP		1	1	0	0	0	1	0	0
0	0	0	0	0	0	1	0								a	81		0			0	0	0	0	0	1	A	C1	0	0	0	0	0	0	1	1
0	0	1	0	0	1	1	0								b	82		0			0	0	0	0	1	0	B	C2	0	0	1	0	0	1	1	1
0	0	1	0	0	0	1	0								c	83		1			0	0	0	0	1	1	C	C3	0	0	1	0	0	0	1	1
0	0	0	1	0	1	1	0								d	84		0			0	0	0	1	0	0	D	C4	0	0	0	1	0	1	1	1
0	0	0	1	0	0	1	0								e	85		1			0	0	0	1	0	1	E	C5	0	0	0	1	0	0	1	1
0	0	1	1	0	1	1	0								f	86		1			0	0	0	1	1	0	F	C6	0	0	1	1	0	1	1	1
0	0	1	1	0	0	1	0								g	87		0			0	0	0	1	1	1	G	C7	0	0	1	1	0	0	1	1
0	0	0	1	1	1	1	0								h	88		0			0	0	1	0	0	0	H	C8	0	0	0	1	1	1	1	1
0	0	0	1	1	0	1	0								i	89		1			0	0	1	0	0	1	I	C9	0	0	0	1	1	0	1	1
0	0	1	1	1	1	1	1								.	4B		0			0	0	1	1	1	1	¬	5F	1	1	0	1	0	1	1	0
															UC	36		0			0	0	1	1	1	0	UC									

Section 2

T/R Code T1	T2	R1	R2	R2A	R5	CY CL	SH	Keyboard Code 0 0	B	A	8	4	2	1	No Shift Char	Hex	Keyboard Code 1 1	P	0	1	2	3	4	5	6	7	Shift Key Char	Hex	T/R Code T1	T2	R1	R2	R2A	R5	CY CL	SH
0	1	2	3	4	5	6	7																						0	1	2	3	4	5	6	7
1	1	1	0	0	1	1	0								&	50		0	x	x	0	1	0	0	0	0	+	4E	1	1	0	1	1	0	1	0
0	1	0	0	0	0	1	0								j	91		1			0	1	0	0	0	1	J	D1	0	1	0	0	0	0	1	1
0	1	1	0	0	1	1	0								k	92		1			0	1	0	0	1	0	K	D2	0	1	1	0	0	1	1	1
0	1	1	0	0	0	1	0								l	93		0			0	1	0	0	1	1	L	D3	0	1	1	0	0	0	1	1
0	1	0	1	0	1	1	0								m	94		1			0	1	0	1	0	0	M	D4	0	1	0	1	0	1	1	1
0	1	0	1	0	0	1	0								n	95		0			0	1	0	1	0	1	N	D5	0	1	0	1	0	0	1	1
0	1	1	1	0	1	1	0								o	96		0			0	1	0	1	1	0	O	D6	0	1	1	1	0	1	1	1
0	1	1	1	0	0	1	0								p	97		1			0	1	0	1	1	1	P	D7	0	1	1	1	0	0	1	1
0	1	0	1	1	1	1	0								q	98		1			0	1	1	0	0	0	Q	D8	0	1	0	1	1	1	1	1
0	1	0	1	1	0	1	0								r	99		0			0	1	1	0	0	1	R	D9	0	1	0	1	1	0	1	1
0	1	1	1	1	1	1	1								$	5B		1			0	1	1	0	1	1	!	DA	1	1	1	1	0	0	1	0
0	0	0	0	1	0	0	0								N/L	15		0			0	1	1	1	0	0	N/L		0	0	0	0	1	0	0	0
1	0	0	0	0	1	1	1								-	60		0			1	0	0	0	0	0	–	6D	1	0	0	0	0	1	1	0
1	0	0	0	0	0	1	1								/	61		1			1	0	0	0	0	1	?	6F	1	0	1	1	1	1	1	0
1	0	1	0	0	1	1	0								s	A2		1			1	0	0	0	1	0	S	E2	1	0	1	0	0	1	1	1
1	0	1	0	0	0	1	0								t	A3		0			1	0	0	0	1	1	T	E3	1	0	1	0	0	0	1	1
1	0	0	1	0	1	1	0								u	A4		1			1	0	0	1	0	0	U	E4	1	0	0	1	0	1	1	1
1	0	0	1	0	0	1	0								v	A5		0			1	0	0	1	0	1	V	E5	1	0	0	1	0	0	1	1
1	0	1	1	0	1	1	0								w	A6		0			1	0	0	1	1	0	W	E6	1	0	1	1	0	1	1	1
1	0	1	1	0	0	1	0								x	A7		1			1	0	0	1	1	1	X	E7	1	0	1	1	0	0	1	1
1	0	0	1	1	1	1	0								y	A8		1			1	0	1	0	0	0	Y	E8	1	0	0	1	1	1	1	1
1	0	0	1	1	0	1	0								z	A9		0			1	0	1	0	0	1	Z	E9	1	0	0	1	1	0	1	1
1	0	1	1	1	1	1	1								,	6B		1			1	0	1	0	1	1	|	4F	0	0	1	1	1	1	1	0

3210 PR-KB CODE TRANSLATION

- The CPU uses the Extended Binary-Coded-Decimal Interchange Code (EBCDIC).
- Input from the 3210 Console Printer-Keyboard comes in the keyboard code.
- Output to the 3210 printers must be in the Tilt/Rotate code of the IBM Selectric® I/O-II printers.
- Because of this, data translation to and from EBCDIC is required for all 3210 input (read) and output (write) operations.
- This translation is accomplished by a microprogram routine using the appropriate translation table in control storage, as shown under "Printing the Punctuation Character, Colon".

The PR-KB transmits 88 printable graphic character codes that must be translated to EBCDIC for CPU use and to the T/R code for printing. These characters are:

26 Uppercase letters A-Z
26 Lowercase letters a-z
10 Digits 0-9
26 Special symbols

Also, certain function (Space, New Line, and Shift Key) codes are also encoded and transmitted.

The shift key code is translated and investigated by the microprogram, but once identified, is disregarded.

This is because shift operations do not use the shift-key code (see "Shift Cycle Operation"), and because the microprogram examines every code sent from the keyboard to see whether it is a legitimate character or function code.

The Space and New Line codes are translated to and from EBCDIC just like any other character. The chart shows the EBCDIC byte assignments for the 3210 graphics and functions.

The full-page drawing on the facing page shows the overall translation scheme for the PR-KB read and write operations, using the punctuation symbol colon (:) as an example.

In read operations, data is translated from the keyboard code to EBCDIC.

In write operations (and every read operation has an accompanying write operation), EBCDIC is translated into the Tilt/Rotate code used by the printer.

Both the EBCDIC and Tilt/Rotate codes are stored in separate sections of control storage. For convenience of addressing, however, these two sections occupy adjacent positions in the same block of storage. The address of the codes in the translate table are:

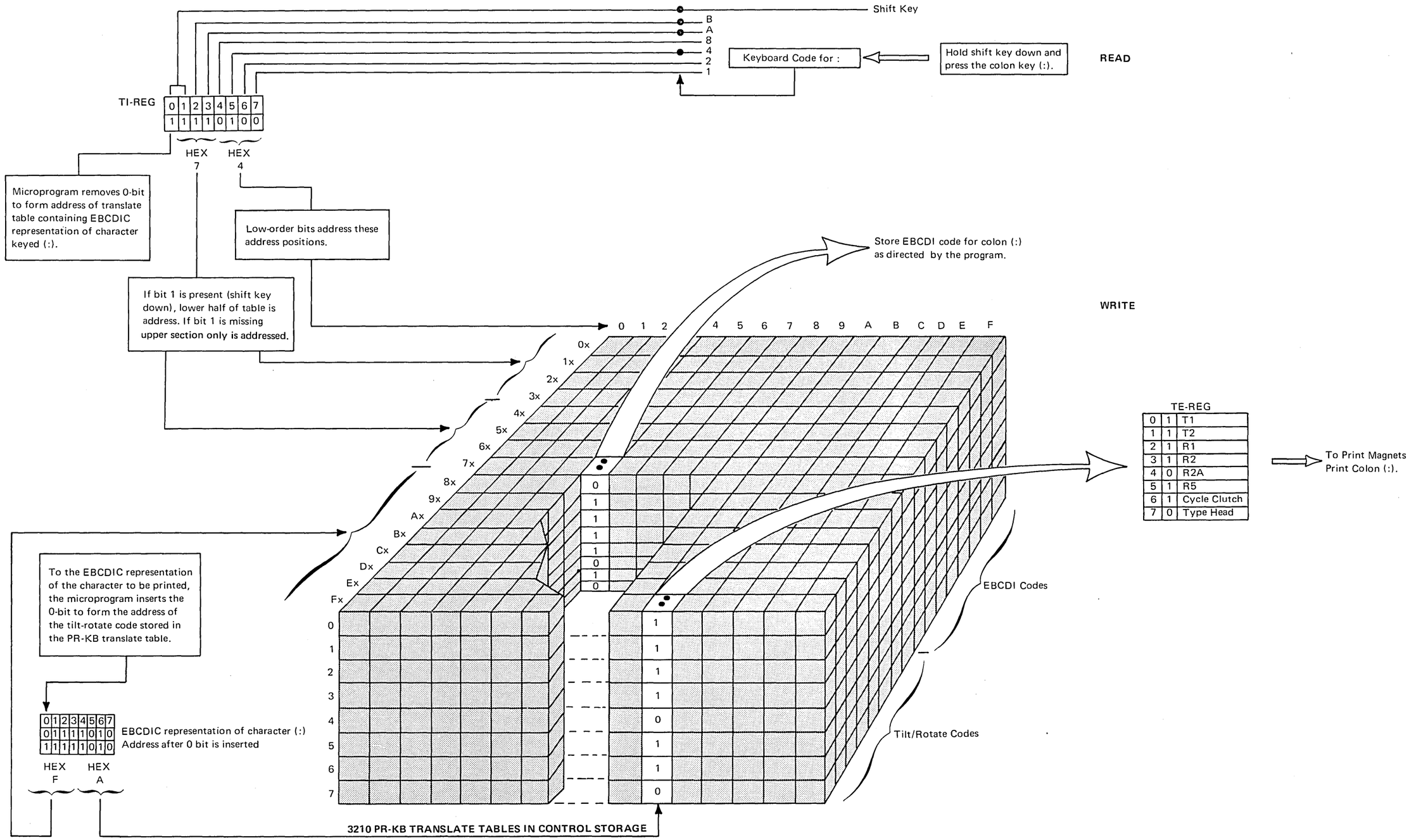
F700-F7FF

Bits 0-1

Bits 4,5,6,7	Bits 2-3 = 00				Bits 2-3 = 01				Bits 2-3 = 10				Bits 2-3 = 11			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	0	0	0	0	SP	&	-	0	0	0	0	0	0	&	-	0
0001	a	j	0	0	A	J	/	1	a	j	0	0	A	J	/	1
0010	b	k	s	0	B	K	S	2	b	k	s	0	B	K	S	2
0011	c	l	t	0	C	L	T	3	c	l	t	0	C	L	T	3
0100	d	m	u	0	D	M	U	4	d	m	u	0	D	M	U	4
0101	e	N/L	v	0	E	N	V	5	e	n	v	0	F	N	V	5
0110	f	o	w	0	F	O	W	6	f	o	w	0	F	O	W	6
0111	g	p	x	0	G	P	X	7	g	p	x	0	G	P	X	7
1000	h	q	y	0	H	Q	Y	8	h	q	y	0	H	Q	Y	8
1001	i	r	z	0	I	R	Z	9	i	r	z	0	I	R	Z	9
1010	0	0	0	0	¢	!	0	:	0	0	0	0	¢	!	0	:
1011	0	0	0	0	.	$	,	#	0	0	0	0	.	$	,	#
1100	0	0	0	0	<	*	%	@	0	0	0	0	<	*	%	@
1101	0	0	0	0	(	)	¬	'	0	0	0	0	(	)	-	'
1110	0	0	0	0	+	;	>	=	0	0	0	0	+	;	>	=
1111	0	0	0	0	|	¬	?	"	0	0	0	0	|	¬	?	"

Bit assignments for the 3210 character sets are shown in the clear areas surrounded by the heavy lines. Characters in the shaded areas may print for their respective bit combinations, but these characters are not guaranteed.

PRINTING THE PUNCTUATION CHARACTER, COLON (:)



This chart shows the contents of the individual positions of the translate-table storage block for the 3210 PR-KBs shown on the preceding page.

The upper half of the block contains the EBCDI codes for each of the 88 graphic (printable) characters plus those for the Space and New Line function characters.

The lower half of the block contains the Tilt/Rotate Selectric codes for the same characters, although they are not arranged in the same order in that half. All unused positions of this half have been filled with zero codes.

The character above the diagonal line in each position of the lower half of this chart shows the character whose Tilt/Rotate code is stored there. The character below the diagonal line is the hexadecimal (HEX) representation of the Tilt/Rotate code for that character (as it would appear when transmitted to the TE-register).

For example, the Tilt/Rotate code for the colon (:) is: 1111 0110, for which the HEX representation is: F6.

Another example: the T/R code for a capital (upper case) W is: 1011 0111, or in HEX,
B 7

To find the Tilt/Rotate code for any of the 88 graphic or two function characters, refer to the charts shown under *3210 PRINTER.*

Note that the cycle-clutch bit (bit 6 in the T/R code) is active (1) for all printable characters, but is inactive (0) for the function characters.

Keyboard to EBCDIC Table

Note: The 0-bit of the keyboard code is removed by the microprogram to form the address of the EBCDIC table.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	SP	a	b	c	d	e	f	g	h	i		.				
1x	&	j	k	l	m	n	o	p	q	r		$	N/L*			
2x	-	/	s	t	u	v	w	x	y	z		,				
3x	0	1	2	3	4	5	6	7	8	9		#	@			
4x	SP	A	B	C	D	E	F	G	H	I		¬				
5x	+	J	K	L	M	N	O	P	Q	R		!	N/L*			
6x	–	?	S	T	U	V	W	X	Y	Z		|				
7x	)	=	<	;	:	%	'	>	*	(		"	¢			
8x	0/C7	a/02	b/26	c/22	d/16	e/12	f/36	g/32	h/1E	i/1A	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7
9x	0/C7	j/42	k/66	l/62	m/56	n/52	o/76	p/72	q/5E	r/5A	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7
Ax	0/C7	0/C7	s/A6	t/A2	u/96	v/92	w/B6	x/B2	y/9E	z/9A	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7
Bx	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7	0/C7
Cx	SP/C5	A/03	B/27	C/23	D/17	E/13	F/37	G/33	H/1F	I/1B	¢/E2	./3F	</C2	(/06	+/DA	|/3E
Dx	&/E6	J/43	K/67	L/63	M/57	N/53	O/77	P/73	Q/5F	R/5B	!/F2	$/7F	*/47	)/82	;/D2	¬/D6
Ex	-/87	'/83	S/A7	T/A3	U/97	V/93	W/B7	X/B3	Y/9F	Z/9B	0/C7	,/BF	%/C6	–/86	>/DE	?/BE
Fx	0/C7	1/C3	2/E7	3/E3	4/D7	5/D3	6/F7	7/F3	8/DF	9/DB	:/F6	#/7E	@/46	'/07	=/FF	"/FE

Character to be printed : / F6 T/R decode for the character (expressed in HEX)

*Note: New Line (N/L) is handled by a special microcode (See GKDT routine).

On a write operation, the N/L character goes into the TE register as a 4-bit only. This initiates both a carrier return and an index (line space) operation but *no print cycle.*

EBCDIC to T/R Table

Note: The 0-bit is added to the EBCDIC character by the microprogram to form its address in the T/R table.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

3210 PR-KB INTEGRATED ATTACHMENT

- The 3210 PR-KB integrated attachment is an electronic package that works with the microprograms to control data flow between the system and the PR-KB.
- The adapter (attachment) contains the four hardware registers: TA,TT, TI, and TE.
- The entire circuitry package comprising the adapter is contained in seven MST cards located on boards 01A-A4 and 01F-A1.

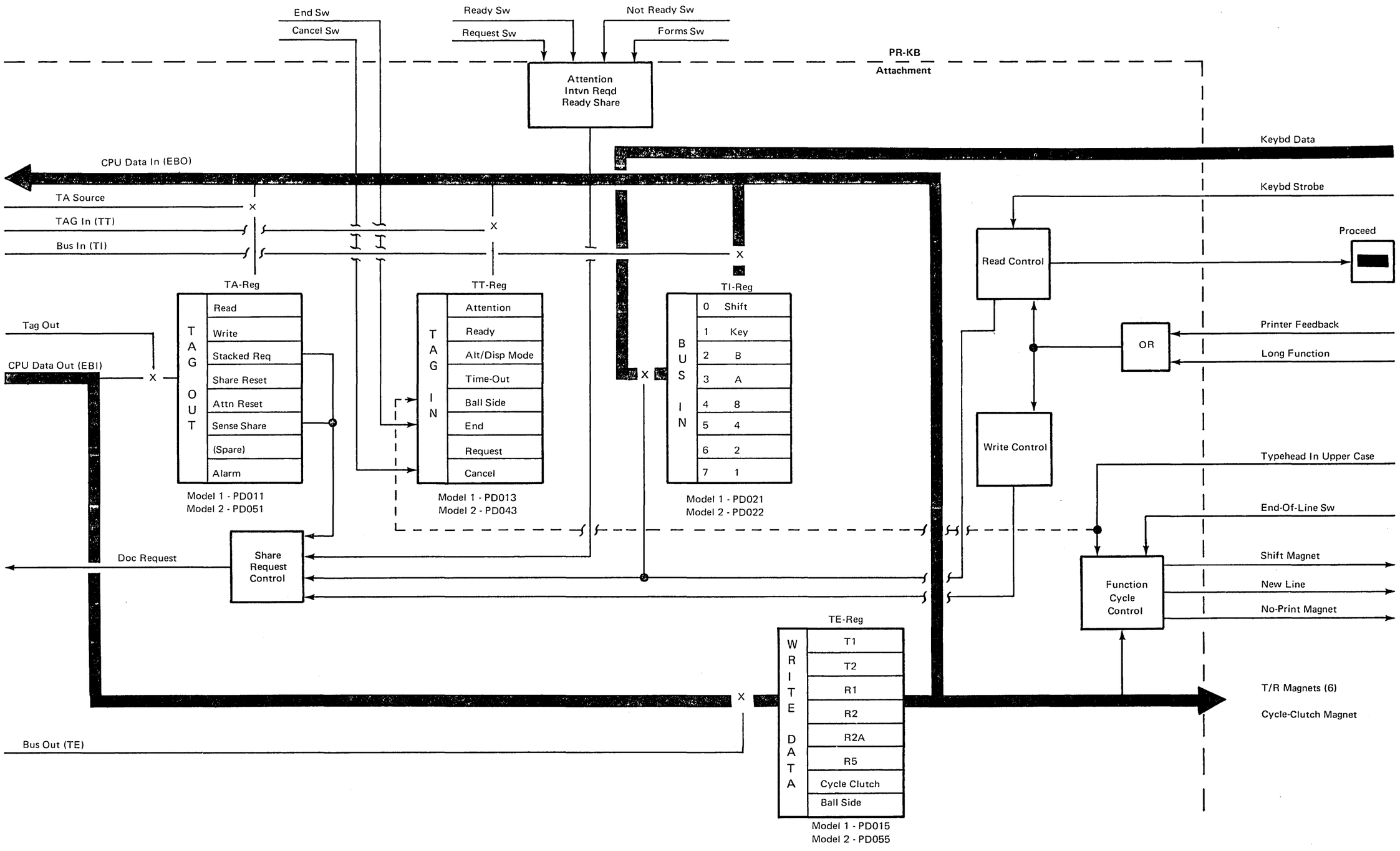
The drawing on the facing page shows the overall scheme of data flow and control provided by the adapter under direction of the system microprograms. The heavy lines at the top are the EBO lines; those at the bottom are the EBI lines. These busses transmit either data or control information, depending on which registers are gated (X). The thin lines denote control signal paths.

Each of the registers in turn is described fully and separately in this section. The small print under each register in the drawing is the page number of the logic (ALD) diagrams where the wiring of each register may be found.

3210 DATA AND CONTROL REGISTERS

- Four registers in the 3210 PR-KB adapter provide control of the printer-keyboard input/output operations. These are the TA (Tag Out), TT (Tag In), TI (Bus or Data In), and TE (Write Data) registers described in the following pages.
- The condition of the individual latches that make up these registers may be observed in the A-register display lights with the roller switch in position 2.

3210 CONTROL AND DATA FLOW



TA-Register

- The microprogram initiates and controls PR-KB operations through the TA-register.

The TA-register (also known as the Tag Out register) consists of eight data-bit latches and a parity-bit latch. Each latch represents one of the bits of external bus-in byte 1.

All of the latches are reset by POWER-ON RESET, or by the output of a common AND circuit, which also gates the latches on. Machine circuitry delays the set gate to the latches to keep it up after the reset line drops, allowing the incoming bits to turn on their respective latches.

The TA-register bit-position latch assignments and their functions are:

TA0 *Read.* This bit sets the read latch to develop read mode and initiate a keyboard (input) operation.

TA1 *Write.* This bit sets the write latch to develop write mode and initiate a PR-KB print (output) operation.

TA2 *Stacked Request.* (Interrupt stacked.) This bit sets the stacked request latch to initiate a request when status is stacked.

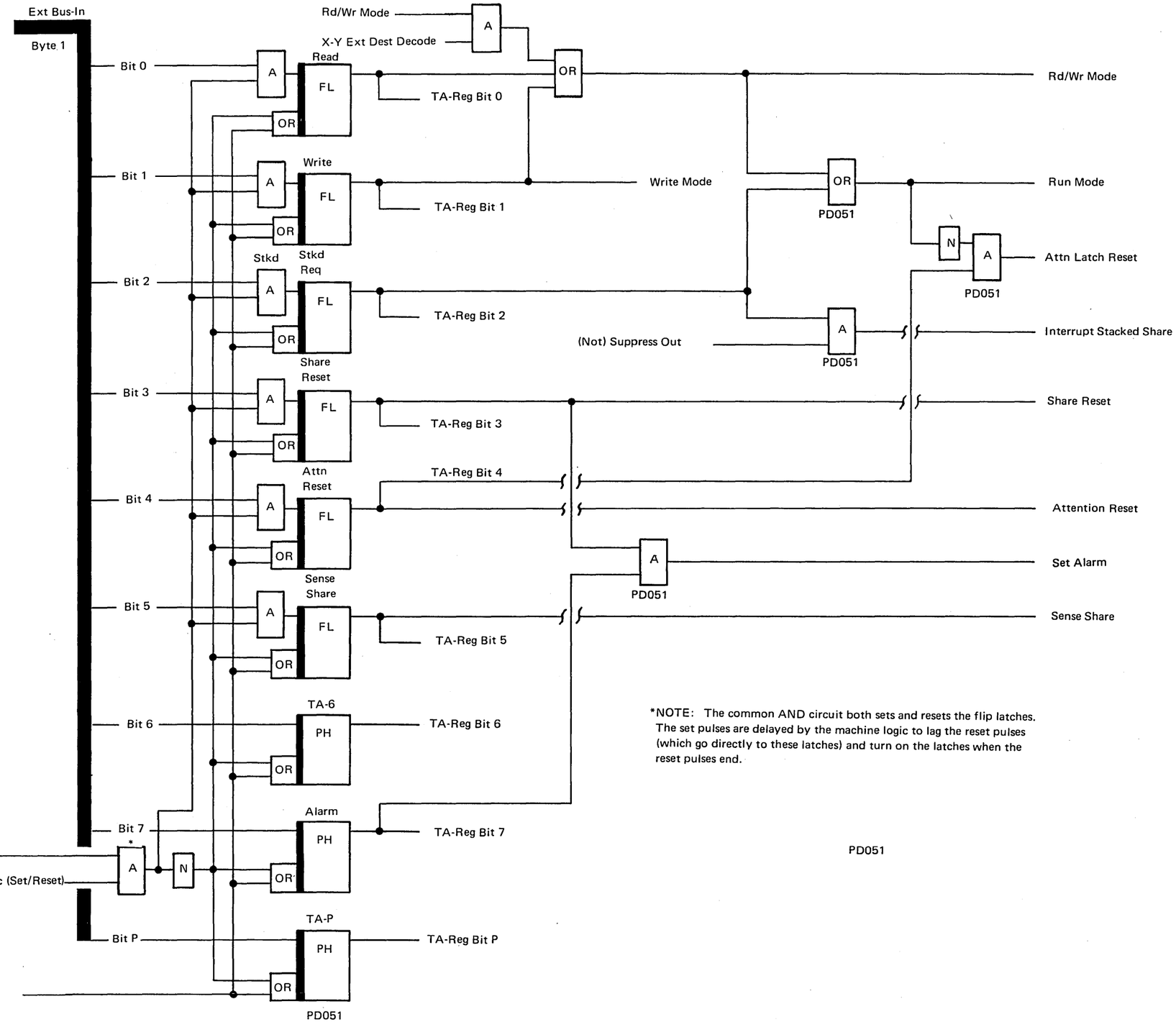
TA3 *Share Reset.* This bit resets the Rd/Wr share (request) latch.

TA4 *Attn Reset.* This bit, when the system is in run mode, resets the end, cancel, and ready share request latches. When in not-run mode, it also resets the Attn latch.

TA5 *Sense Share.* This bit causes a console request and initiates the PR-KB microprograms that perform a sense command. When in Alter/Display (A/D) mode, this bit resets the A/D latch.

TA6 *Spare.* This bit has no assigned function.

TA7 *Alarm.* This bit turns on the audible alarm.



These set-reset lines are developed by decoding the DOC word in the external control assembler to address the TA-register as described under "External Facilities," Chapter 2.

*NOTE: The common AND circuit both sets and resets the flip latches. The set pulses are delayed by the machine logic to lag the reset pulses (which go directly to these latches) and turn on the latches when the reset pulses end.

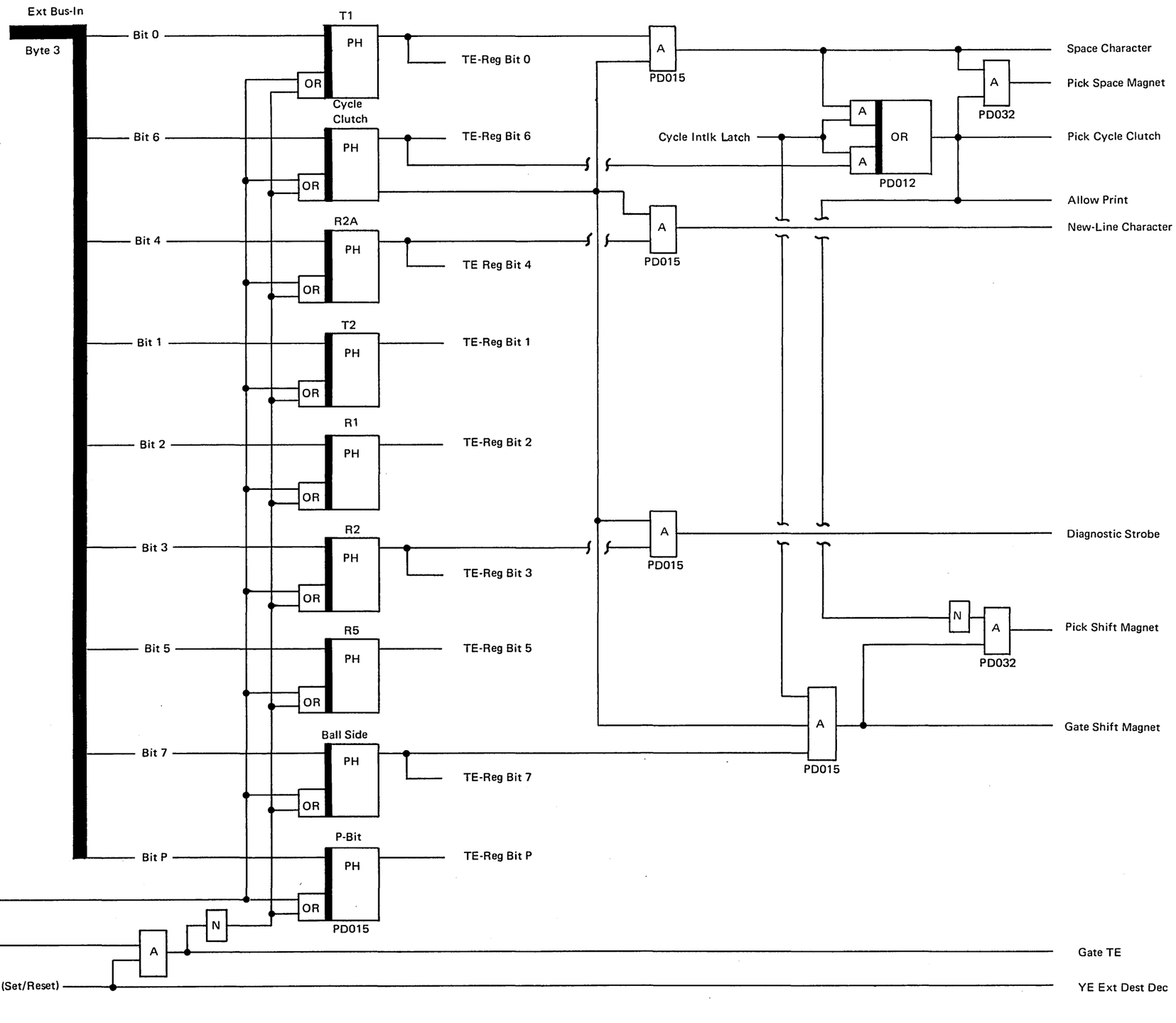
TE Register

- The microprogram translates output data to the console printer T/R codes and sends it to the TE-register for transmission to the tilt/rotate magnets.

The TE-register consists of nine polarity-hold latches that receive the 8-bit (plus parity) tilt/rotate code for each character transmitted from the tilt/rotate translate table in control storage, as described under "3210 PR-KB Code Translation." The TE-register acts as the adapter-to-printer interface, and under control of the microprogram, transmits the tilt/rotate signals directly to their respective magnets in the printer.

In addition to the data-character codes, this register also develops the printer function-character codes for Space and New Line, and develops the following printer-function control lines:

Pick space (no work) magnet,
Pick cycle clutch,
Diagnostic strobe, and
Gate shift magnet.



These set-reset lines are developed by decoding the DOC word in the external control assembler to address the TE register as described under "External Facilities", Chapter 2.

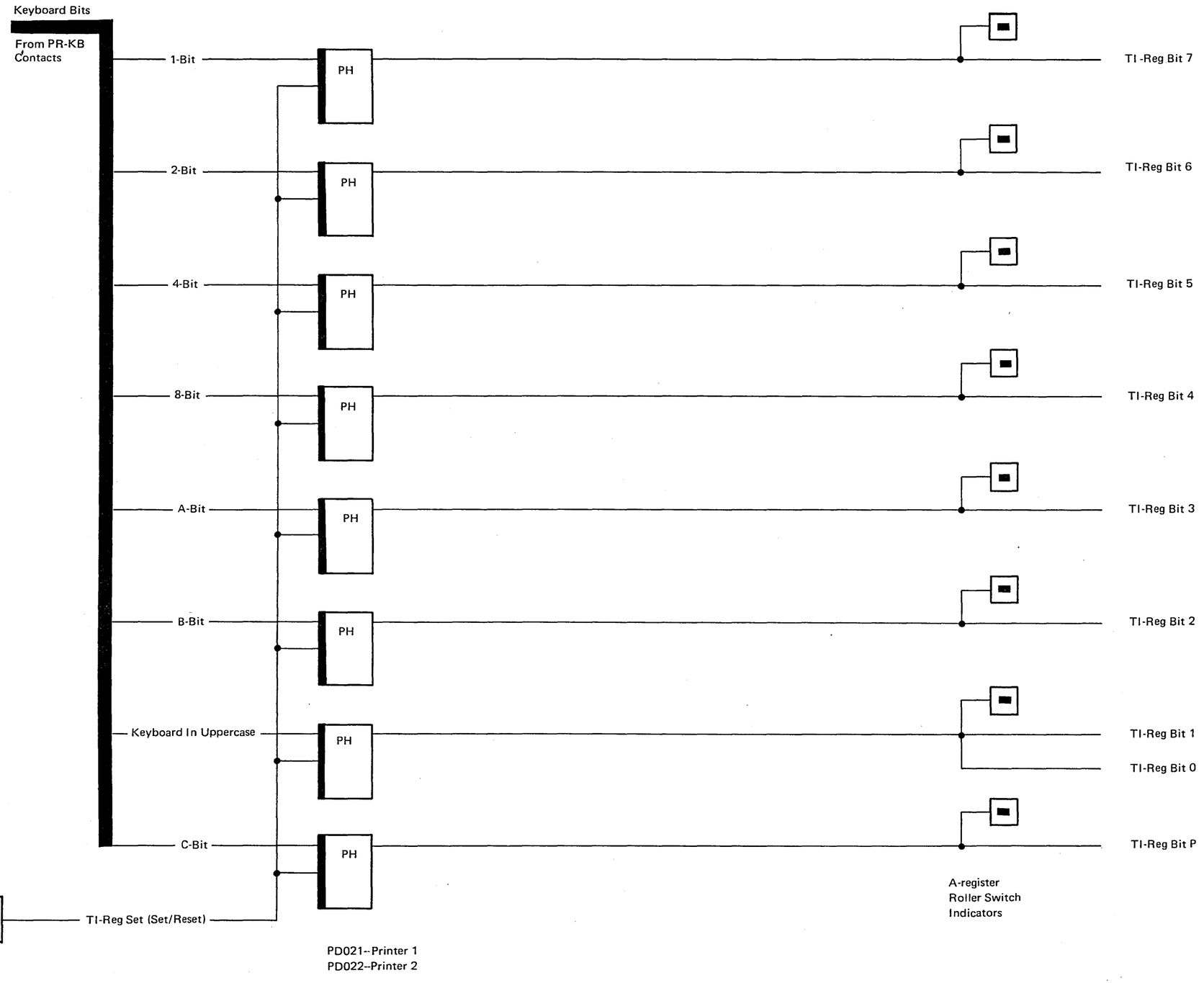
TI-Register

- The TI-register sends coded keyboard characters to the CPU during read or alter/display operations.

The TI (keyboard interface) register consists of eight polarity-hold latches. Six of these accept data from the keyboard interposer contacts in the six-bit Selectric Keyboard code (B-A-8-4-2-1). The seventh latch is set when the shift key is operated (keyboard in uppercase), and the eight latch accepts the C-bit.

The outputs of these latches go to the external assembler for byte 0 where they place the keyboard bits in the low-order bit positions (2 through 7) of the byte. The shift key (keyboard in uppercase) latch goes to both the 0- and 1-bit positions of the assembler. Thus, if the shift key has been operated, bit positions 0 and 1 of the assembler will contain ones; if the shift key was not operated, both of these bit positions will contain zeros. The keyboard C-bit, of course, sets the P-bit latch in the TI-register.

As described under "3210 PR-KB Code Translation," the microprogram ignores the 0-bit in forming the addresses of the EBCDI codes in the look-up table, but uses this bit in forming the address of the tilt/rotate codes.



REMEMBER

There is a Reader's Comment Form at the back of this publication.

TT-Register

- The TT (Tag-In) register stores the status of several input control signals from the PR-KB for examination and use by the microprogram.
- This register is associated with the control keys and indicator lights on the documentary console.
- Unlike the other three registers of the attachment, the TT-register consists of individual latches scattered throughout the logic pages, with each having its own set and reset lines.
  Outputs of these latches are gated to the external assembler for byte 2 (logic pages: BE142, 143, 144, 152, and 153.

The TT-register contains eight bit latches that store the status of as many PR-KB input control conditions generated either automatically in the attachment, or manually by the operator using the control keys on either side of the keyboard. Six of the latches are activated by the control keys shown at the left side of the drawing on the facing page and described under "Control Keys and Indicators" on page 7-6. The other two latches are set by certain combinations of circuit conditions as indicated in the drawing. The five lights that indicate associated signal conditions are shown at the right side of the drawing.

When the branch control word addresses the TT-register, the latch outputs are gated through the external assembler to the A-register for testing.

The TT-register bit-position latch assignments and their functions are:

TT0 *Attention:* This latch turns on (when the request key is operated) to present attention status to the CPU. When the attention latch is on, it also lights the request pending indicator lamp. TA4 resets the attention latch after the microprogram puts the attention bit in the unit-status register.

TT1 *Ready.* This latch indicates that the console printer is ready (forms are in position and the ready key has been operated). If the printer is not ready for any reason, the associated intervention required light will be on.

TT2 *Alter/Display Mode.* This latch indicates that the printer-keyboard is operating in the alter/display (manual) mode. The operator must manually establish this mode of operation by pressing the alter/display key. When this latch is on, the alter/display mode light is on to indicate that the system is ready to accept characters one at a time from the keyboard.

TT3 *Time Out.* Indicates an error condition caused by failure of the printer-keyboard to go busy within about one second of receipt of a character from the adapter during a print operation while either reading or writing.

TT4 *Ball Side.* This latch is operated by a contact in the console printer that indicates which side of the print element (ball) is in position to print (facing the platen). The line from the printer that turns on this latch is called *Typehead in Upper Case.* The output of this latch is compared by the microprogram with the 7-bit of the TE register to determine whether a shift (180-degree rotation) of the print element is required.

TT5 *End.* This latch turns on when the End key is operated to terminate a read, write, or alter/display operation. Formerly known as End of Block (on System/360), this key causes a return to the beginning of the operation so that the operator can start a new manual operation. The End latch issues a new console request.

TT6 *Request.* This bit represents the output of a multi-legged OR block that indicates that the adapter is ready to transmit or receive data, and signals the CPU (by requesting a share cycle request) to begin the operation.

TT7 *Cancel.* This bit indicates that the cancel latch was turned on by operating the cancel key. The cancel latch can be turned on during a read operation only, and sets the unit exception bit in the UCW unit-status byte.

Time-Out Circuit

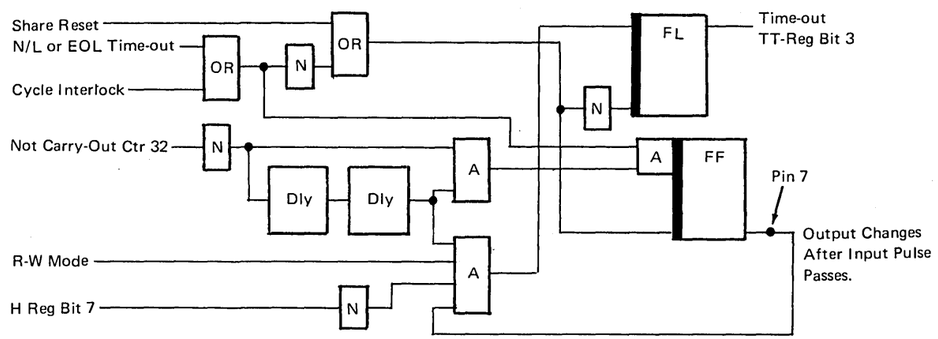
This is a safety circuit designed to reset the PR-KB attachment and notify the CPU if the printer does not respond within approximately one second of receipt of data.

The object is to prevent tieing up the system or allowing magnets to burn out if the printer is down due to such causes as broken belts and jams.

Circuit Description

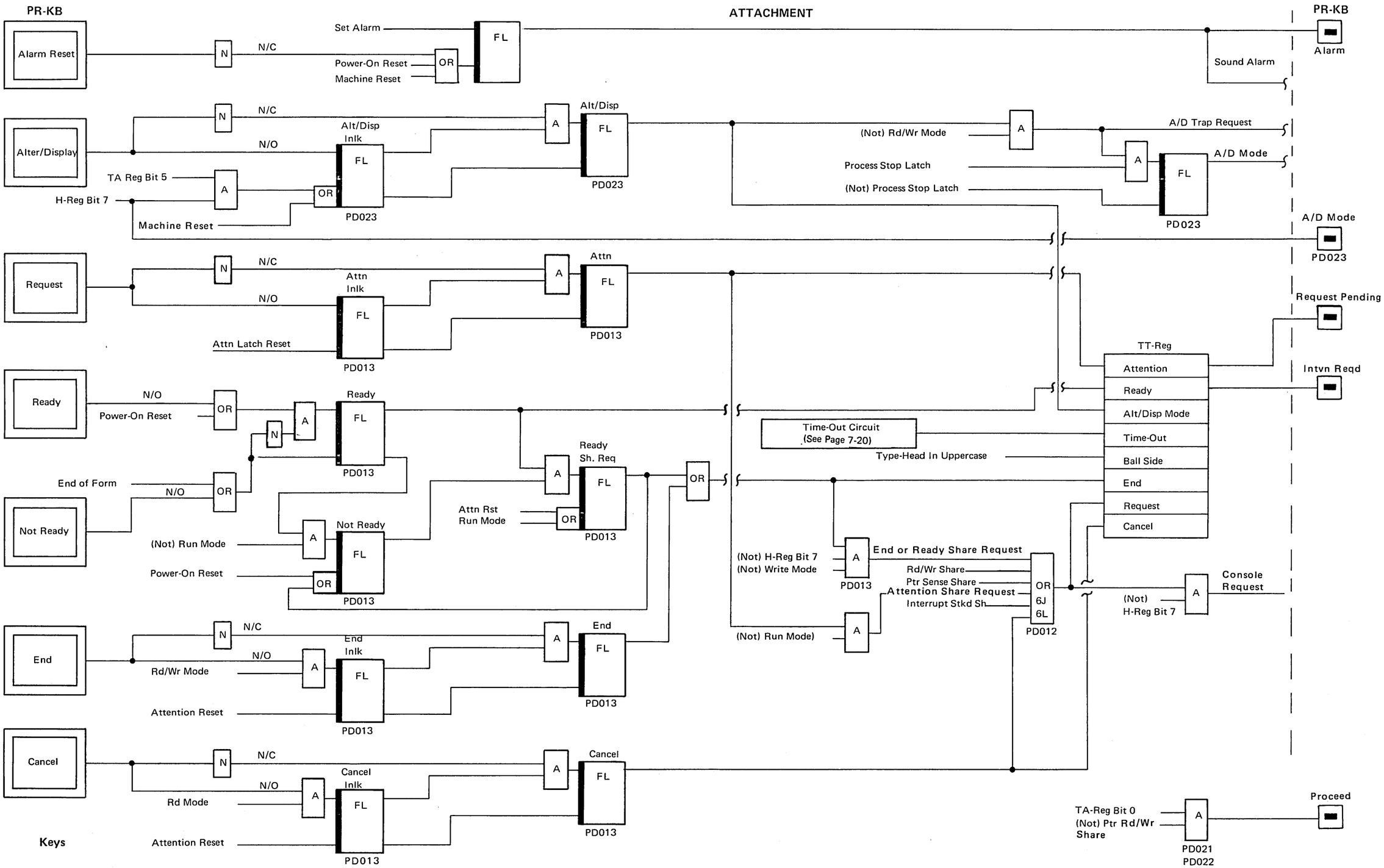
Carry-out counter 32 delivers a pulse from the Time-of-Day clock approximately every second. If the cycle interlock latch has not been reset by the printer's having gone busy, the carry-out counter 32 pulse turns on the flip-flop trigger and, provided the printer is not in A/D mode (H-Reg Bit 7), sets the flip latch to turn on bit 3 in the TT-register and bring up Time-out Share Request line.

When the microprogram detects the time-out bit (TT-3), it resets the PR-KB attachment, sets the unit check bit in the CSW, and turns on the equipment check bit in the PR-KB sense byte.

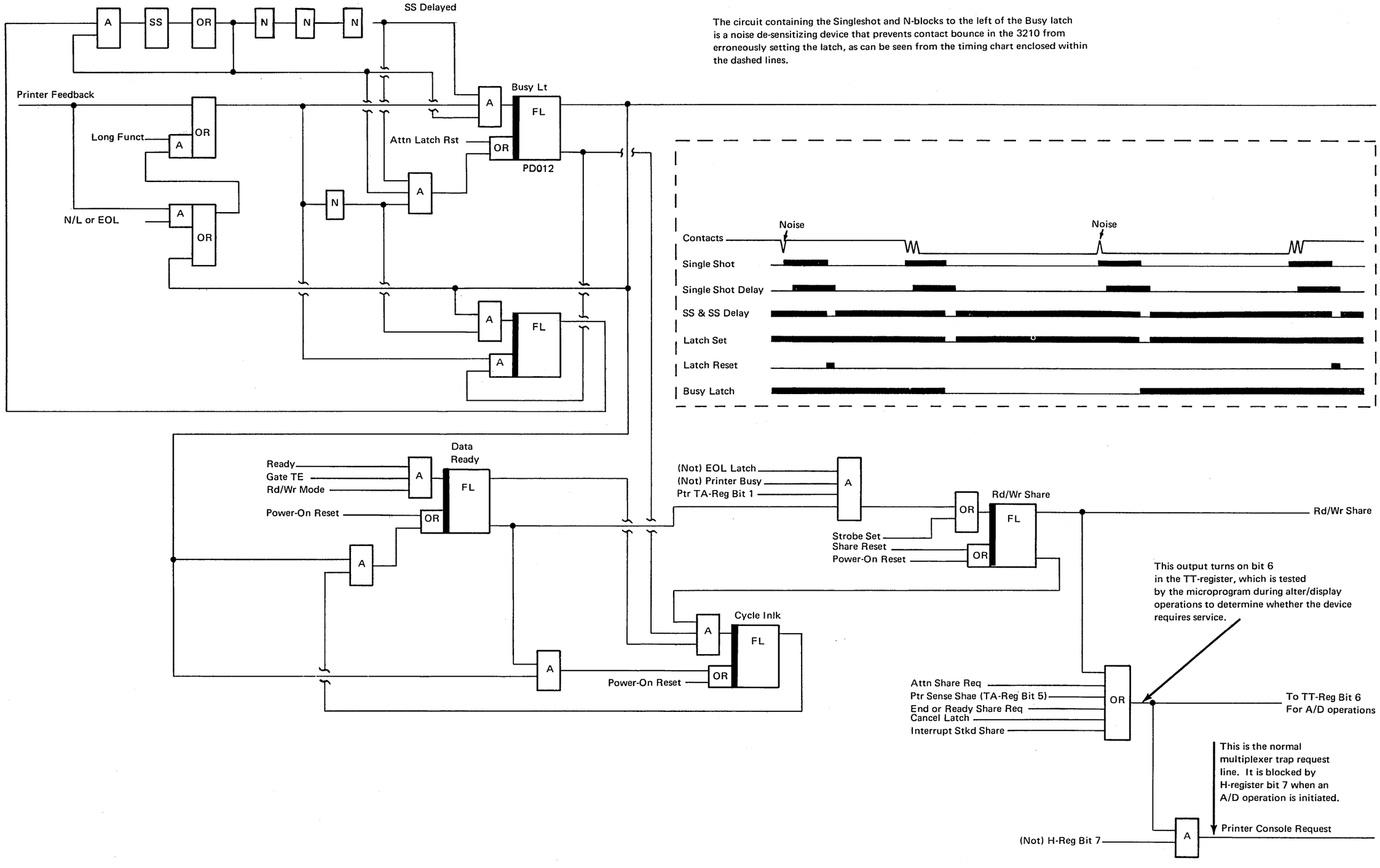


PD 014 for 3210

PN 034 for 3215

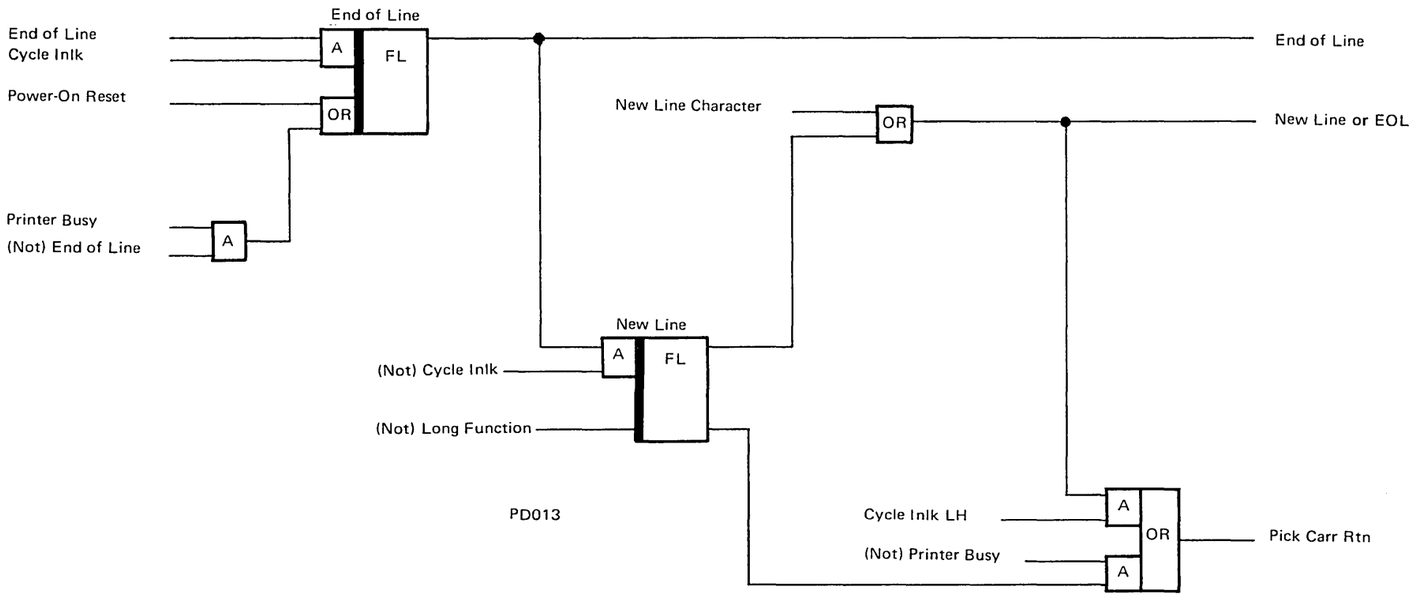

PR-KB
ATTACHMENT
PR-KB
Alarm Reset
N
N/C
Set Alarm
Power-On Reset
Machine Reset
OR
FL
Sound Alarm
Alarm
Alter/Display
N
N/C
N/O
Alt/Disp Inlk
A
FL
PD023
Alt/Disp
TA Reg Bit 5
H-Reg Bit 7
Machine Reset
OR
(Not) Rd/Wr Mode
Process Stop Latch
(Not) Process Stop Latch
A/D Trap Request
A/D Mode
A/D Mode
PD023
Request
Attn Inlk
Attn
Attn Latch Reset
PD013
Request Pending
Intvn Reqd
TT-Reg
Attention
Ready
Alt/Disp Mode
Time-Out
Ball Side
End
Request
Cancel
Ready
N/O
Power-On Reset
Ready
Ready Sh. Req
Time-Out Circuit
(See Page 7-20)
Type-Head In Uppercase
End of Form
Not Ready
(Not) Run Mode
Power-On Reset
Not Ready
Attn Rst
Run Mode
(Not) H-Reg Bit 7
(Not) Write Mode
(Not) Run Mode
End or Ready Share Request
Rd/Wr Share
Ptr Sense Share
Attention Share Request
Interrupt Stkd Sh
PD012
Console Request
(Not) H-Reg Bit 7
End
Rd/Wr Mode
Attention Reset
End
Cancel
Rd Mode
Attention Reset
Cancel
TA-Reg Bit 0
(Not) Ptr Rd/Wr Share
PD021
PD022
Proceed
Keys

Rd-Wr Cycle Control



The circuit containing the Singleshot and N-blocks to the left of the Busy latch is a noise de-sensitizing device that prevents contact bounce in the 3210 from erroneously setting the latch, as can be seen from the timing chart enclosed within the dashed lines.

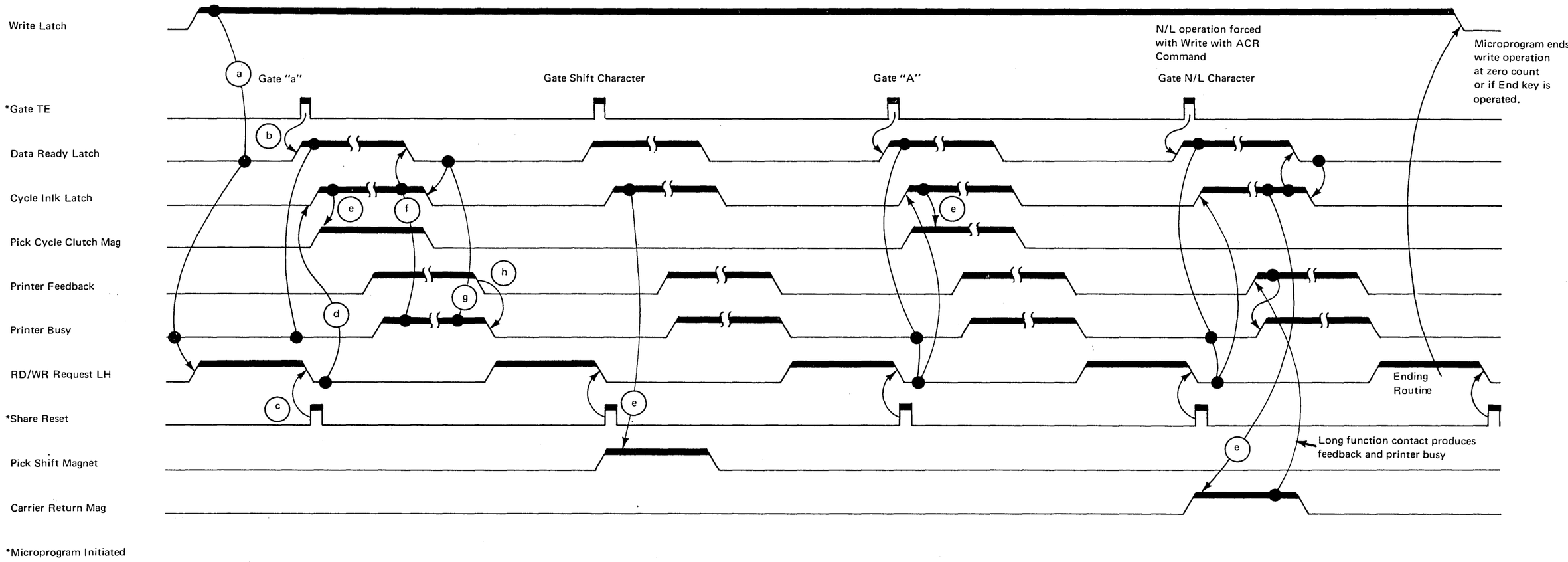
New Line Control



3210 WRITE OPERATION

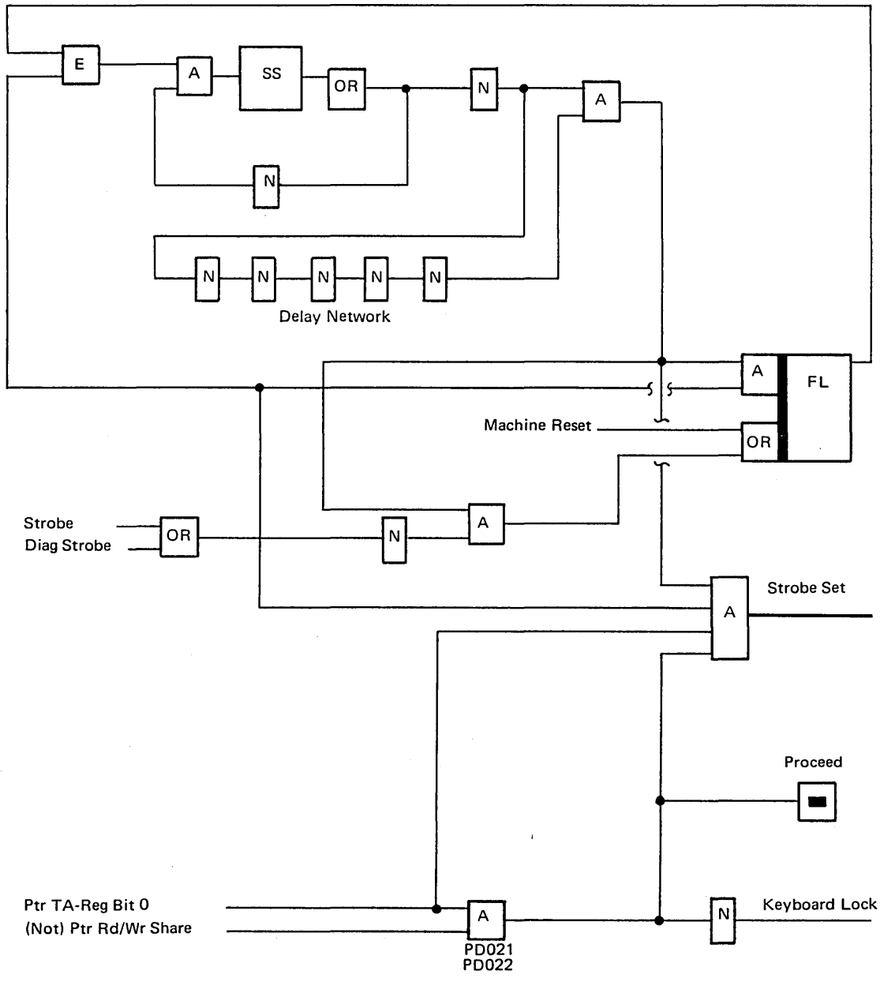
OBJECTIVES: Print the two characters -- a A

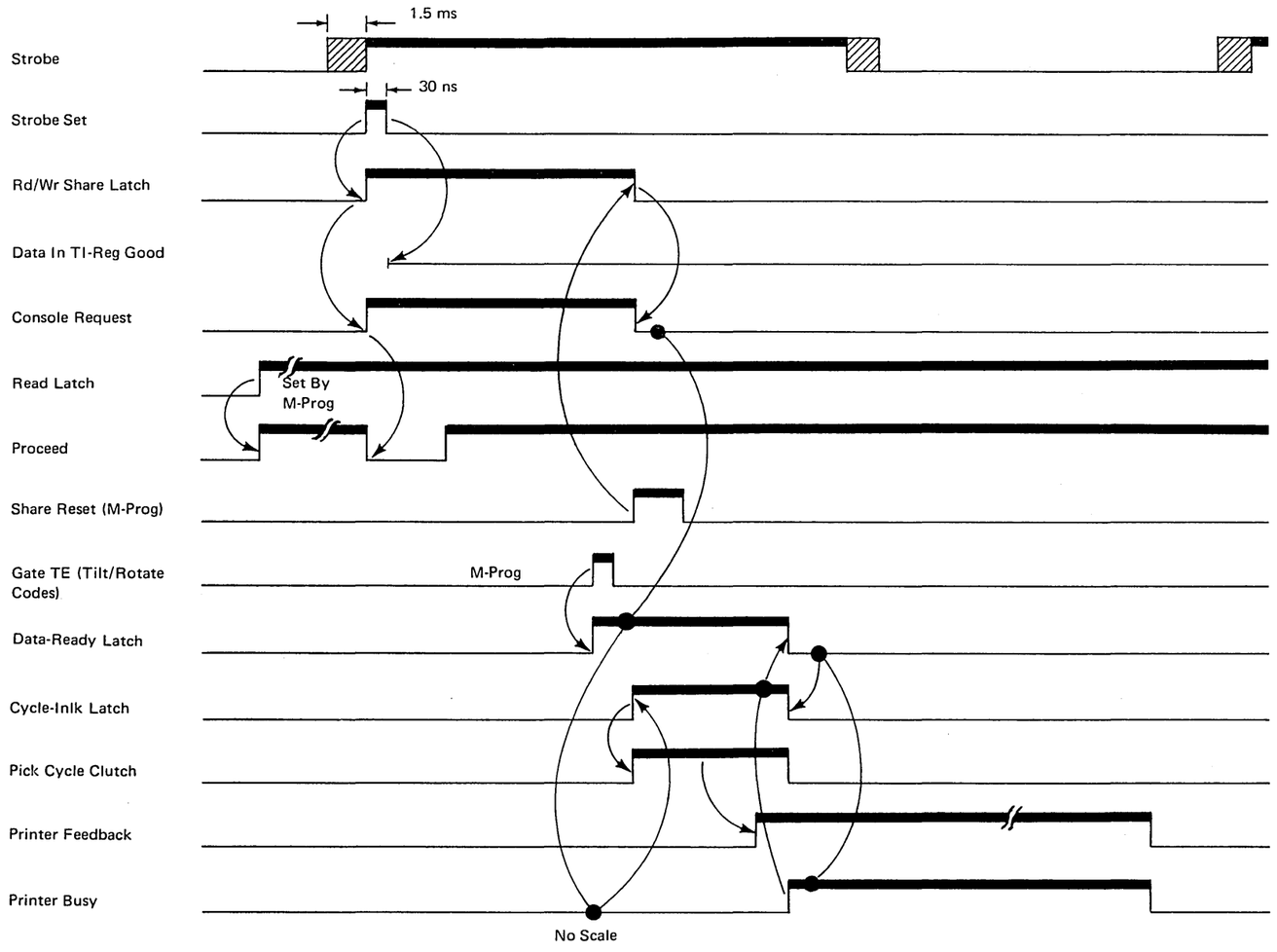
1. Print small letter a
   a. Write latch (TA-Reg bit-1) turned on by write command. Write latch, with Data Ready latch off, (data not ready), and printer not busy, turns on Rd/Wr Share Request latch.
   b. Microprogram generates Gate TE pulse and sets Data Ready latch.
   c. Microprogram (TA Reg bit-3) brings up Share Reset momentarily to turn off Rd/Wr Share Request latch.
   d. With Data Ready latch set, printer still not busy, and Rd/Wr Share Request line down, the Cycle Interlock latch turns on.
   e. Cycle Interlock latch picks the printer cycle clutch to start the print shaft turning in the printer. The feedback contact in the 3210 closes to bring up both the Printer Busy and the Printer Feedback lines.
   f. Printer Busy (printing the character a), with Cycle Interlock latch on, resets the Data Ready latch.
   g. Data Ready line being down with printer still busy, resets the Cycle Interlock latch.
   h. Printer Feedback contact opens, dropping Printer Busy and ending the cycle.
2. Rotate the print element (see **Shift Cycle Operation**).
   a. Write latch on, Data Ready latch off, and printer not busy (same conditions as in step 1a), turns on Rd/Wr Share Request latch.
   b. Microprogram generates Gate TE pulse and sets Data Ready latch. This time, the character in the data register is a shift character. The microprogram recognizes it, discards it, and compares bit-4 of the TT-register with bit 7 of the TE register and finds that they are not the same (not ball-side match).
   c. Microprogram (TA Reg bit-3) brings up Share Reset momentarily to turn off Rd/Wr Share Request latch.
   d. With Data Ready latch set, printer not busy, and Rd/Wr Share Request line down, Cycle Interlock latch turns on.
   e. This time, because we did not get a ball-side match (Step 2b), the Cycle Interlock latch picks the printer shift magnet instead of the cycle clutch magnet.
      The remaining steps in this cycle are the same as in f, g, and h, above.
3. Print capital letter A
   This sequence is exactly the same as for printing the small letter a.
4. Gate New Line character
   This sequence is like that for the Rotate the Print Element sequence except that the carrier return magnet is picked instead of the shift magnet in the printer.



3210 KEYBOARD (READ) OPERATION

The circuit below is an anti-bounce circuit similar to that used in the Rd/Wr cycle-control circuit. Here, too, its purpose is to assure that the strobe set line is not affected by 3210 keyboard contact bounce.





SHIFT-CYCLE OPERATION

The shift-cycle sequence of the 3210 is based on both the new-design shift mechanism of the Selectric I/O-II console printer-keyboards, and a new concept of shift-position identification.

Because the graphic characters are distributed about the print element (print head) in a preferred arrangement in the 3210 PR-KBs, it is no longer correct to relate these characters to the terms "uppercase" and "lowercase." Although these terms persist, they are no longer true for the 3210, and the characters are more precisely related to their physical location on the spherical print elements as side 0 (front) or side 1 (back) characters. Refer to drawing on page 7-8 for the location of the character on the print element. The new shift mechanism has only one magnet; and each time it is pulsed, the print element rotates 180 degrees to position either side 0 or side 1 for printing, and stays there.

In every printing operation, the microprogram compares the character to be printed (TE-register, bit 7) with the side of the print element in position to print (TT-register, bit 4), to see whether the print element needs shifting.

If the two bits compare, the desired character is on the side that is ready to print, and the character prints. The microprogram then adds one to the address counter and subtracts one from the data character counter.

If the bits don't compare, a shift cycle takes place and the counters remain unchanged.

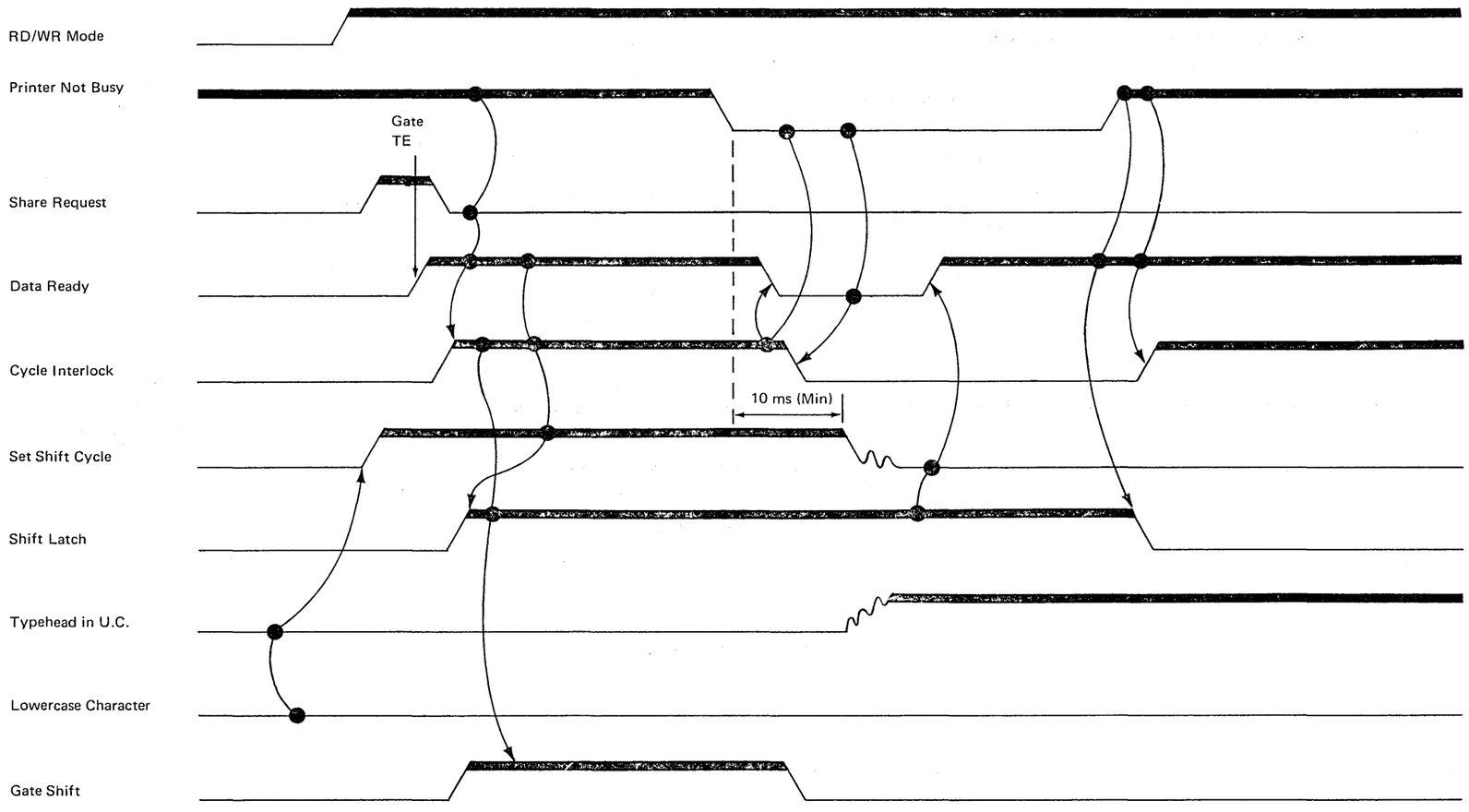
The microprogram calls this comparison the "Ball Side Match." The match results when bit 4 of the TT-register and bit 7 of the TE (Data) register are the same, and the character prints. If the two bits are not the same, then a shift cycle, instead of a print operation, occurs.

The following decode shows the four possibilities that determine whether a print or a shift operation will occur:

TT-register Bit 4	TE-register Bit 7		Resulting Action
0	0	=	Print
0	1	=	Shift
1	0	=	Shift
1	1	=	Print

These registers may be displayed by the lower roller switch on the CPU console. During print operations, if TE bit 7 is on (1), the corresponding light is lit, and those characters on the back side of the print element can print. (Refer to "Printer Tilt/Rotate Codes" for a visual representation of these characters.) The labeling on this switch position is "Ball Side", and the legend explains: "1=PRINT" or "0=print", referring to the uppercase or lowercase characters that can print.

Of course, if a mismatch exists, a shift operation occurs before another CPU cycle is requested to print the character.



3215 CONSOLE PRINTER-KEYBOARD

- The 3215 Console Printer-Keyboard consists of a wire-matrix printer with a separate elastic-diaphragm keyboard.

3215 KEYBOARD

- The keyboard is the elastic-diaphragm-switch keyboard that encodes the characters in EBCDIC for direct transmission to the system. No translation is required.
- This self-contained keyboard is independent of the printer in every way.
- The arrangement of the keys is the same as for the 3210 printer-keyboards.

In external appearance, the 3215 keyboard looks identical to that of the 3210, see "3210 Keyboard." Internally, however, it is quite different. The 3215 keyboard uses permissive-make contacts to encode the characters. An electromagnet restores the keys that are interlocked to prevent operation of more than one at a time. For additional information regarding this keyboard, consult the Theory-Maintenance manual for the *IBM 3215 Console Printer-Keyboard,* SY24-3560.

3215 PRINTER

- The printer uses no-work magnets to operate seven individual print wires arranged in a vertical row in the print head.
- The print head and magnets move back and forth along the print line, propelled by a stepper-motor-driven lead screw.
- As the print head moves, seven bytes of data fire the magnets selectively to construct the characters out of dots (made by the print wires) within a 7 by 7 matrix pattern.
- No uppercase or lowercase shift operations are required by the 3215 because this machine reproduces the characters from their matrix images in storage.
- Two driver cards in the printer provide power for the stepper motor, print-wire magnets, and carriage clutch.
- Pulse emitters on the lead screw provide feedback signals to the system for control of printing and print-head movement.
- Pulse emitters on the lead screw provide feedback signals to the system for the control of printing and print-head movement.
- The printer uses the same pin-feed carriage as the 3210 models.

Although the 3215 Console Printer-Keyboard performs the same functions as the 3210-print, space, and new line (carrier return with line feed), it performs the first two in a new way, using new mechanisms.

To print with the 3215, the adapter translates from EBCDIC to the 7 by 7 matrix code. This code is then placed in the write data (word 1 and word 2) registers and gated, byte-by-byte, to the print-wire magnets.

Additional information on this machine is available in the Theory-Maintenance manual for the IBM 3215 Console Printer-Keyboard, SY24-3560.

3215 Print Operation

- The 3215 prints by firing the print wires according to the character images (matrix codes) appearing successively in the write data (word 1 and word 1) registers.
- The microprogram moves the two-word matrix code from its storage module into the write-data register in two word-cycles for each character, including the space character.
- As the print head sweeps across the print line, print emitter pulses from the printer gate the successive bytes of the matrix code to the print magnets.
- Byte seven, and bit seven of each byte, are not used. These positions always contain zeros and are included only to preserve byte and word formats.

The 3215 is a matrix printer. It prints by firing the print magnets in a specific sequence for each character as the print head moves along the print line. The sequence for each character is contained in the two-word matrix code. This code contains an image of the dot pattern that forms the character.

The microprogram, using the procedures described under "3215 PR-KB Code Translation," locates the matrix code, and in two word-cycles, moves it into the word 1 and word 2 registers. From here, emitter pulses supplied by the printer gate firing pulses to the print magnets for every corresponding bit that is on (set to 1) in the word 1 and word 2 registers.

During the first word cycle, the microprogram moves word 1 into the word-1 position of the word 2 and word 2 registers, and updates the address count by four. (Word 1 also goes into the word-2 position but is not used in this position.) During the second word-cycle, the microprogram moves word 2 from the storage module into the word-2 position of the word 1 and word 2 registers, and starts printing.

If, for some reason the desired character is not ready in the word 1 and word 2 registers, the print head continues to move without printing and the print position is left blank. In this case, the machine hardware initiates an automatic backspace function (this function cannot be initiated manually) and retries the write operation.

The restoration time of the print-magnet armatures prohibits firing the same magnet at consecutive firing points (it can fire only at alternate firing points); therefore the same bit is never on (1) in successive bytes. Successive firings are prevented only by design, as the assigned dot-pattern format of each character (as expressed by the two-word matrix code) takes this limitation into account, and no character contains two horizontal dots that are not separated by at least one firing position. The armatures restore under their own spring tension.

When the print head reaches the end of the print line it operates the right margin-stop switch and the microprogram initiates a New Line function.

3215 PR-KB CODE TRANSLATION

- Code translation for the 3215 involves transmission to the write data (Word 1 and Word 2) registers of the matrix image of the character to be printed.
- This image is stored as a doubleword in the MATRIX CODE table in control storage.
- Using the EBCDIC representation of the character, the microprogram locates the address of the matrix code in the matrix code table.
- No translation is required for the New Line (carrier return with line-feed) function.

The EBCDIC of the character to be printed is transformed into the address of its two-word image in storage. This image, called the matrix code, is a virtual image of the dot pattern of the character. The microprogram executes two word-cycles to send this code to the Word 1 and Word 2 data registers, where it is used to fire the print-wire magnets as described under "Printing the Punctuation Character, Colon (:)"

Transforming the EBCDIC into the address of the matrix code requires two routine or operations, both involving fixed data in storage. The first routine selects the module or block of storage in which the matrix code is stored. The second routine pinpoints the matrix code within the module.

Module Selection

The 88 matrix codes are stored in three modules: B8, B9, and BA. The microprogram branches on bits 1 and 2 of the EBCDIC of each character to select its module according to the following fixed format:

MATRIX CODE
Address

00 = B8xx
01 = B8xx
10 = B9xx
11 = BAxx

Example:

Capital letter G
HEX = C7

EBCDIC =

0	1	2	3	4	5	6	7
1	1	0	0	0	1	1	1

Bits 1 and 2 are 10; therefore module is B9.

This gives us the module (B9) and the high-order portion (byte) of the matrix code address: B9xx.

Address Development

The low-order (xx) portion of the matrix code address is developed by the microprogram, using the EBCDIC of the character to be printed, and a table known as the F9XX Table. All addresses in this table begin with F9.

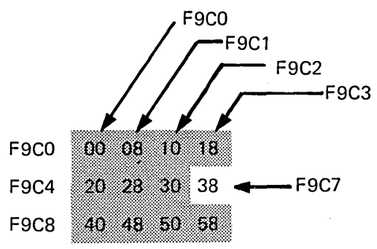
First, the microprogram modifies the EBCDIC by forcing the zero bit to 1 (this is necessary because some characters have a 0 in this position). The modified EBCDIC now becomes the low-order (xx) portion of the address of a byte in the F9XX Table.

Example:

Capital letter G
HEX = C7

Thus, substituting C7 for xx in the table address F9xx, we get F9C7.

F9XX Table



This is the address of the byte in the F9XX table that completes the address of the matrix code within the module. Looking at the F9XX Table, we find that this byte is 38. Therefore, in our example, B9xx becomes B938. This is the address of the low-order byte of the two-word matrix code for capital letter G.

The microprogram now addresses location B938 in storage and locates the first word of the matrix code for capital letter G, and moves this word into the Word-1 register.

It then increments the address by four and moves the second word of the matrix code into the Word-2 register. (Actually, Word 1 also moves into the Word-2 register at the same time that it moved into the Word-1 register but that was an incidental effect of the machine wiring and has no effect upon the operation of the write-data register).

From the Word 1 and Word 2 data registers, the active bits are gated in parallel, byte by byte, to the print-wire magnets.

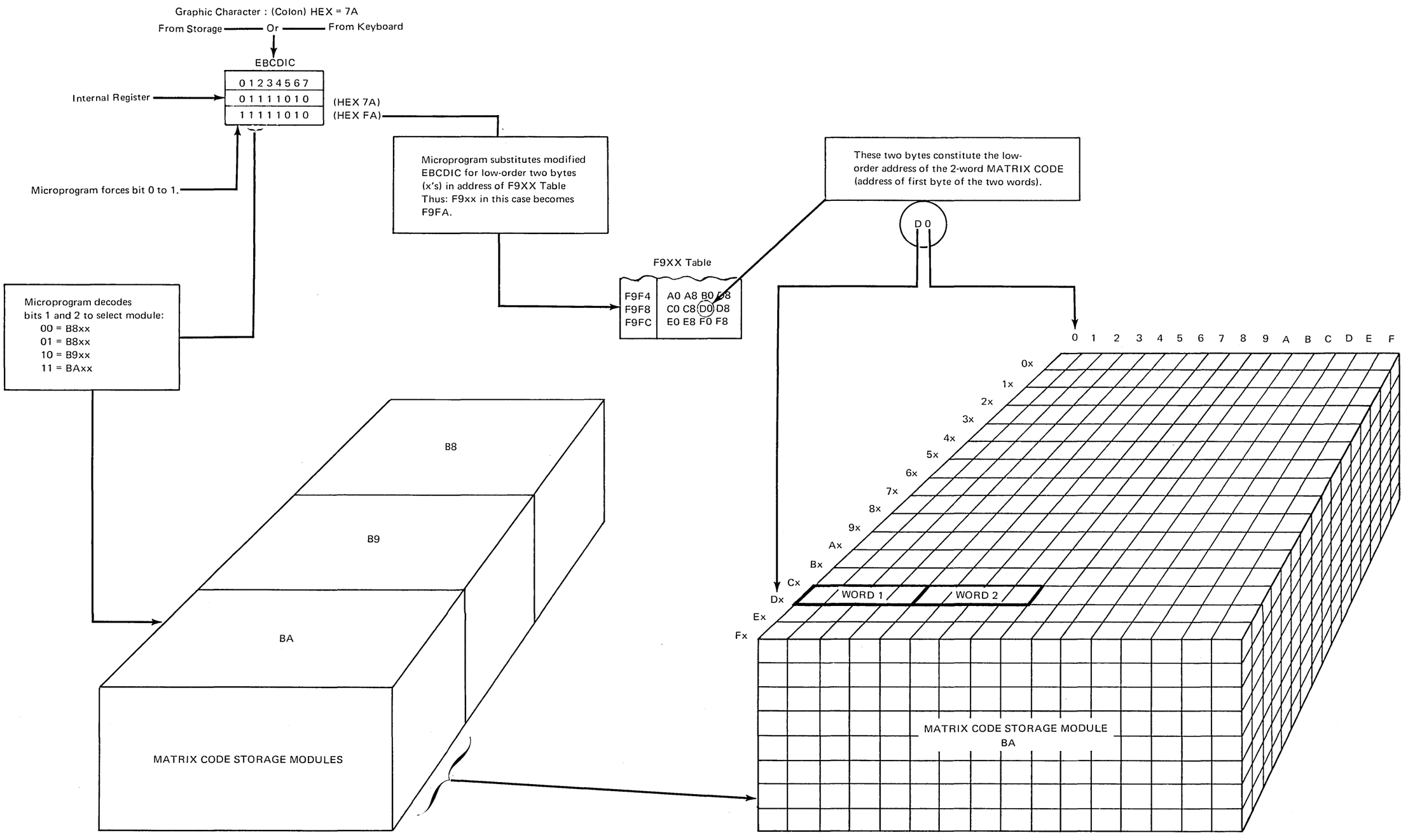
This procedure is repeated for every printable character and for the Space function, for which the two-word matrix code is all zeros.

No translation to the matrix code is required for the New Line function, which can be initiated by gating 08 into the TE-register.

3215 7x7 Matrix Codes

CHARACTER	MATRIX CODE Word 1	Word 2	CHARACTER	MATRIX CODE Word 1	Word 2
a	040A200A	201E0000	A	1E204880	48201E00
b	FE002200	221C0000	B	827C8210	82106C00
c	001C2200	22002200	C	7C820082	00824400
d	001C2200	2200FE00	D	827C8200	82007C00
e	001C2208	22180000	E	FE009200	92008200
f	20007E80	20800000	F	FE09000	90008000
g	00102802	28023C00	G	7C820082	10825C00
h	00FE0020	00201E00	H	FE001000	1000FE00
i	000000BE	00000000	I	008200FE	00820000
j	00020002	BC000000	J	04020002	0002FC00
k	FE000814	22000000	K	FE002010	48048200
l	000000FE	00000000	L	FE000200	02000200
m	1E20003E	00201E00	M	BE402010	2040BE00
n	3E002000	201E0000	N	BE402010	0804FA00
o	001C2200	221C0000	O	7C820082	00827C00
p	3E002800	28100000	P	FE009000	90006000
q	10280028	003C0200	Q	7C820082	08847A00
r	201E0020	00200000	R	FE009000	98046200
s	00102A00	2A040000	S	64920092	00924C00
t	002000FC	02200000	T	8000807E	80008000
u	003C0200	023C0000	U	FC020002	0002FC00
v	00380402	04380000	V	F0080402	0408F000
w	3C02000E	00023C00	W	FC020418	0402FC00
x	00221408	14220000	X	82442810	28448200
y	0020100E	10200000	Y	8040201E	20408000
z	0022042A	10220000	Z	82048A10	A2408200
¢	384400C6	00440000	0	38448200	82443800
•	00000600	06000000	1	004200FE	00020000
<	00102844	82000000	2	42840288	02906200
(	00003844	82000000	3	84028012	A0528C00
+	1000106C	10001000	4	08102840	88760800
|	000000FE	00000000	5	E200A200	A2009C00
&	0C52A052	08040A00	6	0C122052	80120C00
!	000000F2	00000000	7	80028408	9020C000
$	205400D6	00540800	8	6C920092	00926C00
*	10443800	38441000	9	60900294	08906000
)	00008244	38000000	⟩	00008244	28100000
;	0000DA04	D8000000	?	0040800A	90600000
¬	10001000	10001C00	:	00006C00	6C000000
-	10001000	10001000	#	2800EE00	EE002800
/	02040810	20408000	@	38448230	8A403A00
,	00001A04	18000000	'	0000D020	C0000000
%	C204C810	26808600	=	28002800	28002800
_	02000200	02000200	"	0000E000	00E00000

TRANSLATING FROM EBCDIC TO MATRIX CODE



F9XX TRANSLATE TABLE--CHARACTER LOCATIONS

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F90x																
1x																
2x																
3x																
4x																
5x																
6x																
7x																
F98x	SP 00	a 08	b 10	c 18	d 20	e 28	f 30	g 38	h 40	i 48	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00
9x	SP 00	j 50	k 58	l 60	m 68	n 70	o 78	p 80	q 88	r 90	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00
Ax	SP 00	SP 00	s 98	t A0	u A8	v B0	w B8	x C0	y C8	z D0	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00
Bx	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00	SP 00
Cx	SP 00	A 08	B 10	C 18	D 20	E 28	F 30	G 38	H 40	I 48	Ç 50	. 58	< 60	(68	+ 70	| 78
Dx	& 80	J 88	K 90	L 98	M A0	N A8	Ð B0	P B8	Q C0	R C8	! D0	$ D8	* E0	) E8	; F0	¬ F8
Ex	00	/ 08	S 10	T 18	U 20	V 28	W 30	X 38	Y 40	Z 48	SP 50	, 58	% 60	_ 68	> 70	? 78
Fx	0 80	1 88	2 90	3 98	4 A0	5 A8	6 B0	7 B8	8 C0	9 C8	: D0	# D8	@ E0	' E8	= F0	" F8

Example:

Character to be printed : / D0 — Address of the first word of the two-word matrix code within the module.

F9XX Table

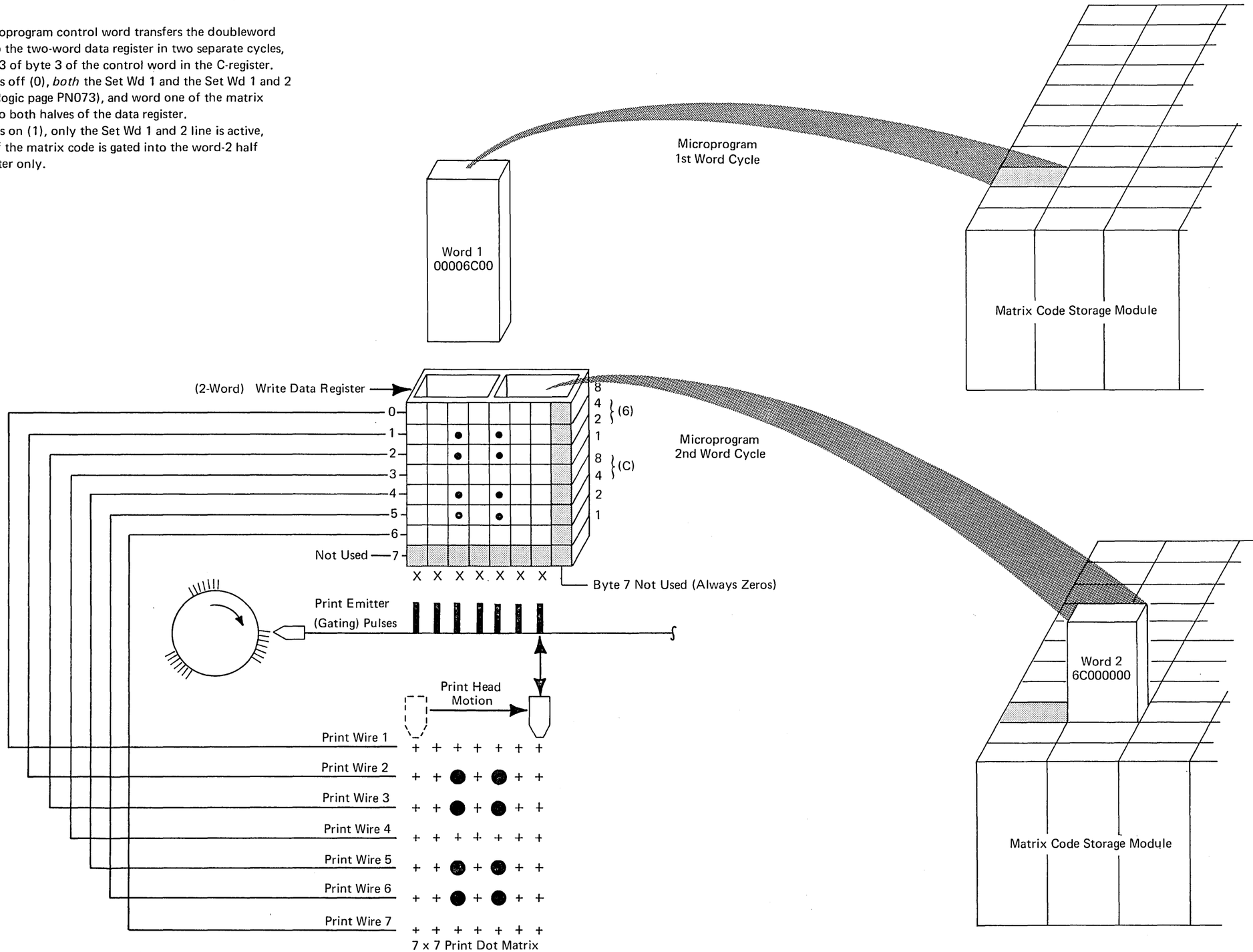
F980	00 08 10 18	F9C0	00 08 10 18
F984	20 28 30 38	F9C4	20 28 30 38
F988	40 48 00 00	F9C8	40 48 50 58
F98C	00 00 00 00	F9CC	60 68 70 78
F990	00 50 58 60	F9D0	80 88 90 98
F994	68 70 78 80	F9D4	A0 A8 B0 B8
F998	88 90 00 00	F9D8	C0 C8 D0 D8
F99C	00 00 00 00	F9DC	E0 E8 F0 F8
F9A0	00 00 98 A0	F9E0	00 08 10 18
F9A4	A8 B0 B8 C0	F9E4	20 28 30 38
F9A8	C8 D0 00 00	F9E8	40 48 50 58
F9AC	00 00 00 00	F9EC	60 68 70 78
F9B0	00 00 00 00	F9F0	80 88 90 98
F9B4	00 00 00 00	F9F4	A0 A8 B0 B8
F9B8	00 00 00 00	F9F8	C0 C8 D0 D8
F9BC	00 00 00 00	F9FC	E0 E8 F0 F8

PRINTING THE PUNCTUATION CHARACTER, COLON (:)

The RDMP microprogram control word transfers the doubleword matrix code into the two-word data register in two separate cycles, governed by bit 3 of byte 3 of the control word in the C-register.

When this bit is off (0), *both* the Set Wd 1 and the Set Wd 1 and 2 lines are active (logic page PN073), and word one of the matrix code is gated into both halves of the data register.

When this bit is on (1), only the Set Wd 1 and 2 line is active, and word two of the matrix code is gated into the word-2 half of the data register only.



REMEMBER

There is a Reader's Comment Form at the back of this publication.

3215 PR-KB INTEGRATED ATTACHMENT

- The 3215 PR-KB attachment is a separate circuitry package integrated into the CPU on Gate B, Board B1.
- The TA-, TT-, and TI-registers generally function like the corresponding registers in the 3210 attachment but with some variations.
- The TI-register accepts data in EBCDIC form (without translation) from the keyboard. For diagnostic purposes, it also accepts signals from the data-register assembler and the control/timing circuits for entry into the system.
- The TE-register is no longer a data register but contains eight latches that provide new control lines for 3215 PR-KB operations.
- Data is read out two words at a time, with the 3215, and placed in the two-word data register by the microprogram.
- The control/timing circuits, using pulses supplied by the console printer, control keyboard interlocking, forms feeding and print-wire-magnet firing.
- The stepping-motor control circuits, using feedback pulses from the printer, control both left- and right-hand motion of the print head across the print line.

The 3215 attachment circuitry supplies all of the control lines necessary for operation of the printer-keyboard with the system. Some of these lines take care of conditions that are peculiar to the 3215. One of these conditions is *late data.*

Because the 3215 operates in a continuous mode while writing (the print head sweeps across the print line without interruption), it is remotely possible that a data character might not be present in the data register at the precise moment that the print head is passing the location where the character should print. In this case, the attachment senses that no character is being printed, and provides stop and backspace signals to make the print head go back one space and try again. This is an unlikely situation, but circuitry is provided in the attachment to handle it if it ever arises. The time required to backspace the print head is more than enough to ensure that even the slowest character will have arrived in the data register in time for the retry.

Certain online and offline diagnostic testing functions are possible with the 3215 that are not provided for the 3210. Again, the 3215 adapter circuitry provides the signals required to perform this testing.

The adapter also provides for starting, running, stopping, and reversing the printer stepper motor. The circuitry that does this is grouped in the "stepping-motor control" block on the overall data flow and control drawing.

Because the 3215 can operate continuously (as when writing), or on a character-by-character basis (as from the keyboard), separate timing charts are included in this section for single- and multiple-character operations. In either mode, and regardless of the direction of print-head motion, the stepping-motor stopping sequence is the same and begins with the setting of the stop latch.

3215 DATA AND CONTROL REGISTERS

- Four registers in the 3215 PR-KB adapter provide control of the printer-keyboard input/output and diagnostic operations.
- These are the TA (Tag Out), TT (Tag In), TI (Bus-In), and TE (Bus-Out) registers.
- The data register is an added two-word storage device that contains the matrix image of each character while it is being printed.

TA-Register

Except for the added bit-6 latch used for diagnostic reset, this register functions just like the TA-register for the 3210.

The new latch (bit 6) generates a simulated power-on reset signal that resets, among other things, the read, write, and stacked request (TA-register bit 2) latches.

TT-Register

This register consists of eight flip latches plus another for parity. These latches are scattered throughout the logic pages.

Bit latches 0, 1, and 2 develop the same 'attention ready', and 'alter/display mode' lines as for the 3210 PR-KBs. In fact, the 3215 uses the actual alter/display mode latch hardware of the 3210 on gate A. This latch, therefore, does not appear in the 3215 logic pages.

The output of bit latches 3 and 4 is decoded by circuits to develop the following lines:

Bit 3	*Bit 4*	*Signal*
0	1	Data Parity
1	0	Time Out (Cycle Interlock)
1	1	PE Sync Check

Bit 5 latch develops 'end' or 'ready share request', depending on whether it is activated by the End key or by making the device ready.

Bit latches 6 and 7 are the request and cancel key latches that function just as for the 3210.

TE-Register

The TE-register has new functions for the 3215. No longer a data register, the TE-register is now an important control device for gating various PR-KB control lines to the system. These lines are mainly diagnostic, to enable the service representative to simulate various signals for both online and offline testing of the 3215 PR-KB.

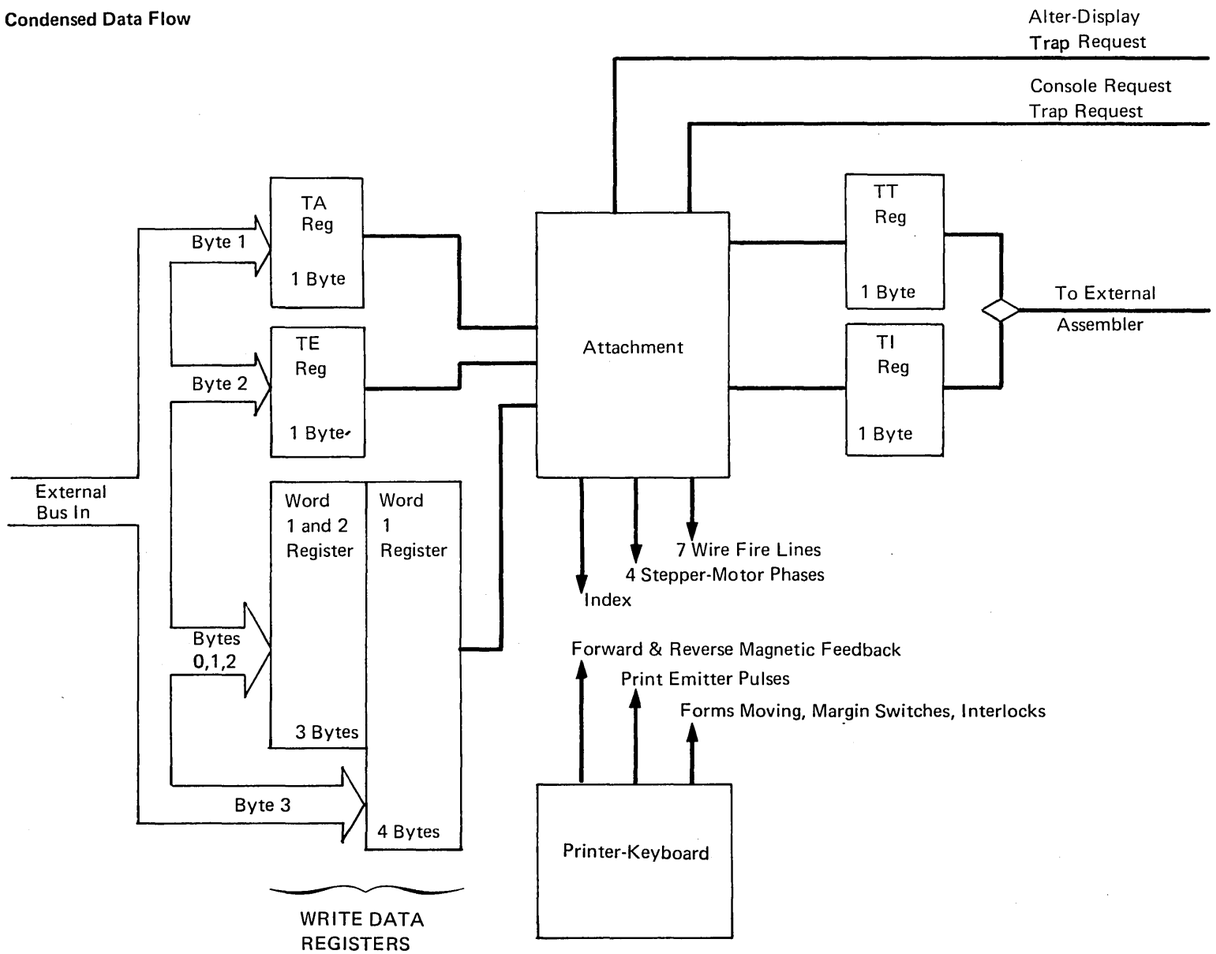
The register consists of eight flip latches, plus another for parity. The outputs of the zero- and one-bit latches are decoded by AND circuits to produce two new diagnostic control lines—'gate diagnostic wire firing', and 'gate diagnostic MP lines'.

Bit-2 latch generates the 'diagnostic PE timing' line.
Bit-3 latch generates the 'diagnostic integrated PE' line.
Bit-4 latch generates the 'new-line character' line.

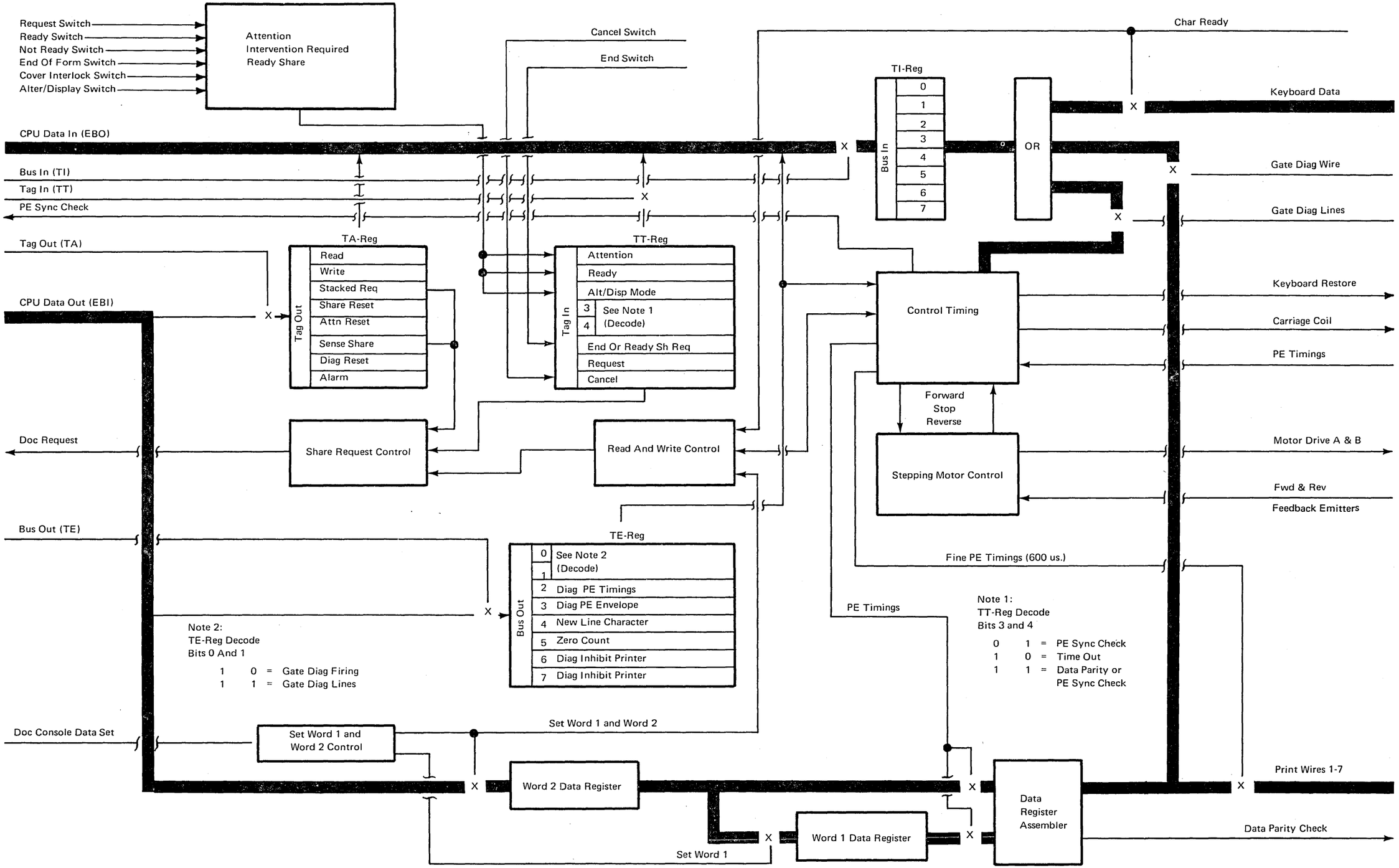
TI-Register

The TI-register for the 3215, in addition to accepting input from the keyboard, also forces up certain lines for testing by gating to the CPU those diagnostic control signals shown on page 7-35.

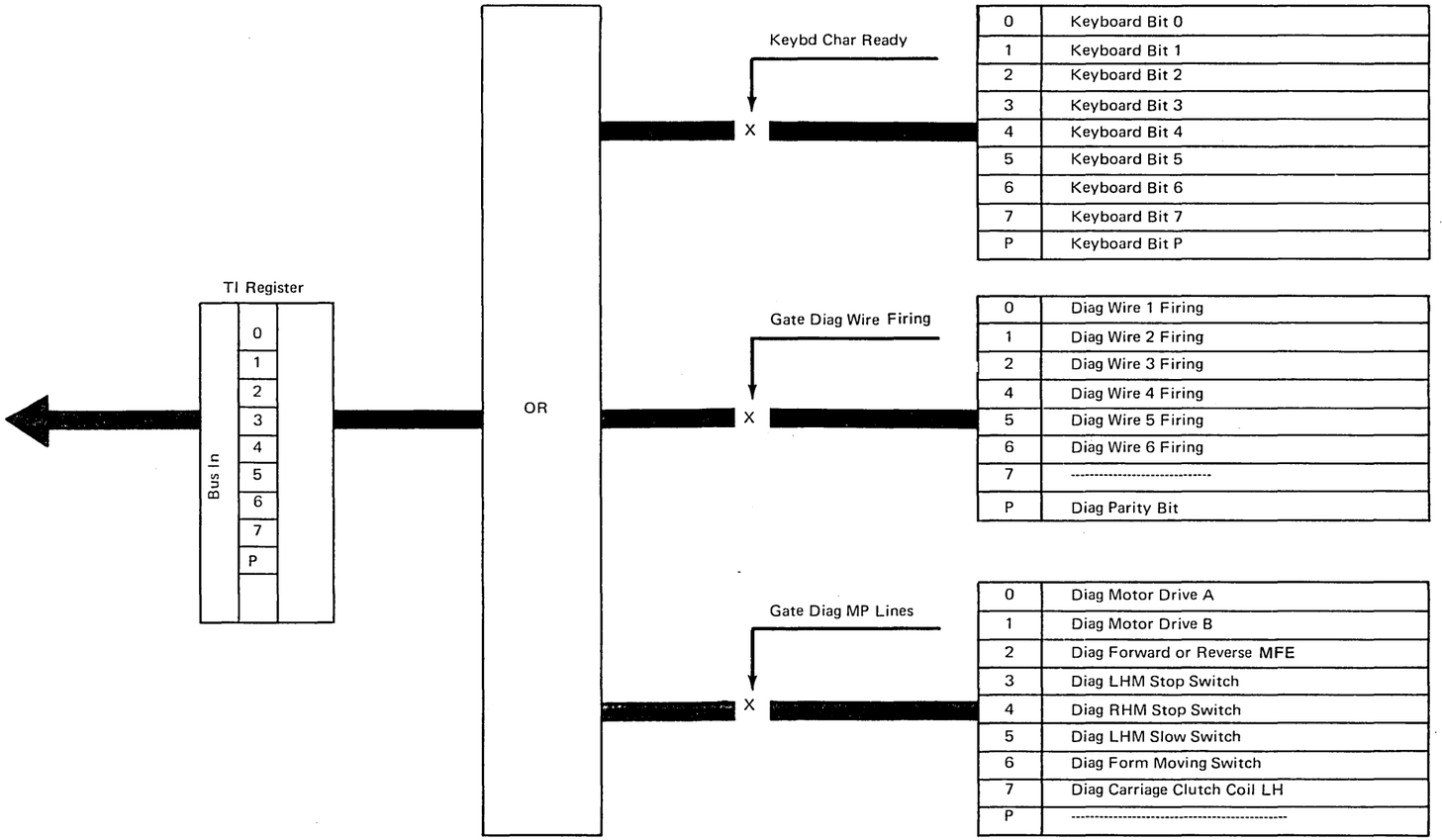
Condensed Data Flow



3215 CONTROL AND DATA FLOW



TI Register Input Gating



0	Keyboard Bit 0
1	Keyboard Bit 1
2	Keyboard Bit 2
3	Keyboard Bit 3
4	Keyboard Bit 4
5	Keyboard Bit 5
6	Keyboard Bit 6
7	Keyboard Bit 7
P	Keyboard Bit P

0	Diag Wire 1 Firing
1	Diag Wire 2 Firing
2	Diag Wire 3 Firing
4	Diag Wire 4 Firing
5	Diag Wire 5 Firing
6	Diag Wire 6 Firing
7	----------
P	Diag Parity Bit

0	Diag Motor Drive A
1	Diag Motor Drive B
2	Diag Forward or Reverse MFE
3	Diag LHM Stop Switch
4	Diag RHM Stop Switch
5	Diag LHM Slow Switch
6	Diag Form Moving Switch
7	Diag Carriage Clutch Coil LH
P	----------

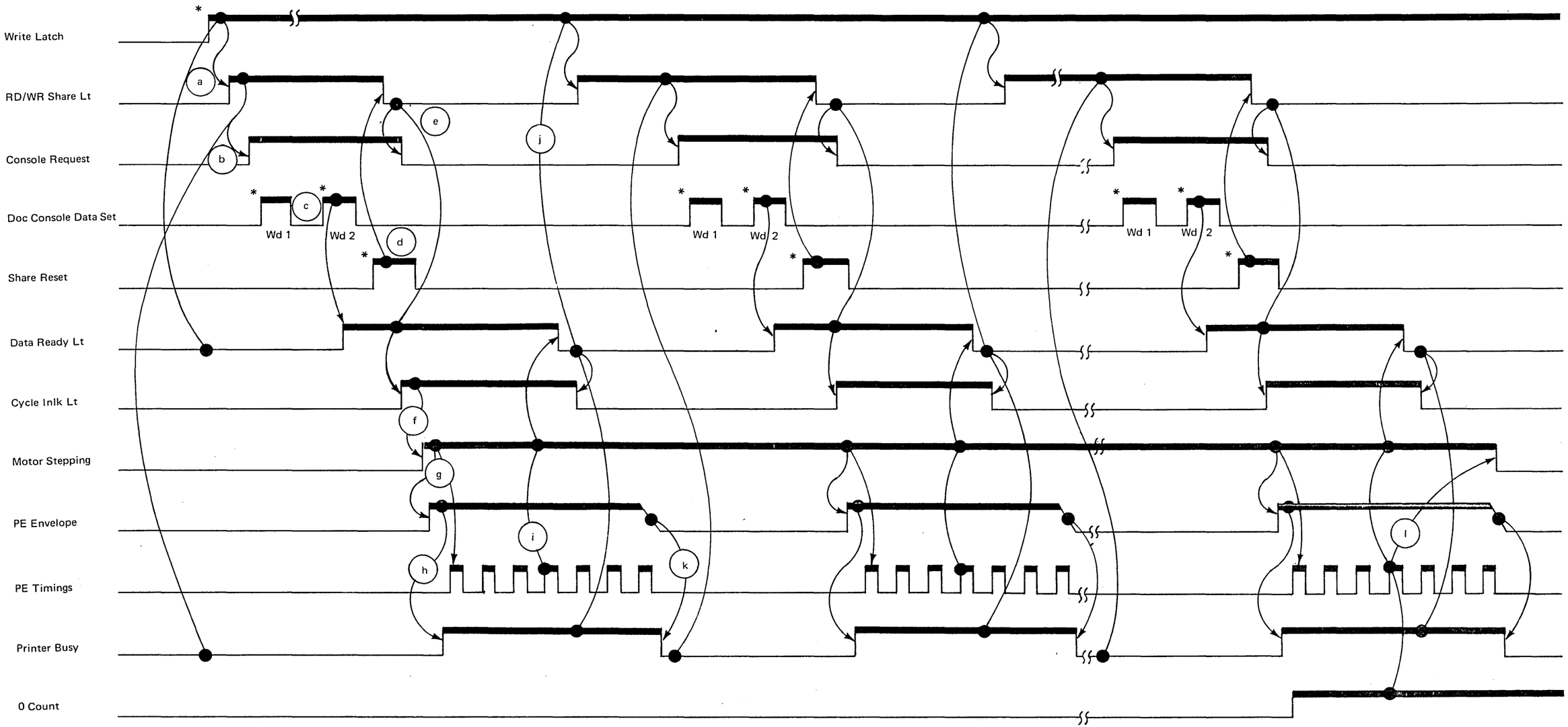
3215 WRITE OPERATION

- Printer operates in continuous mode.
  (Motor stepping stays up until last character prints.)
  a. Write latch (TA-Reg bit 1) turned on by write command.
  b. Rd/Wr Share up and Ptr not busy turn on console request.
  c. Microprogram moves matrix code words from storage module into WD 1 & WD 2 Regs, and brings up data ready latch.
  d. Microprogram brings up Share Reset to drop Rd/Wr Share latch.
  e. Rd/Wr Share latch being off, with Data Ready latch on, turns on Cycle Inlk and drops Console Request.
  f. Cy Inlk on turns on Forward Latch, starts print emitter turning, brings up Motor Stepping line.
  g. Motor Stepping brings up PE Timings and PE Envelope.
  h. PE Envelope brings up Printer Busy.
  i. PE 4 pulse, with Motor Stepping up, turns off Data Ready Lt.
  j. Write latch on with Data Ready off and Printer Busy, turns off Cycle Inlk latch and turns on Rd/Wr Share latch to request another character.
  k. Envelope collapses after 7th PE pulse, dropping Ptr Busy. Cycle now repeats, starting over again with Step b, except that the Motor Stepping line remains up (Step f is omitted).

  This operation continues until the zero-count line comes up, indicating that the last character is in the data register.

  l. This time PE 4 pulse (Step i), because the Zero Count line is up, additionally resets the Motor Stepping line and initiates the 4-pulse stopping sequence. With motor Stepping line down, Ptr Busy drops; the stepper motor stops and cannot again pick Printer Busy (step h),

Write Operation (Write Mode) Not To Scale



* Microprogram

3215 KEYBOARD (READ) OPERATION

- Printer operates in start-stop mode.

a. Read latch (TA-Reg bit 0) turned on by Read command. Read latch, with Keybd Char not ready, turns on Proceed latch.

b. Operator presses key on keyboard. Keyboard contacts in keyboard, with keyboard not restored, bring up keyboard data lines.

c. Keyboard data lines being up, turn on Keybd Char Ready latch.

d. Keyboard Character Ready line up, gates Strobe Set pulse, restores (locks) the keyboard, and turns off the Proceed latch.

e. Strobe Set pulse, turns on Rd/Wr share Lt.

f. Rd/Wr Share latch, with Ptr Not Busy, turns on Console Request.

g. Microprogram takes the keyed EBCDIC data from the TI-Reg and puts it in the Wd1 and Wd2 data registers.

h. Microprogram issues share request pulse to turn off Rd/Wr share latch.

i. Rd/Wr share latch being down with data ready latch on, turns off Console Request and turns on cycle inlk latch.

j. Cycle Inlk turns on Fwd Latch, which starts print emitter turning and brings up motor stepping line (from here on action is the same as for last character in Write op).

k. Motor Stepping brings up PE Timings and PE Envelope.

l. PE Envelope brings up Printer Busy.

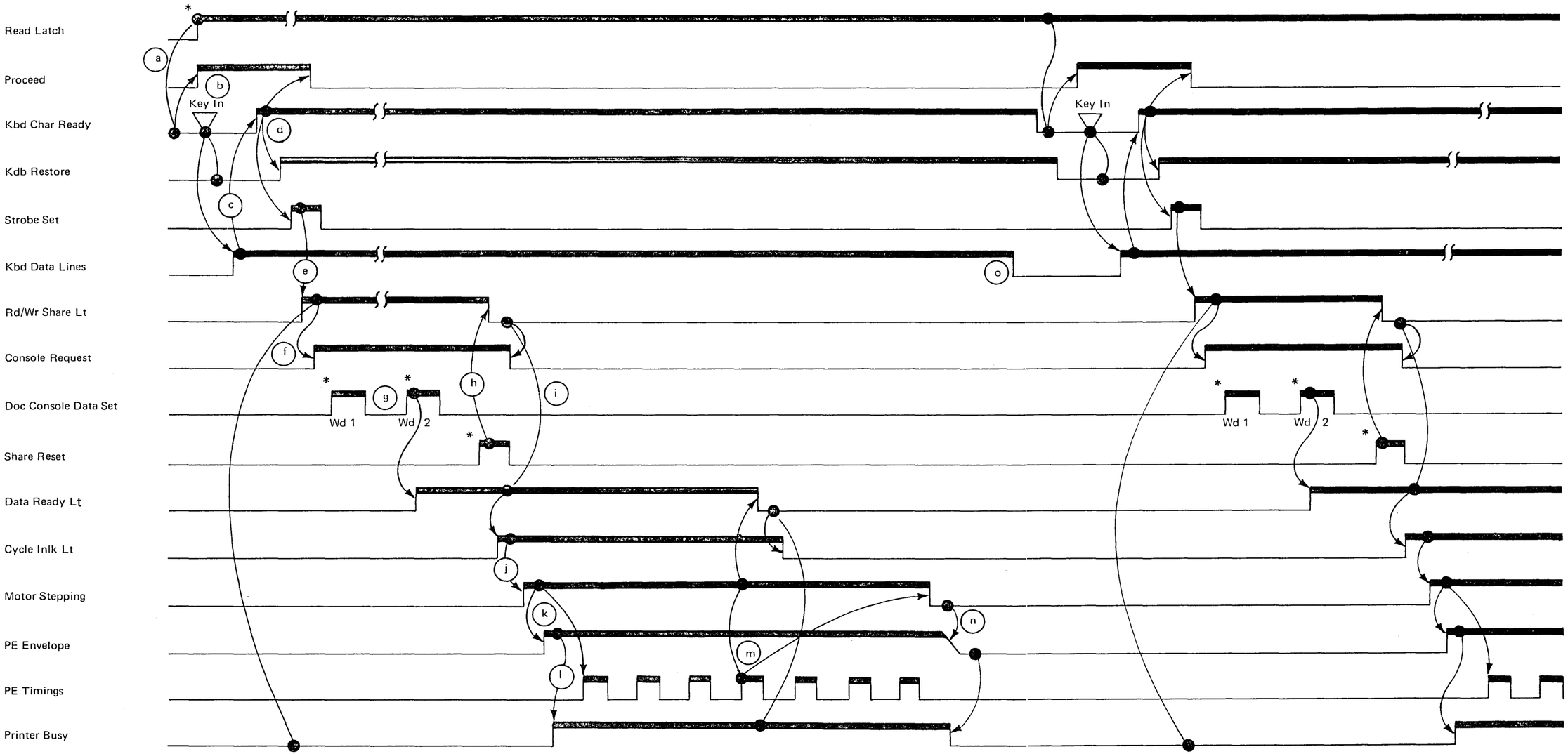
m. PE 4 pulse with Motor Stepping up, resets data ready Lt and drops Motor Stepping line.

n. With Motor Stepping down, PE Envelope collapses, turning off Printer Busy.

o. Keyboard data lines drop; and because read latch is still on, the Proceed light comes on, and another character may be keyed.

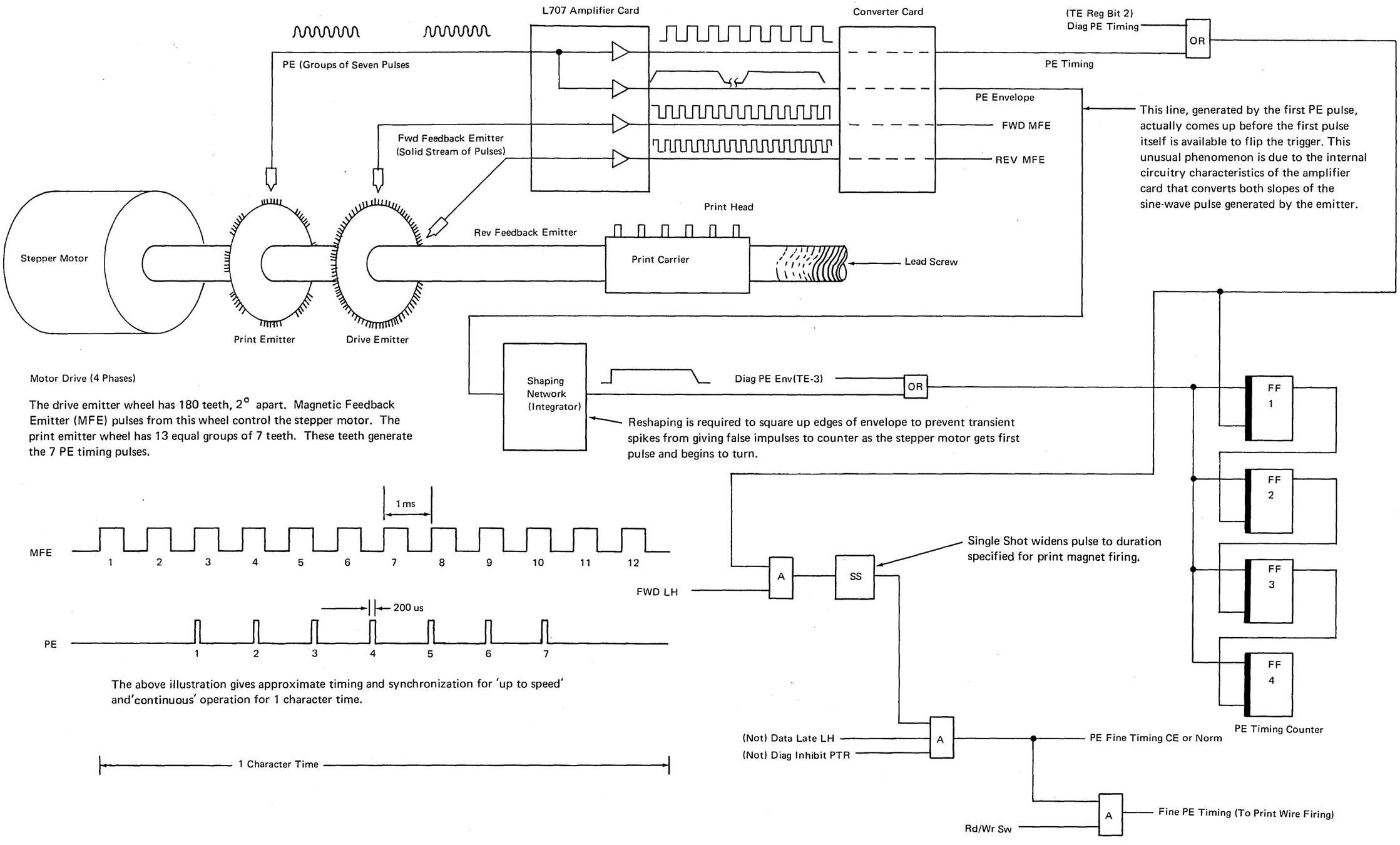
This cycle of events may be repeated until the read latch is turned off by the microprogram.

KEYBOARD OPERATION (READ MODE) NOT TO SCALE



*Microprogram

EMITTER PULSE CONVERSION

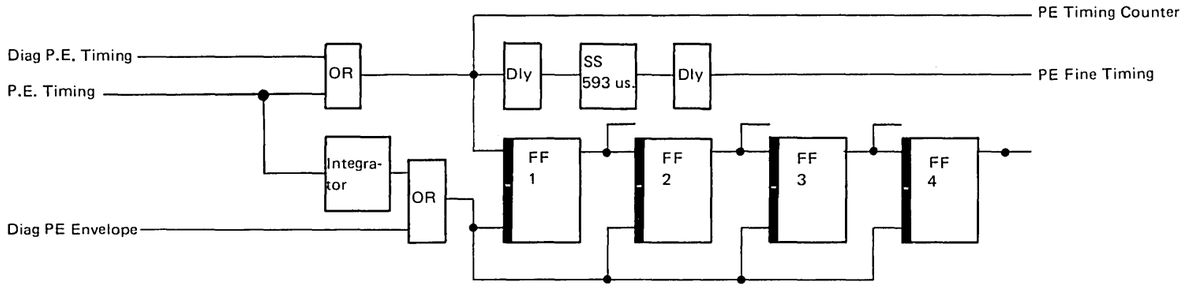


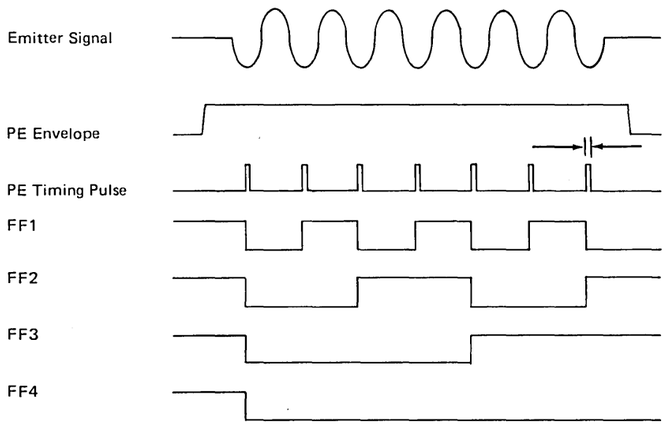
Motor Drive (4 Phases)

The drive emitter wheel has 180 teeth, 2° apart. Magnetic Feedback Emitter (MFE) pulses from this wheel control the stepper motor. The print emitter wheel has 13 equal groups of 7 teeth. These teeth generate the 7 PE timing pulses.

The above illustration gives approximate timing and synchronization for 'up to speed' and 'continuous' operation for 1 character time.

PRINT EMITTER TIMING COUNTER



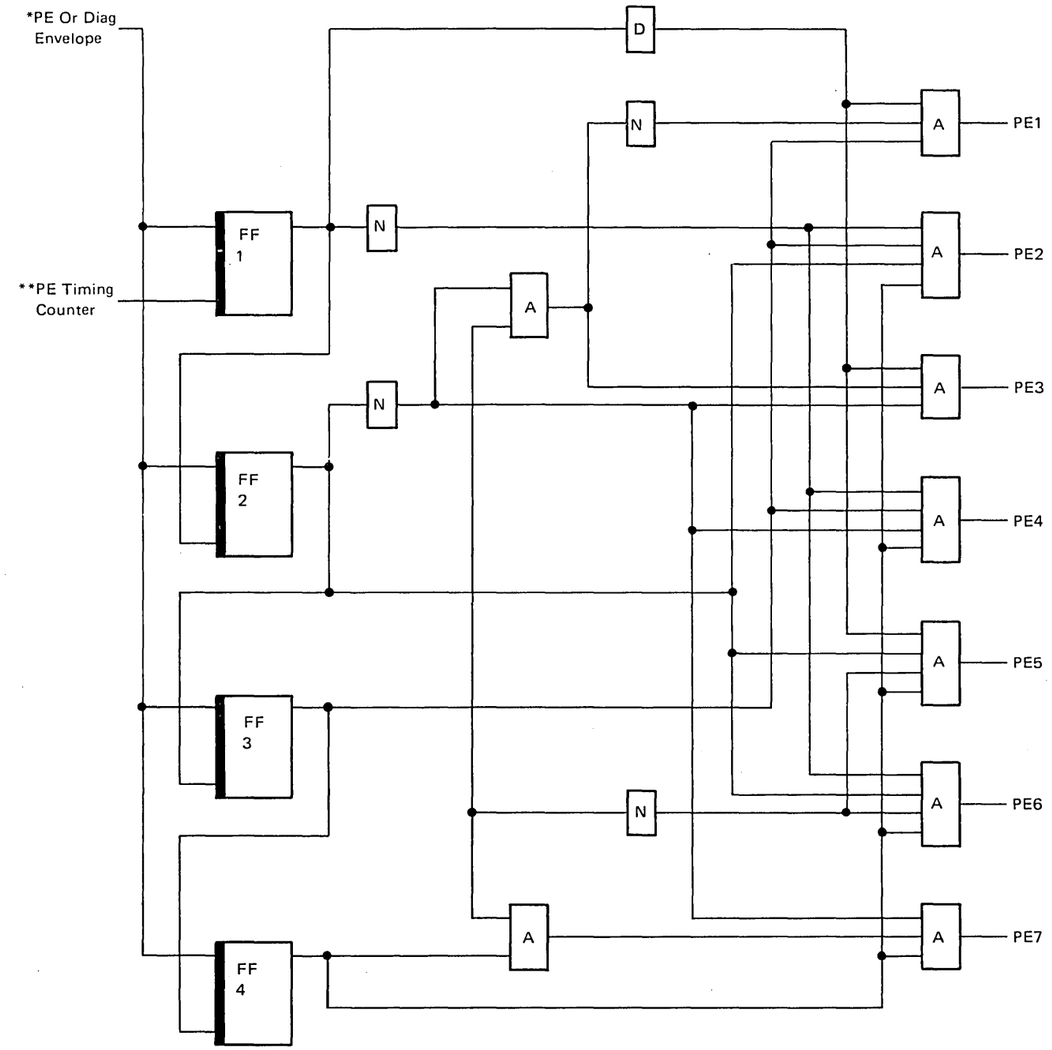


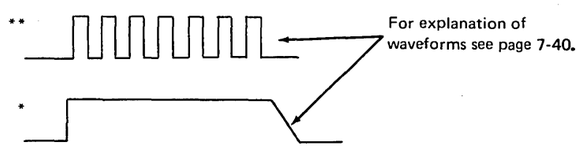
FF1	FF2	FF3	FF4	
0	0	0	0	PE1
1	0	0	0	PE2
0	1	0	0	PE3
1	1	0	0	PE4
0	0	1	0	PE5
1	0	1	0	PE6
0	1	1	0	PE7

The 3215 attachment contains a four-stage binary counter for keeping track of the print emitter (PE) pulses. The counter outputs provide individually identifiable PE pulses 1-7. The PE4 pulse is used to gate all the conditions that set the stop latch (see "Stop Latch Set Conditions," Page 7-47), and to reset the data ready latch for each character during read or write operations.

1. The emitter signal from the printer is amplified, and PE timing pulses are generated from the negative peaks.
2. The positive pulse triggers an integrator circuit that stays positive for a little over one cycle. With emitter pulses arriving at the correct rate, the integrator output, PE envelope, stays positive for all seven cycles.
3. PE envelope being off, sets all the FFs on. PE envelope coming on allows each negative transition at the input of the FF to complement it.

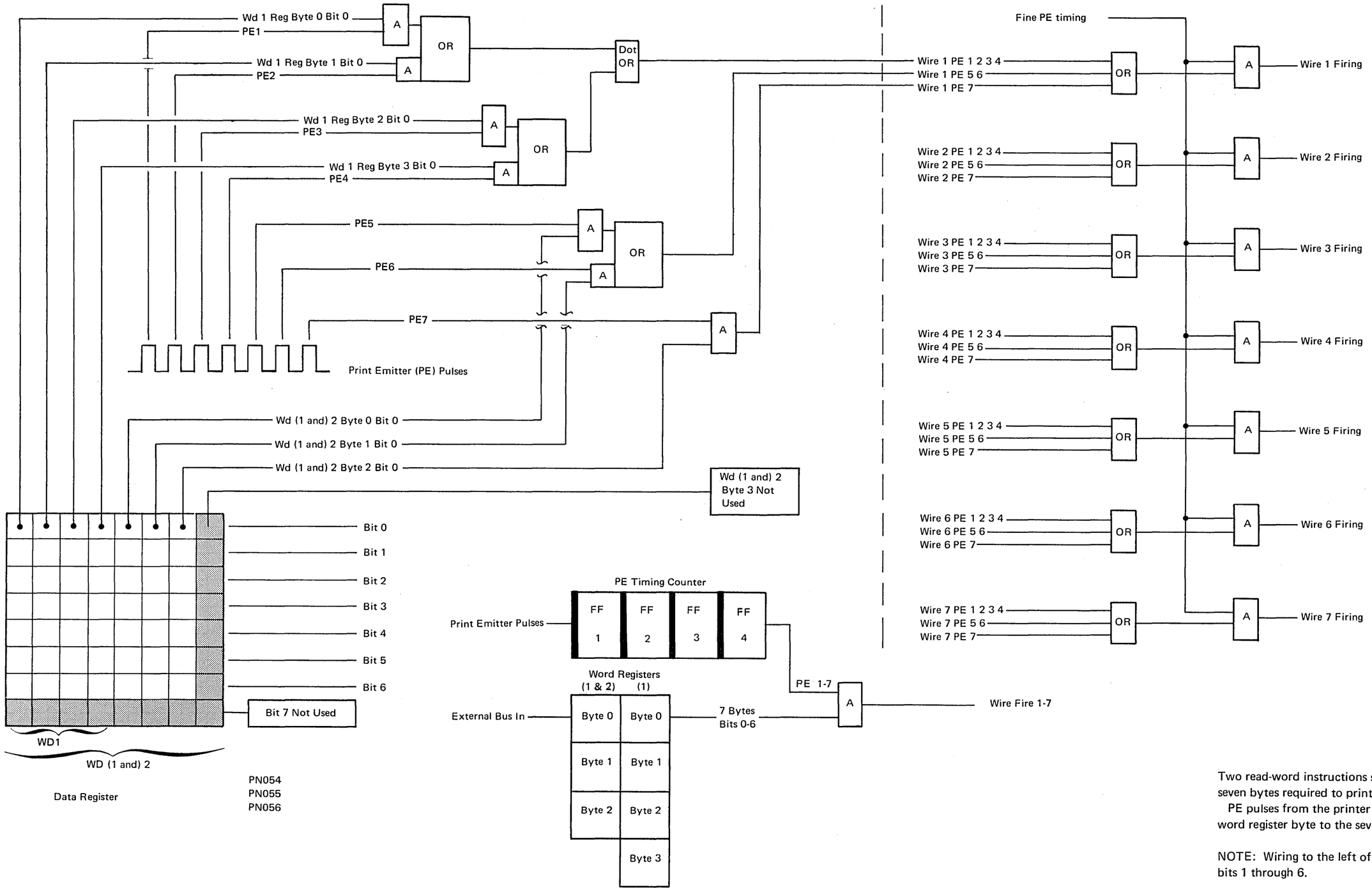
PE 1-7 LINE GENERATION





PN012

3215 Print Magnet Firing Circuits

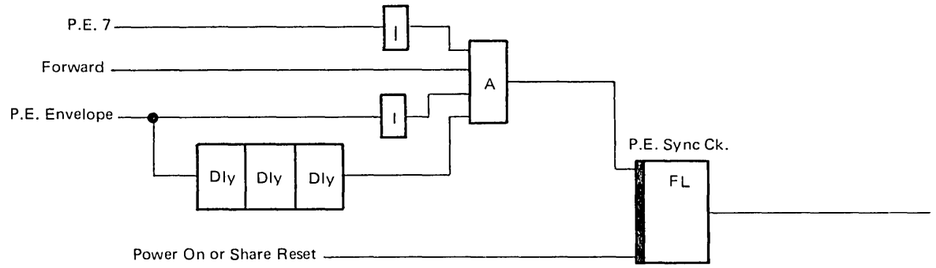


Two read-word instructions set the word registers with the seven bytes required to print the character.

PE pulses from the printer step the counter and gate the word register byte to the seven wire fire lines.

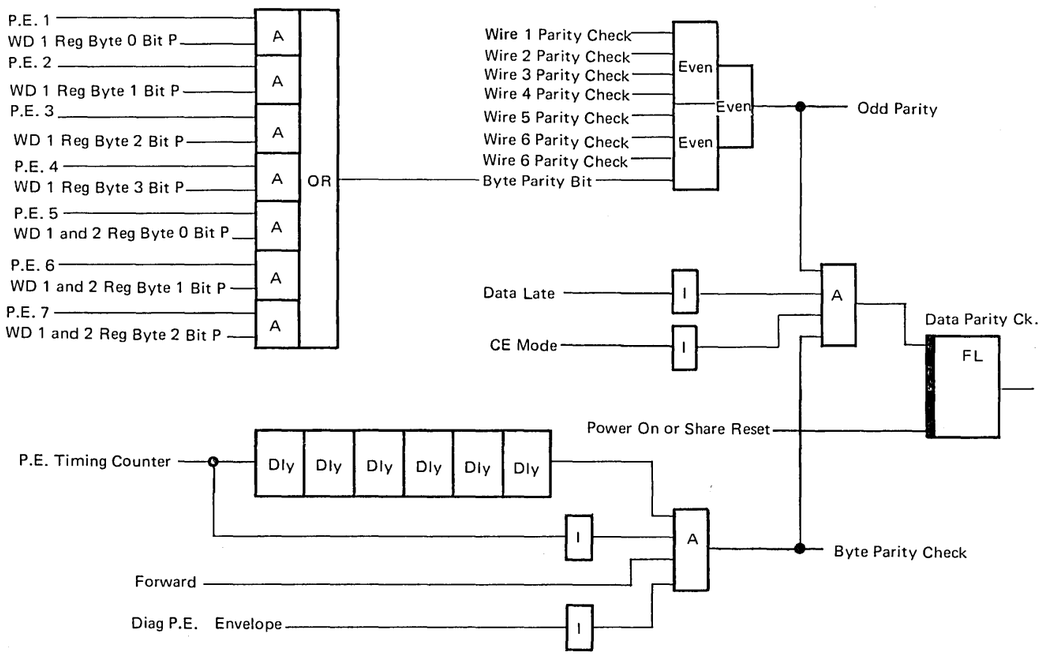
NOTE: Wiring to the left of the dashed line is duplicated for bits 1 through 6.

PRINT EMITTER SYNC CHECK



A PE sync check is a value other than 7 in the PE timing counter when the PE envelope drops . At the time PE envelope falls, a short sampling pulse is generated through a delay block. If PE7 is not active, the PE sync-check latch is set.

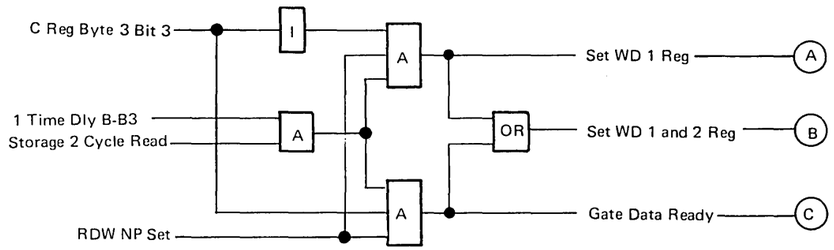
DATA PARITY CHECK



At each PE time, a new byte is parity-checked. A sampling pulse is generated as the PE pulse falls. The parity bit for the byte being printed is gated directly from the word register; the rest of the byte is sampled after it has been gated to the wire-fire lines.

SETTING the WORD 1 and WORD 2 REGISTERS

(1)

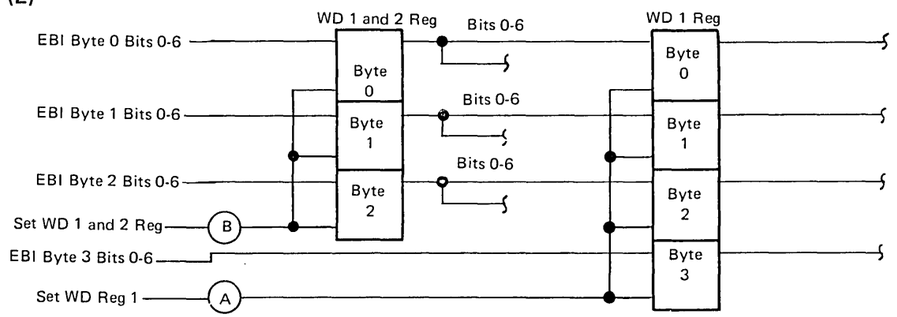


Two read-word instructions gate EBI to the word regs:

	NXTLBL 0,X,X	RDMP	LS--, LS-- +4	Gates (A) and (B)
NXTLBL 0XX	NXTLBL 1,X,X	RDMP	LS--, LS--	Gates (B) and (C)
NXTLBL 1XX	----------			

C-Reg Byte 3 Bit 3 Sets from 0XX and 1XX

(2)



The first of the two RDMP words sets both WD registers.

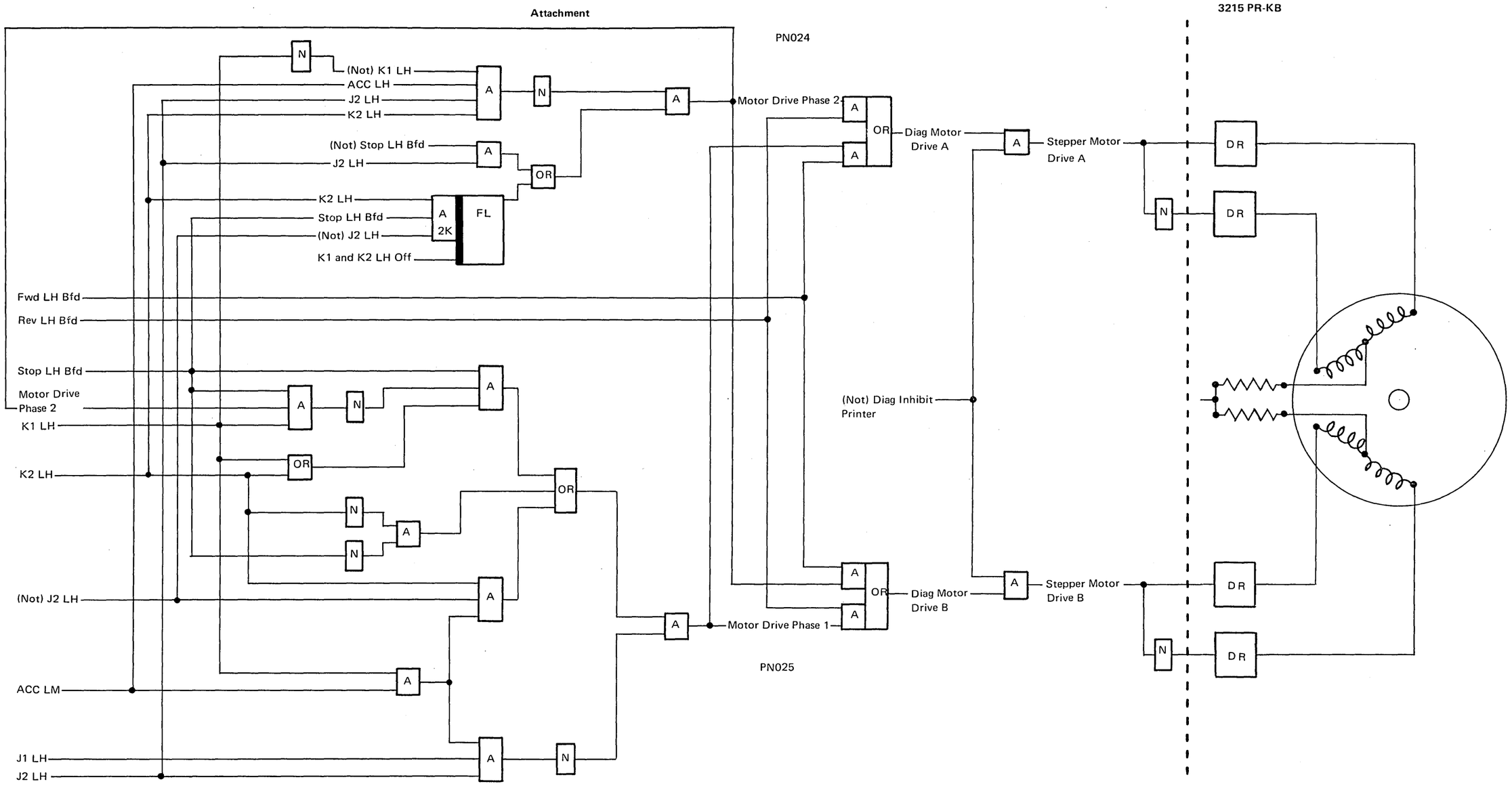
BX, BH, BL at 0XX cause C-register byte 3 bit 3 to be off.

EBI bytes 0, 1, and 2 pass through the WD 1 and WD 2 registers and set in bytes 0, 1, and 2 of the WD 1 register. EBI byte 3 goes directly to WD1 register, byte 3.

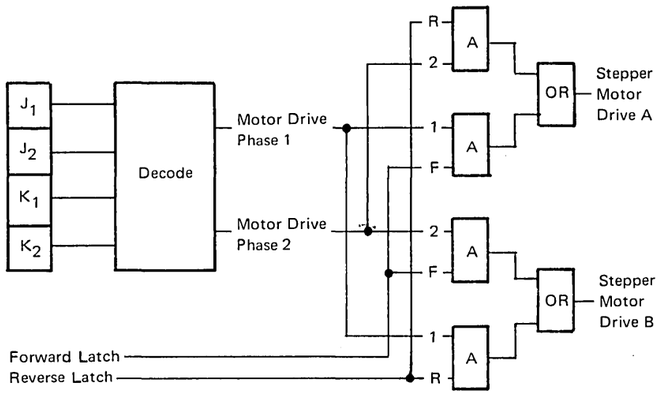
The second RDMP word with BX, BH, BL at 1XX activates the set to WD1 and WD2 registers gates data ready.

EBI byte 3 of the second RDMP word is not used, nor is bit 7 of EBI bytes 0, 1, and 3.

STEPPER MOTOR CONTROL



Motor Control Logic



This diagram shows the effect of forward and reverse latches on stepper-motor drive A and B. Note that for either direction, forward or reverse, the outputs of the JK latches and decode circuits are identical, and only the motor drive phases are switched before sending them to the motor.

The following circuits are found in the 'decode' block above. These circuits perform functions required for efficient and reliable operation of the stepper motor and are presented in the order in which they occur.

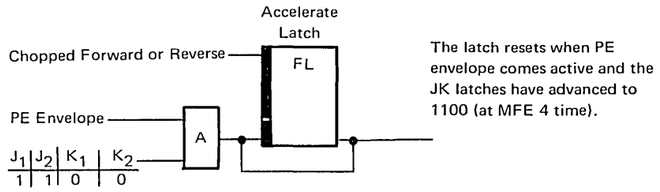
Chopped Forward or Reverse Singleshot

It is fired whenever the forward or reverse latch is turned on. Its purpose is to block possible noise pulses from the magnetic feedback emitter from reaching the JK advance circuits when the motor first starts to turn.

Accelerate Latch

This latch is used to bring the stepper motor up to speed quickly from a stopped state.

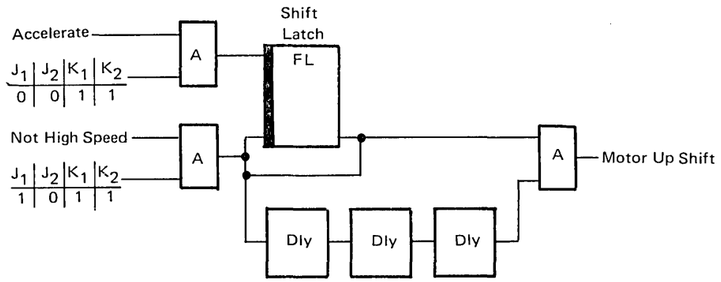
It is set on by the turn on of the forward or reverse latches, and its output is used to gate on the shift latch.



The latch resets when PE envelope comes active and the JK latches have advanced to 1100 (at MFE 4 time).

Shift Latch

This latch inserts an extra pulse into the JK advance circuits, stepping the JK latches, and advancing the motor-drive phasing, which causes acceleration in the stepper-motor speed.

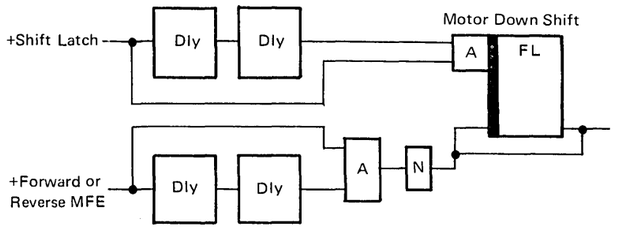


At the time the fourth MFE pulse arrives, the JK latches advance to 0010. This combination ANDs with Accelerate Latch to set the shift latch on. A pulse is generated and sent to the JK advance circuits, advancing the phasing sent to the motor.

At the fall of MFE6 time, the JK latches contain 1011. This combintion ANDs with 'not high speed' (used in carrier-return operations) to reset the shift latch.

Motor Downshift

At the fall of MFE6, continued acceleration is prevented by blanking out the seventh MFE pulse, thereby allowing the JK latches to remain constant during MFE7 time. This moves the phasing back to normal.



When the shift latch resets, a set pulse is generated for the motor down-shift latch. The output of this latch blocks MFE7 from stepping the JK latches, and retards the phasing.

The fall of every MFE pulse generates a reset to the motor down-shift latch, so that the latch resets with the fall of MFE7.

Stopping

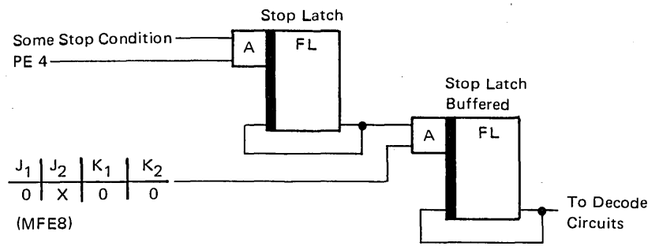
The stepper motor comes to a halt in the same character cycle in which a stop condition is detected. Stop conditions (see also next page) are:

1. New line: Microprogram originated
2. End of line: (EOL) Right-hand margin stop switch
3. Count zero: Microprogram originated
4. Data late: Attachment originated
5. Left-hand margin stop switch
6. Read latch: TA Register bit 0

Stopping is accomplished by changing the phasing to the stepper motor in such a manner as to cause reverse torque to be applied, until motion has nearly ceased; then firing a variable singleshot to provide the 12th and final MFE pulse.

The singleshot is necessary, because motion may be so slight by the end of MFE11, that MFE12 may not be detected, leaving the JK latches out of step.

Stop Latch and Stop Latch Buffered



The stop sequence takes place from the fall of MFE8 through the fall of MFE12.

Although the JK combination of 0X00 recurs at the fall of MFE4, MFE8, and MFE12, the PE4 pulse (which sets the stop latch) comes near the fall of MFE6, assuring the set of the Stop Latch Buffered simultaneously with the fall of MFE8.

The fall of MFE11 advances the JK latches to 1011 . This combination, with K2 on, Motor-Drive Phase 2, and Stop Latch Buffered, sets the inhibit logic feedback latch (which blocks MFE12) and fires the variable-length singleshot (SS2). The output of SS2 is substituted for MFE12 (which may be weak because the emitter-wheel is almost stopped) and steps the JK latches to 0000 .

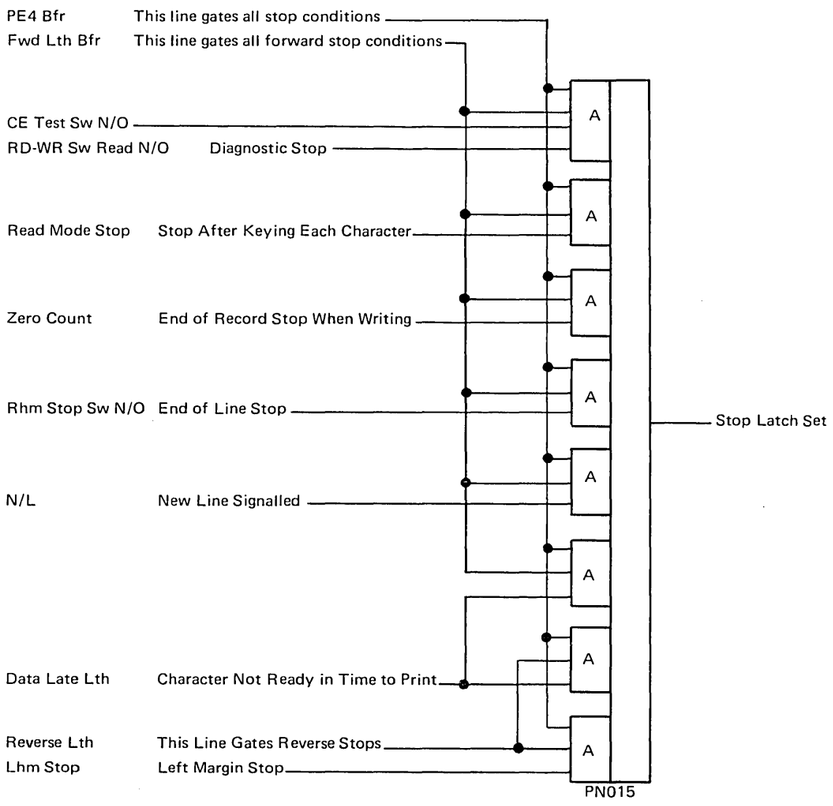
With the completion of the SS2 pulse, another (23 ms fixed-duration) singleshot is fired, preventing another forward or reverse operation from starting during this period.

SS2 Singleshot Adjustment

As soon as possible, set the variable singleshot (SS2) at B-B1-E2-S13 for a plus signal 3.0 milliseconds in duration by either printing in read mode from the CE switches, or by setting the S-register bit 2 on, and the control-address setting to the start of BMA5, subtest 11.

Note: It is recommended that system power be dropped before a card is removed that can cause the printer wire-fire drive line to float (B-B1-D2 or B2).

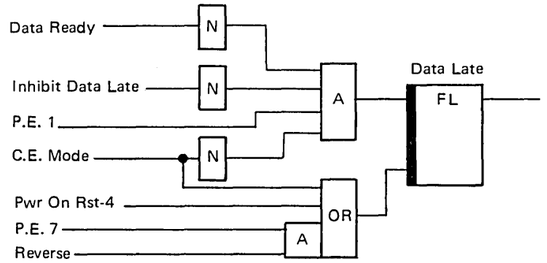
Stop Latch Set Conditions



Data Late

DATA LATE: Blocks the fine PE timing from firing any wires.
Sets the stop latch at PE4 time.
Sets the reverse latch when fully stopped.
Sets the stop latch at PE4 time.
Resets itself at PE7 of the reverse cycle.
Stops and awaits data ready.

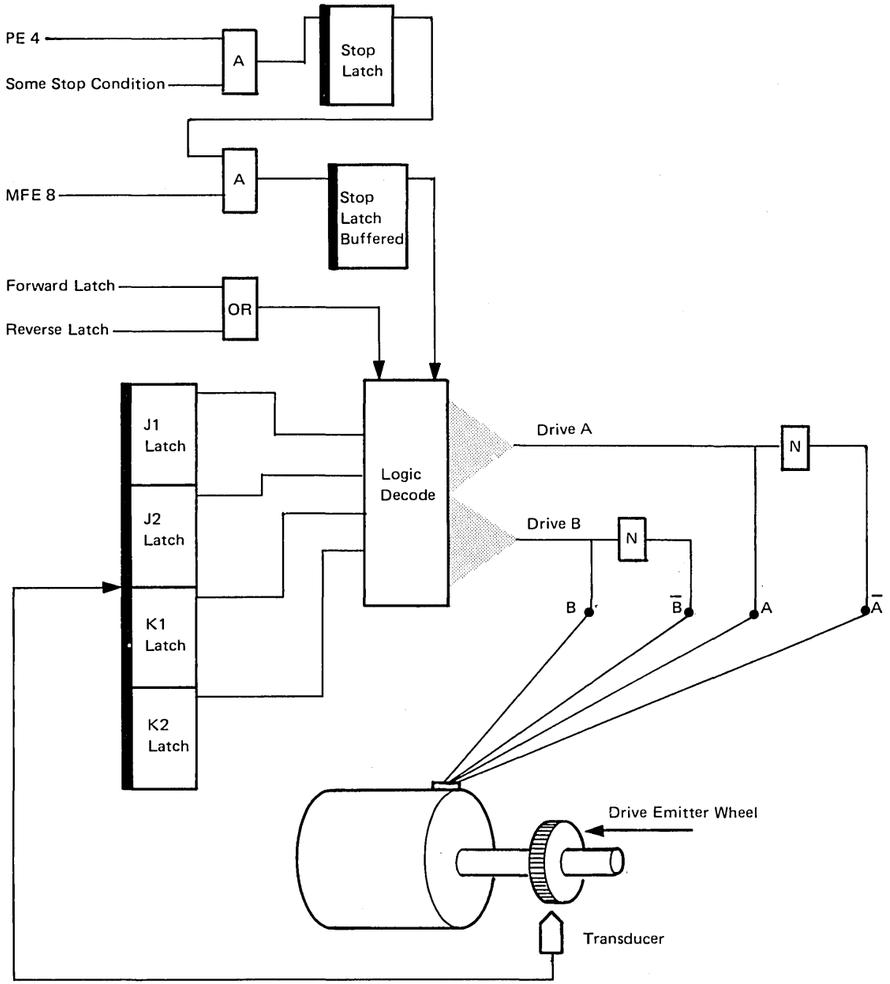
No error condition is flagged as a result of data late.



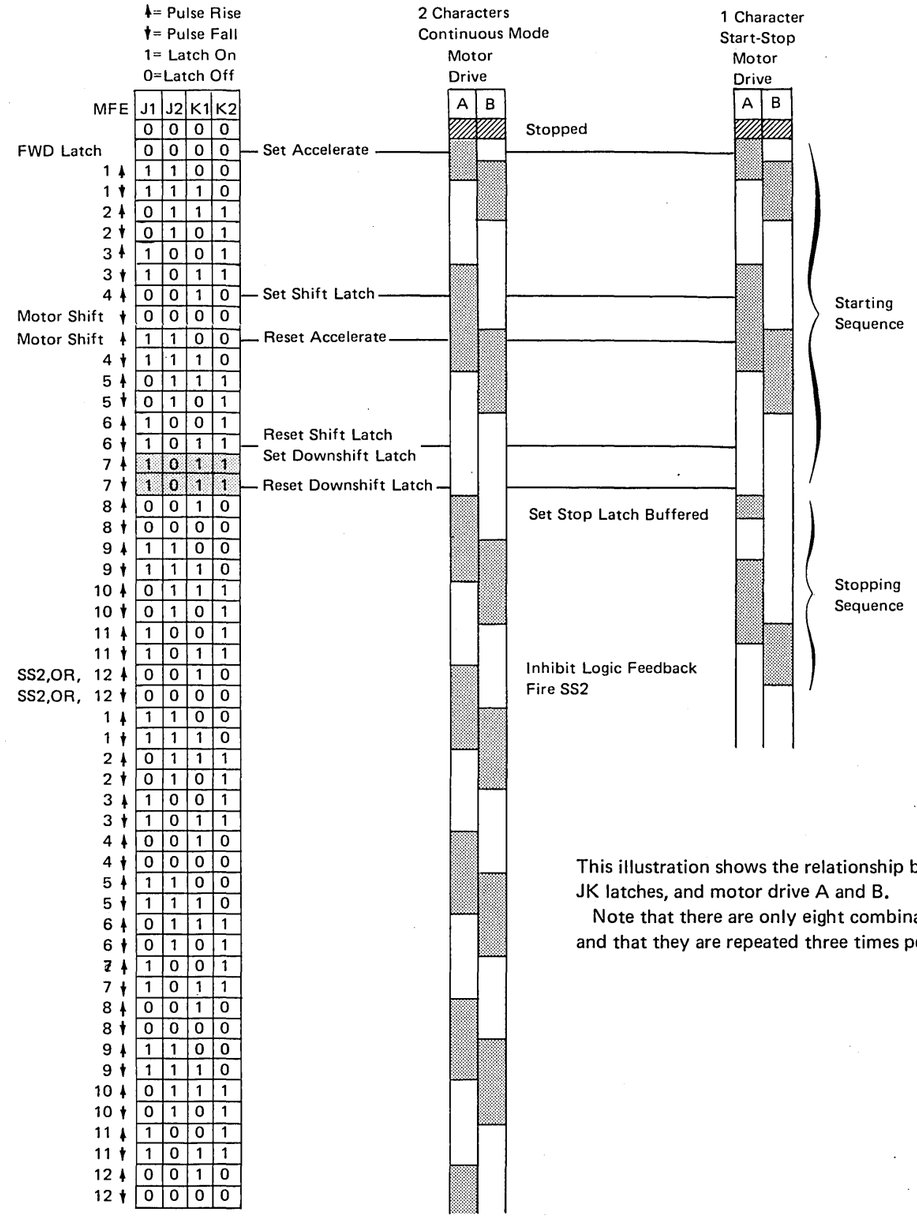
DATA LATE LATCH: This latch detects an overrun condition and stops the printing; then backspaces one position.

During printing in continuous mode, if the microprogram has not set Data Ready by PE1 time, Data Late is set on.

J and K Latch Operation



1. In the static state, the JK latches are off, and phases $\bar{A}$ and $\bar{B}$ are active.
2. Motion is started with the turn-on of the forward or reverse latch, advancing the phasing to the motor to A and $\bar{B}$.
3. The rotation causes the transducer to pick up MFE1, advancing the JK latches, and in turn, the motor phasing.
4. Each MFE pulse from the drive emitter wheel advances the JK latches and drive phasing until the stop latch comes on as a result of:
   a. Count zero
   b. Right-hand margin switch
   c. Data late
   d. New-Line latch (TE4)
   e. Read latch (TA0)
5. Any of the foregoing conditions being on at PE4 time sets the stop latch.
6. At MFE 8 time, Stop Latch Buffered comes on and alters the phasing sent to the motor in such a way as to cause a reverse torque in the motor.
7. By MFE12 time, the stopping sequence has halted forward motion and electrically detents the motor shaft in the home position.

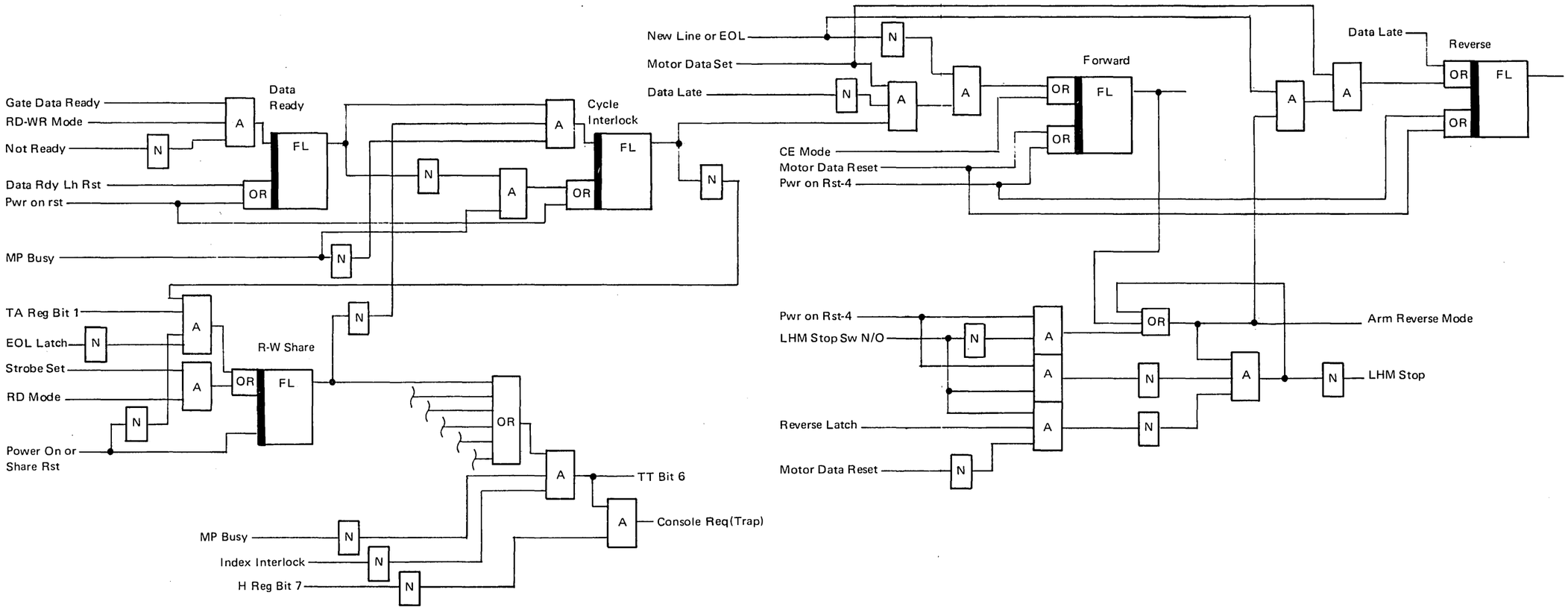


		J1	J2	K1	K2
MFE		0	0	0	0
FWD Latch		0	0	0	0
1	↓	1	1	0	0
1	↑	1	1	1	0
2	↓	0	1	1	1
2	↑	0	1	0	1
3	↓	1	0	0	1
3	↑	1	0	1	1
4	↓	0	0	1	0
Motor Shift	↑	0	0	0	0
Motor Shift	↓	1	1	0	0
4	↑	1	1	1	0
5	↓	0	1	1	1
5	↑	0	1	0	1
6	↓	1	0	0	1
6	↑	1	0	1	1
7	↓	1	0	1	1
7	↑	1	0	1	1
8	↓	0	0	1	0
8	↑	0	0	0	0
9	↓	1	1	0	0
9	↑	1	1	1	0
10	↓	0	1	1	1
10	↑	0	1	0	1
11	↓	1	0	0	1
11	↑	1	0	1	1
SS2,OR, 12	↓	0	0	1	0
SS2,OR, 12	↑	0	0	0	0
1	↓	1	1	0	0
1	↑	1	1	1	0
2	↓	0	1	1	1
2	↑	0	1	0	1
3	↓	1	0	0	1
3	↑	1	0	1	1
4	↓	0	0	1	0
4	↑	0	0	0	0
5	↓	1	1	0	0
5	↑	1	1	1	0
6	↓	0	1	1	1
6	↑	0	1	0	1
7	↓	1	0	0	1
7	↑	1	0	1	1
8	↓	0	0	1	0
8	↑	0	0	0	0
9	↓	1	1	0	0
9	↑	1	1	1	0
10	↓	0	1	1	1
10	↑	0	1	0	1
11	↓	1	0	0	1
11	↑	1	0	1	1
12	↓	0	0	1	0
12	↑	0	0	0	0

This illustration shows the relationship between the MFE pulses, JK latches, and motor drive A and B.

Note that there are only eight combinations of the JK latches, and that they are repeated three times per character.

DATA-TRANSFER CONTROL AND INTERLOCKING



The exchange of data between the CPU and the attachment is controlled by three latches.

R-W Share: Tells the CPU (via Console Request Trap or TT-Reg bit 6) that the attachment is ready to receive (write) or send (read) data.

Data Ready: Tells the attachment that the word registers have been set with data to be printed.

Cycle Interlock: Indicates that an operation has started and blocks another request for data via the RD-WR Share until the current character is printed. The normal starting sequence follows:

Write: Microcode turns on TA-register bit 1, which in turn sets the R-W share latch. Console Request tells the microprogram that the attachment is not busy and the two read words are executed. The second RDMP word sets the data ready latch.

The microprogram now issues a Share Reset (TA bit 3). The R-W share latch resetting, turns on Cycle Interlock, which sets the forward latch, starting the stepper-motor sequence.

Cycle Interlock prevents RD-WR share from requesting another byte of data until MP Busy drops (end of PE Envelope), when Data Ready is reset, allowing Cycle Interlock to reset. Console Request is blocked until MP Busy (PE Envelope) drops.

In continuous mode, the forward latch remains set and does not use Cycle Interlock for the remainder of the characters printed.

Forward Latch: In continuous mode, the first time Cycle Interlock sets on, the forward latch also sets on. It remains on until all characters are printed and the stop latch is set, and the variable singleshot times out.

In read mode, the stop latch is set each PE4 time, so that Forward comes on and off for each cycle.

Reverse Latch: This latch is set from either New Line or Data Late. It causes the stepper motor to go in the reverse direction until the left-hand margin stop is detected.

Reverse Mode: A new-line operation (TE bit 4) may or may not require movement of the print head to the left. Because of difficulty maintaining a consistent left-hand margin using only the left-hand margin stop switch, the switch state is latched as follows:

Power Up: Stop switch made on power up, resets reverse mode.

One char after forward: The forward latch coming on sets reverse mode. This is because at the end of one character cycle, the LHM stop switch is still made. The reverse mode latch ensures one character travel in the reverse direction, even though the LHM stop switch is still made, when the New Line is called for.

Return from more than one character out: As travel to the left closes the LHM stop switch, the reverse mode latch is reset. Motion stops in the same cycle.

Motor Data Set

Gates the set to forward and reverse latches. This line is off from the time Chopped Forward or Reverse turns on the stepping latch until after the time-out of the variable and then the 23 MS singleshots.

Motor Data Reset

Applies a reset to the forward and reverse latches. It is the inverted output of the stepping latch, and is off from the time Chopped Forward or Reverse sets 'stepping' until the fall of the variable singleshot resets 'stepping'.

FORWARD OPERATION—PRINT A SINGLE CHARACTER AND STOP

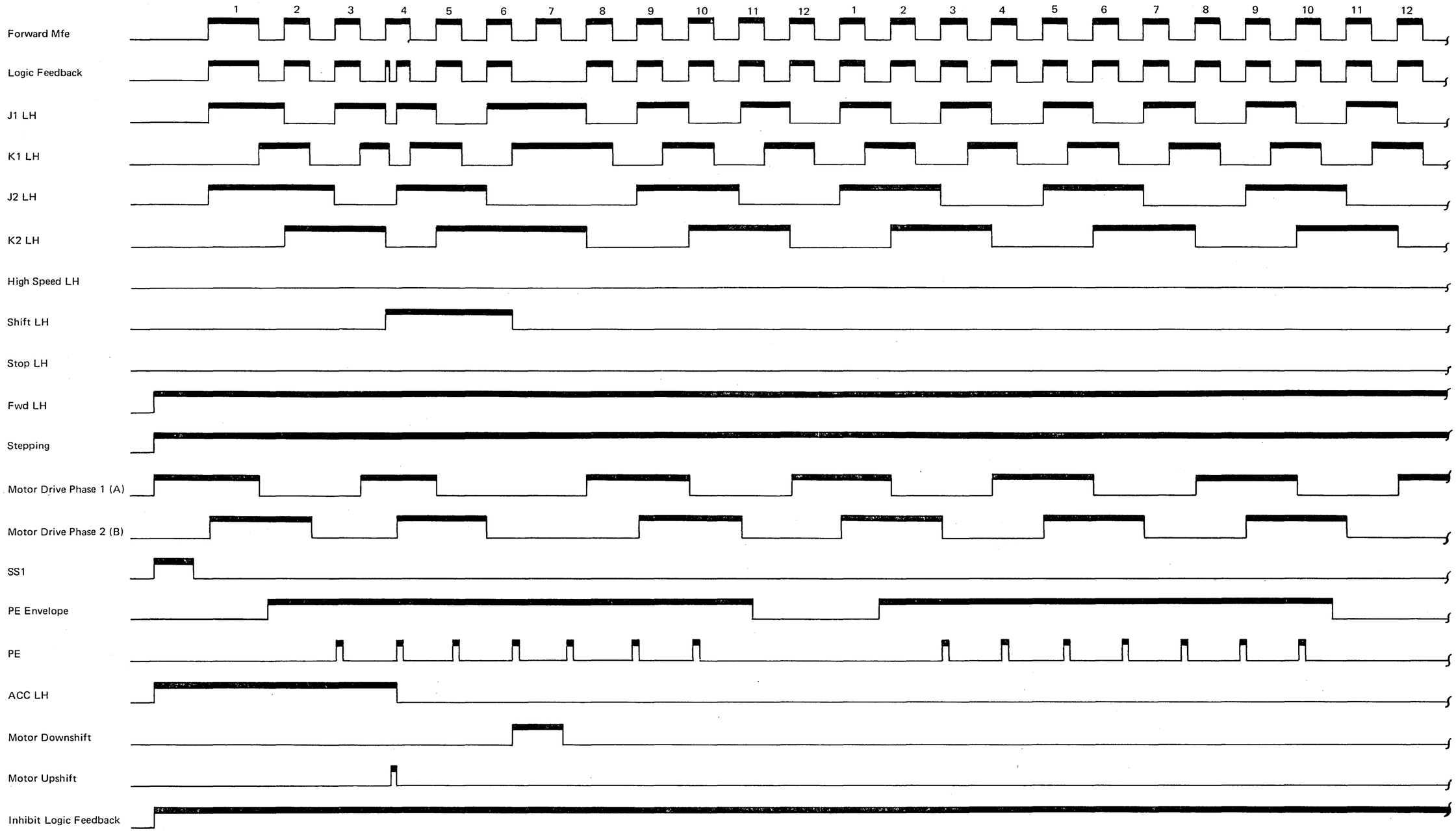
1. Cycle Interlock latch, in response to a Read/Write Share Request, turns on Forward (Go) latch. With motor control latches J1, J2, K1, and K2 off, forward latch brings up Motor Drive Phase 1 (A), SS1, Stepping and Accelerate latches, and drops Inhibit Logic Feedback.
2. With Forward latch on, the motor turns and the forward drive emitter generates the first pulse—Fwd MFE1. The leading edge of the Fwd MFE pulse brings up Logic Feedback and turns on the stepper-motor control latches J1 and J2, and brings up Motor Drive Phase 2(B).
3. The trailing edge of Fwd MFE 1 pulse drops Logic Feedback and turns on motor control latch K1.
4. With motor control latches J1 and J2 both on and K2 off, the K1 latch going on turns off Motor Drive Phase 1 (A).
5. PE Envelope is brought up by the leading edge of the first PE pulse to gate seven print emitter pulses to the counter.
   Note: Drawing is not to scale; refer to "Emitter Pulse conversion," page 7-40, for explanation.
6. Advanced by the K1 latch having come on, the stepper motor turns another increment, generating Fwd MFE 2 pulse, and bringing up Logic Feedback for the second time.
   The leading edge of Fwd MFE 2 pulse turns off motor-control latch J1 and turns on K2.
7. The trailing edge of Fwd MFE 2 pulse drops Logic Feedback and turns off stepper motor control latch K1, dropping Motor Drive Phase 2 (B), advancing the stepper motor another increment.
   Note: Fwd MFE pulses are drawn different lengths to show how they shorten as the machine speed up, and lengthen as it slows down.)
8. The leading edge of Fwd MFE 3 pulse brings up Logic Feedback, turns on motor control latch J1 and Turns off J2.
9. At Fwd MFE 4-pulse time, the leading edge of the pulse brings up Logic Feedback in the usual manner and turns off stepper motor control latch J1.
   With J1 and J2 both off, and the Accelerate latch on, K1 being on turns off K2 and turns on the Shift latch.
   Note: This is not a shift for upper or lower-case printing as for the 3210; it is for a shift in stepper-motor speed.
10. Shift latch output is ANDed with a delay circuit to produce the Upshift pulse.
11. The Upshift pulse interrupts the Logic Feedback line, effectively breaking it into two pulses and turning off K1. The stepper motor responds to this added pulse in an effort to catch up.
12. When the Upshift pulse times out, Logic Feedback again comes up, and with J1, K1, and K2 all off, turns on J2 and resets the Accelerate latch.
13. With K1 and J2 on, and J1 and the High Speed lines down, the Shift latch turns off. (Refer to page 7-48).
14. Shift latch going off brings up Downshift.
15. During this operation, the print emitter has been turning and generating PE pulses.
    At PE-4 time, this pulse gates the Stop latch (to begin the stopping sequence) either because the character being printed on a write operation is the last character to be printed (zero count), or because the machine is operating in start-stop mode in a keyboard (read) operation. In either Read or Write mode, the Stop latch is set at PE-4 time because of an End-of-Line condition (right margin switch has been operated.)
16. Fwd MFE 7 pulse drops the Downshift line.
17. At the end of Fwd MFE-8 time, when Logic Feedback, J1, J2, K1 and K2 lines are down (starting conditions for step 1), and both Motor Drive Phase 1 and 2 lines are down, Stop latch Bfd comes up to start the four-pulse (MFE) stopping sequence.

FOUR-PULSE STOPPING SEQUENCE

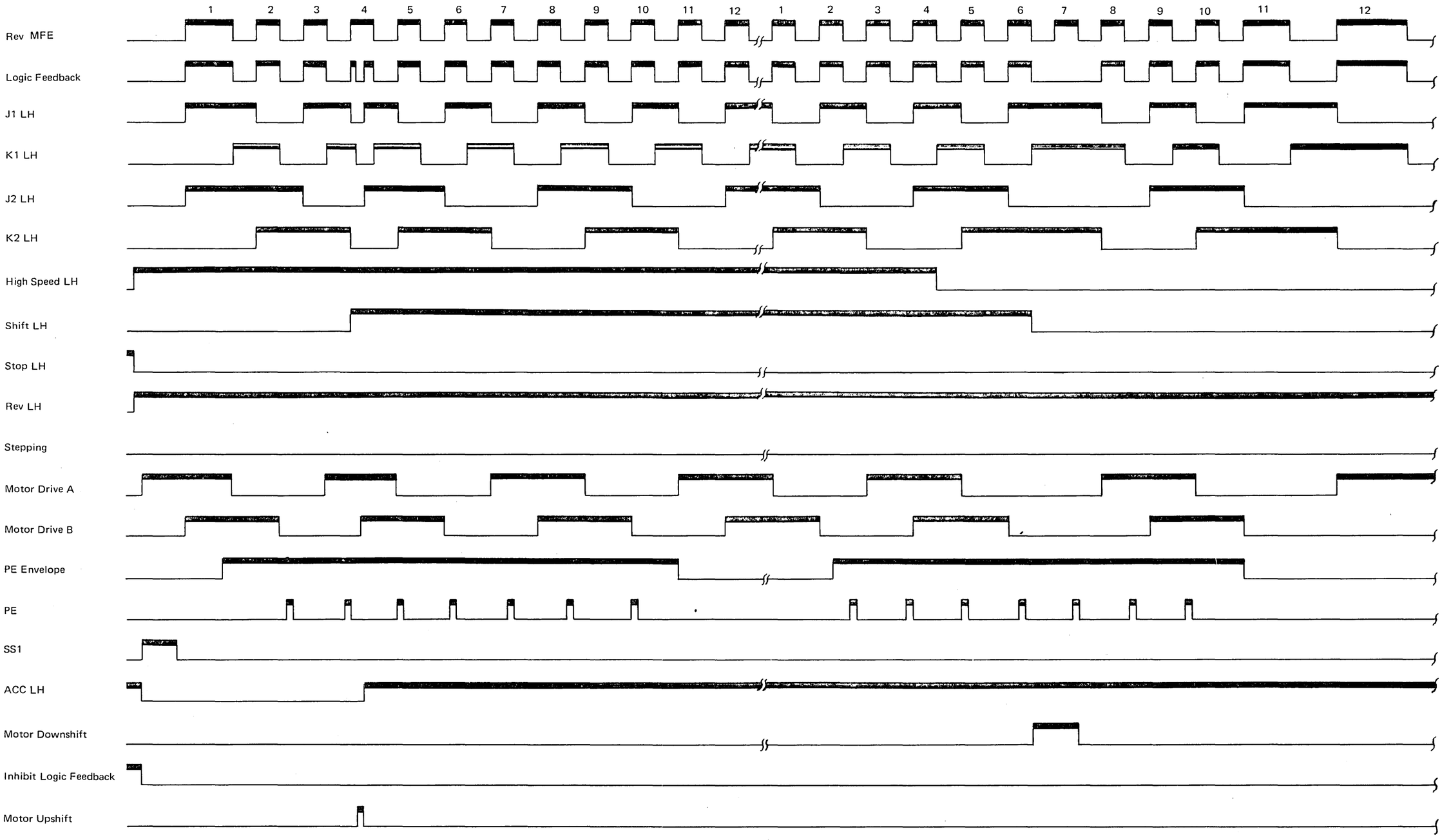
18. Because the Fwd and Stepping lines are still up, the operation continues as in steps 2, 3, and 4 for Fwd MFE pulses 9, 10, and 11.
    At Fwd MFE 11-pulse time, because Motor Control latch K2 in on again, with Motor Drive Phase 2 (B) and Stop latch up, SS2 is fired to bring up inhibit logic feedback and blocks any further feedback of MFE pulses into the logic.
19. When SS2 times out, the Forward (Go), Stepping, Stop latch, and Stop Latch Bfd lines drop, and the stepper motor stops, leaving the logic restored to the starting conditions.

FORWARD OPERATION--SINGLE CHARACTER

FORWARD OPERATION--MULTIPLE CHARACTERS



REVERSE OPERATION, MULTIPLE CHARACTERS



ALTER/DISPLAY OPERATIONS

- Alter/display operations are microprogram-controlled manual operations.

Alter/display operations permit the operator to access and alter data stored in the system, and provide a printed record of the data involved. This capability is not provided for the remote console printer-keyboard (3210 Model 2).

Note: In alter/display operations, alphabetic characters may be entered in either uppercase or lowercase, but will always print out in UPPERCASE.

Mnemonics

Mnemonic	*Function*
AM	Alter Main Storage
AS	Alter Control Storage
AL	Alter Local Storage
AK	Alter Storage Protection Key
AC	Alter Control Registers
AG	Alter General Registers
AF	Alter Floating-Point Registers
AP	Alter Current PSW
DM	Display Main Storage
DS	Display Control Storage
DL	Display Local Storage
DK	Display Storage Protection Key
DC	Display Control Registers
DG	Display General Registers
DF	Display Floating-Point Registers
DP	Display Current PSW

Addressing

Storage Area	*Address*
Main Storage (M)	000000 through 03FFFF
Control Storage (S)	8000 through FFFF
Local Storage (L)	00 through 3F
Storage Protection Key (K)	000000 through 03FFFF
Control Register (C)	0 through F
General Register (G)	0 through F
Floating-Point Register (F)	0, 2, 4, 6
Current PSW (P)	Not Required

Notes:

1. The upper boundary of main-storage address is movable.
2. The lower boundary of control-storage address is movable.
3. For alter/display of the control register, general register, and floating-point register, wraparound of the address is done by the alter/display microprogram.
4. Word or byte address may be used in addressing main storage or control storage. If the starting address is not on a word boundary, the printer spaces and aligns at the byte addressed.

Format

MNEMONIC AND ADDRESS

The two-character mnemonic and the address print on the same line, with one space automatically provided between the mnemonic and the address.

Following the typing of the prescribed addressing character, the alter/display microprogram issues an automatic carrier return (new line) command to the printer.

An optional feature permits addressing without preceding zeros. However, a manual carrier return must be performed by pressing the return key to indicate the end of the address

DATA

Both uppercase and lowercase characters may be used in typing mnemonic or data. The alphabetic characters printed are always in uppercase.

Data printed has the format of eight words per line with two spaces provided automatically between words. An automatic carrier return occurs when eight words have been printed.

In altering or displaying storage-protection keys, each word contains four consecutive storage keys. The address is updated sequentially by 2K.

ALTER OPERATIONS

Data may be altered one hex digit at a time. The use of the Space bar during an alter operation permits nondestructive spacing. The stored data is printed each time the Space bar is pressed.

Termination of an alter operation occurs on a digit basis.

Examples

ALTER MAIN STORAGE

```
AM 480 (Press the Return key)
XXXXXXXX XXXX (Press the A/D key)
```

ALTER CURRENT PSW

```
AP
XXXXXXXX XXXXXXXX
```

DISPLAY OPERATIONS

Following the addressing and automatic carrier return, the addressed data prints continuously until either the A/D key or the End key is pressed.

Termination of a display operation occurs on a word basis.

Examples

DISPLAY MAIN STORAGE

```
DM 00008D
   XXXXXX XXXXXXXX XXXXXXXX ..... XXXXXXXX
XXXXXXXX XXXXXXXX (Press the A/D key)
DM 8D (Press the Return key).
   XXXXXX XXXXXXXX XXXXXXXX ..... XXXXXXXX
XXXXXXXX XXXXXXXX (Press the A/D key)
```

DISPLAY CONTROL STORAGE

```
DS F700
XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX
(Press the A/D key)
```

DISPLAY LOCAL STORAGE

```
DL 01
XXXXXXXX XXXXXXXX (Press the A/D key)
```

DISPLAY FLOATING-POINT REGISTER

```
DF 2
XXXXXXXX XXXXXXXX XXXXXXXX ..... XXXXXXXX
```

DISPLAY STORAGE KEY

```
DK 009000
XXXXXXXX XXXXXXXX XXXXXXXX (Press the A/D key)
```

Note: Each word consists of four consecutive storage keys.

ENDING AN ALTER/DISPLAY OPERATION

1. Pressing the alter/display key terminates the A/D operation but not the A/D mode. A new mnemonic can now be entered for another A/D operation.
2. Pressing the End key terminates both the A/D operation and the A/D mode.
3. Detection of an invalid character or address during an A/D operation terminates the A/D operation but not the A/D mode.
4. Termination occurs upon completion of printing of sixteen words from general registers or control registers, eight words from floating-point registers, and two words of current PSW.
5. A machine check detected terminates the A/D operation.

ERROR MESSAGES

Invalid Character

The error message INVAL CHAR prints if one of the following input errors is made:

1. The first character of a mnemonic is not A, D, or T (see KEYBOARD TEST MODE).
2. The second character is not M, S, L, K, C, G, F, or P. S and L are reserved for service personnel (See item 3 under "Invalid Address".
3. A character other than a valid hexadecimal is keyed when data is addressed or altered.
4. The cancel key is operated.
5. A character with parity error is detected on the PR-KB bus-in line. The character with parity error is neither printed nor stored into the storage. The current operation is terminated following the error message 'INVAL CHAR' printout. The machine remains in A/D mode ready to accept a new operation.

Invalid Address

The error message INVAL ADDR prints if one of the following errors is made:

1. The starting address is invalid.
2. The updated address exceeds the capacity of the specified storage.
3. The operator performs an AS or AL operation when the CE Mode bit is off.

KEYBOARD TEST (T) MODE

Keyboard Test Mode uses the alter/display circuitry to permit the PR-KB to be operated like an electric typewriter for testing the operation of the keyboard. No changes are made to any system facility.

1. Press the CPU stop key or set the rate switch to INSTRUCTION STEP or SINGLE CYCLE HARD STOP.
2. Press ALTER/DISPLAY on the PR-KB console.
3. Type in the mnemonic T to initiate keyboard test mode (no address is necessary). If the 3210 Model 2 is installed, it too, will be ijn this mode, and may be tested simultaneously with the console PR-KB.
4. Check the operation of the character, space, and return keys, observing the results they produce, or the printed output.
5. Press the END key to terminate the operation of the keyboard test mode.
6. Return the rate switch to PROCESS and press the CPU start key to resume processing.

Note: If a parity error is detected on the PR-KB bus-in lines, while in Test mode, the erroneous character is neither printed nor entered into storage, and no error message is printed.

1. Press the CPU stop key or set the rate switch to INSTRUCTION STEP or SINGLE CYCLE HARD STOP. This puts the CPU in the stopped state.

2. Press the alter/display key on the documentary console (not available on the remote printer-keyboard). This requests an A/D trap. When the trap is honored, the alter/display mode light turns on, followed by the proceed light.

3. Type the two-character mnemonic. If the first character is not A, D, or T (see "Keyboard Test Mode"), or the second character is not M, S, L, K, C, G, F, or P, the machine prints out INVAL CHAR. S and L are reserved for service personnel.

4. Type the address. Preceding zeros may be omitted if you press the return key at the end of the address. If you type an incorrect address, the machine prints out: INVAL ADDR.

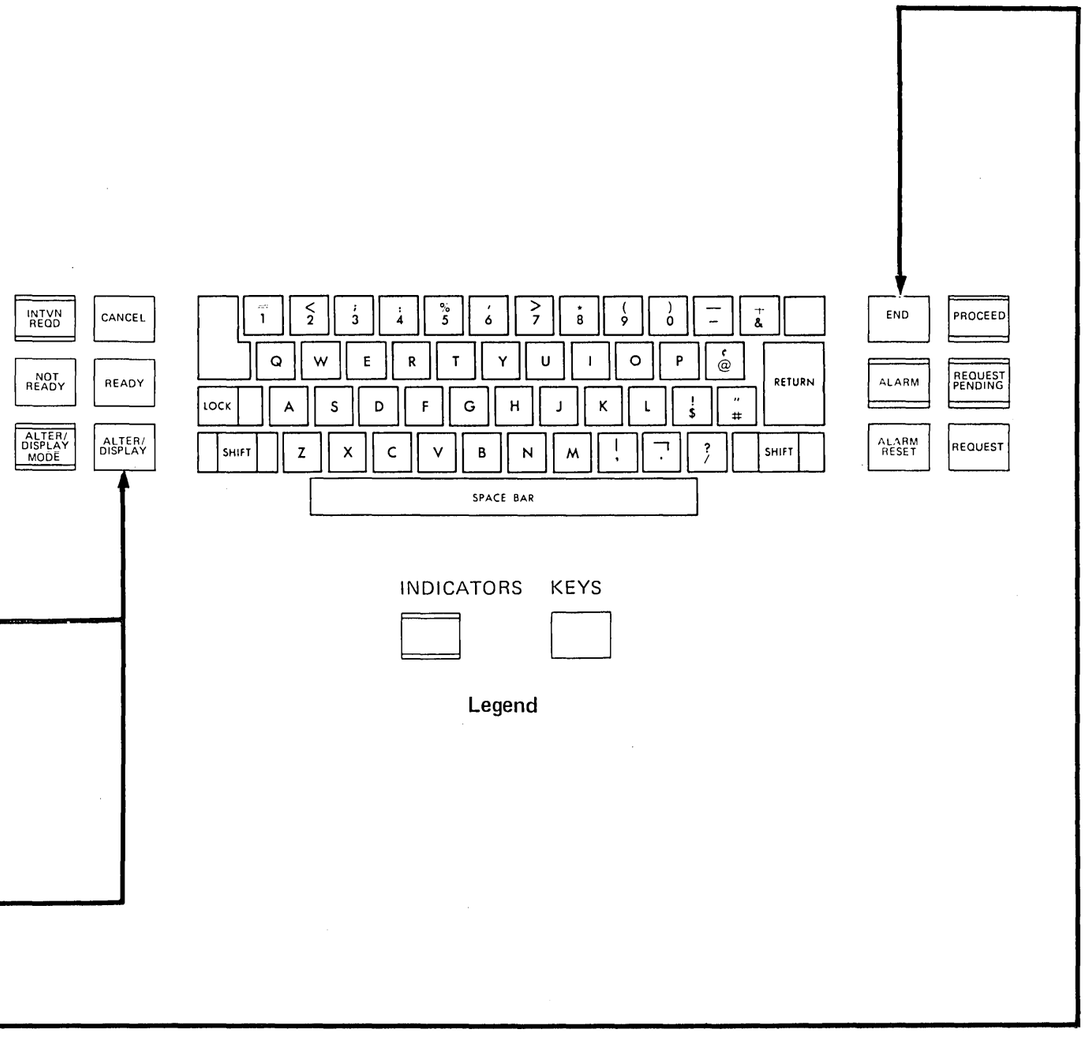
5. For an alter operation:
   Type the data in HEX characters.
   Stored characters print as-is whenever the space bar is pressed (to indicate nondestructive spacing).

   For a display operation:
   The addressed words print out.

6. Press the alter/display key to end the alter or display operation in progress while remaining in A/D mode. To start a new alter/display operation, begin at Step 3 after the proceed light turns on.

7. Press the END key to terminate the A/D mode. This turns off the proceed and alter/display mode lights.

8. Return the rate switch to PROCESS and press the CPU start key to resume processing.



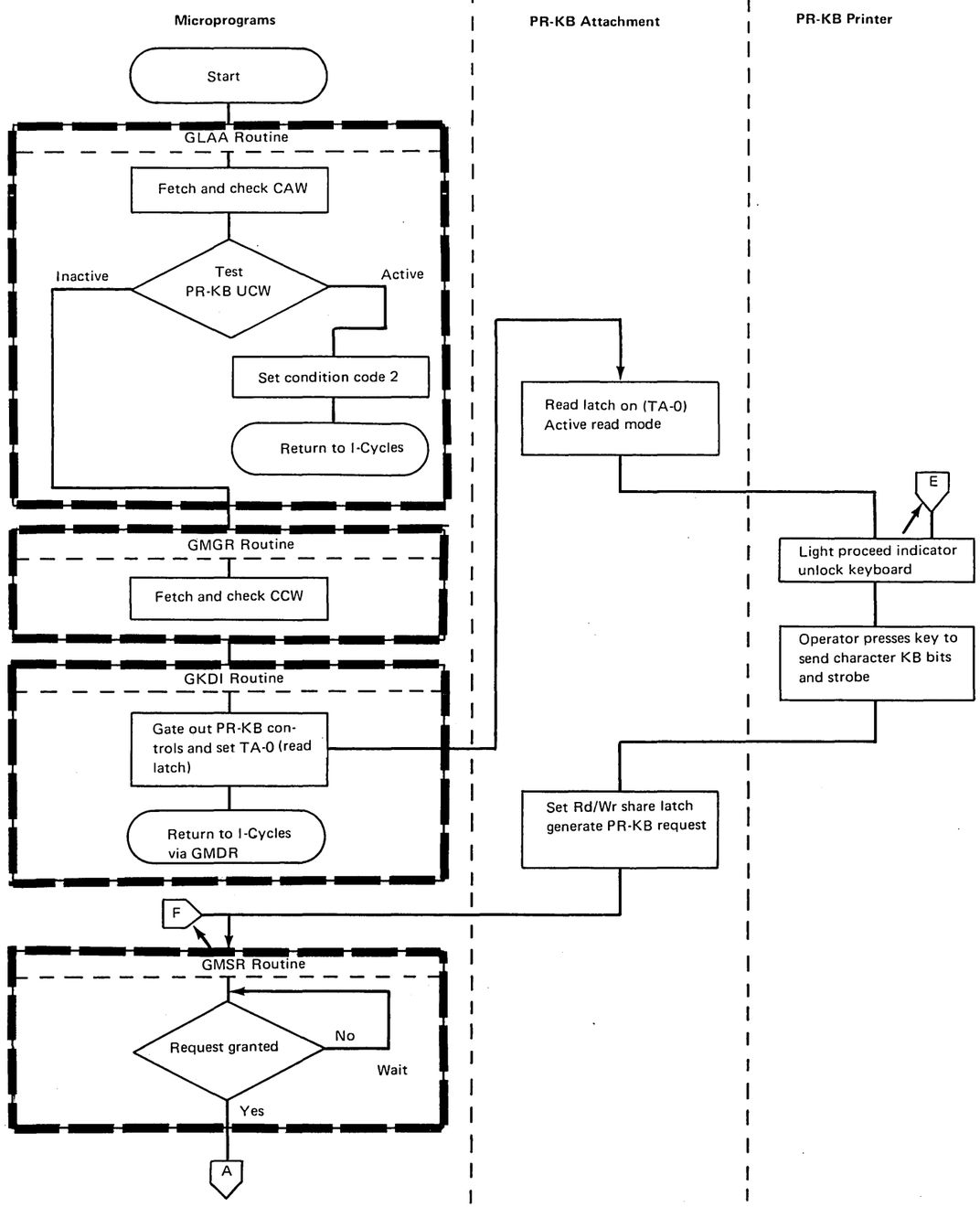
Legend

REMEMBER

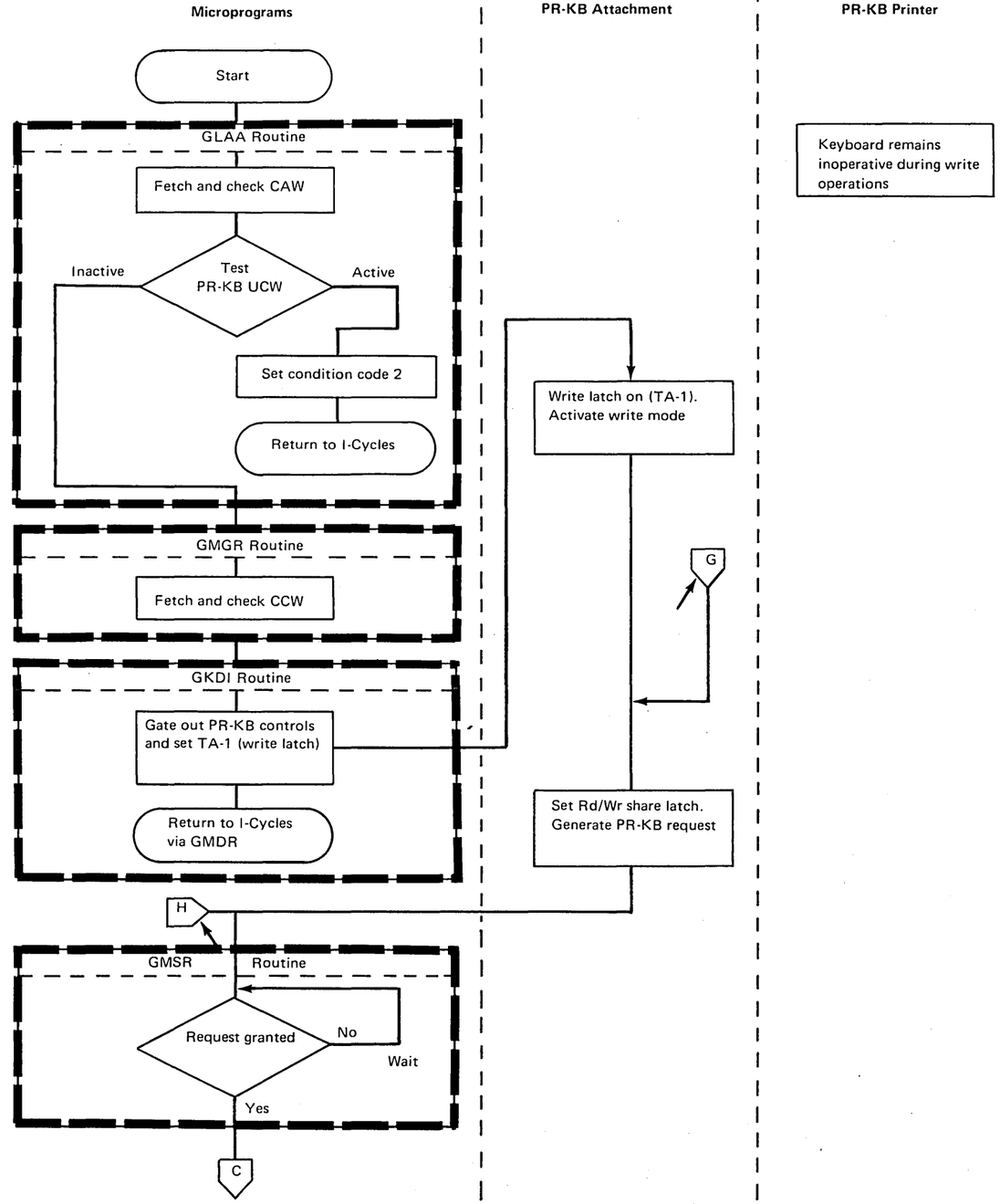
There is a Reader's Comment Form at the back of this publication.

MICROPROGRAM OPERATION

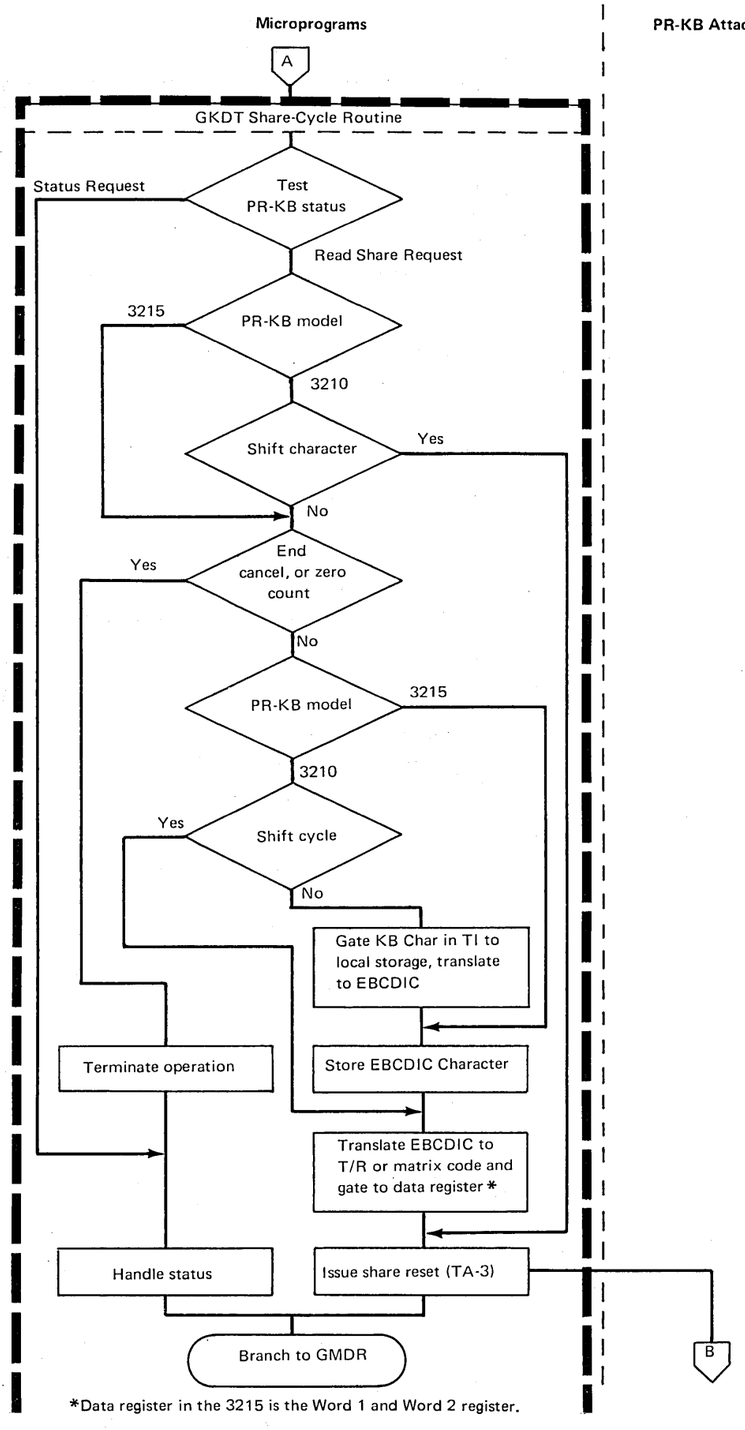
Read (Keyboard) Operation — Part 1



Write Operation — Part 1

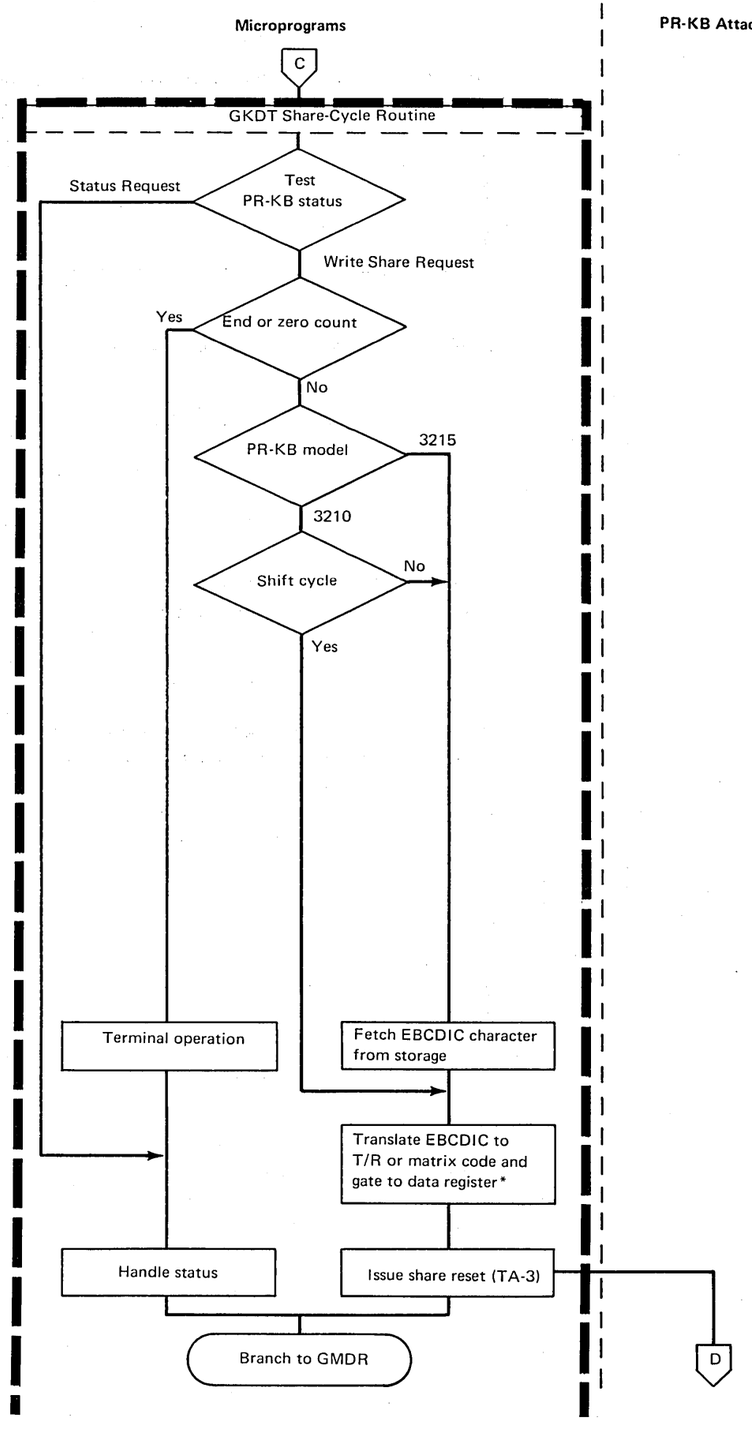


Read (Keyboard) Operation --Part 2

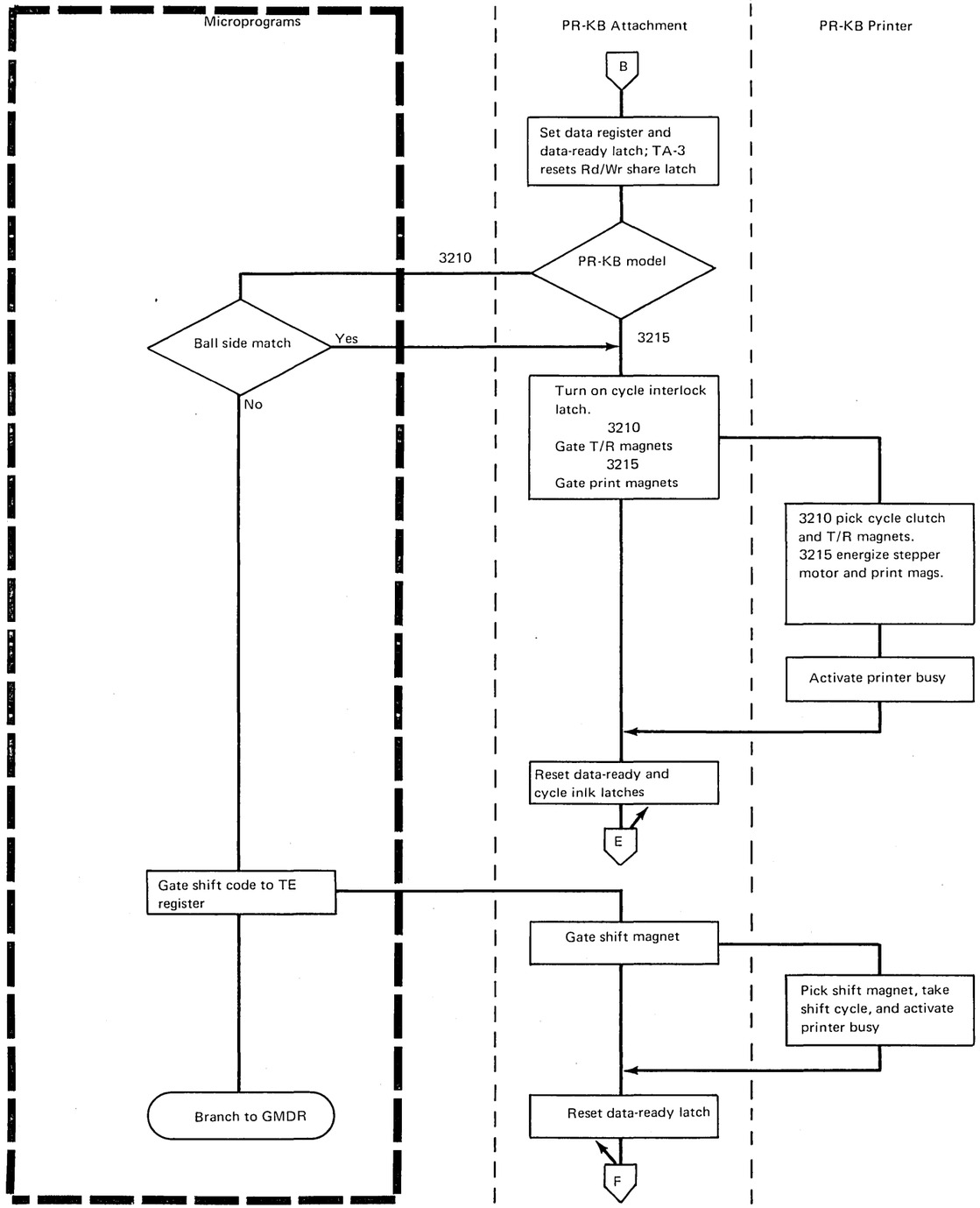


*Data register in the 3215 is the Word 1 and Word 2 register.

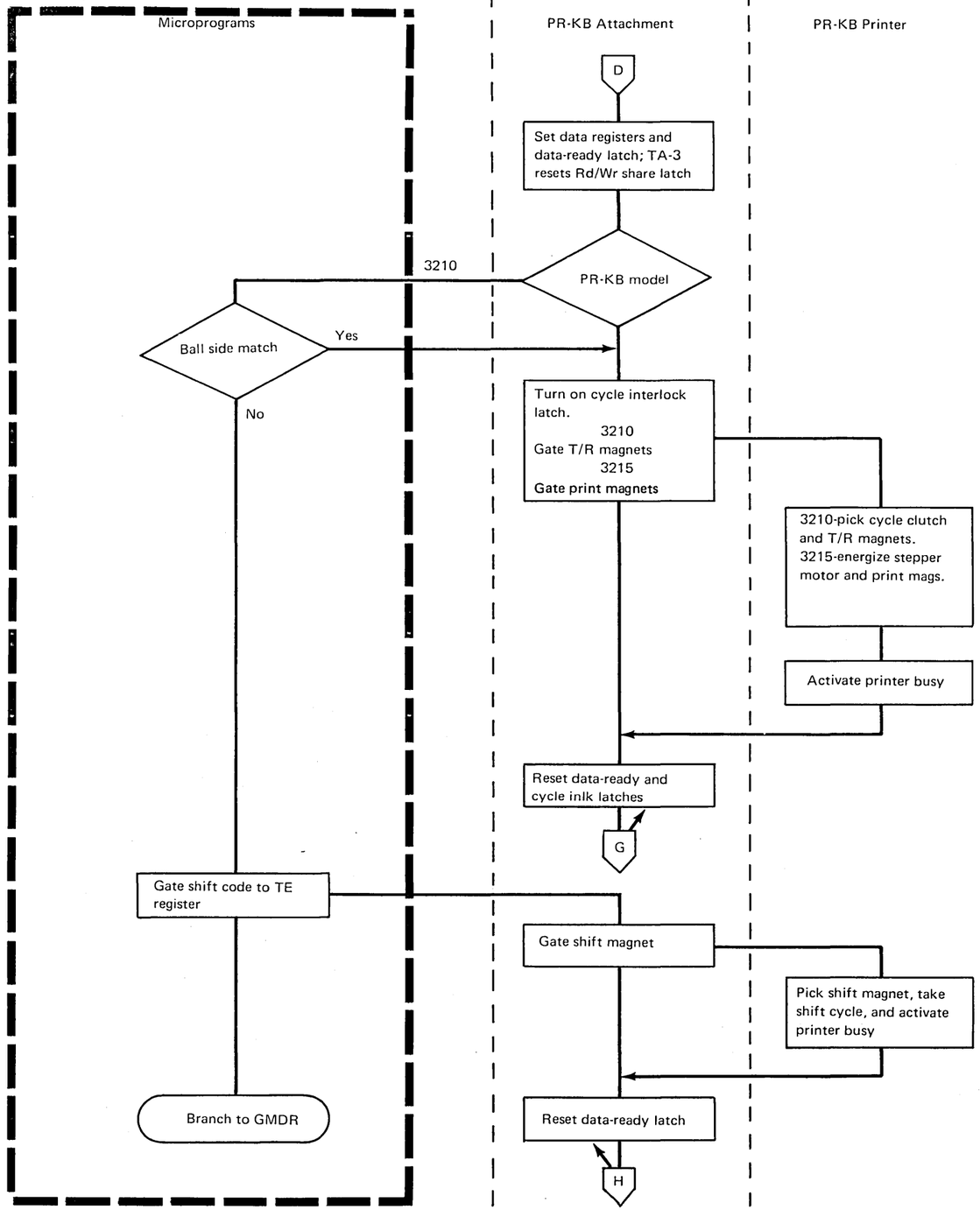
Write Operation -- Part 2



Read (Keyboard) Operation -- Part 3



Write Operation - - Part 3



PROGRAMMING INFORMATION

Except for transfer in channel, printer-keyboard operations written for the 3210 and 3215 are compatible with programs written for the IBM 1052 Model 7 Printer-Keyboard. (Transfer in channel is not defined for the 1052 Model 7.)

Except as specifically noted, system operation with either PR-KB configuration is as follows.

To the programmer, the printer-keyboard appears to be attached to the multiplexer channel with a device address of 01F or 009 for the first console PR-KB, and 01E or 008 for the second. A multiplexer channel UCW is used for printer-keyboard program-controlled operations. The printer-keyboard does not use any of the eight control-unit positions on the multiplexer channel.

Besides program-controlled operations, microprogram-controlled alter/display functions are provided by the 3210 Model 1 or the 3215 PR-KB (not by the 3210 Model 2 remote unit). The alter/display function may be requested by pressing the alter/display key (3210 Model 1 or 3215 only). The primary function of this operation is to provide a means of altering or displaying main storage, general purpose registers, floating-point registers, and certain other facilities. These facilities are described in the section on "Alter/Display Operations."

Status or sense information is not applicable to alter/display functions. However, the alter/display function is not executed until any current program-controlled operation is completed to the point at which status for the operation is presented to the CPU.

If an alter/display operation is started, any CPU instruction execution is delayed until after the alter/display operation is completed. Therefore, initial-selection status for an I/O instruction cannot be obtained while the alter/display operation is in progress, because the I/O instruction cannot be executed.

CHANNEL COMMANDS FOR PRINTER-KEYBOARD

Valid channel commands for the console printer-keyboards are:

Command Code Bits

0 1 2 3 4 5 6 7	*Command Name*
0 0 0 0 0 0 0 1	Write
0 0 0 0 0 0 1 1	No Op
0 0 0 0 0 1 0 0	Sense
x x x x 1 0 0 0	TIC
0 0 0 0 1 0 0 1	Write with ACR (Automatic Carrier Return)
0 0 0 0 1 0 1 0	Read
0 0 0 0 1 0 1 1	Alarm (Audible)

Any command code (except 00, which causes a program check when detected by the multiplexer channel) issued to the documentary console with a bit structure that does not conform to those listed here causes a unit check (bit 6) to be set in the status byte, and a command reject (bit 0) to be set in the sense byte.

Status and sense information is stored for the program-controlled operations when applicable.

Write (without Automatic Carrier Return)

The write command is accepted by the PR-KB attachment only if:

1. The PR-KB is operational (ready).
2. The write command has a valid format (that is, data count is not 0, data address valid, etc), and
3. The PR-KB is not performing some other operation.

If the PR-KB is not ready, condition-code 1 is set, intervention required (bit 1) is set in the PR-KB sense byte, and unit-check status in the Channel Status Word (CSW) is:

1. Stored for the start I/O initiating the write command, or
2. Stored on a subsequent I/O interruption (or a test I/O if chaining to the write command was performed.

If the write command is accepted, the write latch (TA-register bit 1) is turned on, and a console share request occurs because:

1. The write latch is on.
2. The printer can accept a character.

During the resulting share-cycle trap microprogram, the first character to be printed is read out of main storage and used to access the appropriate print translate table—tilt/rotate code for the 3210 printer-keyboards; matrix code for the 3215 printer-keyboards. These tables are located in control storage. The applicable code is sent to the printer. If it represents a printable character, the character prints. If it represents a function character (new line), the function is performed.

The CCW data address (now in the PR-KB UCW) is incremented by one, and the data count is correspondingly decremented,for both function and data characters.

A share-reset condition is developed to reset the share-request control. Another share-cycle request cannot occur until the attachment determines that the printer has completed the character or function. As soon as the print or function (space or new line) operation has been accepted by the printer, another share-cycle request is initiated by the attachment. The entire cycle is then repeated.

At the end of each write operation (count zero, or end), the printer operational line is tested. If the printer is not operational, any chaining called for is not allowed. Unit check, channel end, and device end are set on in the PR-KB status byte (in control storage) and intervention required is set on the PR-KB sense byte (in control storage). Unit check, channel end, and device end are set into the multiplexer channel IB (Interrupt Buffer) if no other device already has status pending in the IB. A subsequent I/O interruption operation (or test I/O instruction that addresses the PR-KB stores the status in the CSW. If, however, another device already has status in the IB, the PR-KB attachment circuits are conditioned to cause a share request when IB becomes available.

A zero-count condition is checked for during each write share-cycle after data is transferred and address and count updates are performed. If a zero-count condition is detected and data chaining is indicated, a branch is made to the multiplexer chain-data microprogram to load the new CCW.

If data chaining is not indicated:

1. A zero count condition is set in the PR-KB UCW.
2. Channel end and device end are set on in the PR-KB unit status in the next share cycle, and
3. The write latch is reset.

If command chaining is indicated, the unit-status byte is set to 00 and a branch is made to the multiplexer chain-command microprogram to load the new CCW.

An Incorrect Length (IL) indication is given if the SLI flag is off for any write command except a write command that has the chain-data flag on. This IL indication occurs because the PR-KB requests one more share cycle after the data count (for a write command) has been decremented to zero.

If the end-of-line switch is activated, a new-line function is automatically initiated. In this case another share cycle is not requested until after the new-line function is completed.

The keyboard is blocked during a write operation.

Operation of the END key terminates any write command operation that is in progress (in the same manner as for a read command). When the write command is terminated, an asterisk is printed, and a new-line operation takes place.

No Op (No Operation)

No Op is an immediate command. This command is processed whether or not the PR-KB is operational. The unit check and intervention required bits are not set on when a No Op is executed.

Channel end and device end are set in a CSW stored for a start I/O that indicates a No Op (if command chaining is not indicated).

Sense

The sense command is processed whether or not the PR-KB is operational. The sense byte is read from control storage and placed in the main-storage location specified by the address in the sense command. If not operational, unit check and intervention required are not set on when a sense command is executed. The data count in the sense command should equal one. If the count is greater than one, an Incorrect Length (IL) indication results if the Suppress Length Indication (SLI) flag is off in the sense command.

Channel-end and device-end status are presented in the CSW stored by a subsequent I/O interruption (or cleared by a test I/O) for the sense operation.

Transfer in Channel (TIC)

This command is included here only for sake of compatability with the 1052 Printer-Keyboard Model 7. It functions in the usual manner as described in the SRL manual *IBM System/360 Principles of Operation*, ***GA22-6821.***

Write with ACR

The write with Automatic Carrier Return (ACR) command functions in the same basic manner as a write command. However, a new-line function (carrier return and line feed) is performed after the data count reaches zero.

The new-line function is performed as follows.

1. In the share cycle that occurs after the last character has been sent to the printer, an N/L (New Line) character is automatically generated and sent to the printer by the micro-program, and channel end is set on in the PR-KB status byte.
2. The new-line function is performed by the printer.
3. On completion of the ACR, another share cycle is requested to set device end in the unit status.

If the end-of-line switch is operated after the last character is printed, two new-line functions occur: one for the end-of-line switch indication, the second for the write-with-ACR command.

Read

The read command is accepted by the attachment only if:

1. The PR-KB is ready, and
2. The PR-KB is not performing some other operation.
3. The read command has a valid format.

If the PR-KB is not ready, unit-check status set in the CSW is:

1. Stored for the start I/O initiating the command, or
2. Stored on a subsequent I/O interruption (or a test I/O) if chaining to the read command was performed (condition-code 1 is set).

If the read command is accepted, the read latch (TA register bit 0) is set on and the proceed indicator (on the PR-KB console) lights.

Operation of a character key then results in a share-cycle request. A 9-bit pattern (including a parity bit) is sent from the keyboard to the CPU via the TI-register, and parity is checked in the A-register. (A set-ABCK status set is used in the control word for which the data parity is checked in the A-register.) If a parity check is detected, the PR-KB unit-check status and equipment-check sense bits are set on, but the operation is not terminated until its normal ending point. A machine check is not indicated, and no instruction retry or

machine-check trap is taken. A character is printed by the PR-KB whether or not a parity check is detected. (The character that is printed is unpredictable for a parity-error condition.)

The 9-bit pattern from the 3210 keyboard is translated to the appropriate EBCDI Code and stored in main storage. The translation is effected by the microprogram through use of translation tables in control storage.

The nine-bit pattern from the 3215 PR-KB is already encoded in EBCDIC and is stored without translation.

The microprogram then uses the EBCDI code to address another section of storage containing the tilt/rotate codes for the 3210 PR-KBs or the matrix-image code for the 3215 PR-KB. These codes are transmitted to the write-data register for printing.

If the end-of-line switch is operated after a character is printed, a new-line function occurs but the new line-character bit pattern is not sent to main storage. The proceed light is off until the new-line function is completed.

If the End key is pressed, the read operation is ended. (If the count is not 0 and the SLI flag is off, IL is indicated in the CSW stored for the operation.) The End character bit pattern is not sent to main storage, and nothing is printed as a result of an End-key operation.

If the cancel key is pressed, unit-exception status is set on. (If the data count is not 0 and the SLI flag is off, IL is indicated in the CSW stored for the operation.) A bit pattern is not sent to program storage, but an asterisk is printed and the carrier returns.

If the data count equals zero, the operation is ended the next time any key is operated. The character bit pattern is not sent to main storage, and nothing is printed. If any key other than the end or cancel key is operated, IL is indicated in the CSW stored when the SLI flag is off.

At the end of each read operation (count zero, or end of data), the printer-ready condition is tested. If the printer is not ready, chaining is not allowed. Unit check, channel end, and device end are set on as ending status in the PR-KB sense byte.

Alarm (Audible)

This control command is an immediate command and functions in the same manner as a No-Op, except that an alarm sounds in the CPU when the control command is executed.

Under program control, the audible alarm signals the operator when the system requires manual attention. When the program issues the alarm command, the feature emits an audio tone and turns on the alarm indicator on the printer-keyboard. The tone sounds for about 1.5 seconds, but the indicator remains lighted until the operator presses the alarm reset key on the printer-keyboard. The alarm intensity is adjustable.

An alarm command is executed even if the PR-KB is in the not-ready state.

CHANNEL STATUS BYTE FOR PRINTER-KEYBOARD

The channel status byte for PR-KB operations is set up in the UCW used for the PR-KB. The channel status byte portion of the CSW is stored as a result of processing a test I/O instruction, only if an interruption condition is outstanding for the PR-KB, or as a result of an I/O interruption executed for the PR-KB.

Program-Controlled Interruption (PCI)—(CSW Bit 40)

The following are characteristics of the PCI.

1. The PCI does not affect the progress of the current operation.
2. PCIs are not stacked. If one or more PCIs in a chain have not been processed (prevented by the system mask), only the latest PCI is processed.
3. A PCI bit in a CCW causes the PCI to remain pending (until processed) throughout the chain if the PCI cannot be taken when first detected.

Incorrect Length (IL)—(CSW Bit 41)

1. IL is set in the CSW (stored by a test I/O or I/O interruption) as a result of a read command during which the end or cancel key is operated, or a result of a write command during which the END key (not cancel) is operated, when the count does not equal zero and the SLI flag is off. Any chain is terminated.
2. IL is set in the CSW (stored by a Test I/O or I/O interruption) as a result of a read command during which the count equals zero and any key other than the end or cancel is operated and the SLI flag is off. Any chaining is terminated.
3. An intervention-required condition occurs (out-of-forms or not-ready switch is operated), and the count does not equal zero when either:
   a. The SLI flag is off, or
   b. The SLI flag is on and data chaining is also specified.

   Any chaining is terminated.
4. IL is set for any write command if the SLI flag is off for that operation. This results from the fact that the PR-KB attachment requests one more data cycle after the CCW data count has decremented to zero. Any chaining is terminated if the IL indication occurs. If, however, the chain-data flag is on in the current write command, the IL indication will not occur for that write command. But, the last write command in the data chain should have its SLI flag on in order to avoid the IL indication.
5. IL is set for a sense command if the data count specified is greater than 1 and the SLI flag is off.

Program Check (CSW Bit 42) and Protection Check (CSW Bit 43)

Program check and protection check are set as defined for the multiplexer channel.

Channel Data Check (CSW Bit 44)

This bit does not apply to PR-KB operations.

Channel Control Check (CSW Bit 45)

This bit is set as defined for the multiplexer channel.

Interface Control Check (CSW Bit 46)

A share request is received and none of the following conditions exists.

1. A not-ready-to-ready sequence has not been performed, or
2. The request key has not been operated, or
3. No program-controlled operation is in progress, or
4. No status is outstanding for the PR-KB.

Chaining Check (CSW Bit 47)

This bit does not apply to PR-KB operations.

STATUS BYTE FOR PRINTER-KEYBOARD

The status byte for the printer-keyboard is kept in control storage. Status is presented in the CSW, only for not-ready-to-ready, device-end, channel-initiated, and attention-status operations (that is, not for alter/display functions).

Attention (Status Bit 0)

This bit is set when the request key is pressed if no other operation is in progress. If another operation is in progress, pressing of the request key causes the attention status bit to be turned on after status for the other operation has been cleared (that is, accepted by the CPU program). If the other operation is an alter/display operation (for which operation status is not presented), attention is not set on until the alter/display is completed.

When the attention bit (status bit 0) is on, bit 32 of the CSW will be set on for the following.

1. If an I/O interruption for the PR-KB is processed.
2. If a start I/O is executed before the I/O interruption can be processed. Busy (bit 35) is also set on.
3. If a test I/O is executed before the I/O interruption can be processed.

The preceding Items 1, 2, and 3 clear the status at the PR-KB.

Status Modifier (Status Bit 1) and Control Unit End (Status Bit 2)

Unit status bits 1 and 2 are not used for PR-KB operations.

Busy (Status Bit 3)

Busy is set in the CSW (bit 35) stored as a result of execution of a start I/O for the following conditions only.

1. A program operation (other than a No-Op or alarm command) has been completed to the point at which channel end has been accepted by the CPU (and I/O interruption or test I/O instruction has been processed to store the channel end in a CSW), but device end is not outstanding. Device end (CSW bit 37) accompanies busy in the CSW for the start I/O, and the status at the PR-KB is cleared.
2. Attention status (resulting from a request-key operation) is outstanding for the PR-KB (that is, the attention has not yet been cleared by an I/O interruption or test I/O operation). Attention (CSW bit 32) accompanies the busy bit in the CSW stored for the start I/O.
3. A device end for a not-ready-to-ready sequence (the ready switch has been operated) is outstanding at the PR-KB. Device end (CSW bit 37) accompanies busy in the CSW stored for the start I/O.
4. A program operation has been completed to the point at which channel end has been accepted by the CPU (an I/O interruption or test I/O instruction has been processed to store the channel end in the CSW), but device end is not yet available. The busy bit alone is presented in the CSW for the start I/O, and the PR-KB status is not affected.

Busy is stored as a result of a test I/O instruction, only if it is executed after channel end for a command is accepted, but before device end for that same command occurs.

Channel End (Status Bit 4)

Channel end is set on in the PR-KB attachment for any of the following conditions.

1. A zero data count has occurred for a write, write-with-ACR, read, or sense command.
   (For write or write-with-ACR, channel end is set on during the share cycle after the one in which the zero data count is detected.)
2. At initial selection during execution of a No-Op or alarm command when that command is accepted by the PR-KB attachment.
3. The END key or the cancel key has been operated during a read command.
4. The END key is operated during a write command.
5. If a count of greater than 1 is specified in a sense command; the operation is terminated after 1 byte is transferred.

If channel end alone is in the multiplexer channel Interruption Buffer (IB) or has been stacked at the PR-KB, it is cleared by an I/O interruption (or by a test I/O) and stored in the CSW.

Device End (Status Bit 5)

Device end is set on for any of the following reasons.

1. When carrier-return motion has begun for a read or write-with-ACR command.
2. On the service request (share cycle) following the one in which a zero-count condition occurs for a write (with no ACR) command.
3. When the PR-KB attachment accepts a No-Op or alarm command.
4. When the ready key is pressed while the PR-KB is in the not-ready condition. (The key must produce a ready condition to set device end.)
5. During the service request in which a sense byte is sent to the CPU.

If a device end has been generated or stacked at the PR-KB, it is cleared by initial selection for a start I/O, only if channel end (as a result of the operation) has already been stored in the CSW by an I/O interruption or test I/O. Busy accompanies device end in the CSW stored for the start I/O.

Test I/O clears any outstanding device end at the PR-KB.

A halt I/O does not clear a device end at the PR-KB.

Unit Check (Status Bit 6)

Unit check is set for any of the following reasons.

1. When a character with even parity is sent from the keyboard to the CPU during a read command operation. Equipment check, sense bit 3, is also set on for this condition.
2. If a parity error is detected on data during a write operation with the 3215, a check condition is indicated in the same manner as for other multiplexer-channel operations.
3. If the PR-KB goes not-ready because the cover is open, or the printer is out of forms, or the not ready key is operated:
   a. At initial selection for a read or write (with or without ACR) command, or
   b. During execution of a test I/O instruction to the PR-KB.
   c. At the end of a printing operation.

   (Intervention required, sense bit 1, is also set on for this condition.)
4. If a command byte not defined for the PR-KB is sent to the PR-KB. (Command reject, sense bit 0, is also set for this condition.)
5. If the printer fails to print within approximately two seconds, equipment check (sense bit 3) is also set.
6. If the 3215 printer has a print-sync check. Equipment check (sense bit-3) is also set.

Unit Exception (Status Bit 7)

This bit is set on if the cancel key is operated during a read command operation. The read operation is terminated (channel-end status is set on). If the count is not zero and the SLI flag is off for the read command, the incorrect-length indication (CSW bit 41) is also given during a subsequent I/O interruption or test I/O operation.

SENSE BYTE FOR PRINTER-KEYBOARD

The PR-KB sense byte is kept in control storage. Unit check status is set whenever any one or more of the following bits are set.

Command Reject (Sense Bit 0)

This bit is set on if a command not defined for the PR-KB is issued.

Intervention Required (Sense Bit 1)

This bit is set on only for a read or write command in which:

1. The not-ready switch has been operated to place the PR-KB in a not-ready condition, or
2. The forms switch indicates that the PR-KB required forms.
3. The PR-KB cover is open.
4. AC power is off in the 3210 Model 2.
5. The 3215 is in CE Mode (printing H's, offline).

Bus-Out Check (Sense Bit 2)

This bit is not set for PR-KB read operations.

Equipment Check (Sense Bit 3)

This bit is set on:

1. When even parity is detected on a character code sent from the keyboard to the CPU during a read operation.
2. If the printer fails to print.
3. If the 3215 printer had a print-sync check or incorrect data from the keyboard.

   If Equipment Check is set because of bad parity from the keyboard, the operation is not terminated until its normal ending point.

   If Equipment Check is set because the printer fails to print (during a read or write operation), the operation is terminated.

Sense Bits 4-7

These bits are not set for PR-KB operations.

SUGGESTED RESTART PROCEDURES FOR PRINTER-KEYBOARD

An I/O error causes an interruption condition, which is indicated in the CSW. The CSW is located in main-storage locations 40 through 47 (hexadecimal). Bit 38 of the CSW, when on, indicates a unit-check condition. This is bit 6 of the byte at main-storage address 44 (hexadecimal).

When a PR-KB unit check is detected by the program, a sense command should be executed for the PR-KB. Sense information sent from the attachment provides more detailed information concerning the cause of the unit check. As a result of program analysis of the sense information, an error message should be made available to the operator to indicate the condition.

The following information describes the actions that should be performed when the program detects unit-check status in the CSW. The actions are related to particular sense indications that can occur. These bits are analyzed by the program.

Command Reject (Sense Bit 0)

Provide an operator message and exit from this error-recovery procedure. Command reject indicates that an invalid command was received at the PR-KB attachment.

Intervention Required (Sense Bit 1)

The PR-KB enters a not-ready condition (intervention-required light on) becuase one of the following has occurred:

1. The not-ready key was operated to place the PR-KB in a not-ready condition.
2. The PR-KB has run out of forms.
3. The PR-KB cover is open.
4. The AC power switch in the 3210 Model 2 is off.
5. The 3215 is in CE Mode (offline).

If possible, provide a message to instruct the operator to:

1. Check that paper is in the printer.
2. Close the printer cover.
3. Make sure the ac power switch for the 3210 Model 2 is on.
4. Make sure that the 3215 printer is not in CE (printing H's) Mode.
5. Press the ready key on the PR-KB console.

Even if it is not possible to provide a message, the INTVN REQD light will be on.

Equipment Check (Sense Bit 3)

Provide an operator message to indicate the failure to read the input message and provide for at least one retry at execution the command that resulted in the equipment check.

CHAPTER 8. CHANNELS

INTRODUCTION

- The CPU of the System/370 Model 145 communicates with I/O devices through channels.
- The channels are connected to, and communicate with, the I/O control units via a standard interface.
- The channels are a physical part of the CPU but are functionally separate.
- Each I/O device attached to a channel must have an associated control unit.

Control of the operation and transfer of data between the CPU and I/O devices are via the standard interface of 38 lines. The CPU has a group of control circuits and microprograms to execute the required channel sequences. Each I/O device operates through an I/O control unit that interprets the standard interface signals and provides the necessary control lines for the I/O devices.

Each device can have its own control unit, or several devices can be controlled by one control unit. The control unit acts as a buffer and compensates for the difference in the rate of flow of data, or the time of occurrence of events when transferring information between the 3145 and an I/O device.

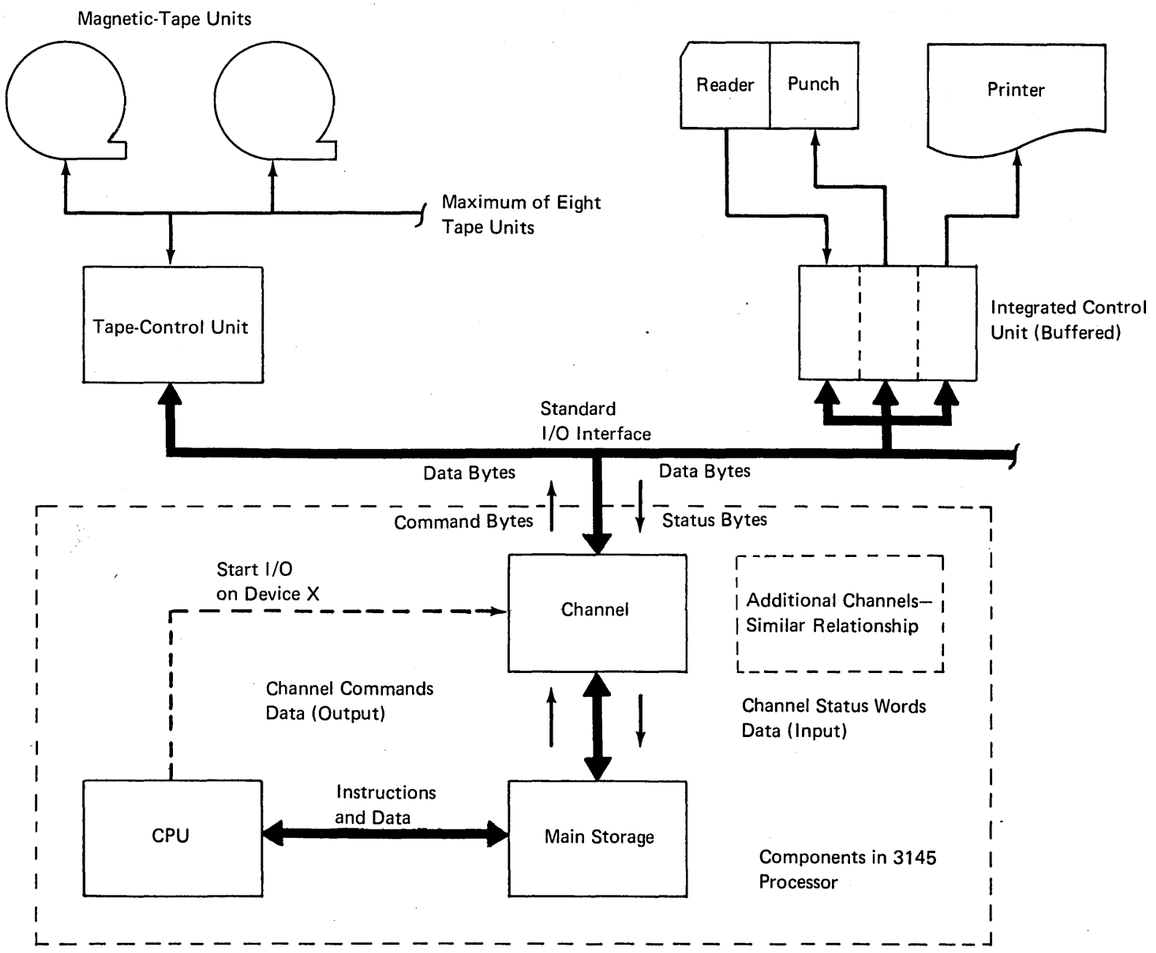
In some cases a single byte of data is buffered in a register, and other cases a complete record is stored in a storage unit.

A channel can have up to eight I/O control units attached. Only one of the control units is logically connected to the interface at a given time.

The period of connection varies, depending on the size of the transfer and the speed of the I/O device. Devices operating in byte and multibyte modes on the multiplexer channel disconnect from the channel after each byte or group of bytes. The device control unit requests a new connection when it is ready for an additional transfer. During these intervening periods, other channel devices may use the interface.

For identification, each I/O device attached to the standard interface is assigned a unique address during the installation of the device. This address is set on the interface lines by the channel when it initiates an operation with the device, or by the device control unit when the device is requesting service.

Overview of Channel Operation



1. (CPU in Supervisor State)
2. Execute start I/O instruction
3. Fetch CAW
4. Fetch CCW
5. Send command to device
6. Set condition code 0
7. Channel transfers data
8. End operation
9. Take I/O interruption
10. Store CSW
11. Store old I/O PSW
12. Get new I/O PSW

MODEL 145 I/O CHANNELS

- Several channel configurations are available:

One byte-multiplexer and one selector channel are standard.

Special features: Integrated File Adapater (IFA), addressed as selector channel 1. When IFA is installed, selector channel 2 is standard. Additional channels: For IFA configuration, selector channel 3 may be added for non-IFA configuration, selector channels 2, 3, and 4 are available. The word buffer feature is available on selector channels. Block-multiplexer channels are available instead of any or all installed selector channels. The Channel-to-Channel Adapter Feature is available.

The byte-multiplexer channel can operate in either multiplex or burst mode. The mode is determined by the characteristics of the device operating on the channel. If a device is capable of forcing burst mode operation and that device is started, no other device can operate with the byte-multiplexer channel until the burst-mode operation is completed.

In multiplex mode, the single data path of the byte-multiplexer channel can be time-shared by a number of low-speed I/O devices operating simultaneously. The channel multiplexes, on demand, data to or from these devices (one device at a time) in groups of bytes (or singly) as determined and specified by the particular I/O device being serviced.

Any selector channels are capable of handling high-speed I/O units of many types. Operations between a selector channel and an I/O unit proceed until channel-end status for the last channel command word in the operation is received by the channel. Another unit on that channel may then be started (depending on how the operation is programmed). Selector-channel accesses to main storage during a data transfer cycle are on a byte basis, or if the Selector Channel Word Buffer feature is installed, on a word (four-byte) basis.

The block-multiplexer channel operation differs from selector channels in the way in which command-chained channel programs are handled. In selector mode during a command-chained channel program, the channel is busy during the entire time the channel program is in operation, whether data transfer is occurring or not. The block-multiplexer channel executing a command-chained channel program has the ability to disconnect from the operational channel program during certain non-data transfer operations. Thus, the channel can be freed during a nonproductive activity, (for example, during disk seeking and most record positioning), thereby allowing more data to be transferred per unit of channel busy time.

The channels operate from the same I/O instruction and command formats used for System/360 models (except Model 20).

The channels, except the IFA channel, operate with attached I/O control units in accordance with signal sequences defined for System/360 and System/370. Refer to Chapter 10 of this manual for IFA operation.

INSTRUCTIONS

- There are seven instructions—all use SI format (32 bits).
- Instructions specify channel, control unit, and I/O device.

Seven program instructions are used to initiate I/O operations on channel. All of them use the I/O instruction format as defined in the System/360 Principles of Operation.

The instructions defined are:

Start I/O (SIO)
Start I/O Fast Release (SIOF)
Test I/O (TIO)
Halt I/O (HIO)
Halt Device (HDV)
Test Channel (TCH)
Store Channel ID (STIDC)

Start I/O (9C): This instruction is used to initiate all I/O operations that control device operation, transfer data, or obtain sense information.

Start I/O Fast Release (9C and bit 15): This instruction is executed as start I/O in the Model 145.

Test I/O (9D): The test I/O instruction is used for testing the state of the addressed channel, subchannel, and I/O device. TIO sets a condition code in the current PSW.

Halt I/O (9E): This code is used to discontinue the operation between the addressed channel and whatever I/O device is in operation on the channel.

Halt Device (9E and bit 15 on): This code is a variation of Halt I/O. Halt device is used to discontinue the operation between the addressed channel and a particular device.

Test Channel (9F): The Test Channel instruction is used for testing the state of the addressed channel. The state of the channel is not affected, and no action is caused.

Store Channel ID (B203): This instruction causes information identifying the designated channel to be stored in the two bytes, 168-169.

All seven I/O instructions are executed only in the supervisor state and use the SI format in a 32-bit word:

Bits 0-7: Op code.

Bits 8-15: Used for additional Op code modifiers for some instructions.

Bits 16-19: This four-bit field designates the general register in auxiliary storage that is to be used to develop the channel and device addresses. The register contains the base address (B1), with a possible 32 bits.

Bits 20-31: This twelve-bit field contains the literal value to be used as the displacement (D1) in developing the channel and device addresses for the operation.

The value of B1 from the specified general register is added to the D1 value specified to form a 32-bit address. The resultant address is used to identify the channel, subchannel, and I/O device as follows:

Bits 0-15: Not used for channel addressing.

Bits 16-23: These eight bits can address a theoretical 256 channels. In the Model 145, addresses 0, 1, 2, 3, and 4 are valid.

Bits 24-31: These eight bits can address a possible 256 I/O devices per channel.

The seven I/O instructions specify only the address of the channel, and the I/O device to be used in the operation. The start I/O instruction requires additional operating information for its execution. The channel gets this information from a channel address word (CAW) in main-storage address location 48 and from channel command words (CCW) also in main storage.

CHANNEL CONTROL WORDS

There are four control words associated with I/O operations:

1. Channel Address Word (CAW)
2. Channel Command Word (CCW)
3. Channel Status Word (CSW)
4. Program Status Word (PSW)

CAW

The CAW is one word (four bytes) of information stored in location 48. It is used during the start I/O instruction to identify the location in program storage where the CCW concerned is located. It also contains the storage-protection key for subsequent data transfers and CCW fetches.

CCW

The CCW is a doubleword; its location is specified by the CAW. The CCW is fetched during execution of the start I/O instruction. One or more CCWs are used to make the program of channel operations. The CCW contains the command code, data address, byte count, and flag bits. CCWs are fetched sequentially except that the Transfer In Channel (TIC) command may direct otherwise.

CSW

The CSW keeps the program informed of the status of an I/O device and the conditions under which an I/O operation has been terminated. The CSW is generated or modified during execution of the start I/O, test I/O, halt I/O, and halt device instructions, and by an I/O interruption. The CSW is placed in (main) storage location 40 and is available to the program at this location until an I/O instruction replaces it or an I/O interruption occurs. When the CSW is stored because of the I/O interruption, the I/O device that caused the interruption is identified in the old PSW.

PSW

The PSW contains information equired for correct execution of the program. The PSW (or part of it) is stored and replaced when the conduct of the program is to be changed because of an interruption. The stored information is brought back to control the state of the CPU and the conduct of the program when the original activity is resumed. There are two PSWs for each of the five interrupt conditions. The two PSWs are referred to as *old* and *new.*

The PSW contains mask bits to mask I/O interruptions, provides a place in the *old* I/O PSW for the device address, and contains the supervisor bit that controls when I/O instructions can be executed.

Refer to the System/360 Principles of Operation and the Reference Data Card for additional information concerning format and content of the control words.

SUBCHANNELS

- The channel facilities required for sustaining a single I/O operation are termed a *subchannel.*
- The subchannel consists of the control-storage area used for recording the addresses, count, and any status and control information associated with the I/O operation. For selector channels, the hardware and local storage constitute the single subchannel facility of the channel.
- Four-word Unit Control Words (UCWs) in control storage serve as subchannels for byte-multiplexer channels.
- Block-multiplexer channel subchannels require two-word UCWs.

Subchannels are commonly referred to as UCWs. The mode in which a channel can operate depends on whether it has one or more subchannels. The byte-multiplexer channel contains a minimum of 16 and a maximum of 256 subchannels and can operate in either multiplexer or burst mode. In multiplex mode, one or more devices may operate concurrently, each on a separate subchannel. A selector channel has one subchannel and always forces the I/O device to transfer data in burst mode. A selector channel cannot Disconnect Command Chain (DCC).

The number of block multiplexer channel subchannels required depends on the number and type of I/O devices in the System configuration. The maximum number of block-multiplexer UCWs possible per system is 512. The actual number must be specified at order time and must be a multiple of 16.

Subchannels on the byte-multiplexer and block-multiplexer channel may be either unshared or shared. Details of shared and unshared UCW assignment and method of addressing is explained in the individual sections describing byte-multiplexer and block-multiplexer channel operation respectively.

STANDARD INTERFACE

Bus-Out A

- 'Bus-out' transfers information from the channel to a control unit.
- Information on 'bus-out' can be either data, control, or address.
- Tags-out identify the information on 'bus-out' lines. For example, the 'address-out' tag designates the information on 'bus-out' as an address.
- Tag lines control the period during which 'bus-out' lines contain valid information. Data on 'bus-out' lines must be valid before the rise of the associated tag line and until the rise or fall of the associated input tag line. For 'address-out', the address must be on the bus at least 250 ns before 'address-out'.

Bus-In B

- 'Bus-in' transfers information from a control unit to the channel.
- 'Bus-in' information can be either data, address, or status.
- Tags-in identify the information on 'bus-in' lines.

Mark 0 In C

The 'mark 0 in' line is used only for the Command Retry feature.

When the command being executed encounters a condition requiring retry, the control unit indicates it by raising 'mark 0 in' and 'status-in' while presenting 'unit check' and 'status modifier' ('retry status') together with 'channel end' (meaning the control unit or the device is not yet ready to retry the command), or with 'channel end' and 'device end' (meaning the control unit and device are prepared for immediate retry of the command).

Tags-Out D

- Tags-out identify the information on 'bus-out' lines.
- Only one of these lines can be active at a time. (Address-out can be up with another tag for an Interface Disconnect sequence.)
- An active tag-out line remains active until a tag-in line responds.

Address-Out

- 'Address-out' identifies information on 'bus-out' as an address or instructs the control unit to disconnect from the interface (halt I/O instruction).
- When 'address-out' is active during initial selection, all attached control units decode the address on 'bus-out'.
- 'Address-out' drops when it receives a response of 'Operational-in' (control unit selected), 'select-in' (no control unit selected), or 'status-in' (control unit busy).
- During a halt I/O instruction, when interface disconnect is required, 'address-out' is active until 'op-in' falls. The I/O operation proceeds to the normal end, but no data is transferred across the interface.

Command-Out

- 'Command-out' identifies information on 'bus-out' as a command.
- During a control-unit-initiated initial-selection sequence, a 'command-out' response to 'address-in' signals the control unit to proceed (present status or data).
- In response to 'status-in', it signals the control unit to stack status.
- In response to 'service-in' (or 'data-in'), it signals the control unit to stop data transfer.

Service-Out

- 'Service-out' is the response to 'service-in' on read or write operations and to 'status-in' during a status sequence.
- 'Service-out' indicates to a control unit that information on 'bus-in' was accepted by the channel or that the channel has provided the requested information on 'bus-out'.

Data-Out

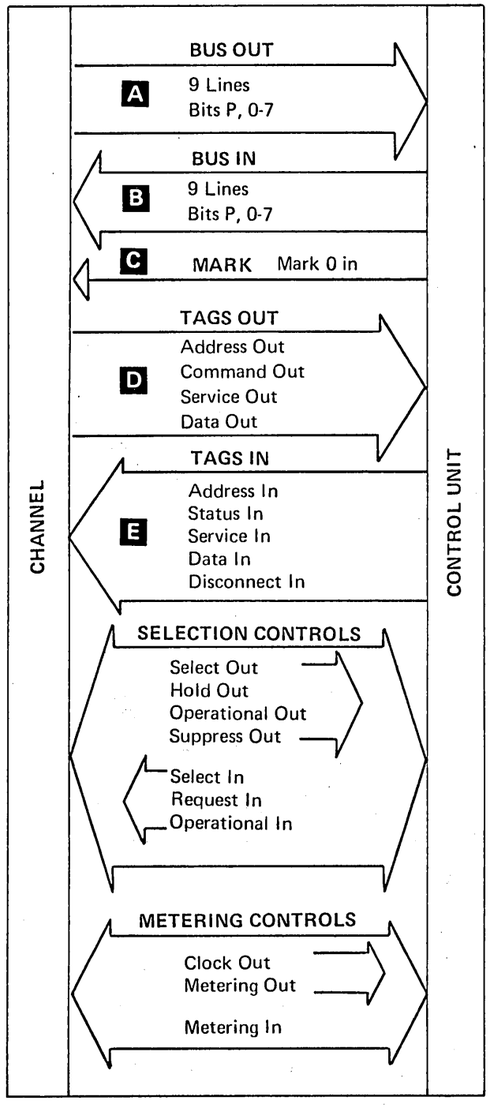
- 'Data-out' is the response to 'data-in'.
- 'Data-out' indicates to a control unit that data on 'bus-in' was accepted by the channel or that the channel provided the requested data on 'bus-out'.
- 'Data-out' is effective with selector/block-multiplexer channels only.

Tags-In E

- Tags-in identify the information on 'bus-in' lines.
- Only one of these lines can be active at a time.
- An active tag-in line remains active until a tag-out line responds.

Address-In

- 'Address-in' identifies the information on the 'bus-in' lines as an I/O unit address.



Status-In

- 'Status-in' identifies the information on the 'bus-in' lines as a status byte.
- 'Status-in' remains active until the channel responds with 'service-out' (channel accepted status) or 'command-out' (stack status) or drops Select Out for a short control-unit busy sequence.

Service-In

- On a read operation, 'service-in' indicates that data is available on the 'bus-in' lines.
- On a write operation, it indicates that the control unit is ready to receive data.
- This line remains active until the rise of either 'command-out', 'service-out', or (during an interface disconnect) by 'address-out'.

Data-In

- During read and sense operations 'data-in' rises when data is available on 'bus-in'. During write and control operations, 'data-in' indicates that the control unit is ready to receive data.
- 'Data-in' indicates to the channel that data on 'bus-out' was accepted by the control unit or that the control unit provided the requested data on 'bus-in'.
- 'Data-in' is effective with selector/block-multiplexer channels only.

Disconnect-In

- 'Disconnect-in' provides control units with the ability to alert the system of malfunctions.
- The channel responds to 'disconnect-in' by performing a selective reset.

Selection Controls F

Selection controls are used to scan, or select, attached I/O devices. They establish communication between the channel and the control units on a priority basis. Selection controls permit only one control unit at a time to communicate with the channel.

Select-Out and Select-In

- 'Select-out' is connected serially through each control unit and is used to select the control unit.
- Priority is established by internal wiring in each control unit. The control unit allowed to interrogate the 'select-out' signal first has the highest priority. This is not necessarily the first physically cabled control unit.
- When a control unit receives the 'select-out' signal, it raises 'operational-in' (control unit selected), or propagates 'select-out' to the next control unit (control unit not selected).
- The return path for 'select-out' is 'select-in'; and when active indicates to the channel that a control unit was not selected (all control units propagated 'select-out').

Request-In

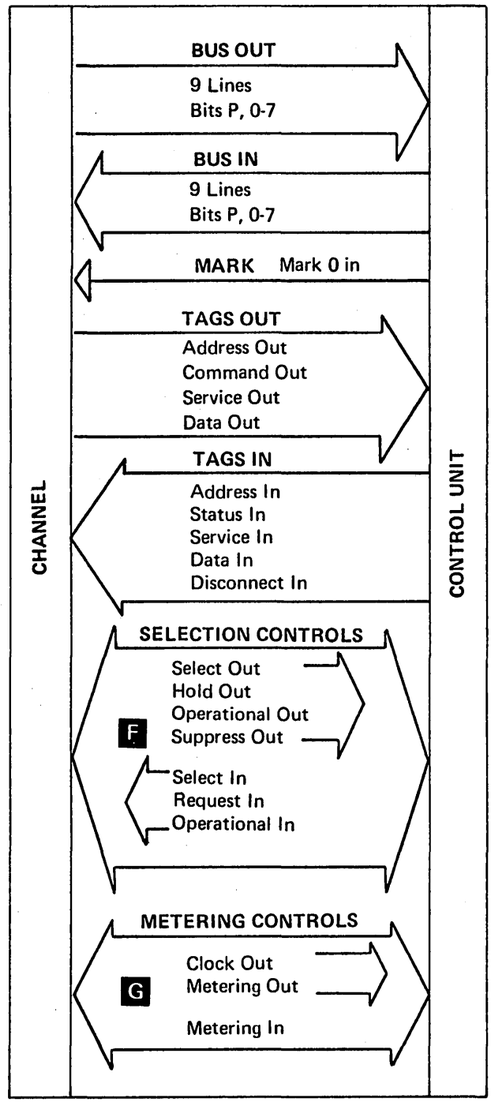
- 'Request-in' indicates to the channel that a control unit requires service.
- When the channel receives the 'request-in' signal, selection is started.
- If more than one control unit raises 'request-in' concurrently, the control unit with the highest priority is serviced first.

Hold-Out

- 'Hold-out' signal allows additional channel control over the polling operation by controlling the effect of 'select-out'.
- When the channel holds 'select-out' active and cancels 'hold-out', no control unit can make use of 'select-out'. Therefore, the channel can interrupt the polling sequence and cause 'select-out' to fall in all control units in the shortest possible time.

Operational-Out

- 'Operational-out' gates all outbound tag lines. It is raised with CPU power on reset.
- When the channel drops 'operational-out' (with 'suppress-out' up) while a control unit is operating, that control unit is reset (selective reset).
- When 'operational-out' is dropped and 'suppress-out' is down, all control units online are reset (general reset).



Operational-In

- 'Operational-in' signals the channel that a device is selected and prevents another control unit from connecting to the interface (blocks the propagation of 'select-out').
- Signals on 'bus-in' and all inbound tag lines are valid only when 'operational-in' is active, except in the case of the short control-unit-busy sequence.

Suppress-Out

- 'Supress-out' is used alone or with an outbound tag line to suppress status, suppress data transfer, for chain command control, or for selective reset.
- Suppress Status: When a control unit ends an I/O operation, it transmits a status byte on bus-in lines and conditions the 'status-in' tag line to the channel. A channel may respond to 'status-in' with 'command-out', causing the control unit to stack the status information.

  When 'select-out' rises at a control unit holding stacked status, that control unit will not capture the interface to present the status byte if suppress-out is active. To ensure that the stacked status data is not transmitted, the channel must condition 'suppress-out' before the control unit receives 'select-out'. 'Suppress-out' prevents a control from raising 'request-in' to present stacked or suppressible status. If a control unit conditions 'request-in' to offer status, and 'suppress-out' rises before the control unit receives 'select-out', 'suppress-out' drops 'request-in'.
- Suppress Data Transfer: For buffered I/O devices that can wait for data transfers without indicating an overrun condition, 'suppress-out' blocks data transfer ('service-in' or 'data-in'). To ensure that the subsequent request for data or offer of data will be suppressed, the channel must condition 'suppress-out' before the previous 'service-out' tag drops.
- Chain Command Control: Command chaining is indicated if 'suppress-out' is up when 'service-out' is raised in response to 'status-in'. When command chaining is indicated at the time 'device-end' is presented, this indication is valid until reselection is made or until 'suppress-out' falls.

  If device-end does not accompany channel-end, then the I/O device that presents 'channel-end' must be the next device from that control unit to present device-end status.
- Selective Reset: If the channel conditions 'suppress-out' before allowing 'operational-out' to fall and holds 'suppress-out' active until after 'operational-out' rises again, only the I/O device presently operating on the interface is reset.

Metering Controls G

Metering controls are used for conditioning usage meters located in the various attached units.

Clock-Out

- 'Clock-out' designates when a control unit is allowed to change to CE mode ('clock-out' down).
- 'Clock-out' indicates that the processing unit is not in a halt or wait state.

Metering-Out

- 'Metering-out' enables the control-unit usage meters to record time.

Metering-In

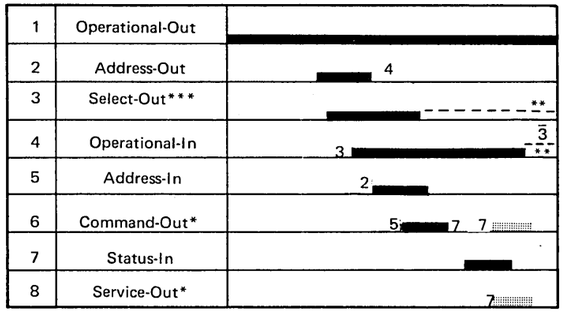
- 'Metering-in' indicates to the channel that the control-unit customer usage meter is recording time.
- 'Metering-in' causes the customer meter to accumulate time even though the processing unit may be in the halt or wait condition.

INTERFACE OPERATION SUMMARY

Initial Selection Sequence

- During the initial selection sequence, the channel selects a control unit and specifies the operation to be performed.
- This sequence is essentially the same for all control units and operations.

The interface signal sequence in which the channel selects a control unit and I/O device and specifies an operation to be performed is called the *initial selection sequence.* Regardless of the unit selected or the operation designated, the signal sequence in the initial selection is standard.



* A multiplexer channel responds to status-in with either command-out or service-out. Normally, a selector channel M11 responds to status-in with only service-out.

** Depending on the channel controlling the operation, select-out might drop during the initial selection sequence or remain active after the sequence is complete. Operational-in cannot drop until select-out is inactive.

*** Select-Out and Hold-Out are up together.

The channel begins the initial selection sequence by transmitting an address byte on the 'bus-out' lines and raising 'address-out'. The address byte selects the unit to execute the operation. Each control unit attached to the interface attempts to decode the address; but, because all interface addresses are different, only one unit can interpret the coded byte.

When 'select-out' rises at the control unit that successfully decodes the address byte, that control unit conditions either:

1. 'Status-in', indicating that the selected unit is busy and cannot execute another operation, or
2. 'Operational-in', indicating that the designated unit will complete the initial selection sequence.

If no control unit decodes the address byte (specified control unit is offline; the address byte is invalid, etc.), the control unit with the lowest priority propagates 'select-in' to the channel when its incoming 'select-out' is conditioned. The 'select-in' or 'status-in' reply to 'select-out' and 'address-out' causes the channel to drop 'select-out' and 'address-out' and terminate the selection sequence.

When 'operational-in' causes the channel to drop 'address-out', the selected control unit then transmits an address byte on 'bus-in' lines and conditions the 'address-in' line. The channel compares this address to the address it placed on the 'bus-out' lines to ensure that the right device has been selected.

After checking the address, the channel responds to 'address-in' by transmitting a command byte and conditioning 'command-out' to the control unit. The command byte designates one of seven operations (read, read-backward, write, control, sense, test I/O, or the No-Op special control), and establishes conditions to control execution of the operation.

The control unit must then drop 'address-in'; and after 'command-out' falls, the control unit places its status information on 'bus-in' and raises 'status-in'. If the I/O device is available and capable of executing the command, the status byte is zero. If the channel accepts this status byte, it responds with 'service-out'. This signal completes the initial selection sequence.

Nonzero status does not necessarily mean that the device is unavailable. It can also occur during selection for handling interruptions or for a TIO instruction. For start I/O, the selection is completed, a CSW is stored, and condition code 1 is set. Nonzero status can also occur on initial selection with a control immediate command.

When a control unit that does not have 'operational-in' up requires service, it raises its 'request-in' line to the channel. The next time that 'select-out' rises at any control unit requiring service and no I/O selection is being attempted by the channel ('address-out' down), the control unit places the address of the device on 'bus-in'. It then signals on both the 'address-in' and the 'operational-in' lines and removes the 'request-in' signal.

When the channel recognizes the address, a 'command-out' signal is sent to the control unit, indicating proceed. After 'address-in' drops, the channel responds by dropping the 'command-out' signal.

If the service request is for data, the sequence proceeds as described in "Data Transfer." If the service request is for status information, the sequence proceeds as defined for the status cycle in the ending procedure.

Data Transfer

- A control unit can send data to or request data from the channel.
- 'Service-in' and 'service-out' are the controlling tag lines.
- 'Data-in' and 'data-out' can also be used with selector/block-multiplexer channels.

Data transfer may be requested by a control unit after a selection sequence. To transmit to the channel, the control unit places a data byte on 'bus-in' and raises 'service-in'. The tag and the data on 'bus-in' must be held until an outbound tag is raised in response. To request data from the channel, 'service-in' is raised. The channel places the data on 'bus-out' and signals with 'service-out'. The channel maintains the validity of 'bus-out' until 'service-in' falls. When 'service-in' falls, the channel responds by dropping 'service-out'.

End Operation

- An operation is completed when the control unit and device present ending status to the channel.
- The channel acknowledges 'status-in' with 'service-out' or 'command-out'.

When any I/O operation, except test-I/O and No-Op, has proceeded to its normal end, the control unit assembles and transmits a status byte to the channel.

To acknowledge 'status-in', the channel conditions either 'service-out' or 'command-out'. 'Service-out' indicates that the channel has accepted the status data and resets the operation. 'Command-out' causes the control unit to stack the status byte.

If the channel conditions 'suppress-out' 250 nanoseconds before 'select-out' rises at the control unit holding stacked status data, the control unit does not transmit the status byte again until 'suppress-out' drops. When the channel cancels 'suppress-out', and 'select-out' to the unit rises, the control unit sends its address, and re-transmits the status byte to the channel.

If the channel does not condition 'suppress-out', the control unit initiates another cycle to transmit the status byte again by raising 'request-in'.

BYTE-MULTIPLEXER CHANNEL

- The byte-multiplexer channel shares data-flow registers with the CPU.
- The byte-multiplexer channel consists mainly of microprogram routines.
- The byte-multiplexer channel can sustain operations with several I/O devices on a data interleaving basis.
- The byte-multiplexer channel may operate in either byte or burst mode.

Byte-multiplexer channel operations with attached I/O units are executed under microprogram control. A particular status, data, or control communication with a device is coordinated by the standard-interface signal sequences between the device and the byte-multiplexer channel. The I/O instructions issued to the byte-multiplexer channel (Start I/O, Test I/O, Halt I/O, Halt Device, and Test Channel) are all executed by the channel microprogram routines.

The byte-multiplexer channel may operate in either one of two modes, byte or burst. In burst mode, the channel selects and operates with one I/O device only, until the entire CCW or chain of CCWs is executed. In byte or data interleave mode, the channel selects and operates with a device until one or more bytes have been transferred. The channel has no control over its mode of operation; nor in the case of byte mode, does it determine the number of bytes to be transferred on each selection of an I/O device. Both of these conditions are determined by the I/O device involved.

The basic byte-multiplexer channel operates under control of 16 subchannels (UCWs). The number of subchannels can be any of the following: 16, 32, 64, 128, or 256. The subchannels maintain the operating information for each I/O device being operated by the channel.

Upon the initiation of an I/O operation by the CPU, the channel microprogram reads out the channel address word (CAW). Besides reading out the CAW, this routine also checks the PSW to ensure that it specifies supervisory state.

If the PSW does not specify supervisory state, a program interruption occurs. If the PSW is in supervisory state, the channel begins the execution of the instruction.

The channel initiates an operation by sending the desired address out on the channel. If the device is available, it returns the address for verification. A command is then sent to the device to initiate the operation. The device status (zero if device is available) is returned to the CPU, a condition code is set, and the operation continues. The sequence from this point varies, depending on the I/O device. Unless the initial command calls for a data transfer, the channel disconnects to allow operation of other channel devices or CPU routines.

If the device forces burst mode, the peration may be done before the condition code is set. The condition code cannot be set until the device disconnects from the channel.

Whenever the CPU returns to executing other instructions, the operating information for a selected device is stored in the UCW. When the device requires service, the channel breaks into the current microprogram routine, performs a branch and link to store the S, P, N2 and N3 registers in local storage, obtains the operating information from the UCW, and operates with the device until it disconnects from the channel. When the I/O control unit disconnects from the channel, the channel updates the subchannel information and allows the CPU to continue the execution of the current instruction.

The device (or its control unit) controls the points of channel disconnect. When the device starts transfer of data, it may operate in burst mode and transfer the entire record, or it may operate in byte (or multibyte) mode and transfer one (or several) characters, then disconnect.

When the entire CCW sequence specified by the instruction has been completed, the channel interrupts the current PSW instruction sequence if allowed to do so by the current PSW system mask. The channel then stores the required information in the Channel Status Word (CSW). After the CSW has been generated, the I/O new PSW takes the place of the current PSW. The CPU now executes the I/O new PSW instruction sequence.

The I/O program normally prevents other I/O interruptions by masking them off with the system mask. It interrogates the CSW and executes some type of error routine if the CSW indicates that the instruction was not executed correctly. Otherwise, the I/O program normally allows the CPU to begin the execution of any pending I/O instructions and then return the CPU to processing the PSW instruction sequence that was discontinued by the I/O interruption.

BYTE-MULTIPLEXER DATA FLOW (INPUT)

Two cycles are required to place data in main storage when an MPX channel input operation is done.

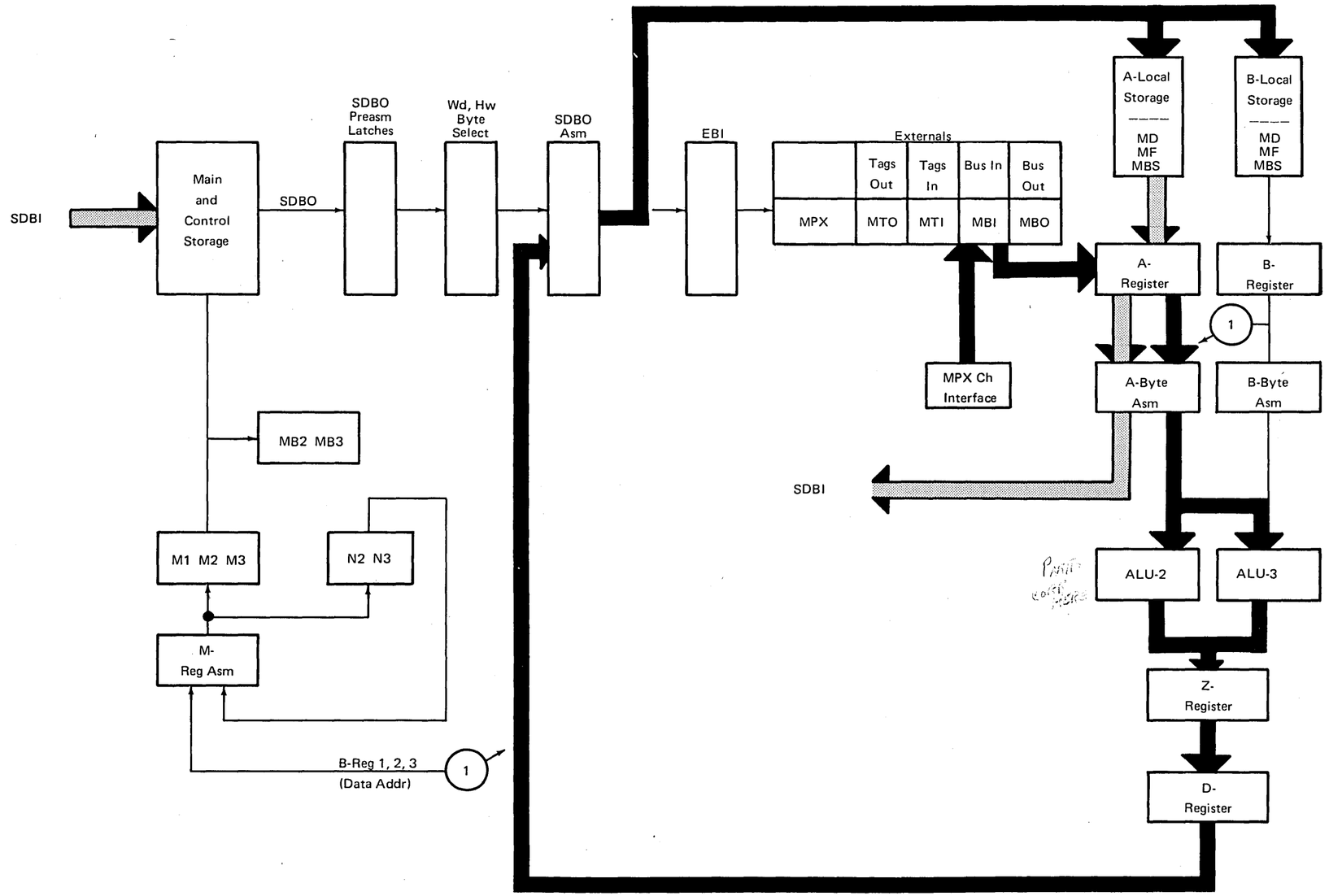
The control words used are:

1. MBS3 = MBI, 0E, 0 ABCK
2. STB MBS3 MD+1, DCNT

The two cycles of data flow are represented by:

1st Cycle

2nd Cycle

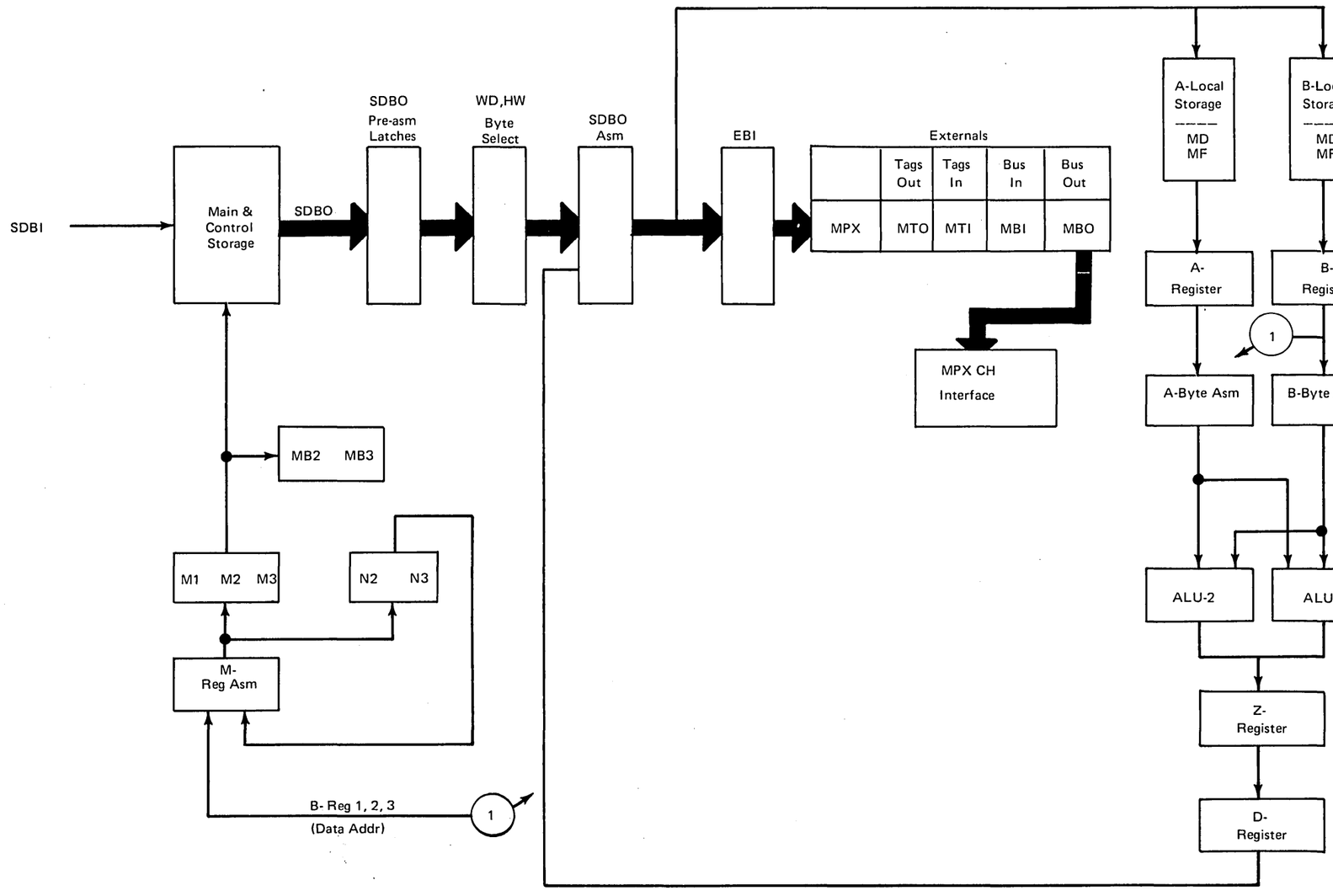


BYTE-MULTIPLEXER DATA FLOW (OUTPUT)

One cycle is required to take data from main storage and place it on the Bus-Out lines, and modify address and decrement count. The control word used is:

RDB MBO MD+1, DCNT

The Data Flow is represented by:



I/O DEVICE AND UCW ADDRESSING

Before a command for operation of an I/O device can be sent to its control unit, the device must be addressed. The address is derived from the I/O instruction and consists of the two low-order bytes of the developed address. The channel address is in the high-order byte, and the device address is in the low-order byte.

The numbering scheme for subchannels relates to I/O device addresses. The byte-multiplexer channel uses the Exxx module in control storage for its UCWs. Because each UCW requires four words, the address must be Exx0. The two hex digits (xx) are derived from the unit address. In the UCW addressing table and related text, the high-order E and low-order 0 are omitted.

One device address is used for each unshared subchannel. This is sometimes referred to as *single-unit addressing* and is abbreviated SUA. Several device addresses may be used with a shared subchannel. This is called *multi-unit addressing* (MUA), because the devices share a single control unit on that shared subchannel. The devices may be selected for operation one at a time, but not concurrently.

On the byte-multiplexer channel with either 16, 32, 64, or 128 subchannels, provision exists for subchannels 00 through 07 to be used as shared subchannels. Addresses with a 1 in the high-order bit position of the device address field (bit 0) specify subchannels that are shared. This means that on the systems with 128 or fewer subchannels no unit using an unshared subchannel can have its high-order address bit set to a 1. The byte-multiplexer channel microprogram routines recognize the high-order 1 as specifying a unit on a shared subchannel. Shared subchannels are not used on systems with 256 subchannels. In this case, the number of unshared subchannels can be a maximum of 256 (000 through 255).

Sharing is allowed on subchannels 00 through 07 on the systems with 128 subchannels or fewer. Each of these subchannels can attach up to 16 devices. No two devices are allowed to have the same UCW number *unless* they are sharing the same subchannel. The address range is as follows:

UCW	Shared Addresses		
00	80	–	8F
01	90	–	9F
02	A0	–	AF
03	B0	–	BF
04	C0	–	CF
05	D0	–	DF
06	E0	–	EF
07	F0	–	FF

Bit 0 must be a 1. Bits 1, 2, and 3 provide eight unique control-unit/UCW addresses. Bits 4, 5, 6, and 7 are ignored and thus allow multiple devices (up to 16) to share each UCW. For example, devices 90, 91, 92, and 93 would each address UCW 01.

Refer to the UCW Addressing Table.

Example 1. Assume a 32 UCW configuration and device address 75. Find address 75 in column H. Move down column H into the lower area of the chart to the row that pertains to "32 UCW systems." The letter B indicates that the device addresses in column H fold into column B; in this case, device address 75 accesses UCW 15.

Example 2. Assume a 16 UCW configuration and device address 5A. This address is in column F. Moving into the lower area of the chart to the appropriate row (opposite 16 UCW) you find the letter A indicating that 5A folds into row A, thus addressing UCW 0A.

As a general observation, note that the validity of any address for any size UCW configuration can be checked by using this table. This table also lends itself to checking shared UCW addresses. The notes that appear vertically in columns I through P indicate which UCW is addressed by any or all addresses in each column. Observe also that if any of the UCWs that can be shared *are* shared, then it logically follows that this UCW must not be addressed as an unshared UCW. For example, in a 64-UCW system, suppose that UCW 00 is shared. This automatically invalidates any unshared address that folds into UCW 00. In this example, this includes address 40 in row E. In a 16-UCW system under similar circumstances, addresses 10, 20, 30, 40, 50, 60, and 70 would all be invalid because all these addresses fold into UCW 00.

UCW Addressing Table

Column ID	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
UNIT/UCW ADDRESSES	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
	01	11	21	31	41	51	61	71	81	91	A1	B1	C1	D1	E1	F1
	02	12	22	32	42	52	62	72	82	92	A2	B2	C2	D2	E2	F2
	03	13	23	33	43	53	63	73	83	93	A3	B3	C3	D3	E3	F3
	04	14	24	34	44	54	64	74	84	94	A4	B4	C4	D4	E4	F4
	05	15	25	35	45	55	65	75	85	95	A5	B5	C5	D5	E5	F5
	06	16	26	36	46	56	66	76	86	96	A6	B6	C6	D6	E6	F6
	07	17	27	37	47	57	67	77	87	97	A7	B7	C7	D7	E7	F7
	08	18	28	38	48	58	68	78	88	98	A8	B8	C8	D8	E8	F8
	09	19	29	39	49	59	69	79	89	99	A9	B9	C9	D9	E9	F9
	0A	1A	2A	3A	4A	5A	6A	7A	8A	9A	AA	BA	CA	DA	EA	FA
	0B	1B	2B	3B	4B	5B	6B	7B	8B	9B	AB	BB	CB	DB	EB	FB
	0C	1C	2C	3C	4C	5C	6C	7C	8C	9C	AC	BC	CC	DC	EC	FC
	0D	1D	2D	3D	4D	5D	6D	7D	8D	9D	AD	BD	CD	DD	ED	FD
	0E	1E	2E	3E	4E	5E	6E	7E	8E	9E	AE	BE	CE	DE	EE	FE
	0F	1F	2F	3F	4F	5F	6F	7F	8F	9F	AF	BF	CF	DF	EF	FF
									These Addresses Share UCW 00	**These Addresses Share UCW 01**	**These Addresses Share UCW 02**	**These Addresses Share UCW 03**	**These Addresses Share UCW 04**	**These Addresses Share UCW 05**	**These Addresses Share UCW 06**	**These Addresses Share UCW 07**
Column ID	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
16 UCW s, any other address up to 7F folds into A.		A	A	A	A	A	A	A	When there are 128 UCW s or less, the columns of addresses above are reserved for shared UCW addressing. The note in **bold type** relates the column of addresses to the correct UCW.							
32 UCW s, any other address up to 7F folds into A or B.			A	B	A	B	A	B								
64 UCW s, any other address up to 7F folds into A, B, C, or D.					A	B	C	D								
128 UCW s, no address folding.																
256 UCW s There are 256 unique UCW addresses - No folding permitted - No shared unit addresses - All valid unshared																

In Summary:

1. If shared subchannels are used, (128 UCWs or fewer), unshared subchannels are restricted to:
   a. UCWs in the 0 through 7 range that are *not* used as shared subchannels, and
   b. UCWs that are specified by unit addresses in which the high-order bit is a 0.
2. If shared subchannels are not used (as in a system with 256 UCWs) all addresses are valid and address a unique unshared UCW.

The number of subchannels used is also dependent upon the characteristics of the devices on the channel as pointed out in the following description.

The electrical characteristics of the channel allow for up to eight I/O control-unit positions, connected serially. Positions for units on the I/O interface can be thought of in three ways, depending upon the units involved:

1. A single control unit that controls one I/O unit can be connected to the channel. An example of this type of unit is the IBM 1443 Printer Model N1. One control-unit position is needed.
2. A single unit that contains several control units can be connected to the channel. An example is the 2821 control unit, which has separate control units for each attached 1403 Printer, one for the 2540 Reader, and one for the 2540 Punch. One control-unit position is needed.
3. A single control unit that services the requirements of several I/O units (one at a time) can be connected to the channel. An example is a tape-control unit that controls several tape drives. One control-unit position is needed.

Assume for example, that eight 1443 Model N1 printers are attached to the byte-multiplexer channel. All eight positions are then used, and no other unit can be attached to the channel. Any eight of the subchannels could then be used for the printers, according to the device address.

As a second example, assume that eight shared control units, each controlling eight I/O units, are connected to the byte-multiplexer channel:

1. Eight positions are required for the eight control units.
2. Eight UCWs (0-7) are required, one for each shared control unit.
3. No other positions are then available.

Note: The printer-keyboard does *not* use one of the eight positions on the byte-multiplexer channel, but does require a UCW.

FUNCTIONAL UNITS

Unit Control Word Format

The maximum number (256) of UCWs require 1,024 words (4,096 bytes) of control storage. The format of the UCW is:

Word	Address	Byte 0	1	2	3
1	xxx0	Flags & Ops	UCW/Chnl Status	Count High	Count Low
2	xxx4	Key	Data address		
3	xxx8	Key	Next CCW address		
4	xxxC	Dev Adr			

Word 1. Byte 0 is set up under microprogram control. As the operation proceeds, the microprogram can examine this byte to determine what function is to be performed. The meanings of the bits in byte 0 of word 1 are:

Bit	Value	Indicates
0	1	Data-Chaining (CDA)
1	1	Command chaining (CC)
2	1	Suppress-Length-Indication (SLI)
3*	1	SKIP or status next
4	1	Program-Controlled-Interruption (PCI)
5	1	Reserved
6*	1	Output (write to I/O)
	0	Input (read from I/O)
7	1	Decrement data address (for read backward operation)
	0	Increment data address (for write or read operations)

*Bits 3 and 6 specify:

Bits 3	6	Specify
0	0	Input
0	1	Output
1	0	Status next or SKIP
1	1	Stop or count zero

Byte 1 in word 1 contains UCW or channel status, depending upon the operation. Channel status is in this byte when a CSW-store operation (to store channel status) is to be performed. The channel status is moved to a multiplexer-channel working area (from byte 1 of the UCW) before the store-CSW operation is performed. (Note that unit status--channel end, unit check, etc--is not kept in a UCW. Unit status is placed in another area area of local storage when unit status is received during a multiplexer-channel share cycle.) The meanings of bits in byte 1 of word 1 are:

Bit	Value	UCW Status	Channel Status
0	1	Active subchannel	Program-controlled interruption
1	1	IL (Incorrect Length)	IL (Incorrect Length)
2	1	Program check	Program check
3	1	Protection check	Protection check
4	1*	Status queued	Channel data check
5	1	Channel control check	Channel control check
6	1	Interface control check	Interface control check

*When this byte is used for UCW status, bits 4 and 7 signify to the microprogram:

Bits 4, 7	Signifies
00	Handling data
01	Status next
10	Status stacked to control unit
11	Unit status in IB (Interrupt Buffer) in local storage

Bytes 2 and 3 of word 1 contain the data count for the operation in progress.

Word 2. Byte 0 of this word holds the CCW command when a CCW is loaded into local storage for execution. The ops (in byte 0 of word 1) are then set up from this CCW command code to designate the operation to be performed.

Next, the protect keys are moved into word 2, byte 0 from another local-storage location. As the operation proceeds, the keys in UCW word 2 are used for protection purposes as data is transferred to or from main storage for the I/O operation.

Word 2. Bytes 1, 2, and 3 contain the data address of the main-storage area from/to which data, if any, is to be transferred during the operation. If fetch protection is specified, both the read-byte and the store-byte operations are protected. If storage protection without fetch protection is used, only the store-byte operation is protected.

Word 3. Byte 0 of this word contains the protect keys.

Bytes 1, 2, and 3 of Word 3 contain the last CCW address +8. Note that because word 3 contains the protect key in byte 0, CCWs accessed from main storage (by use of a read-word storage-word operation) are storage-protected (fetch protect only). This is the same key used for data transfer to/from main storage during an I/O operation. Therefore, the main-storage area from which CCWs are accessed must have the same protect-key value as the area to which or from which data is transferred, but only if the CCWs are fetch-protected.

Word 4. Byte 0 contains the device address when the UCW is active. Bytes 1-3 are spares.

External Facilities

The external facilities that are used for multiplexer-channel operations are assigned external word address 0E. The microprogram communicates with the channel circuitry via word name MPX, or byte names MTO, MTI, MBO, and MBI. The bytes MTI and MBI cannot be used as a destination.

Word MPX	Byte 0 MTO	Byte 1 MTI	Byte 2 MBI	Byte 3 MBO
	0 Operational-out	0 Operational-in	B	B
	1 Select-out	1 Address-in	U	U
	2 Address-out	*2 Status-in signal	S	S
	3 Command-out	*3 Service-in signal		
	4 Service-out	4 Select-in	I	O
	5 Interrupt	5 MPX-request in	N	U
	6 Suppress-out	6 MPX or Cons Req In		T
	7 MPX check	7 Disconnect-in		

Note:
Bits 2 and 3 of MTI are logical functions of the interface 'service-in' and 'status-in' lines.

*Bits 2-3
00 = Op-in up
01 = Service-in up
10 = Status-in up
11 = Operational-in down

MTO (Multiplexer Tags-Out)

The MTO register consists of eight latches that are fed from the external bus-in byte 3. Byte 0 of the MPX word allows the microprogram to communicate with the register.

The state of any of the seven multiplexer tags-out can be tested by the Branch and Module Switch word. The MTO register can be set or reset by the Branch word.

MTI (Multiplexer Tags-In)

The MTI register consists of eight polarity holds that are fed from the interface receivers. The register may also carry information gated from the bus-out register during channel-diagnostic operations. Byte 1 of the MPX word allows microprogram communication with the register.

The status of any of the MTI register bits can be tested by the Branch or the Branch and Module Switch words. The MTI external cannot be used as a destination (tags-in cannot be set or reset by the microprogram).

MBI (Multiplexer Bus-In)

The MBI external facility is not an actual register. It consists of the nine lines that carry the information from the channel interface. The information may be either data, address, or status.

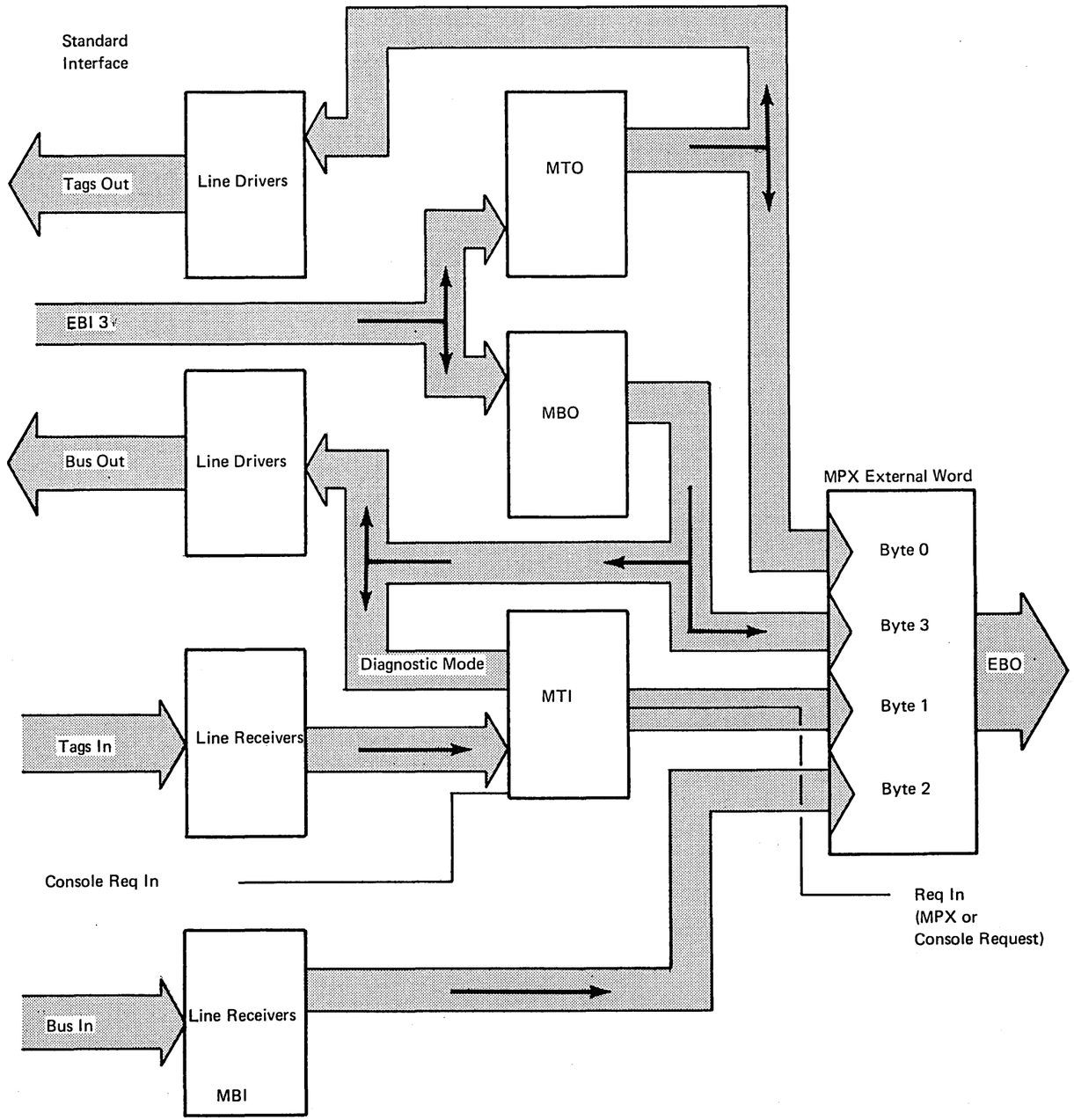
The information on the bus-in lines can be read by the arithmetic word type 10 or 11. The MBI cannot be used as a destination.

Refer to "Arithmetic Word Types" in Chapter 4 for a description of the ABCK function as it relates to parity errors.

MBO (Multiplexer Bus-Out)

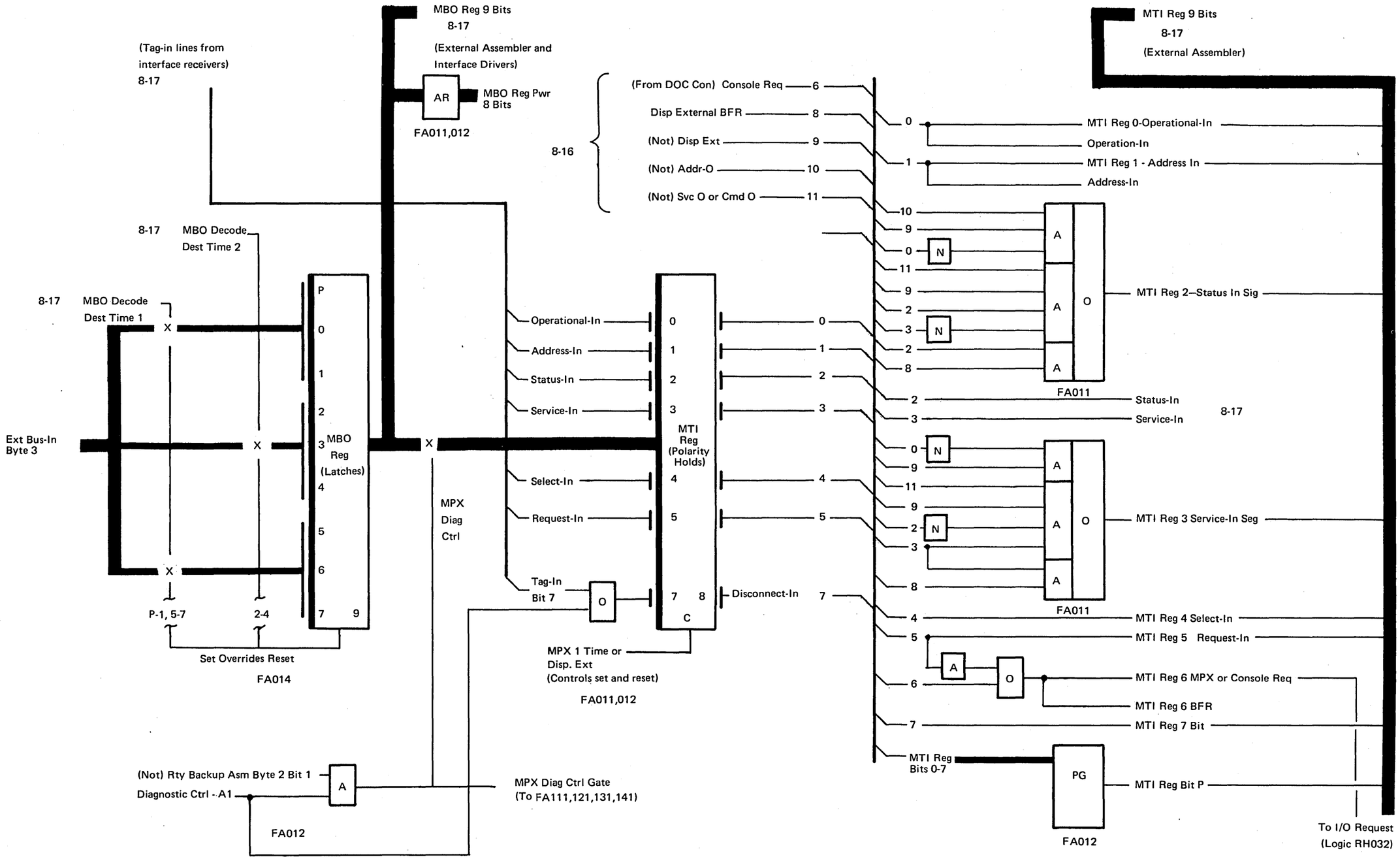
The MBO register consists of nine latches that are fed by external bus-in byte 3. The information in the register is made available to the I/O device as input (address, command, or data) or may be gated to the MTI register when in diagnostic mode.

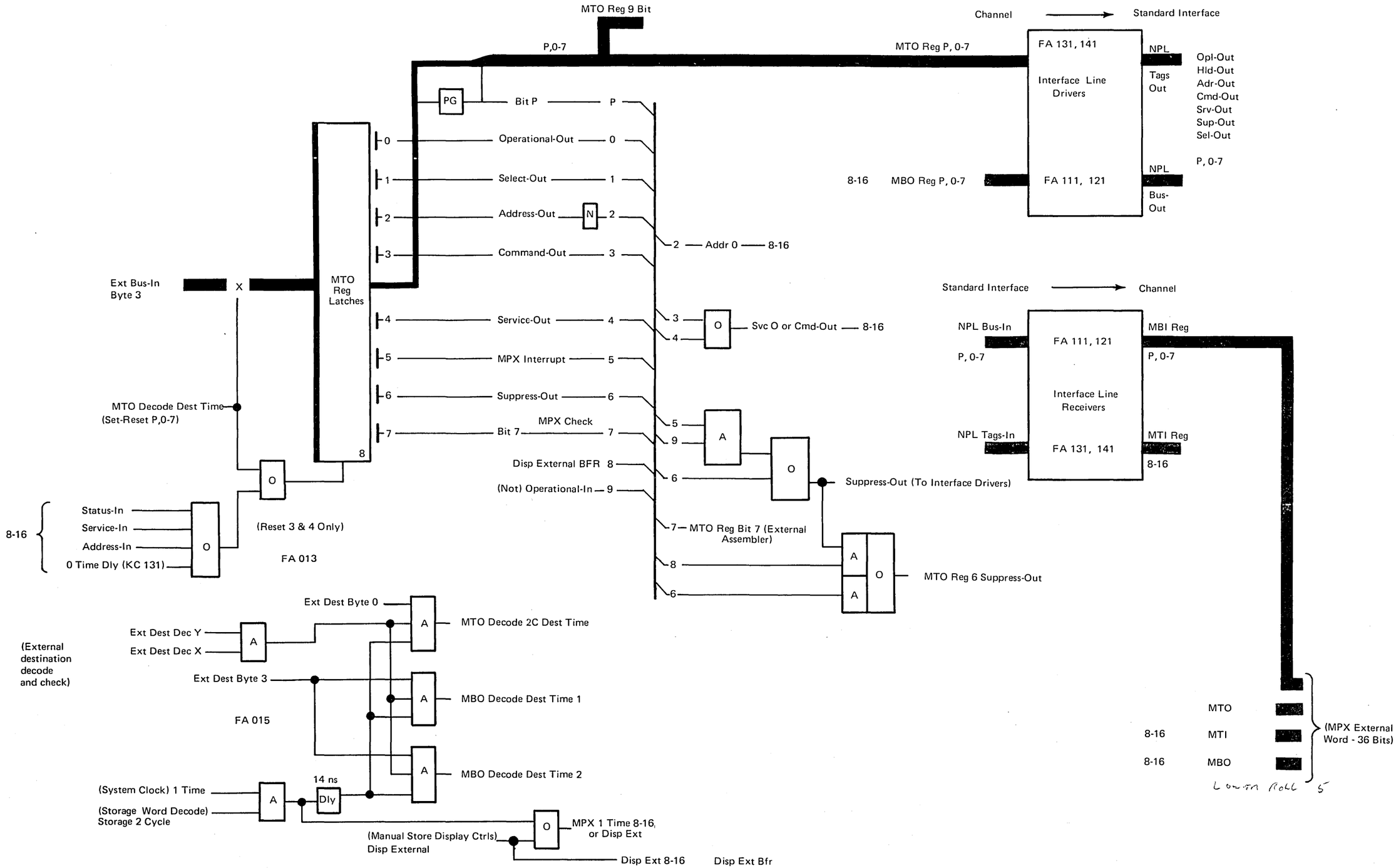
Byte 3 of the MPX word allows communication between the microprogram and the MBO-register. The MBO register is set with an address or command by the arithmetic word type 10. Data is read from the register to storage by a storage word. The register is reset by the arithmetic word type 11.



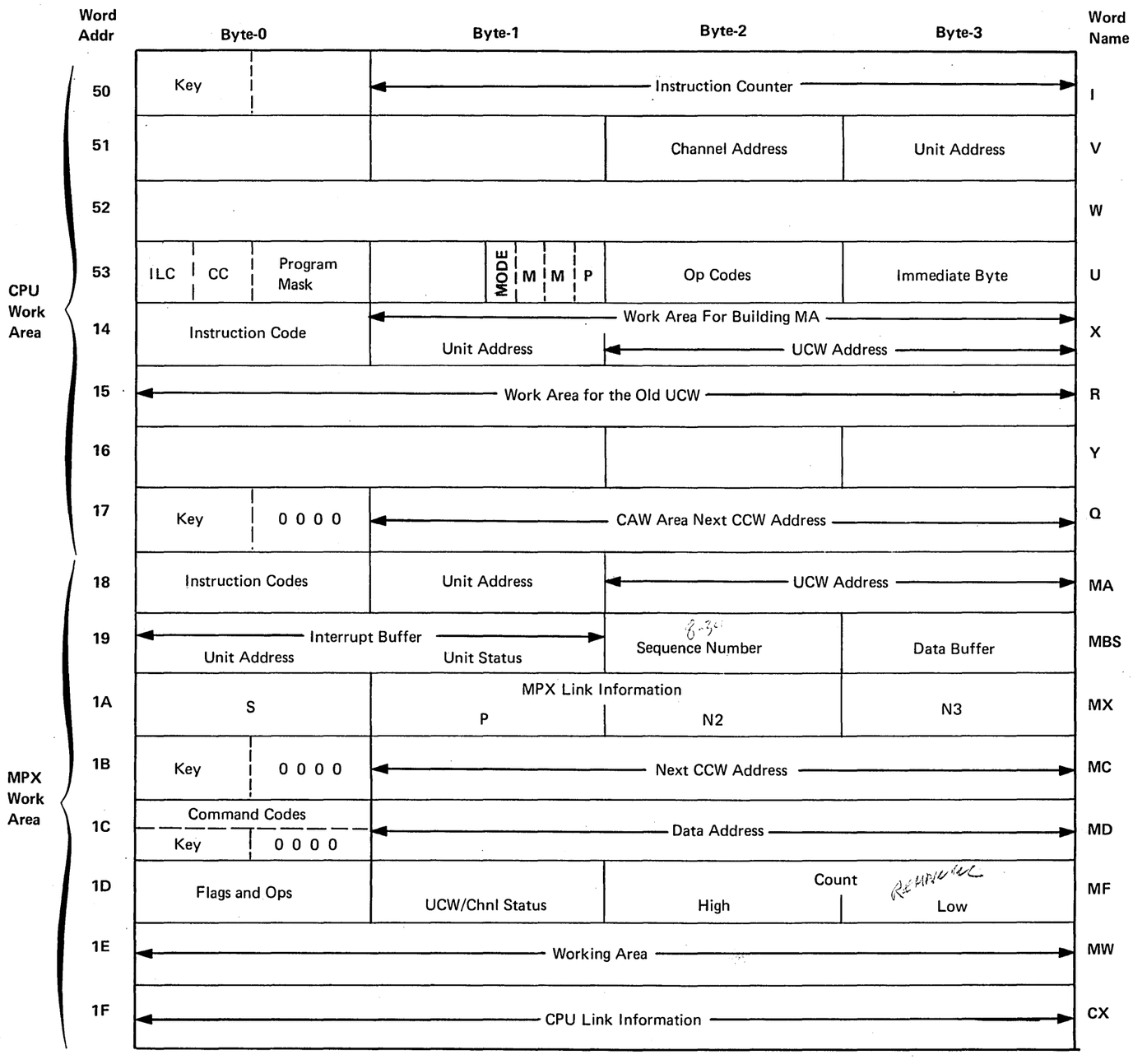
Second-Level Diagrams

MBO, MTI




MTO Reg 9 Bit
P,0-7
MTO Reg P, 0-7
Channel
Standard Interface
FA 131, 141
Interface Line Drivers
NPL
Tags Out
Opl-Out
Hld-Out
Adr-Out
Cmd-Out
Srv-Out
Sup-Out
Sel-Out
P, 0-7
NPL
Bus-Out
8-16
MBO Reg P, 0-7
FA 111, 121
PG
Bit P
P
0
Operational-Out
0
1
Select-Out
1
2
Address-Out
N
2
3
Command-Out
3
2
Addr 0
8-16
Ext Bus-In Byte 3
X
MTO Reg Latches
4
Service-Out
4
3
4
O
Svc O or Cmd-Out
8-16
5
MPX Interrupt
5
6
Suppress-Out
6
MTO Decode Dest Time (Set-Reset P,0-7)
MPX Check
7
Bit 7
7
8
Disp External BFR
8
5
9
A
6
O
Suppress-Out (To Interface Drivers)
(Not) Operational-In
9
7
MTO Reg Bit 7 (External Assembler)
8
A
6
A
O
MTO Reg 6 Suppress-Out
O
Status-In
Service-In
Address-In
0 Time Dly (KC 131)
8-16
(Reset 3 & 4 Only)
FA 013
Standard Interface
Channel
NPL Bus-In
P, 0-7
FA 111, 121
MBI Reg
P, 0-7
Interface Line Receivers
NPL Tags-In
FA 131, 141
MTI Reg
8-16
Ext Dest Byte 0
Ext Dest Dec Y
Ext Dest Dec X
A
MTO Decode 2C Dest Time
(External destination decode and check)
Ext Dest Byte 3
A
MBO Decode Dest Time 1
FA 015
A
MBO Decode Dest Time 2
14 ns
(System Clock) 1 Time
(Storage Word Decode) Storage 2 Cycle
A
Dly
O
MPX 1 Time 8-16, or Disp Ext
(Manual Store Display Ctrls) Disp External
Disp Ext 8-16
Disp Ext Bfr
MTO
8-16
MTI
8-16
MBO
(MPX External Word - 36 Bits)

Local Storage



During byte-multiplexer channel operations, the information from the UCW identified by the address of the I/O unit involved in the operation is read from control storage and placed in local storage by the microprogram.

Local storage is addressable by word name. The areas of local storage shown are either designated for byte-multiplexer channel operations or are classified as CPU work areas and are used during channel operations as shown.

Flags & Ops Byte MF0 — Byte Details

Bit	Value	Indicates
0	1	Data-Chaining (CDA)
1	1	Comman-Chaining (CC)
2	1	Suppress-Length-Indication (SLI)
3*	1	Skip or Status Next
4	1	Program-Controlled-Interruption (PCI)
5	1	Not Used
6*	1	Output (write to I/O)
	0	Input (read from I/O)
7	1	Decrement data address (for read-backward operation)
	0	Increment data address (for write or read operation

*Bits 3 and 6 are used under microcode control to provide additional information as follows:

3	6	Specify
0	0	Input
0	1	Output
1	0	Skip or Status Next
1	1	Stop (Zero Count)

Instruction Codes † Byte X0/MA0

Bit	Value	Indicates
0	1	Channel Loaded
1	1	Note 1
2	1	Chaining
3	1	Share Request
4	1	Interrupt
5	1	Halt I/O (HIO)
6	1	Test I/O (TIO)
7*	1	Start I/O (SIO)

†Developed in microroutines.
*Bit 7 (SIO) is reset when good status is received on initial selection.

Note 1: Byte X0/MA0; Bit 1: Used during initial selection to indicate an immediate control command. Used during chaining to keep track of the operation (Ex= 1st CCW indicator)

Byte MF1 (from 1st UCW word) +

UCW Status

Bit	Value	Indicates
0	1	Active
1	1	Incorrect Length (IL)
2	1	Program Check
3	1	Protection Check
4*	1	Status Queued
5	1	Channel Control Check
6	1	Interface Control Check
7	1	Status Next

Channel Status

Bit	Value	Indicates
0	1	PCI
1	1	IL
2	1	Program Check
3	1	Protection Check
4	1	Channel Data Check
5	1	Channel Control Check
6	1	Interface Control Check
7		Spare

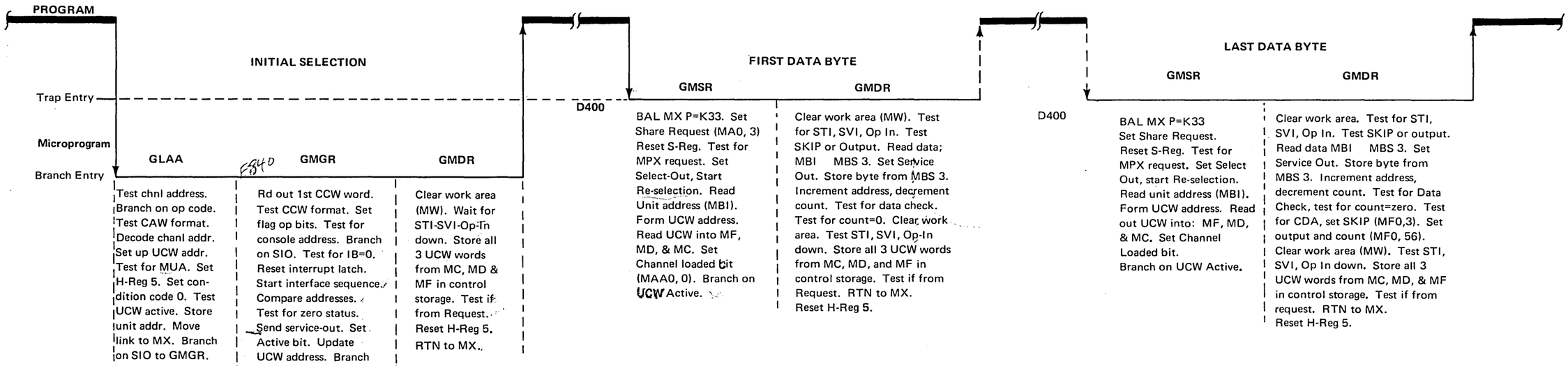
+MF1 indicates UCW or channel status depending on the operation. During I/O initialization, MPX interrupt, etc., it indicates UCW status. Channel status is in this byte when a CSW-store operation is to be performed.

*When this byte is used for UCW status, bits 4 and 7 signify to the microprogram:

4	7	Indicates
0	0	Handling data
0	0	Status next
1	0	Status stacked to control unit
1	1	Unit status in interrupt buffer

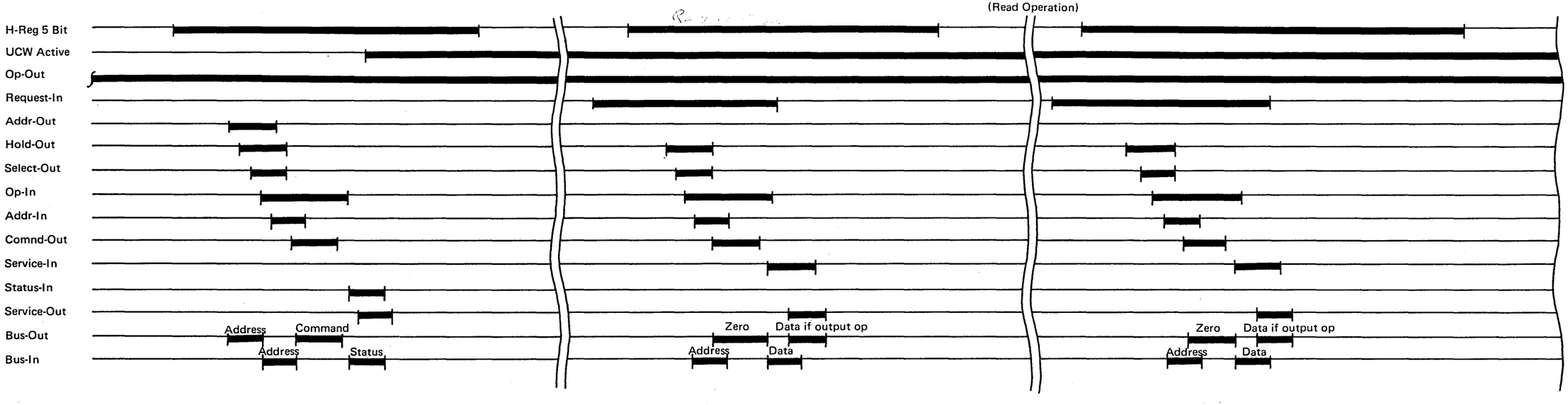
OPERATIONS

Operational Overview

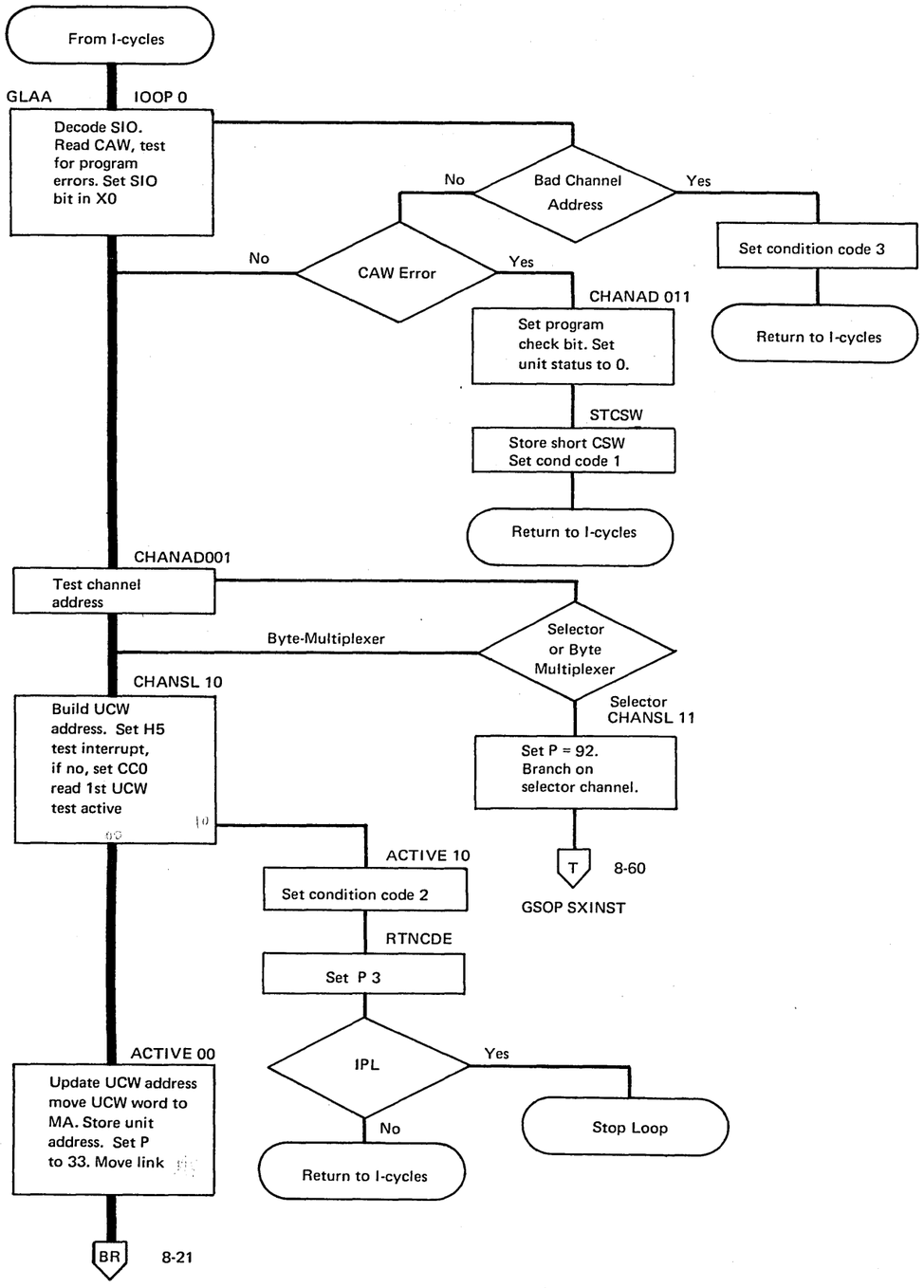


SEE 6-14

UCW Word 1 LS1B - MC	Key	0 0	Next CCW Address	
UCW Word 2 LS1C - MD	Command			
UCW Word 3 LS1D - MF	Key	0 0	Data Address	
	Flag/OP's		UCW/CHNL Stat	Count-HI	Count-LO



Start I/O (Part 1 of 4)



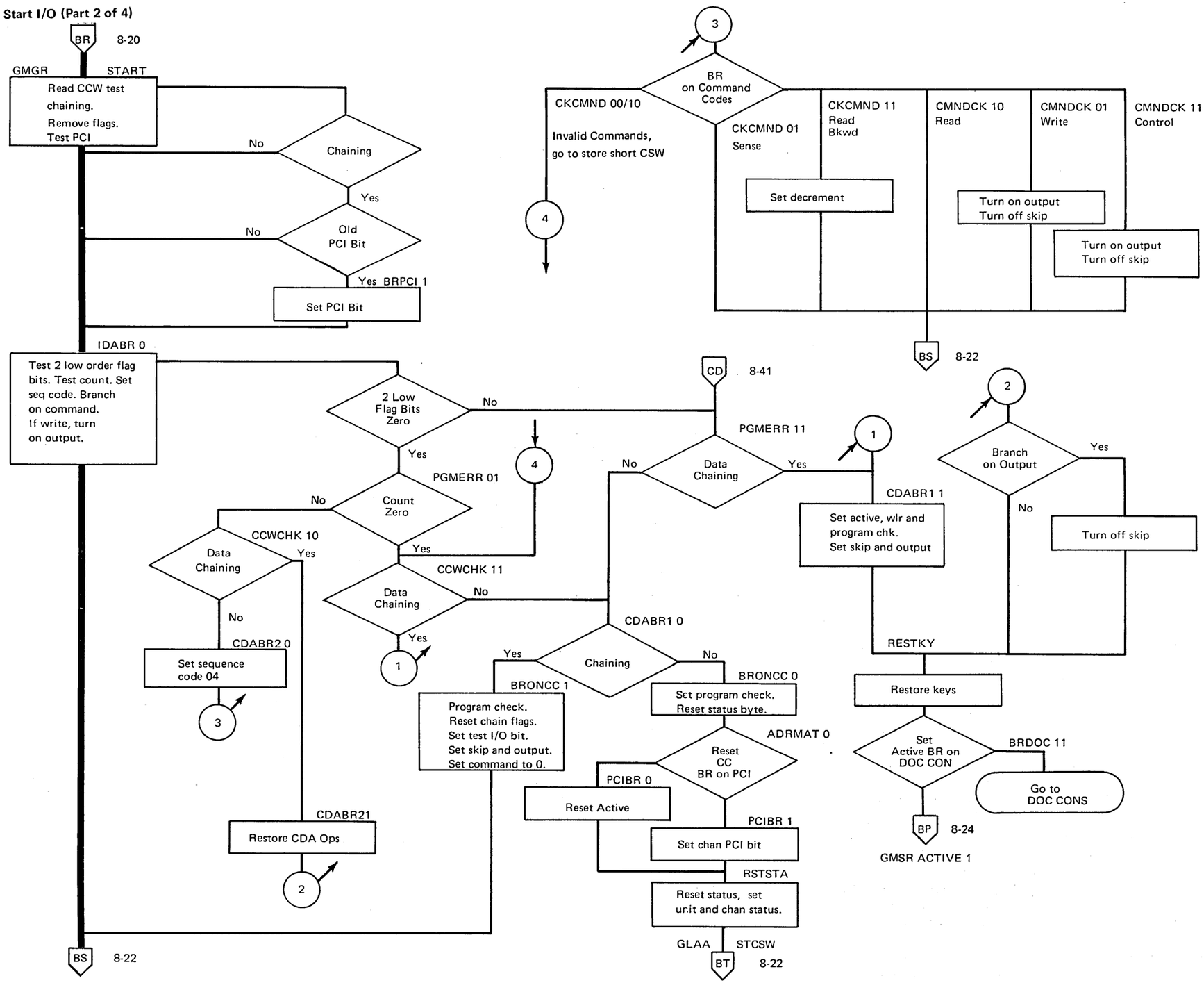
Start I/O

- The system must be in supervisor state for all I/O operations.
- The initial selection routine places the I/O device in service on the channel.
- A trap routine handles each data byte (in byte mode).

All I/O instructions are decoded in the I-cycles (GAIC) routine and enter a common I/O decode (GLAA) routine. In GAIC, the processor tests whether the system is in the supervisor state. (If not, the GLAA IOOP 1 entry point is used, causing a branch to the common interrupt routine.) The GLAA routine determines the channel involved and enters the CAW data into local storage. The CAW is analyzed to determine the address of the first CCW. The UCW address is now constructed in the X-register, and the unit address is checked for shared subchannels (multiunit addresses). If the address is valid, condition code 0 is set and H5 is set to block other trap requests.

Next, the first UCW word is read into the R-register, and the active bit is tested. Active bit on means that the subchannel is in operation; thus is unavailable. Assume that the active bit is not on — update the UCW, move the first UCW word into MA (in LS), restore the unit address to the X-register and branch to GMGR (the general routine).

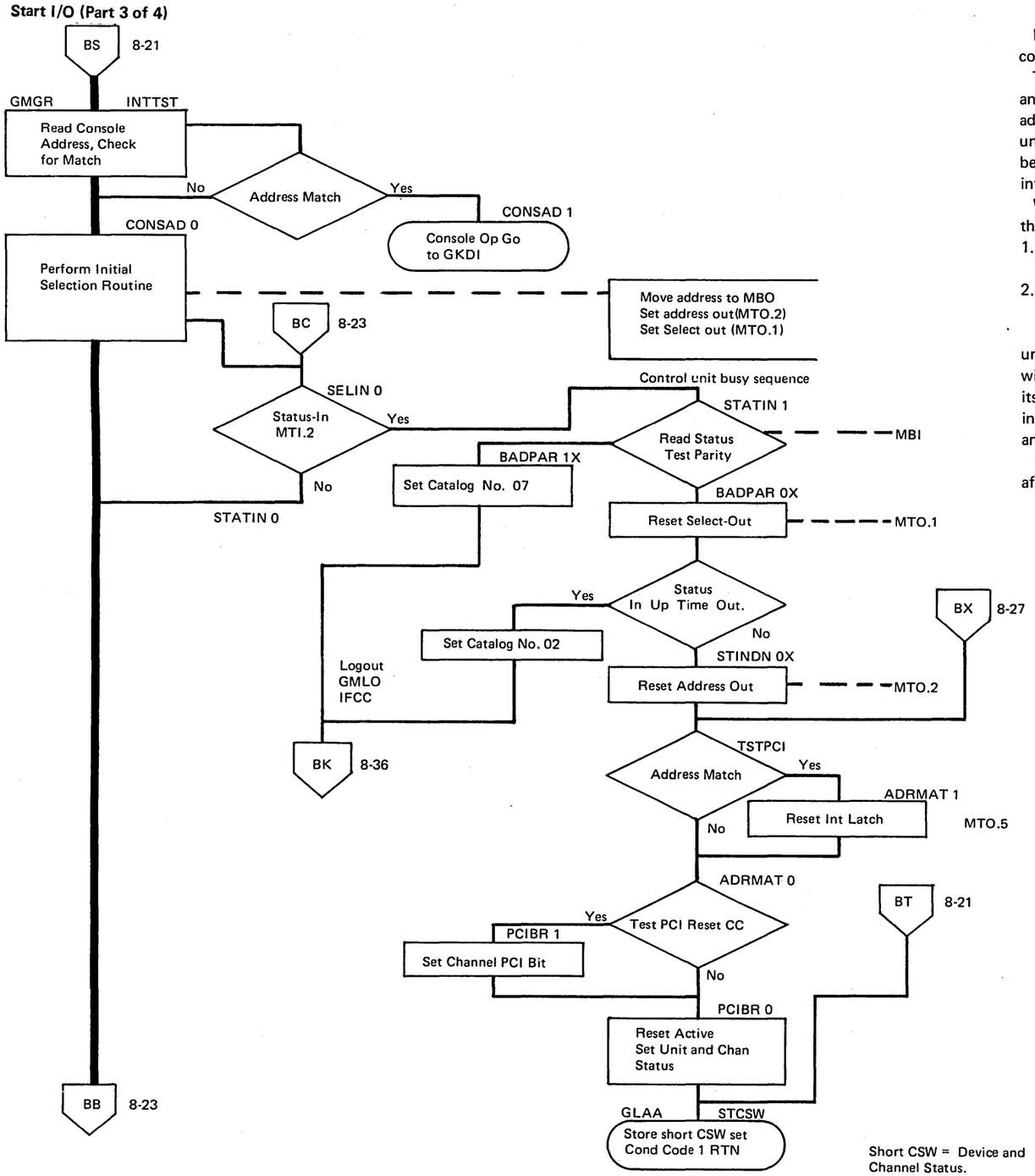
Start I/O (Part 2 of 4)



The first function of the GMGR routine is to fetch and analyze the CCW at the address specified by the CAW. The following information is checked in the CCW.

1. The two low-order bits of the flag byte are tested to ensure that they are zero.
2. The initial byte count is tested for nonzero state.
3. Test for (not) a TIC during a SIO.
4. Test for invalid command.
5. Ensure that a TIC is to a CCW on a doubleword boundary.
6. Ensure that a TIC is not to a second TIC.

When it is determined that all the foregoing conditions are satisfied, a test is made to determine the type of command involved and whether a console operation is being performed.



Short CSW = Device and Channel Status.

Initial selection is performed next, placing the I/O device concerned in service on the interface.

The channel begins the initial selection sequence by transmitting an address byte on the 'bus-out' lines and raising 'address-out'. The address byte selects the unit to execute the operation. Each control unit attached to the interface attempts to decode the address; but, because all interface addresses are different, only one unit can interpret the coded byte.

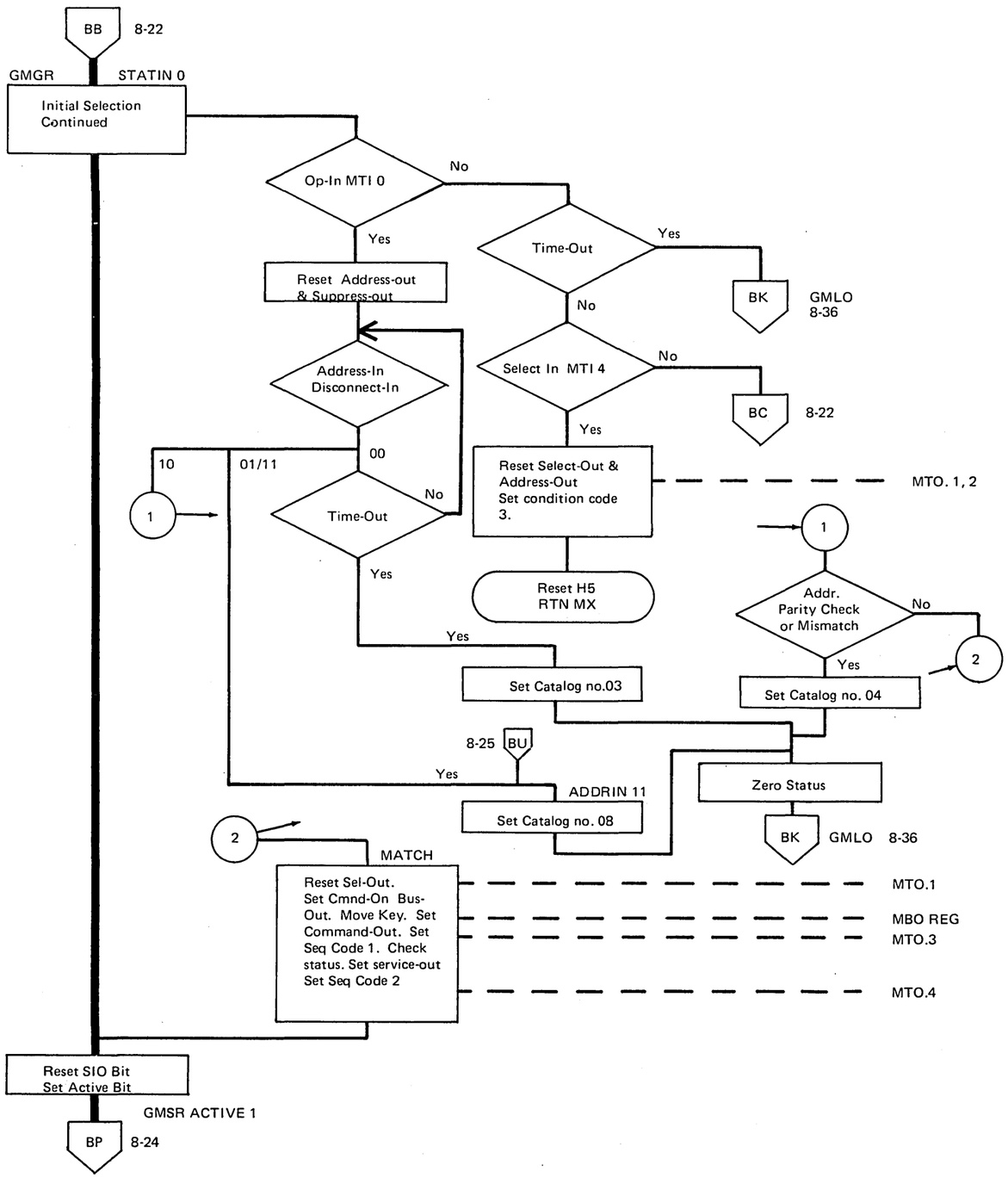
When 'select-out' rises at the control unit that successfully decodes the address byte, that control unit conditions either:

1. 'Status-in', indicating that the selected unit is busy and cannot execute another operation, or
2. 'Operational-in', indicating that the designated unit will complete the initial selection sequence.

If no control unit decodes the address byte (specified control unit is offline; the address byte is invalid, etc.), the control unit with the lowest priority propagates 'select-in' to the channel when its incoming 'select-out' is conditioned. The 'select-in' or 'status-in' reply to 'address-out' causes the channel to drop 'address-out' and terminate the selection sequence.

If 'status-in' was raised, 'select-out' drops and 'address-out' drops after 'status-in' drops.

Start I/O (Part 4 of 4)



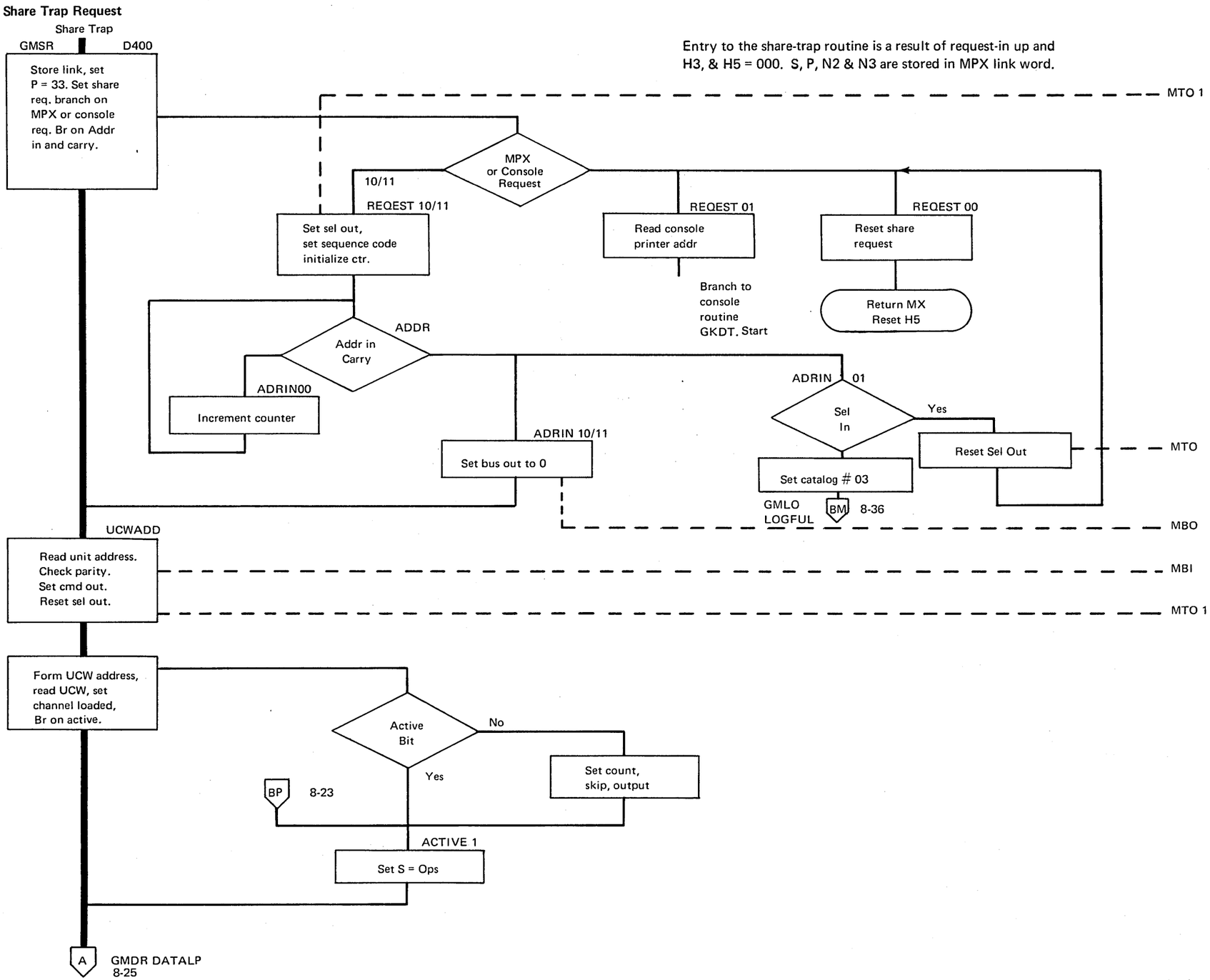
When 'operational-in' causes the channel to drop 'address-out', the selected control unit then transmits an address byte on 'bus-in' lines and conditions the 'address-in' line. To ensure that the right device has been selected, the channel compares this address to the address it placed on the 'bus-out' lines.

After checking the address, the channel responds to 'address-in' by transmitting a command byte and conditioning 'command-out' to the control unit. The command byte designates one of seven operations (read, read-backward, write, control, sense, test I/O, or the No-Op special control), and establishes conditions to control execution of the operation.

The control unit must then drop 'address-in'; and after 'command-out' falls, the control unit places its status information on 'bus-in' and raises 'status-in'. If the I/O device is available and capable of executing the command, the status byte is zero. If the channel accepts this status byte, it responds with 'service-out'. This signal completes the initial selection sequence.

If the selection is completed successfully, the key is moved from MD to MC in local storage, the active bit is turned on, UCW address is updated and moved to the working area and stored back in control storage, and the Op bits are reset.

The microroutine now exits from GMGR to GMSR to handle disconnecting from the channel, if byte-mode operation; or handling data, if burst-mode operation. For byte mode, the microprogram returns to I-cycles so that other functions may be performed.



When the first data byte is ready for transfer, 'request-in' is developed and the microprogram traps to GMSR — the share-trap routine. This routine handles reselection, and then exits to the GMDR routine to handle the byte of data.

Share Trap — Data Handling

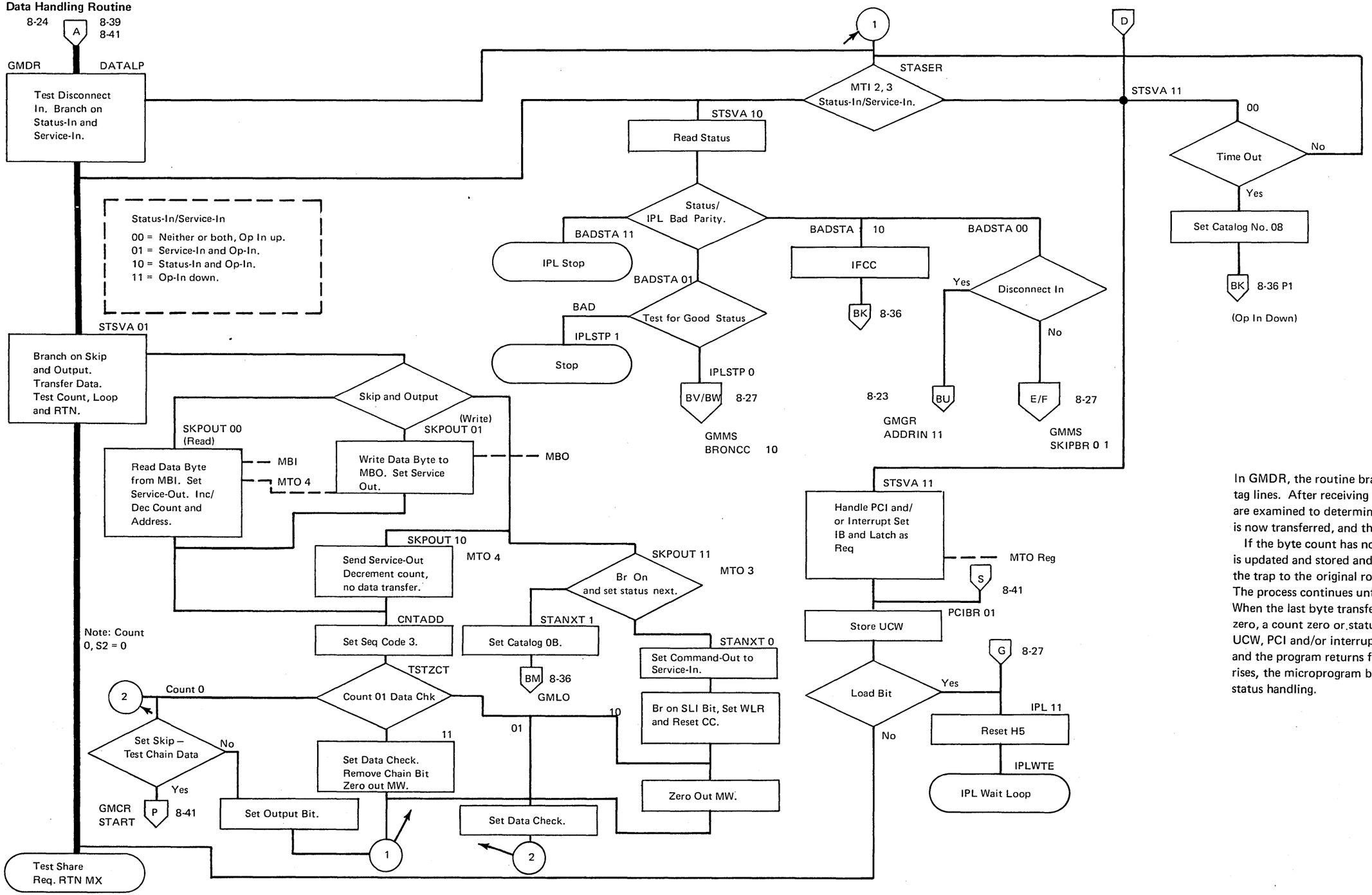
- A share-trap routine handles each data byte.
- When byte count is reduced to zero, data transfer ends.

When the first data byte is ready for transfer, 'request-in' is developed and the microprogram traps to the GMSR (share trap) routine. Before the byte of data is transferred, the following functions are handled:

1. Store link information, set share request bit.
2. Test for console request or MPX.
3. Reselect the device.
4. Develop UCW address and read UCW into local storage.
5. Test active bit.

The microprogram now branches to the GMDR routine to handle the byte of data.

Data Handling Routine



In GMDR, the routine branches on 'status-in' and 'service-in' tag lines. After receiving 'service-in', the skip and output bits are examined to determine the operation. The byte of data is now transferred, and the byte count is modified.

If the byte count has not been reduced to zero, the UCW is updated and stored and the microprogram returns from the trap to the original routine to await another share trap. The process continues until all the data has been handled. When the last byte transferred reduces the byte count to zero, a count zero or status next condition is set in the UCW, PCI and/or interrupt are handled, the UCW is stored, and the program returns from the trap. When 'status-in' rises, the microprogram branches to GMMS to initiate status handling.

Ending Status and Interruption Handling

PROGRAM

Microprogram

Trap D400

STATUS BYTE (Channel End)

GMSR

Store link. Set P=33. Set Share Request. Reset S-Register. Test for MPX request. Set Select Out, start Re-selection. Read unit address. Form UCW address. Read out UCW into MF, MD, & MC. Set channel loaded. Branch on UCW active.

GMDR

Clear work area. Test STI, SVI, & Op In. Branch on STI/SVI Read status into MBS3. Test for bad status. Test for count 0 indication Branch to GMMS.

GMMS

Test for bad status. Test for Cmd Chain. Test for IPL. Set Addr in MD1. Test IB empty. Set interrupt latch (MTO,5). Set unit address in interrupt buffer. Test UCW active. Set status indicators (MF1 bit 4 & 7). Set status in interrupt buffer (MBS3 MBS1). Set service-out. Set count zero. Branch on Op-In down. Go store UCW.

GMDR

Store all 3 UCW words from MC, MD. & MF into control storage. Test if from request. Rtn to MX. Reset H5.

GAAI

Interruption Request

HANDLE INTERRUPTION REQUEST

GICM

(P=52) Start interruption priority sequence. Test timer. Test I/O or external. Test for selector-channel interrupt. Assume multiplexer channel interruption Set unit address in X0, X3. Set channel address. Branch to GLAA IENTRY

SET UP CSW

GLAA

Set interrupt bit. Set unit and channel address into V-Register Test for MUA. Set up UCW address in X2 and X3. Set H-Reg 5. Read 1st UCW. Test active bit. Move word X into MA. Store unit address. Move link to MX. Move 1st UCW word to MF. Read 2nd UCW word into MD. Read 3rd UCW word into MC. Test for logout. Move IB status into MBS 3. Branch to GMSW START.

GMSW

Reset interrupt latch. Clear MBS 1 and MW. Test active. Test status queued. Restore UCW address. Remove status bits. Reset active bit and store UCW. Move unit status, store CSW. Test for logout. Move extended status. Test EC mode, test interrupt. Reset H-Reg. 5 bit. Go load PSW (GICM, GIPW).

PSW HANDLING

GICM

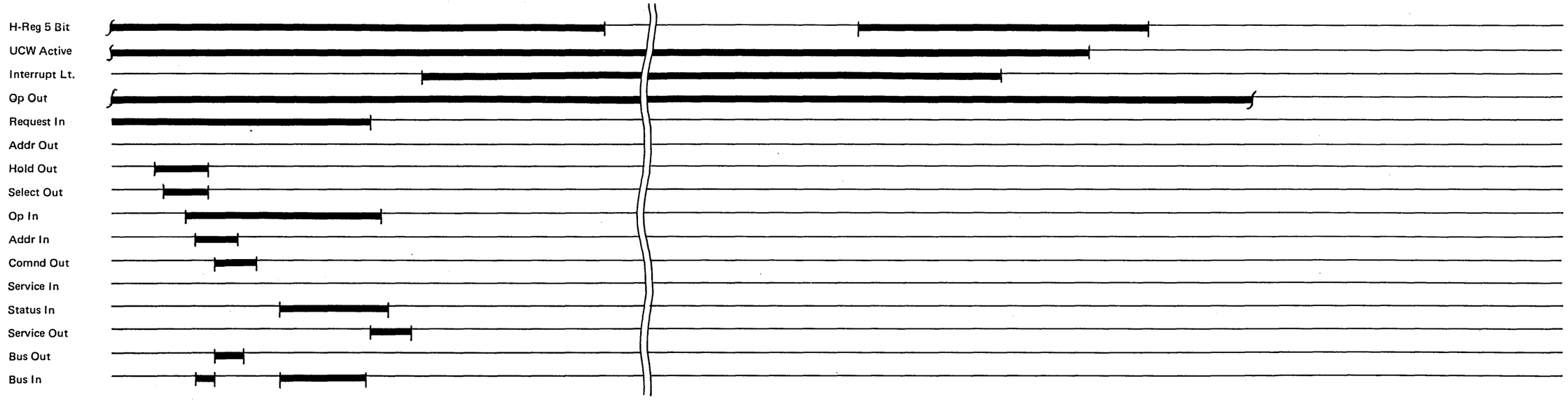
Store PSW in I/O Old PSW location. Set up address of I/O Old PSW location. Branch to GIPW.

GIPW

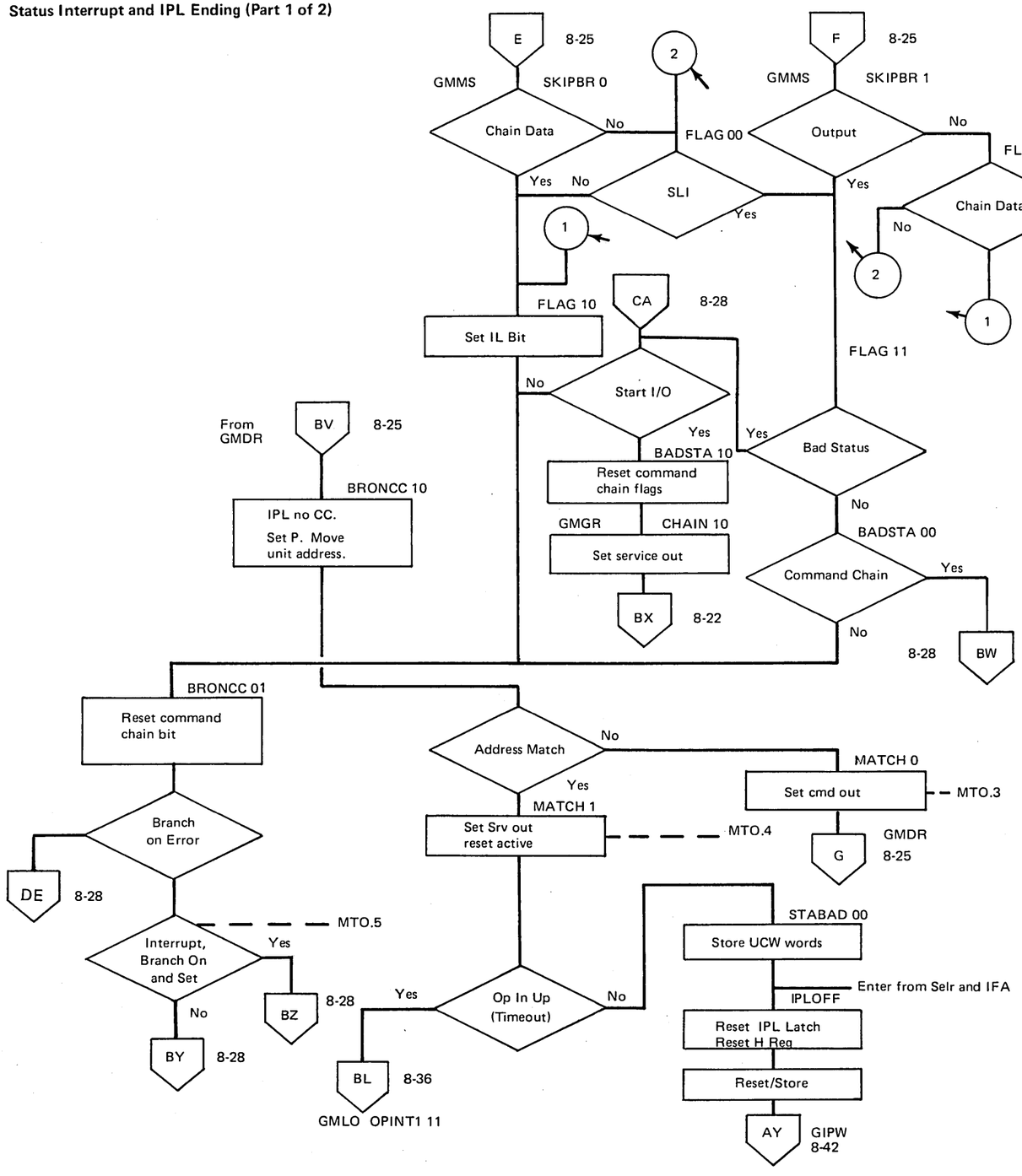
Assembly and store I/O Old PSW. Increment address in V-Reg for Input/Output new PSW (78) location. Read I/O new PSW double-word. Load new PSW (mask, OMWP CC, key, etc.) Reset ILC. Load key for 2nd op and 1st op. Test for Machine Check, Interrupt Pending and Wait State. No Wait, RTN LNK to GAAI.

Start sub-routine for test CSW

GAAI



Status Interrupt and IPL Ending (Part 1 of 2)



Ending Procedure

- Channel-end signals end of data handling.
- Device-end signals end of command.
- Ending status may be stored or queued.

An I/O control unit signals the channel that it has executed the data transfer portion of a CCW command by sending channel-end. Sending device-end signals the channel when the selected I/O device has fully executed the CCW command. Channel-end (CE) and device-end (DE) are status conditions and are identified as such by the 'status-in' tag line.

Channel-end and device-end status may occur in the same status byte or in separate status bytes depending on the CCW command and the I/O unit involved.

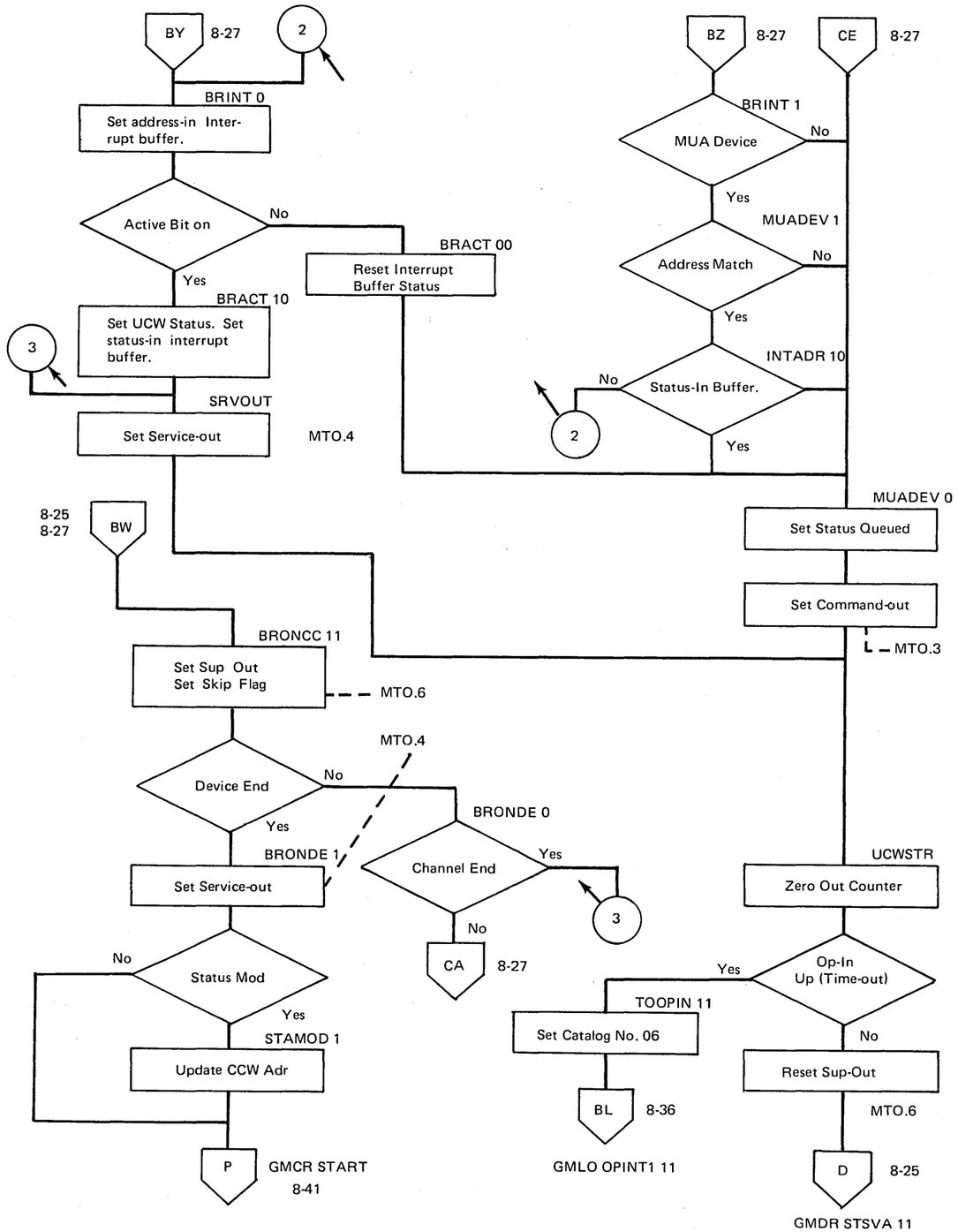
A control unit may present ending status at three different times:

1. During initial selection.
2. Immediately following a data-handling sequence.
3. By raising 'request-in'.

When the channel receives ending status, it may either request the CPU to interrupt the current instruction sequence and store the required information in the CSW, or it may obtain a new CCW and start its execution. This depends on the UCW flag bits and the type of status the channel receives.

Consider normal ending status for an input operation. Assume that the channel receives 'request-in' causing a share trap and normal reselection. The control unit then presents 'status-in'. Refer to the GMDR routine and assume 0-count (count field equal to data bytes) giving a branch to the GMMS routine at the SKIPBR 1 entry point.

Status Interrupt and IPL Ending (Part 2 of 2)



The status is read from bus-in (MBI) and tested for bad status (any bits on other than status modifier, CE, or DE). Assume not CC or IPL. The interrupt bit is tested in the Ops and flags byte of the UCW. If the interrupt bit is on (interrupt buffer full) and it is not a multiunit address, the routine activates 'command-out' to indicate to the control unit that its status byte was not accepted. When a control unit receives 'command-out' it queues or stacks its status (that is, it holds the status in its status register) and deactivates 'operational-in'.

Assuming that the interrupt bit is not on (IB is not full), the microprogram turns on the interrupt bit, places the unit address in the IB (MBS0) in local storage, and tests the active bit. For this example, assume that the UCW is active, meaning that the status now being presented is the first status received from the I/O unit for the command sequence it is presently executing. In this case, the microprogram places the status in the buffer and activates 'service-out'. When 'operation-in' falls, the program branches to the GMDR routine to store the UCW, and returns to the MPX trapped address. The CPU handles the interrupt request when it has the opportunity to do so.

When the UCW is not active, this means that the unit is presenting status a second time, normally device-end status. In this case, the microprogram sets the IB status to 0 and sets the status-next and status-queued bits; then activates 'command-out'. When 'operational-in' falls, the program branches to the GMDR routine, stores the UCW and returns to I-cycles. The CPU handles the interrupt when it has an opportunity to do so. Refer to the next section for details of MPX channel interruptions.

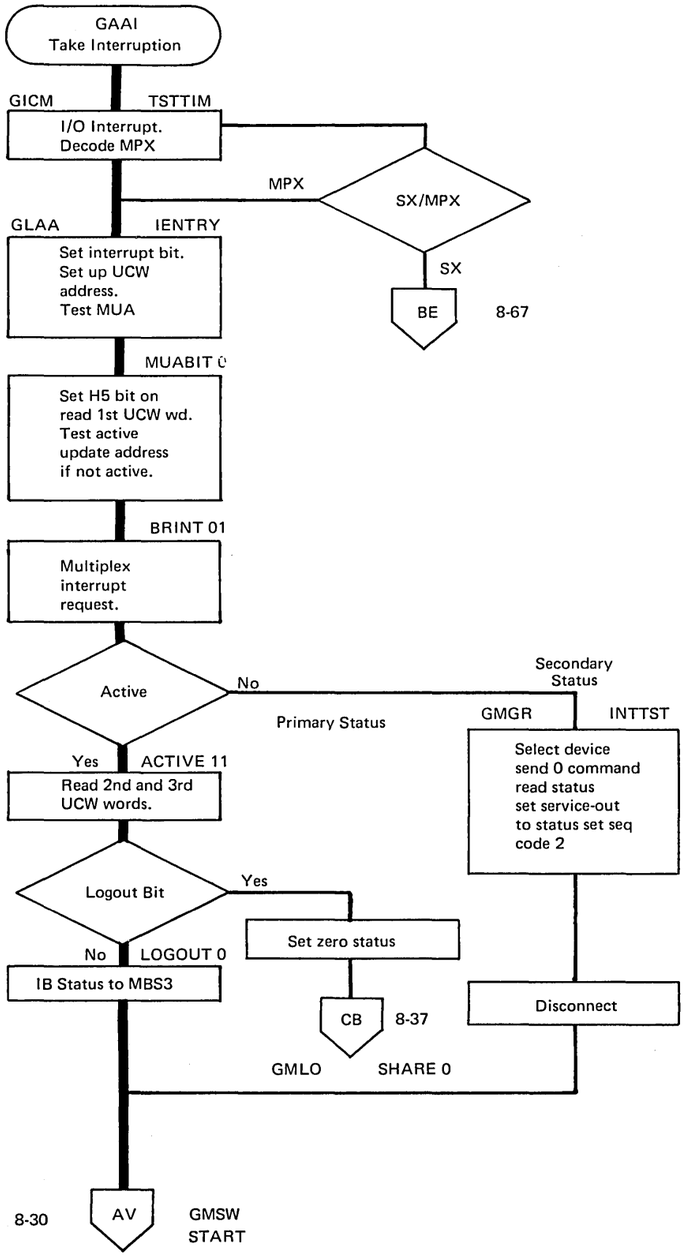
Byte-Multiplexer-Channel Interruptions

- A byte-multiplexer-channel interruption can be caused:
  1. At the normal end of an operation for status presentation.
  2. If the PCI bit is on in an active subchannel.
  3. If an error occurs during an I/O operation.
  4. When a device presents unsolicited status that is accepted by the channel (such as not-ready to ready device-end).
- An interruption buffer (IB) is provided in local storage for MPX channel interruptions.
- There are two types of Status:
  a. Primary: normal channel end.
  b. Secondary: normal device end.

Normal Primary Ending Status Interruption

The MPX channel initiates a normal ending status interruption when it receives ending status from a control unit (and operations are not command-chained). It does this by turning on the MPX channel interrupt latch during the GMMS (main status handling) routine.

Byte-Multiplexer Channel Interruption



The CPU cannot honor an interrupt request until it has completed the current instruction. Execution of the interrupt operation begins in I-cycles (GAAI) when an interrupt is detected, and then branches to the common interrupt routine (GICM), where the interrupt is identified as being on the MPX channel. The microprogram then branches to the I/O routine (GLAA) where the following objectives are accomplished:

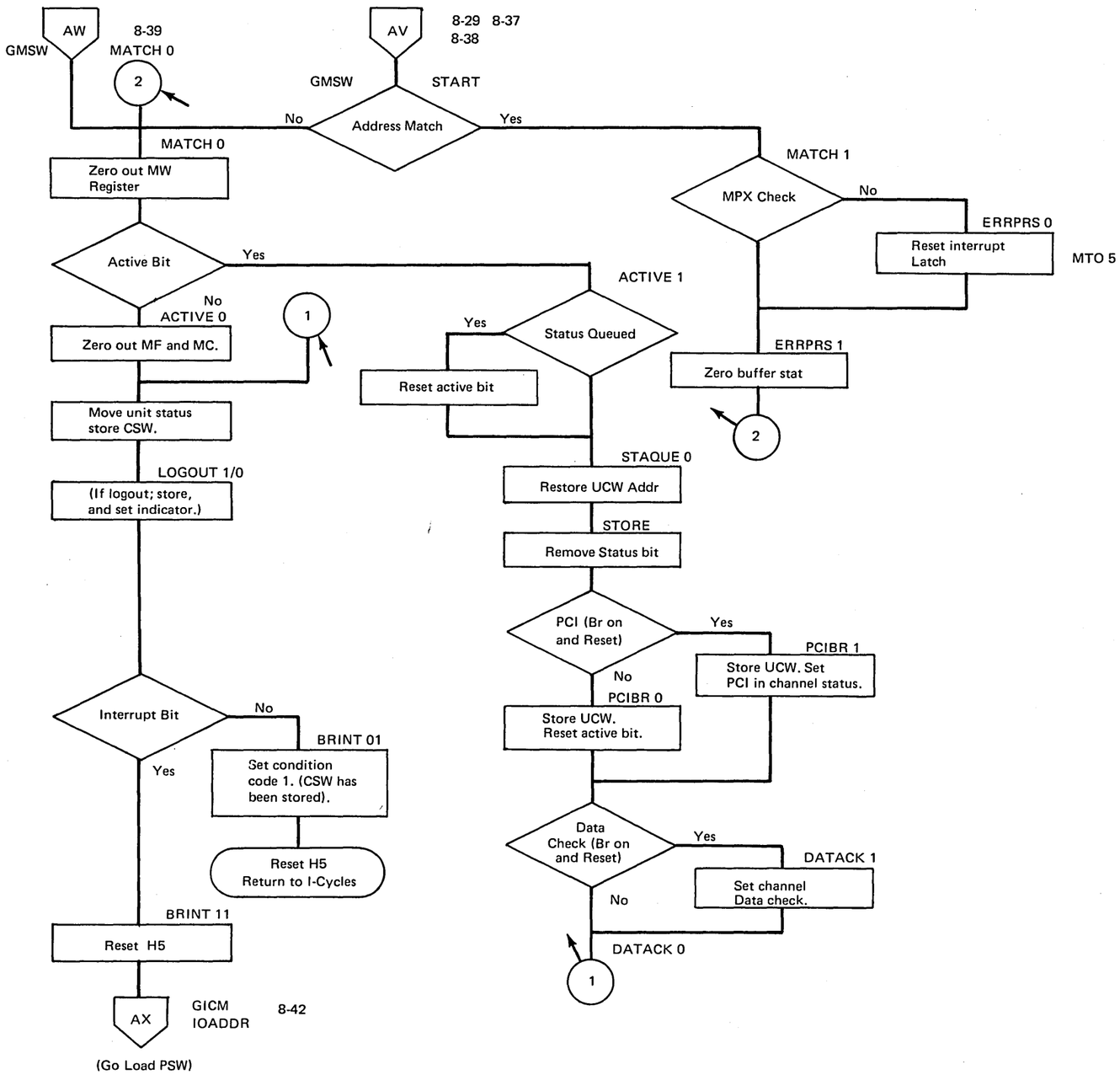
1. Set the H5 bit ON to block MPX share traps.
2. Access the UCW for the interrupting device and test the active bit. If the UCW is active, branch to the GMSW routine and perform the remaining objectives of the interrupt routine.
3. Reset the interrupt latch.
4. Reset the active bit.
5. Remove buffer status bits.
6. Handle PCI if specified.
7. Set data check if required.
8. Store full CSW (all bytes meaningful) and reset H5. Go load new PSW.

Normal Secondary Ending Status Interruption

If, in item 2, the UCW is not active, it indicates that the device is presenting status a second time, normally device-end status. For example, suppose a control unit initiated an interrupt by presenting channel-end status alone. When the CPU executes this interrupt, it turns the UCW active bit off. Therefore, when the CPU executes an interrupt initiated by device-end status from the control unit, the UCW is inactive.

For this condition the microprogram branches to the GMGR routine and performs the initial selection of the unit in order to obtain its status. Following this sequence (send 0 command — read status — set 'service-out' to status — then disconnect) the program branches to GMSW and performs the CSW Store (zero out CSW, store status only).

After the MPX channel interrupt latch and H5 have been reset, any unit to which status has been stacked may have its status accepted into the IB, causing an interrupt request.



CSW Store

ENTRY POINT, GMSW START

From GLAA;

- This is the entry point from the GLAA routine honoring an interruption request.
- START is also the entry point from the GLAA routine when a test I/O instruction determines that the device specified has presented status. Entry here fulfills the major objective of TIO (store CSW).

From GMGR;

- GMSW START is entered from the GMGR routine during a start I/O instruction when initial selection is not completed because the control unit is busy.
- START is the entry point from GMGR during a test I/O instruction when the UCW is active and a STatus Queued condition exists.
- From GMGR during test I/O when the UCW is not active and non-zero status exists.

From GMLO;

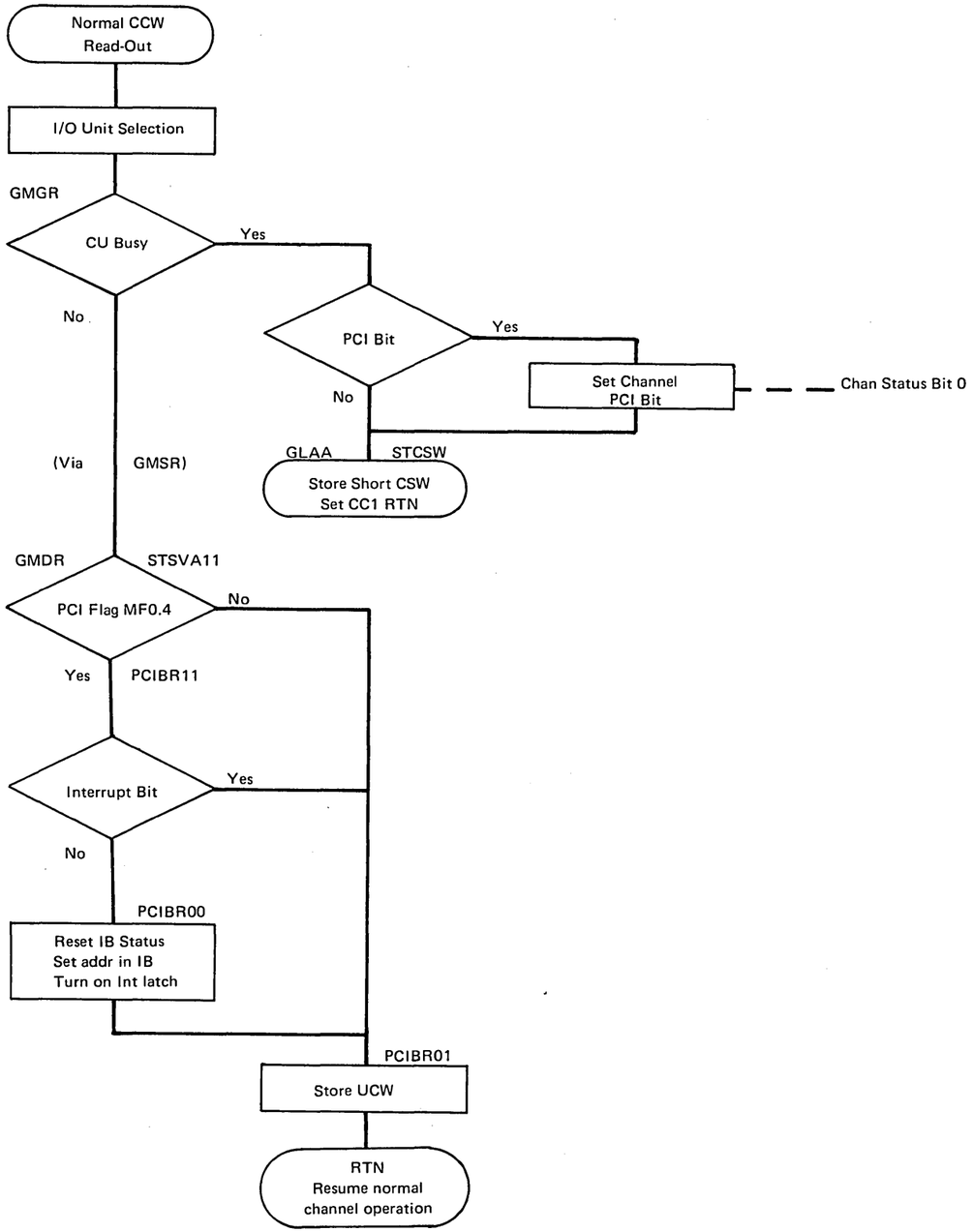
- START is the entry point from GMLO at the end of an error logout operation.

From GKDI;

- START is the entry point from GKDI for a halt I/O operation on the documentary console.

ENTRY POINT, GMSW MATCH 0

- The MATCH 0 entry point is used from the GMGR routine during a test I/O instruction when the control unit is busy.



Program-Controlled Interruption (PCI)

- The PCI bit allows the program to initiate an interruption.
- The PCI bit sets up an I/O interruption when the interruption buffer latch is off (IB empty).
- When the CPU executes an interruption initiated by the PCI bit, the PCI bit becomes part of the channel-status byte that is stored in the CSW.

The program-controlled interruption bit is bit 4 of the CCW flag byte. It also becomes bit 4 in the UCW Op and flags byte that is generated for the CCW. The PCI bit in the CCW initiates an interruption without, in any other way, affecting the channel operation. It may be used by the program to initiate an interruption at any point in a chain of CCWs to determine how the operation is progressing.

The PCI bit is effective at two different times:

1. During the execution of the CCW in which it occurs.
2. During the CSW store routine of an I/O interruption. In this case, the PCI bit does not initiate any action but becomes bit 0 of the channel status byte that is stored in the CSW. (The CSW is then interrogated by the program, and action is taken according to the conditions found).

When the CPU executes and interruption initiated by the multiplexer channel, it resets the PCI bit in the UCW associated with the unit that initiated the interruption. This occurs whether or not the interruption was caused by the PCI bit. If PCI interrupt is not allowed, the PCI flag is propagated to the next CCW.

Interruption Conditions

The following are conditions, during an I/O operation, that can result in a byte-multiplexer channel interruption:

1. A parity error detected on data during a read operation will cause a data check. (The data byte received from the bus-in register 'MBI' is checked for parity in the B-register.) A channel-control check does *not* occur for this type of error.
2. Status received from an I/O unit for:
   a. A chain-data operation in which the data count has not yet reached zero.
   b. A command-chain operation in which the data count has not yet reached zero and the SLI flag is off.
   c. No chaining, but the data count has not yet reached zero and the SLI flag is off.

   In each of these cases the IL bit is set on in the UCW/channel status byte of the UCW.
3. During chaining, one of the following CCW program errors occurs.
   a. An invalid CCW command code (bits 4-7 in command code equal 0000). This condition is not checked if data chaining is being performed.
   b. A transfer-in-channel (TIC) command branches to another TIC.
   c. The two low-order CCW flag bits do not equal zero (bits 38 and 39). Not checked on a TIC command.
   d. Initial CCW data count is all zero. Not checked on TIC command.
   e. The three low-order CCW address bits on a TIC command do not equal zero (bits 29, 30, and 31).

Note that device-end status has already been received from a device before command-chaining to a new CCW occurs. If the interrupt latch is already on for some other interruption condition on the byte-multiplexer channel, the interruption condition for a CCW program check is set up as follows:

a. A test I/O *command* is issued (by the microprogram) to the device for which the CCW program check occurred.
b. Command-out is sent to the device when that device attempts to present status in response to the test I/O. This action stacks the status (which can be all zero) to the device. The status is not accepted until the interrupt latch is off. When the status is accepted, the interrupt latch is turned on and the interruption can be processed.

In the byte-multiplexer channel, *channel-control check* is caused by a CPU check condition detected during execution of a byte-multiplexer channel microprogram in which H-register bit 5 is on if retry fails to obtain correct operation or if the MPX EXT REG (OE) is involved in the operation.

Channel-control check is set on in the UCW/channel status byte of the UCW for use in subsequent storing of the CSW. If operational-in (from the unit with which the channel is communicating) is down, the operation proceeds to its normal ending point (if succeeding errors do not prevent such ending). If operational-in is up after the channel control check is set, an interface disconnect for the affected unit is performed by the channel. (Select-out down with address-out up is the sequence for an interface disconnect.) If the interface disconnect is successful, the affected control unit disconnects from the channel and, at its normal ending point, the control unit attempts to present status. In this case, then, the channel-control check is indicated in the CSW (stored as a result of processing the interruption condition caused by acceptance of the ending status).

If the interface disconnect is unsuccessful (that is, Op-In fails to drop in 32 microseconds), the channel performs a selective reset (suppress-out up and operational-out down). The characteristics of the I/O device determine whether that device will present device end after a selective reset. Therefore, in this case, the interruption condition may not occur if the device does not subsequently present status. However, it is not likely that CPU errors (which caused the channel control check) and interface errors with the unit will both occur together. Note also that a logout is performed for this error.

An *interface control check* occurs when an error is detected in the signal sequence between the channel and an I/O control unit as follows.

1. Bad parity on address or status byte from interface.
2. No address match on initial selection.
3. The device appeared nonoperational during command chaining.
4. Invalid signal sequence on interface.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

If the unit can subsequently present status to the channel, processing of the resulting interruption condition will cause the interface control check to be stored in the CSW. However, if the channel cannot communicate with the device, the interruption condition may not be generated. However, a logout associated with the device in error is performed to indicate the error.

That is, if the IB is in use for some other device's status, the channel cannot place status in the IB. If the unit for which the channel (or interface) control check occurred cannot be communicated with, then that unit cannot generate the required interruption condition.

An address for a CCW or an address for data that references (during an I/O operation) a main-storage area beyond the capacity of the system (*address-check* condition), causes the program check bit to be set (in the UCW/channel status byte of the UCW). If the storage access is for a CCW *during command chaining,* unit status (device end) has already been received befory the access to main storage is made for the CCW. If, for this condition, the interruption latch is off, it is turned on and unit status and address are placed in the IB. When the interruption condition is processed, the program check bit is available in the CSW.

If, however, the interrupt latch is on, a test I/O *command* is sent by the channel to the working unit. When that unit responds with status, command out is activated to stack that status to the unit. The interrupt latch will be turned on when the status is subsequently accepted (that is, when it is presented *and* the interrupt latch is off) by the channel. Again, processing of this (that is, an address check occurred) interruption condition will result inprogram-check status in the CSW.

If the address check (program check) occurs during data transfer (or for CCW access during data chaining), the unit is told to stop data transfer and present status. After the status is presented, processing of the interruption condition results in program-check status in the CSW.

A protection check for CCW or data accesses is processed in the same manner as address check (program check), except that the resulting indication in the CSW is the protection check bit.

Note that chaining check status is not used in multiplexer channel operations.

Channel Logout

Channel-logout facilities provide a means of preserving for analysis a record of pertinent conditions that exist at the time a malfunction involving channels occurs. Logout is basically a "snapshot" of conditions that existed at the time the error occurred.

Essentially two levels of channel logout occur, depending on the nature of the malfunction and on the setting of control register 14, bit 2. The bit determines, under control of the program, whether extended logouts can occur as part of the I/O interruption, or whether only limited channel logouts are allowed.

I/O Communications Area

The I/O Communications Area (IOCA), a permanently assigned area in main storage, is utilized by the logout operation for limited channel logout, to point to the address of the extended log, and for other logout control information. The IOCA is located in main storage 160-191 (A0-BF).

Channel ID: (Locations 168-171.) When stored during the execution of the Store Channel ID instruction, this area contains information that describes the addressed channel.

I/O Extended Logout Pointer: (Locations 173-175.) The I/O Extended Logout (IOEL) pointer field is program-set to designate an area to be used by channels to store the extended logout information. The low-order three bits of the pointer are ignored so that the extended logout always begins on a doubleword boundary. Logout information may be stored in the IOEL area only when the IOEL mask bit (CR14 bit 2) is set to 1.

Limited Channel Logout: (Locations 176-179.) The limited channel logout field is also called the ECSW. This field contains model-independent information related to hardware errors detected by the channel. The ECSW is used to provide detailed machine status when errors have affected I/O operations. The field is accessible to the CPU program. The ECSW field may be stored only when the CSW or a portion of the CSW is stored. It may (or may not) be accompanied by the extended channel logout.

(Locations 180-184): Not used.

(Locations 185-187): Reserved for future use.

(Locations 188-191): Not used.

Refer to the sections "Byte-Multiplexer Channel" and "Selector/Block-Multiplexer Channels" for logout information as it applies to individual channel types.

Stop After Log

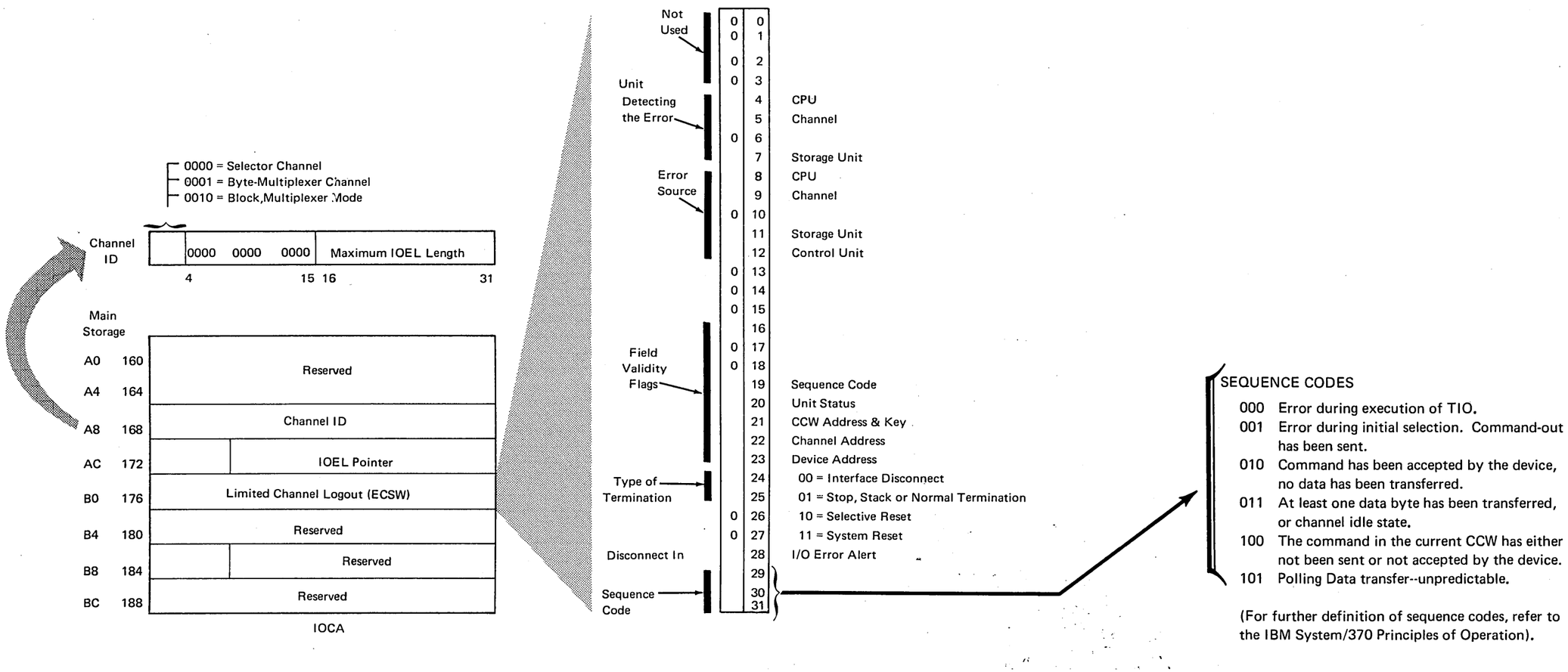
The stop after log switch on the console overrides the control register and allows extended channel logging when there is an interface control check or channel control check.

Stop after log causes the following action on a channel error:

- The IOEL pointer is set to 100 (hex).
- The ECSW is stored in the IOCA and the extended log is stored at 100.

Both the ECSW and the extended channel log are always available when running in the stop after log mode.

The CPU places an identifier in main storage locations 32 and 33 (machine check old PSW) to identify the failing channel. The identification is 1450-1454 for channels 0 through 4 respectively and 145F for the CPU. The system stops with at least one red light and the log present indication.



Error Handling and Logout

Byte-multiplexer channel check facilities are incorporated in the microcode routines. When Interface Control Check (IFCC) errors occur and are detected by the microcode, a catalog number is set to identify the type of failure and then the routine branches to the logout routine to perform the logout.

Channel-control checks are detected in CPU hardware as a result of unretryable errors. These errors are also handled by the logout routine, but there is no catalog number. The machine-check registers are logged for channel-control checks, whereas for IFCC errors, these words in the logout area are set to 0.

The byte-multiplexer channel extended logout is located in main storage in the area designated by the IOEL pointer. The pointer is in storage location 172, in the IOCA. Logout is under control of CR14 bit 2 unless the system is in Stop After Log mode.

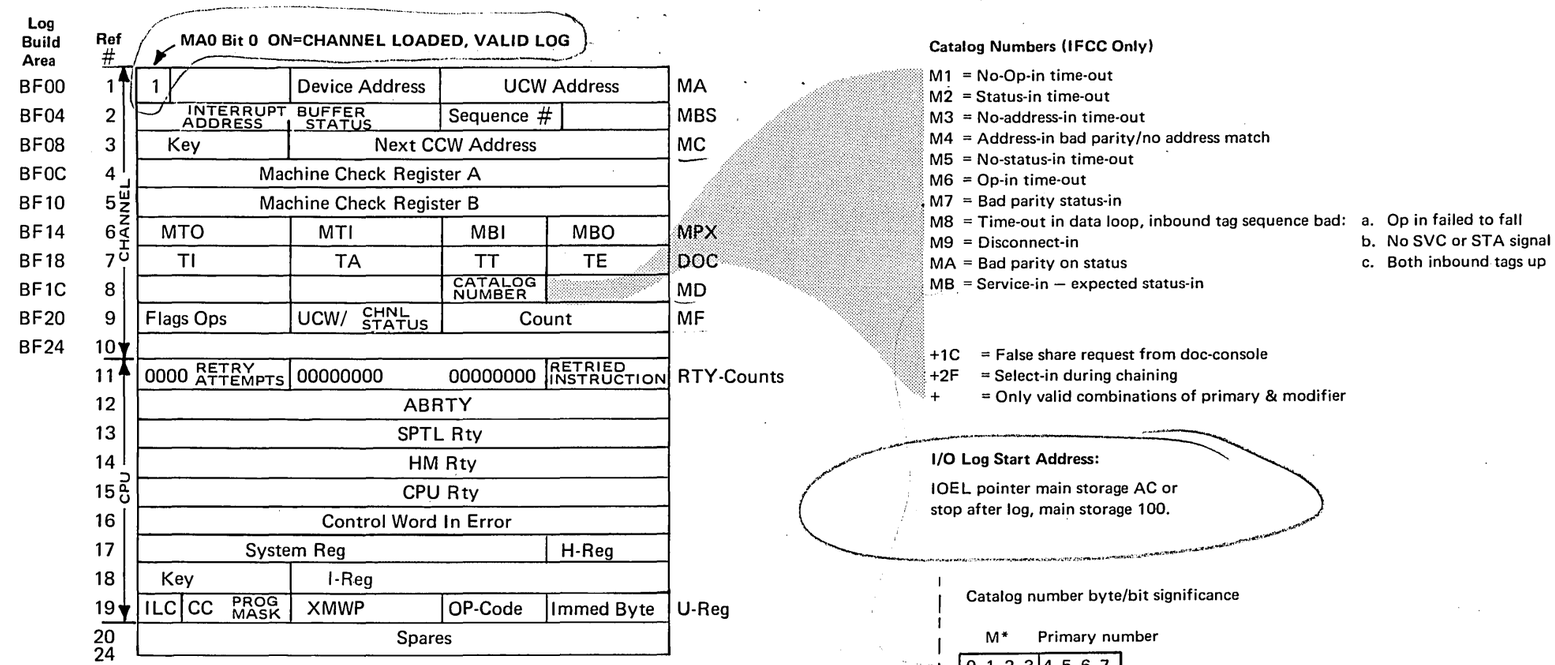
The low three bits of the IOEL pointer are ignored so that the log always begins on a doubleword boundary. If the pointer is set such that any part of the log would be out of storage, no part of the log is stored.

The channel logout is attempted anytime a CSW is stored with an interface control check or a channel-control check. This is done at I/O interruption time and at condition code 1 time to an I/O instruction. A channel error does not cause a machine-check interruption.

Channel logs are lost under any of the following circumstances.

1. CR14 bit 2 is set to 0 (unless Stop After Log is active).
2. The IOEL pointer is pointing to an address that is out of storage or allows insufficient space for the maximum logout.
3. A second channel error occurs before the first is cleared; in this case, the first log is preserved.
4. A machine-check log is pending or a machine check occurs before the I/O log is cleared. In this case, the first ten log words are valid, but the 14 CPU words that may be appended are taken as part of the machine check.

For most conditions causing logout, the format and contents of the log are as shown in the diagram. There is the possibility, however, of some error conditions or a false logout at which the usage of local storage might vary from this pattern. The identification of the local-storage log and external registers are as shown regardless of content.



The sequence number in MBS2 may be used to construct the sequence code in the Limited Channel Logout area.

Catalog Numbers (IFCC Only)

M1 = No-Op-in time-out
M2 = Status-in time-out
M3 = No-address-in time-out
M4 = Address-in bad parity/no address match
M5 = No-status-in time-out
M6 = Op-in time-out
M7 = Bad parity status-in
M8 = Time-out in data loop, inbound tag sequence bad: a. Op in failed to fall
M9 = Disconnect-in b. No SVC or STA signal
MA = Bad parity on status c. Both inbound tags up
MB = Service-in — expected status-in

+1C = False share request from doc-console
+2F = Select-in during chaining
\+ = Only valid combinations of primary & modifier

I/O Log Start Address:

IOEL pointer main storage AC or stop after log, main storage 100.

Catalog number byte/bit significance

M* Primary number

0 1 2 3 4 5 6 7

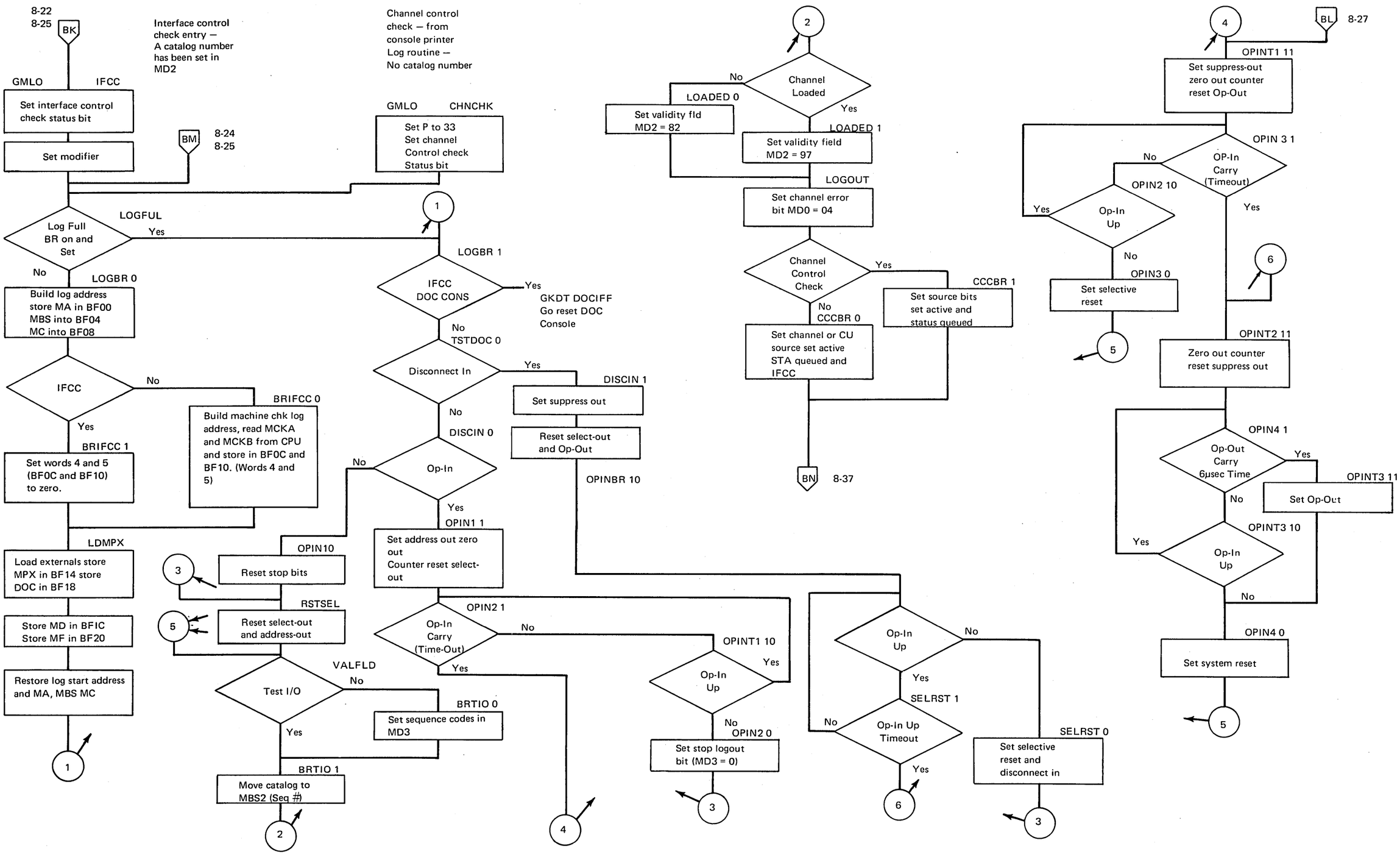
1 = Share trap
1 = Chaining

M = 0 = Initial selection or burst mode

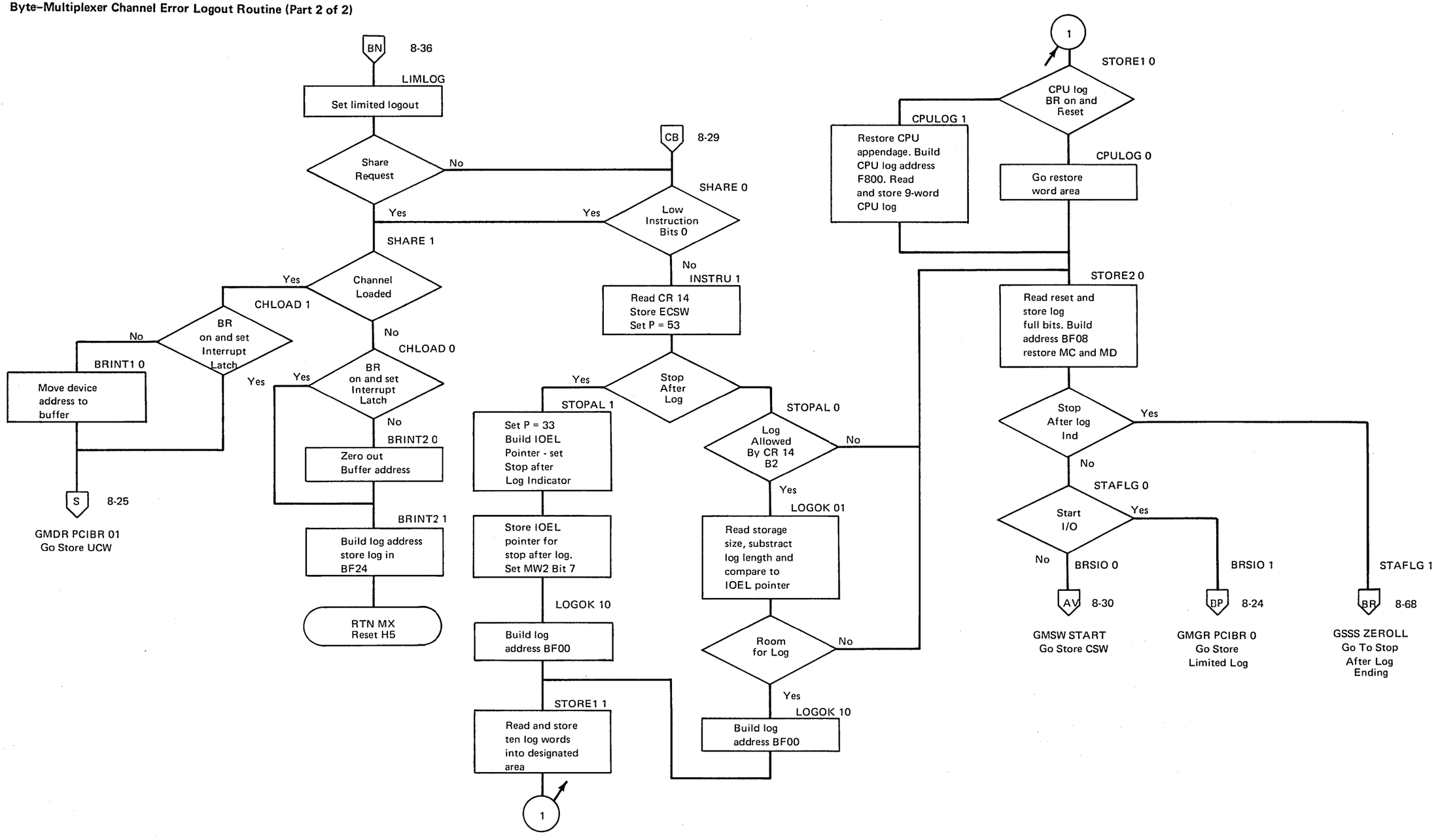
Stop after log ID for SEREP

Interruption code in old MCK PSW
1450 = byte-multiple
Log present and some red
Check light

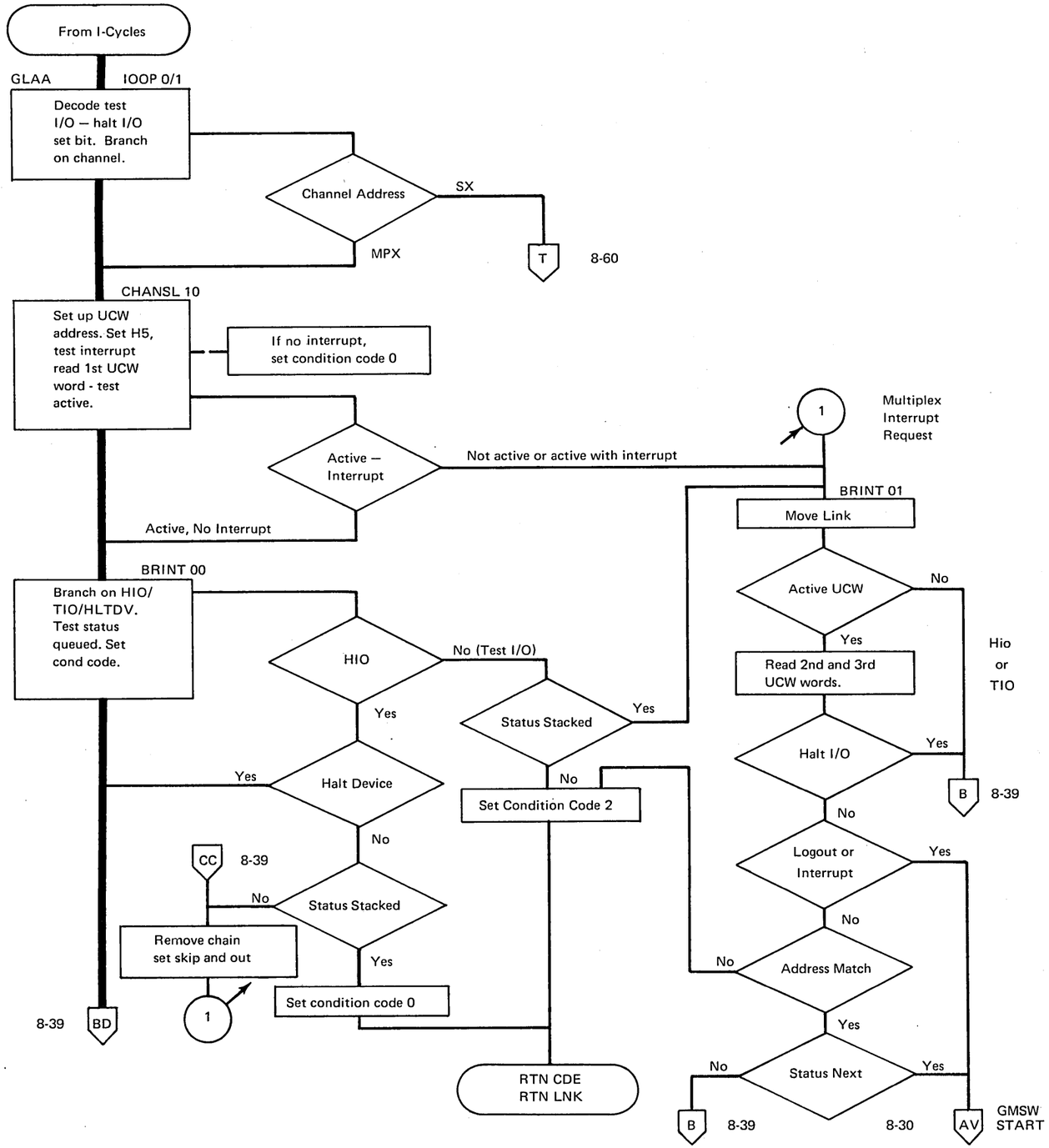
Byte-Multiplexer Channel Error Logout Routine (Part 1 of 2)



Byte–Multiplexer Channel Error Logout Routine (Part 2 of 2)



Test I/O--Halt I/O--Halt Device (Part 1 of 2)



Test I/O — Halt I/O — Halt Device

- Test I/O (TIO) obtains the status of an I/O device.
- Halt I/O (HIO) discontinues operation between the addressed channel and whatever I/O device is in operation on that channel.
- Halt Device (HDV), a variation of HIO, discontinues operation between the addressed channel and a particular device.

Test I/O: The test I/O instruction is decoded, and the execution begins in routine GLAA by setting the TIO bit in the instruction code byte (byte 0) in the X-register. The UCW address is developed.

Set H5 bit ON to block requests, test the interrupt bit, and if not ON, set condition code 0. The first UCW word (Flags and Ops byte) is read and the 'active UCW' bit is tested. The UCW active bit ON indicates that the subchannel is unavailable. Active bit OFF means that either the I/O device involved has status stacked or is free to perform other operations.

When the UCW is available (active bit OFF), the channel must execute an initial selection sequence with the I/O device involved and issue a zero command. When the device responds to 'command-out' with 'status-in', the channel tests the status, (set 'service-out' to status, and waits for the fall of 'op-in'). If status is zero, set condition code 0 and return to I-cycles. If status is not 0, reset the status-next bit, store a CSW, set condition code 2, and return to I-cycles.

When the channel receives 'status-in' in response to 'address-out' (control unit busy) it reads 'status-in' and resets 'select-out'. When 'status-in' falls, the routine branches (on TIO) to GMSW to store the unit-status byte (store CSW).

When the active bit is on, it means one of three things:

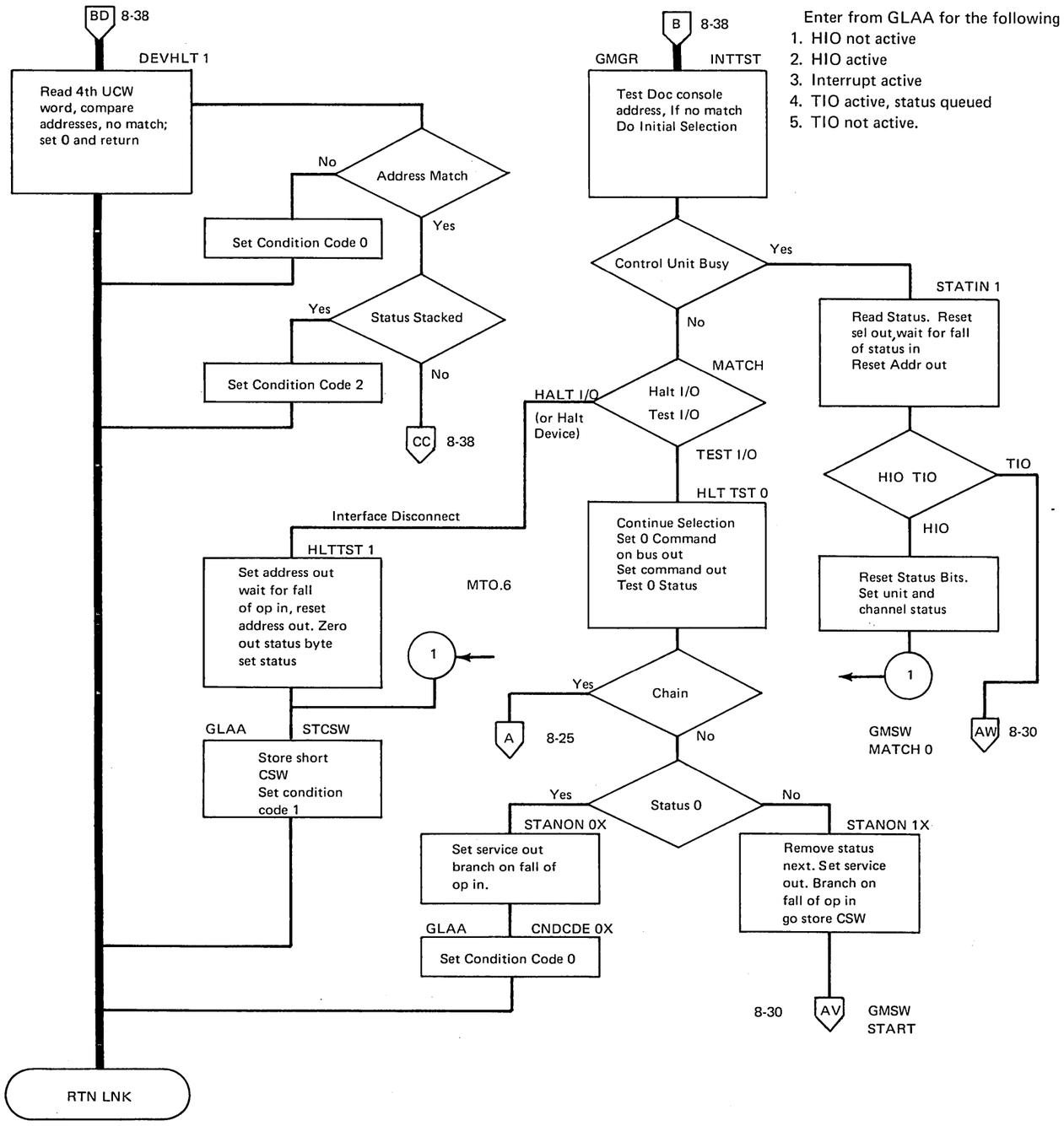
1. The subchannel is busy, either with the specified device or another device.
2. The subchannel has an interrupt condition waiting, or
3. Status is stacked.

When the UCW is active, the interrupt bit is tested. If an interrupt is waiting, the UCW word and link information is moved in LS, the second and third UCW words are read out, and the routine branches to GMSW to store the CSW.

If there is no interrupt or logout, the high-order device address is compared with the control-unit address. Equal comparison means that the device specified by the TIO instruction has presented status; the routine branches to GMSW to store status. For unequal address comparison, the channel might have received status from another device associated with the same UCW as the I/O device specified by the test I/O instruction. If this is so, the channel sets condition code 2 and returns the CPU to I-cycles.

Halt I/O, Halt Device: The channel begins the execution of a halt I/O or halt device instruction in the same manner as other I/O instructions. The PSW is checked to ensure that the CPU is in supervisory state, and a UCW address is set up and tested for validity. Condition code is set, H5 bit is turned on, and the first UCW word is read and tested.

Test I/O--Halt I/O--Halt Device (Part 2 of 2)



Enter from GLAA for the following conditions:

1. HIO not active
2. HIO active
3. Interrupt active
4. TIO active, status queued
5. TIO not active.

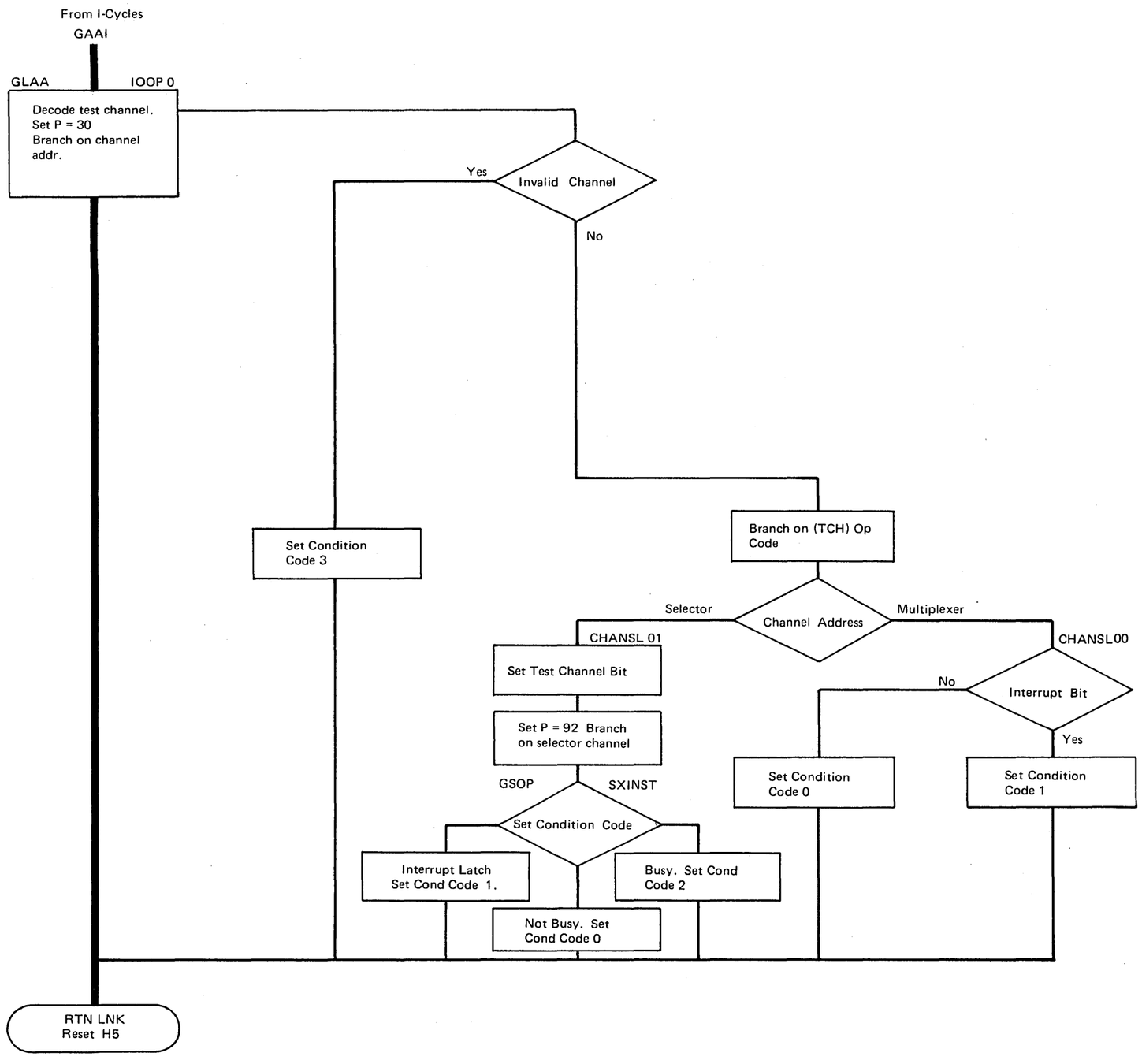
UCW Not Active: When the active bit is not on, the channel does not have to test the status byte for an interrupt. When the channel finds the UCW available, it branches to GMGR, executes the halt I/O interface test (checks for console Op), and then does initial selection.

If response to 'select-out' is 'status-in', branch to control unit busy sequence. Read status, reset 'select-out' and wait for fall of 'status-in'. (Go to logout if time-out). Assume fall of 'status-in'. Reset 'address-out', branch on HIO and turn off status bits. Set new unit and channel status in MBS2 and MBS3, move it to the V-Reg, and then do short CSW store, set condition code 1 and return to I-cycles.

If on initial selection, response to 'address-out' with 'select- out' is 'op-in' (selection is normal) followed by 'address-in' and equal comparison, reset 'select-out', and then branch on HIO to perform the interface disconnect routine. This consists of setting 'address-out', waiting for fall of 'op-in', resetting 'address-out', and status. Set new status, do short CSW store, set condition code 1 and return to I-cycles.

UCW Active: Test for HDV (bit 15 of immediate operand). If HDV operation, read the fourth word of UCW and check for address match. If no match, indicating that the device addressed is not active, set condition code 0 and return. If address matches, indicating that the device addressed is active and therefore should be halted, test status queued (if on set condition code 2 and return) if no status queued, remove chain flags, set skip and output, restore unit address, move link information, read 2nd and 3rd UCW words, then go do initial selection. Procedure then is the same as described for UCW not active.

Test Channel



Test Channel

- Test Channel (TCH) is executed to determine the status of a particular channel.
- A condition code is set to indicate the status of the addressed channel.

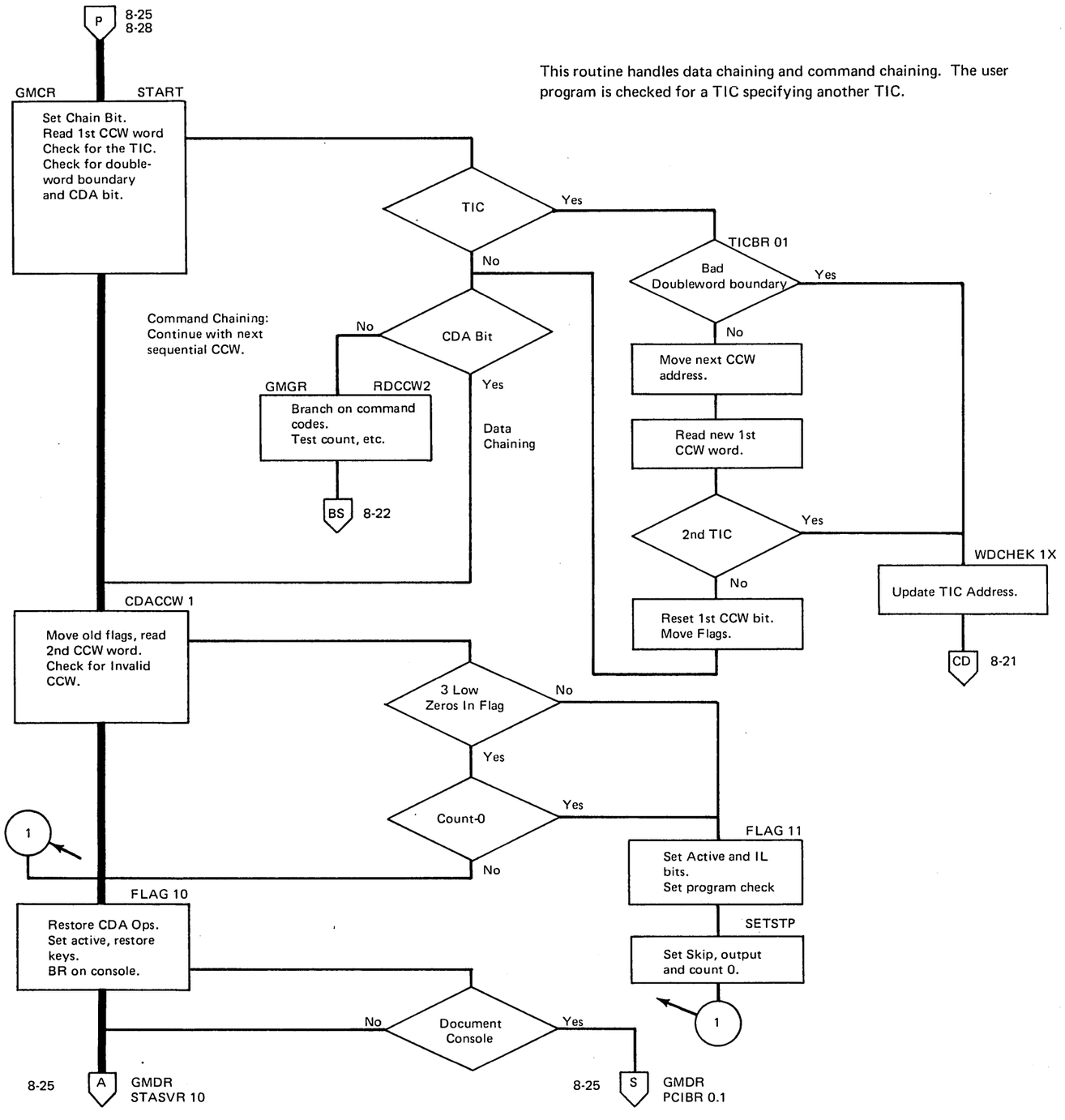
The test channel instruction is executed (when the CPU is in the supervisor state) to determine the status of a particular channel. It tests whether the channel is operating in burst mode, is aware of any outstanding interruptions, or is not operational. When none of these conditions exists, the available state is indicated. In executing TCH, no device is selected, and on the multiplexer channel the subchannels are not interrogated.

The TCH instruction is detected as an I/O operation during I-cycles, and the microprogram branches to the GLAA routine. Test channel in other than supervisor mode is a programming error and causes a branch to GICM.

The operation follows a common path for either MPX or SX channels until the actual channel address is tested. At this point, if it is an SX channel, the test channel bit is set and the microprogram branches to the GSOP routine for completion of execution and setting of appropriate condition code.

For MPX channels, testing is completed in GLAA with either condition code 0 or 1 being set. The condition code meanings are as follows:

0 - Channel available
1 - Interruption pending in channel
2 - Channel operating in burst mode
3 - Channel not operational



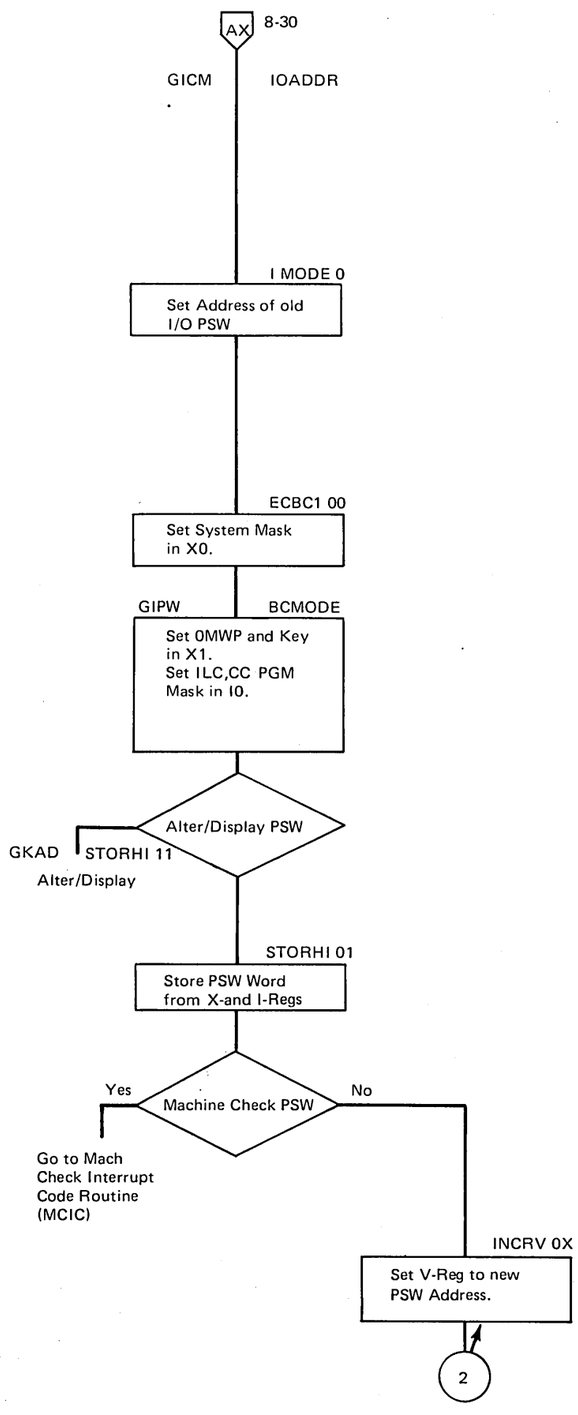
Data Chaining

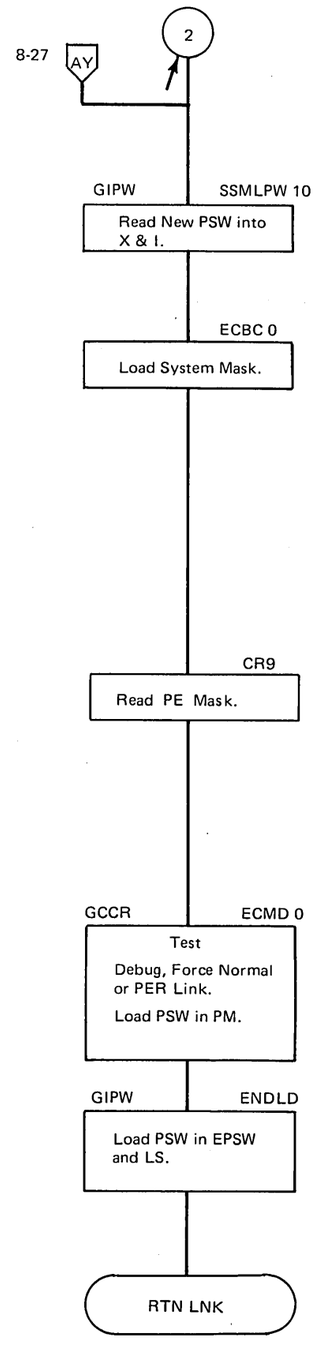
- Entry to the chaining routine is from GMDR following zero count/CDA flag detection.
- Data chaining permits reading from or writing to storage locations that are not adjacent.
- Data chaining is controlled by the CDA flag bit (bit 0 of Flags and Ops byte, MF0).
- During any data loop cycle in which the working count reaches 0, the CDA flag is tested; if *on*, branch to GMCR to obtain replacement for working count, data address, and flags from the next CCW.

Command Chaining

- Entry to the chaining routine is from status handling (GMMS) when suppress-out is up and DE status and CC flag bit are present.
- Command chaining allows the channel to execute several commands on an I/O device with one start I/O instruction.
- Command chaining is specified by the CCW CDA flag bit off and the CC flag bit on.
- When the channel completes error-free execution of a CCW and receives DE status, the channel reads out the next CCW and begins its execution.
- Status modifier with DE requires the channel to bypass the next CCW in favor of the one following it.
- During command chaining, suppress-out prevents other I/O units from beginning a reselection sequence.
- A program check on a command-chaining operation causes the execution of an interruption (store CSW).

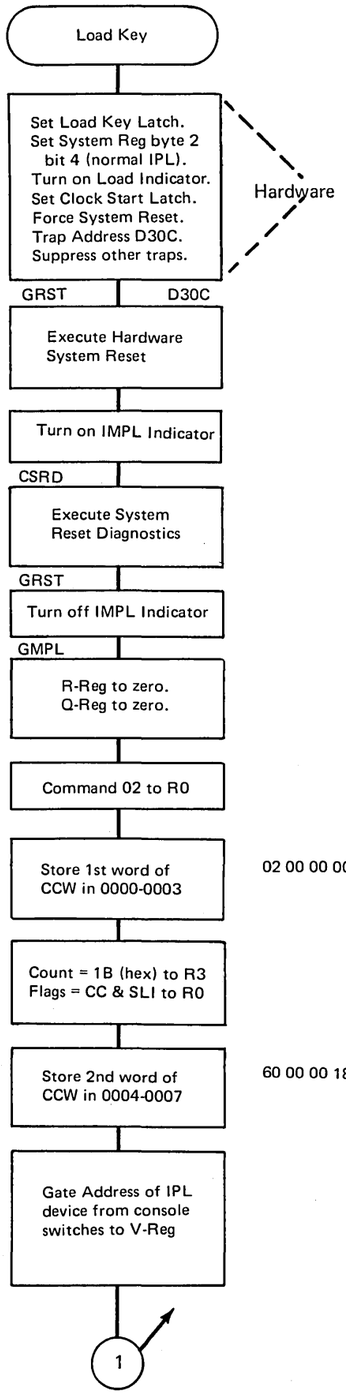
Store and Load PSW





Store and Load PSW

- The PSW store routine stores the current PSW in the I/O old PSW location.
- The PSW load routine puts the new PSW into local storage and external registers.
- When the PSW is loaded, the microprogram branches to I-cycles and starts execution of the I/O program.



- IPL is started by selecting an input device and pressing the load key on the console.
- IPL provides the facility for loading a program into main storage.
- IPL can be from devices on either byte-multiplexer channel, selector/block-multiplexer channel or IFA (if installed).

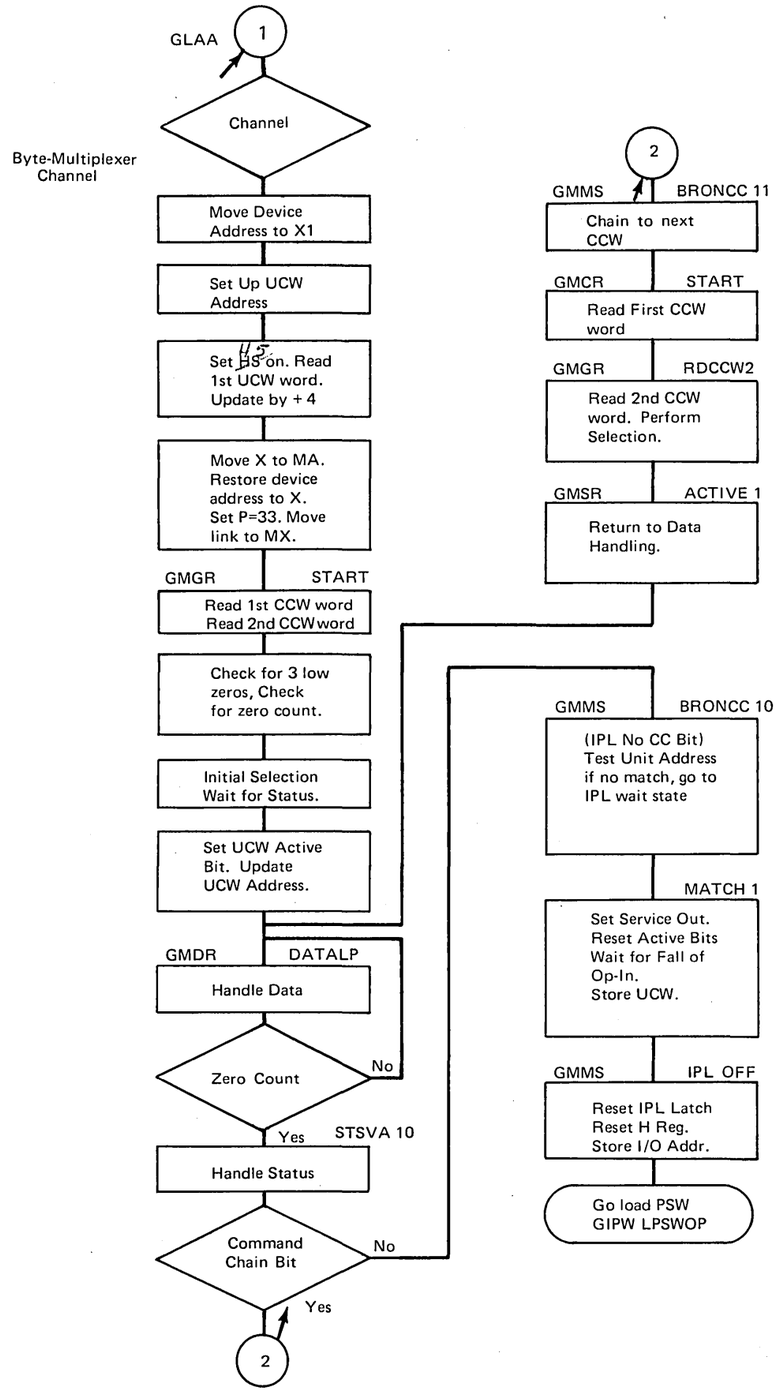
The IPL procedure resembles a SIO instruction in which the selected I/O device and a zero protection key is specified. The CCW for this instruction has a read command, zero data address, a byte count of 24 command chain flag on, SLI on, and command address 0.

The IPL operation reads new information into the first 3 doublewords of storage. The remainder of the program can be placed elsewhere.

The first doubleword contains the starting PSW for the entering program. The next 2 doublewords contain 2 CCWs that are command chained. The operation continues with these and other chained CCWs to enter the full program.

Selection, handling of data and status, etc. follow the same general pattern as for SIO. This diagram is intended to give overall flow of the IPL procedure. Refer to other parts of this manual for details of selection, data handling, status handling, etc. Good status and correct programming are assumed.

At ending time, (CC bit not on) all channels including IFA branch to the GMMS (IPL OFF entry) to reset the IPL latch and end the IPL operation. The microprogram then branches to GIPW to load the PSW and return to I-cycles.

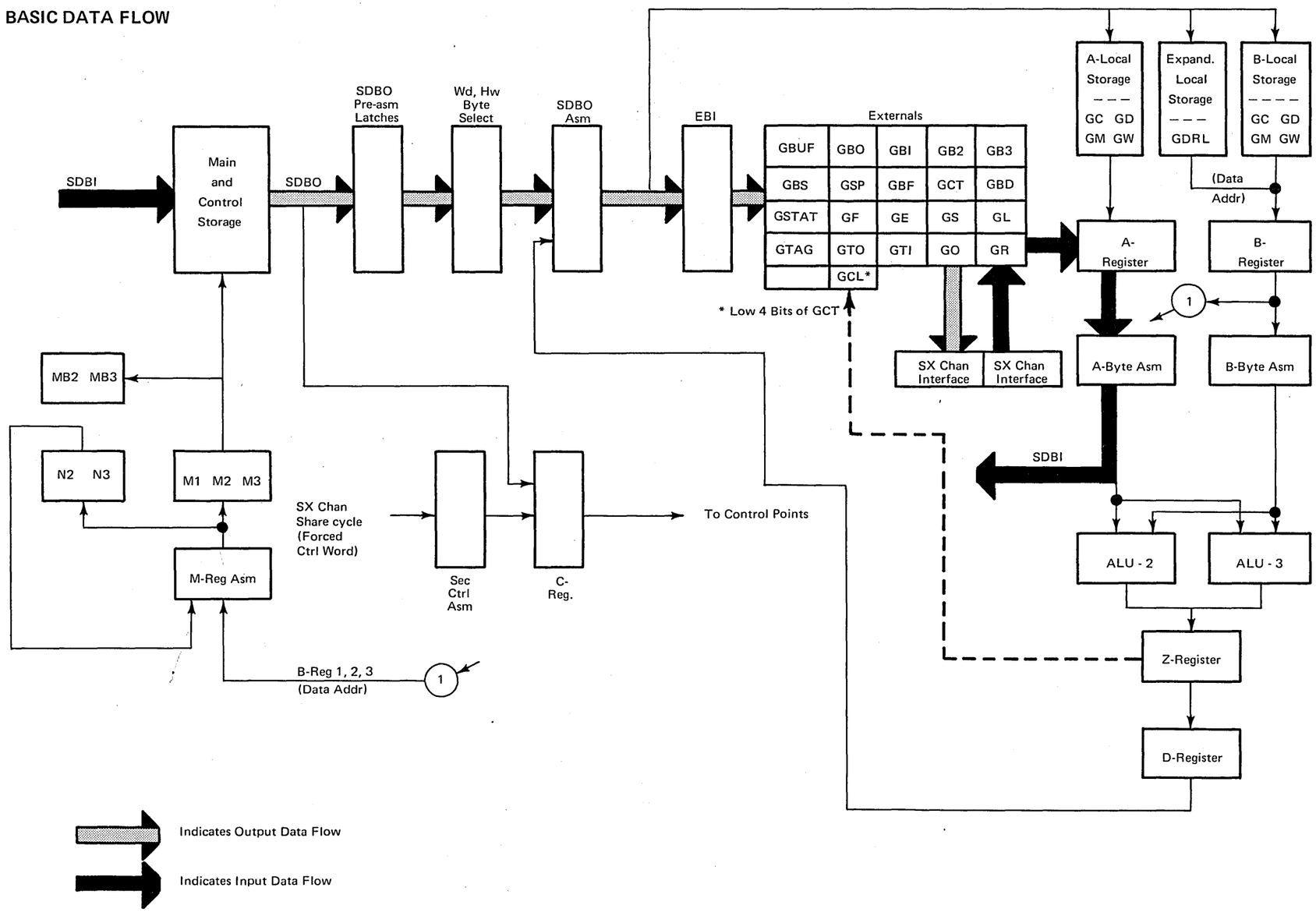


Byte-Multiplexer Channel IPL

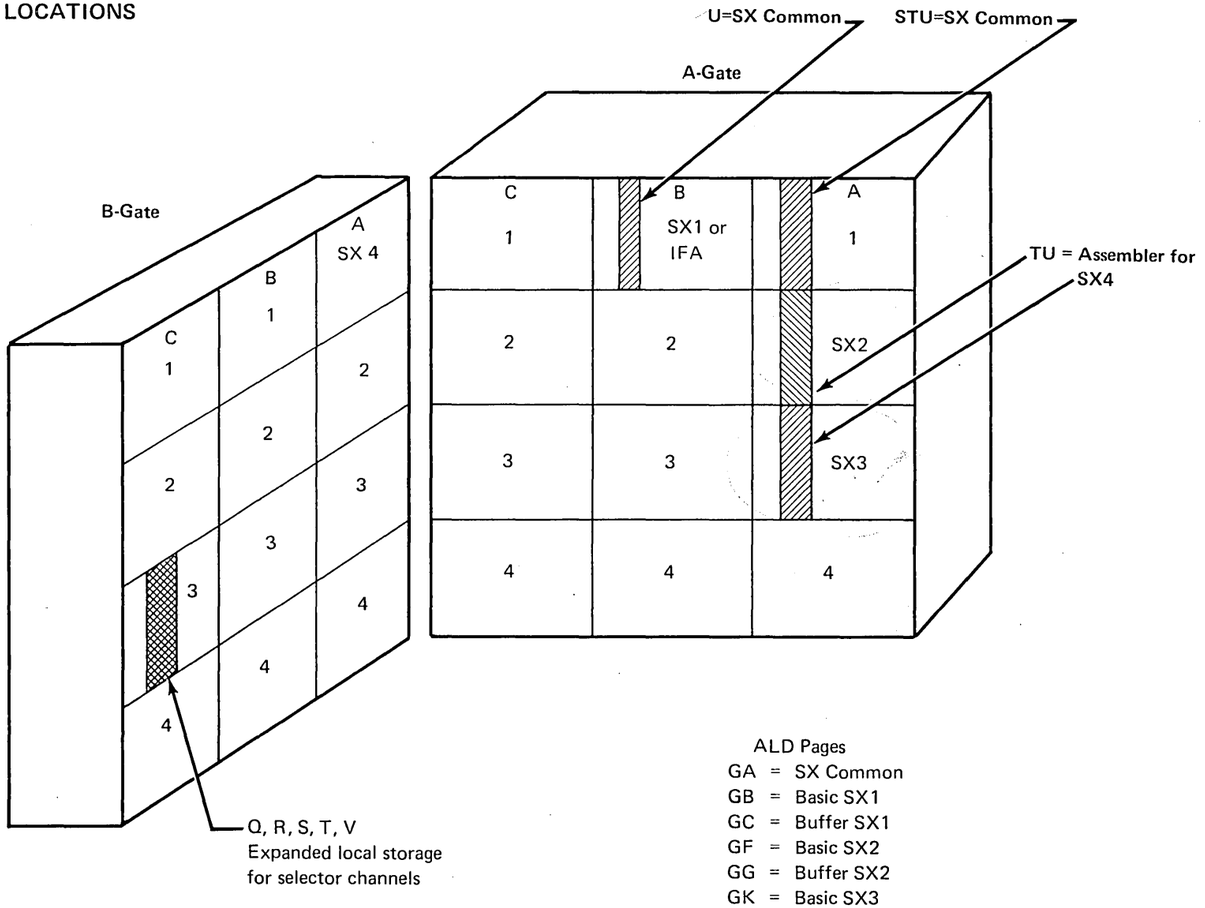
SELECTOR CHANNELS

- The Model 145 may have from one to four selector channels.
- Selector channels are designed primarily for use with high-speed devices.
- Each selector channel can control one operation at a time. Each operation is performed in burst mode.
- The CPU local-storage registers and special microprogram routines in control storage are used with selector channel and CPU circuits to perform selector-channel operations.
- Selector share cycles are used for data transfers (including sense and control information).
- The word buffer feature enables assembling four bytes of data before requesting a share cycle.
- There is time between share cycles to allow for CPU processing and for operating the integrated I/O devices.
- Selector-channel addresses are 01-04. Device addresses on each channel may be any of the 256 possible bit combinations of a byte.
- The maximum data rate possible with the word buffer installed is 1.85 million bytes per second; without the buffer it is 820,000 bytes per second.

BASIC DATA FLOW



LOCATIONS



ALD Pages

GA = SX Common
GB = Basic SX1
GC = Buffer SX1
GF = Basic SX2
GG = Buffer SX2
GK = Basic SX3
GL = Buffer SX3
GP = Basic SX4
GQ = Buffer SX4
GW = Expanded LS for SX

Control Storage Map

FFFC	K-Addressable	ABKA
FF00		
F870	CPU Logout	GHLC
F800		
F4BC	Control Registers	GRGS GCCR
F480		
BF30	I/O Logout	GMLO GSER
BF00		
B3FC	Blk Mpx UCW Pointers	GSTB
B000		
9FFC	Sel Channel UCW Pool	GSTB
9000		
8000		

FUNCTIONAL UNITS

Branch Word GA Function

- Conditions channel circuitry response to subseuqnet control words.
- Controls Tags-Out and Poll control.
- Permits access to selector-channel circuits that cannot be accessed by normal microprogram.

h K Field	Set GAL GA, OR, 0h	Reset GAL GA, A-, 0h	Set GAH GA, OR, h0	Reset GAH GA, A-, h0
1	Set Poll Cntl (Sft)	Reset Poll Cntl	Set Channel 1	Channel Reset
2	Set Poll Cntl (Hrd)	Reset Retry Holdup	Set Channel 2	Chain Reset
3	Set Cmnd Retry		Set Channel 3	Machine Reset
4			Set Channel 4	
5	Set Count Ready	Start I/O Reset	Set Channel Loaded	Reset Channel Loaded
6	Set Protect Check	Set Control Check	Set CC, Reset CD	(Diag) Buffer Shift
7	Set Program Check	WLR Sample	Set PCI	Rst PCI & Intp Lth
8	Set Intrpt Latch	Set DCC Mode	(Diag) Set Blk Shr Req	RST DCC/(DIAG), RST BLOCK SHARE REQUEST
9	Set Select Out	Rst Out & Prmd	Set Diag Stat	Reset Diag Stat, Rst Intrp Lth
A	Set Suppress Out	Reset Suppress Out	*Set Channel Primed	Set Channel Tried
B	Bus-In to GR		*Set Data Out	
C	Set Oprnl Out	Reset Oprnl Out	*Set Command Out	Set Address Out
D	Infce Ctrl Check	(Diag) Service Signl	*Set Service Out	Reset PCI
E	Set Diag Mode	Reset Diag Mode	*Set Halt I/O	Reset Halt I/O
F				

*Set Retry Holdup (Also)

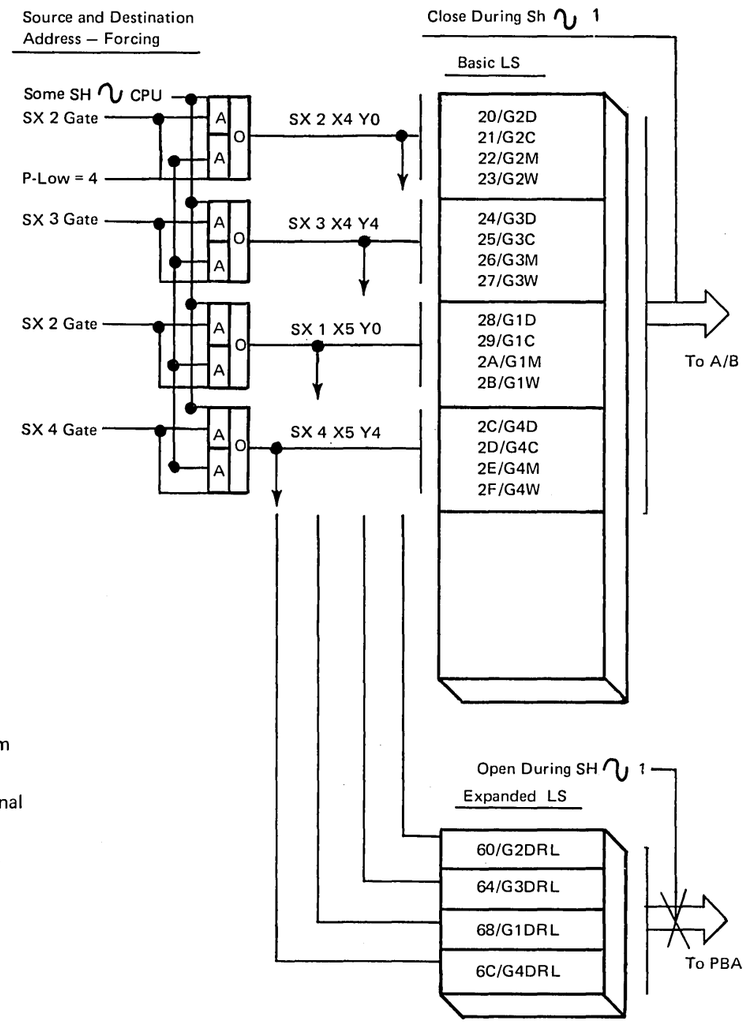
Selector-Channel Addressing Principles

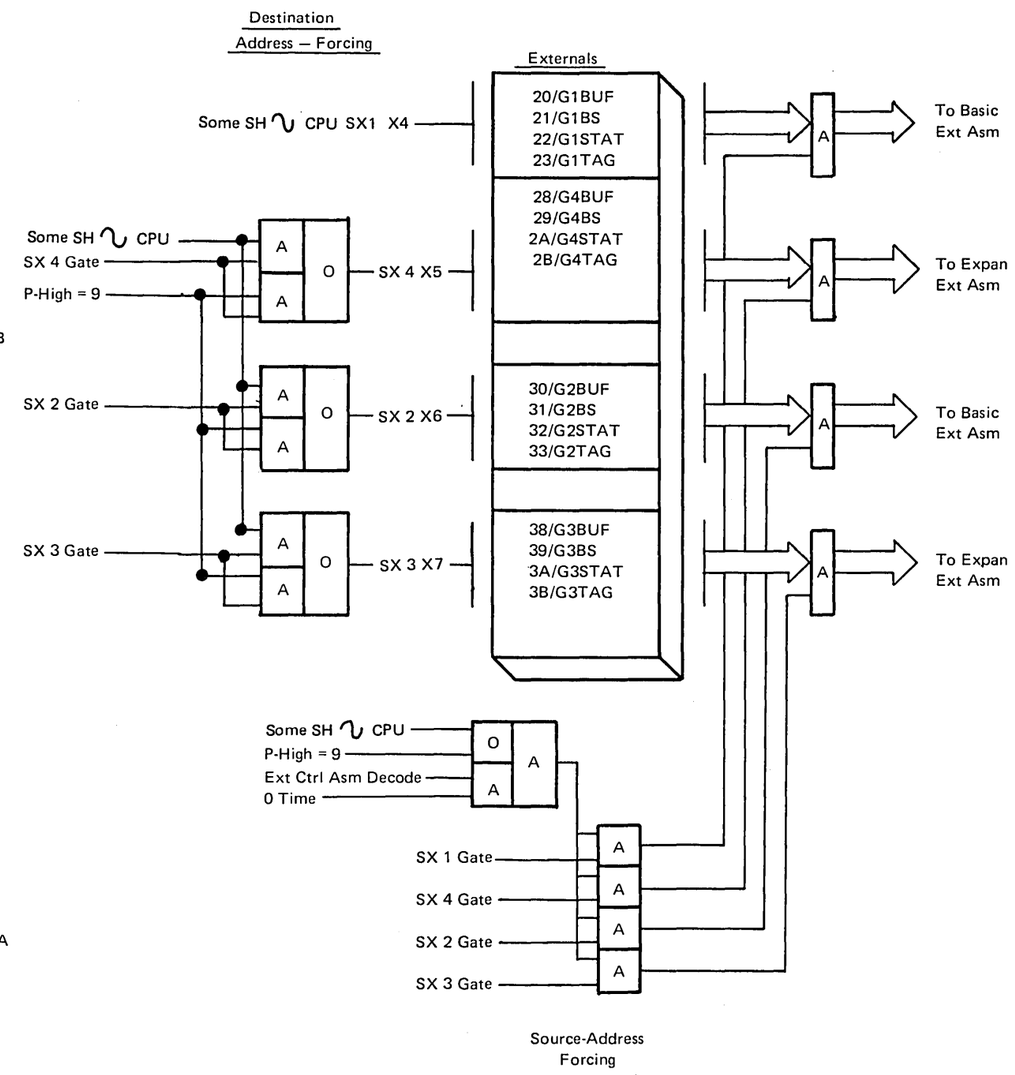
RTN Name	Selector Channel Micro Program
GSER	Error Cleanup Rtne
CSES	Exceptional Status Trap Rtne
GSHT	Halt Rtne
GSIN	Channel Interrupt Handling
GSLD	Block MPX Load Rtne
GSOP	I/O Instructions Rtne
GSSS	Status Store Rtne
GSTB	Block MPX UCWs/Pointers
GSTR	Trap Rtne
GSAP	Address or Protect Check

Note: The selector channel microprogram handles all four selector channels:

P = 9X forces the correct channel external register addressing.

P = X4 forces the correct channel local-storage addressing.





Selector-Channel Local-Storage Assignments

	Word Addr	Word Name	Word Assignments Byte-0	Byte-1	Byte-2	Byte 3
			Selector Channel 2			
SX2	20	GD	Protect Key	CCW bytes 1-3		
	21	GC	Flags	CCW Byte 5	Count	
	22	GM	Protect Key	CCW Address + 8		
	23	GW	Marks		Time Out Ctr	Unit Address
			Selector Channel 3			
SX3	24	GD	Protect Key	CCW bytes 1-3		
	25	GC	Flags	CCW Byte 5	Count	
	26	GM	Protect Key	CCW Address + 8		
	27	GW	Marks		Time Out Ctr	Unit Address
			Selector Channel 1			
SX1	28	GD	Protect Key		CCW bytes 1-3	
	29	GC	Flags	CCW Byte 5	Count	
	3A	GM	Protect Key	CCW Address + 8		
	2B	GW	Marks		Time Out Ctr	Unit Address
			Selector Channel 4			
SX4	2C	GD	Protect Key	CCW bytes 1-3		
	2D	GC	Flags	CCW Byte 5	Count	
	2E	GM	Protect Key	CCW Address + 8		
	2F	GW	Marks		Time Out Ctr	Unit Address

Each of the four selector channels has four Local Storage words that contain the UCW for the operation. The microroutine specifies only the letters of the name to allow one routine to operate for all channels. The channel selection develops address bits 4 and 5 to select the correct word in local storage.

The four local-storage locations of each channel are used as general working registers and thus have multiple uses throughout the operation of the channel.

The "Standard Usage" map is valid only when the count ready latch (GL bit 0) of that channel is on.

SELECTOR-CHANNEL EXTERNAL WORD ADDRESS AND BIT ASSIGNMENTS

Word Addr	Word Name	Byte-0	BYTE Byte-1	NAMES Byte-2	Byte-3
	Selector Channel 1				
20	G1BUF	G1B0	B2B1	G1B2	G1B3
21	G1BS	G1SP	B1BF	G1CT	G1BD
22	G1STAT	G1F	G1E	G1S	G1L
23	G1TAG	G1TO	G1T1	G1O	G1R
24					
25					
26					
27					
	Selector Channel 4				
28	G4BUF	G4B0	G4B1	G4B2	B4B3
29	G4BS	G4SP	G4BF	G4CT	G4BD
2A	G4STAT	G4F	G4E	G4S	G4L
2B	G4TAG	G4TO	G4TI	G4O	G4R
2C					
2D					
2E					
2F					
	Selector Channel 2				
30	G2BUF	G2B0	G2B1	G2B2	B2B3
31	G2BS	G2SP	G2BF	G2CT	G2BD
32	G2STAT	G2F	G2E	G2S	G2L
33	G2TAG	G2TO	G2TI	G20	G2R
34					
35					
36					
37					
	Selector Channel 3				
38	G3BUF	G3B0	G3B1	G3B2	G3B3
39	G3BS	G3SP	G3BF	G3CT	G3BD
3A	G3STAT	G3F	G3E	G3S	G3L
3B	G3TAG	G3TO	G3TI	G3O	G3R
3C					
3D					
3E					
3F					

NOTE: Microcode that addresses externals and local storage is of the form G1BUF, G2D, G3BS, G2M, etc., for microcode that is not hardware-switched. (Example: GM=G2DRL,SF).

Microcode that is hardware switched is of the form GBUF, GD, GBS, GM, etc. (Example: RDW GM DC, 88).

SX External Word Address and Bit Assignments

External Bit Assignments

GBUF

0	
1	
2	FWD = B0
3	
4	
5	BKWD=B3
6	
7	

GB1

0	
1	
1	FWD = B1
3	
4	
5	BKWD = B2
6	
7	

GB2

0	
1	
2	FWD = B2
3	
4	
5	BKWD = B1
6	
7	

GB3

0	
1	
2	FWD = B3
3	
4	
5	BKWD = B0
6	
7	

GBS — GSP

0	Spare
1	Spare
2	Spare
3	Spare
4	Spare
5	Space
6	Spare
7	Spare

GBF

0	Spare
1	BF 6
2	BF 5
3	BF 4
4	BF 3
5	GB 2
6	BF 1
7	BF 0

GCT

0	Buffer Ctrl Chk
1	Count through 0
2	Chain Data Request
3	Buffer Partition
4	GCL 0
5	GCL 1
6	GCL 2
7	GCL 3

GBD

0	Spare
1	GDRL 6
2	GDRL 7
3	GB P
4	GB 0
5	GB 1
6	GB 2
7	GB 3

GSTAT — GF

0	Chain Data
1	Command Chain
2	Sup Length Ind
3	Skip
4	Allow Halt
5	Input FWD
6	Input BKWD
7	Output

GE

0	Prog Ctrl Intrp
1	Incorrect Length
2	Program Check
3	Protection Check
4	Data Check
5	Chan Ctrl Chk
6	Intf Ctrl Chk
7	Chaining Chk

GS

0	Channel Loaded
1	Interrupt Latch
2	Poll Control
3	Channel Primed
4	DCC Mode
5	Command Retry
6	Gate A/Share Req*
7	Gate B/Trap Req*

*(Diag)

GL

0	Count Ready
1	Count Zero
2	GR Full
3	Interrupt Cond
4	Share Error
5	Spare
6	} Position Code
7	} Position Code

GTAG — GTO

0	Operational Out
1	Select Out
2	Address Out
3	Command Out
4	Service Out
5	Data Out
6	Suppress Out
7	Spare

GTI

0	Operational In
1	Address In
2	Status In
3	Service In
4	Select In
5	Data In
6	Request In
7	Disconnect In

GO

0	
1	
2	Chan Bus Out
3	
4	
5	Chan Bus Out
6	
7	

GR

0	
1	
2	Communication
3	with Data Flow
4	(Data, Command,
5	Address, Status)
6	
7	

GBUF External Word (Bytes GB0-GB3)

The selector-channel word buffer external facility (GBUF) consists of the seven byte-registers that are located logically between the normal channel GR-register and the GO-register. The GR-register is functionally part of the buffer. This configuration allows space within the buffer for two complete words. Each position of the buffer has a buffer-full bit in addition to the data bits and parity bit. (Refer to "GBS External Word.")

Under normal operating conditions, the GBUF external word is not addressed by the microprogram. The data-handling functions of the buffer register operate within circuitry via the normal data paths and are active during a share cycle to move data from the SDBO to the external bus or from the external bus to the SDBI.

When the selector-channel word buffer feature is present, there are complete buffer facilities for each installed selector channel. Refer to "Selector Channel Word Buffer."

GBS External Word (GSP, GBF, GCT, and GBD)

GSP: This byte has no assigned function.

GBF (Buffer Full Byte): The buffer-full bits control the shift of data through the buffer. This facility is addressed by the microprogram for a logout operation.

GCT: The GCT facility consists of the GCL register, buffer control check latch, and the block buffer at 4-latch. These facilities are buffer-control circuitry; they are not addressed by the microprogram except for logout. Refer to "GCL Register."

GBD: The GBD facility consists of bits 30 and 31 of the GDRL register and the buffer byte counter. Refer to "GDRL Register."

GSTAT External Word (GF, GE, GS, and GL)

GF: The GF facility consists of eight latches, also called *flag bits.* Some of the latches are set/reset by circuitry; others are set/reset by the GA microword.

GE: The GE facility is primarily for identifying check conditions. It also indicates the presence of a program-controlled interrupt. The facility consists of eight latches.

GS: The GS facility consists primarily of share-trap controls and indicators that are set and reset by microprogram routines via the GA function. Bits 6 and 7 are used for diagnostics also.

GL: The GL facility consists of eight latches that are used to indicate various conditions and share cycles.

GTAG External Word (GTO, GTI, GO, and GIR)

GTO: The GTO external facility makes Tags-Out information associated with selector-channel operation available to the microprogram. There are seven latches (one position of the byte is unassigned). The latches are set and reset by the GA function, with the exception of some that are reset by circuitry. The conditions of these latches may be tested by the microcode.

GTI: The GTI external facility provides communication between the selector-channel tags-in register and the microprogram. The tags-in register consists of eight polarity holds that represent the eight tags-in bits. The condition of the GTI can be tested by the microcode.

GO: The GO register (bus-out) receives data bytes either directly from the GR register (bus-in) or from the GR register via the word buffer circuits. The GO register can be tested by the microcode.

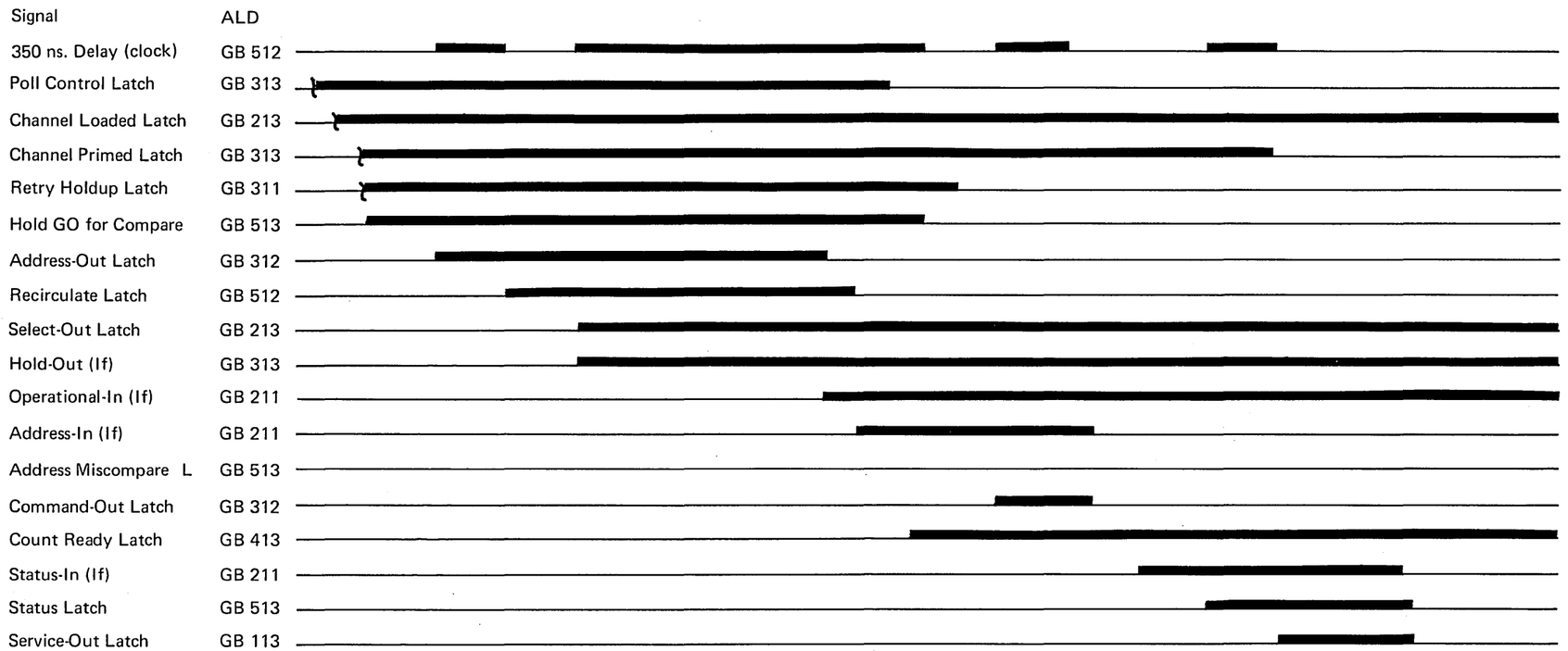
GR: The GR register can be gated to accommodate any one of the following: a) Channel Bus-In; b) Channel Bus-Out; or c) Status. GR is the only byte of the selector-channel externals that can be used as a destination in microprogramming.

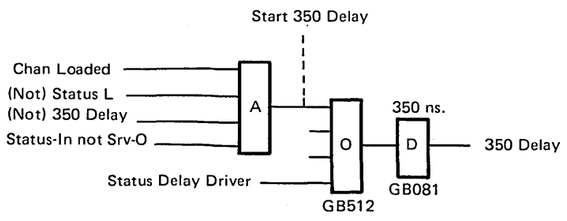
The inbound information can be:
- Address from unit
- Status from unit
- Data from unit.

The outbound information can be:
- Address to select unit
- Command to unit
- Data to unit.

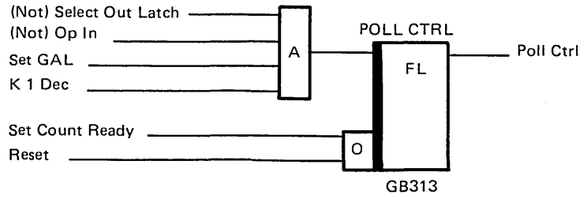
Outbound information is always passed to Bus-Out through the GO register.

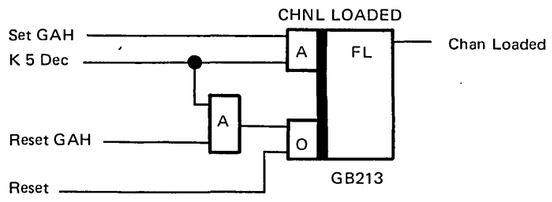
Initial Selection Polling Controls





The only clock in the polling controls is a 350 ns. delay block. By controlling the input gating and the selection of the output level, the controls are made to follow in sequence with the accepted delays for the interface exchange.



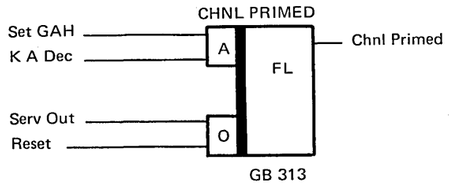


During the initial selection from the start I/O instruction, the hardware initial selection controls select and confirm the device address, issue the first command, and test the returned initial status. With a zero status condition, the operation is allowed to continue into the data transfer. The initial selection sequence microprogram provides the address and the first command along with the interlocking signals. For SIO, after the microroutine has supplied the device address and command, the routine loops with a time-out test until the controls return the signal to indicate that the polling was successful. If the polling is unsuccessful, the operation ends either with a time-out condition or a returned initial status from which the microroutine sets the appropriate condition code.

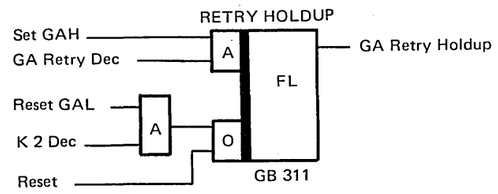
Initial selection on command chain does not use the micro-program loop.

The microroutine first sets the poll-control latch to prevent a channel device from gaining control of the channel through the request-in line. This latch must remain set until after the interface select-out line has been raised. The microroutine next sets the channel-loaded latch to indicate a channel in-use condition for subsequent operations. This latch remains set throughout the channel operation.

Initial Selection Polling Controls

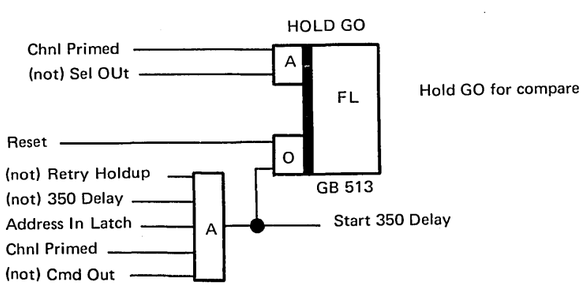


The channel-primed latch is set by the microroutine after the device address has been loaded into the GR register and transferred to the GO register. The output of this latch initiates the polling sequence. The latch remains set throughout the polling, and its reset indicates a successful polling to the microroutine.

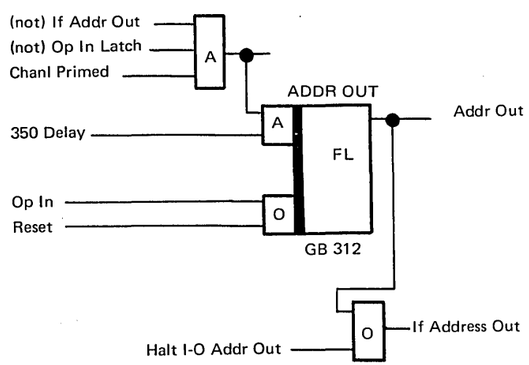


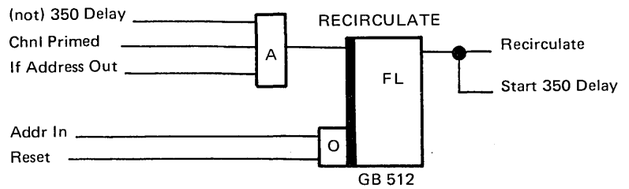
The retry-holdup latch is set at the same time as the channel-primed latch through a special set decode gating. The latch output serves as an interlock to be reset by the microroutine when the command has been placed in the GR register.

If a microword sets a latch, (such as Service-Out) that is reset by hardware asynchronous with the microcode, the holdup latch is also set. The holdup latch prevents the hardware reset to ensure that microretry does not cause the latch to set multiple times and give invalid signal sequences.

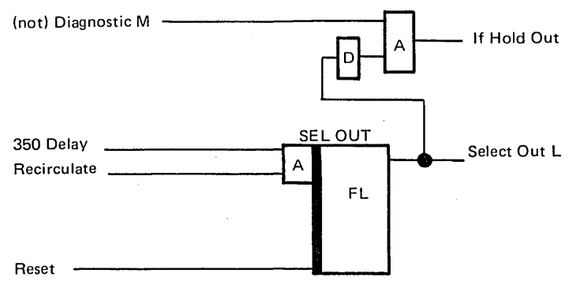


The hold-GO-for-compare latch is set immediately following the set of the channel-primed latch to enter the address in the GO register and to ensure that it remains until the address has been compared. At the same time that the latch is set, the initial gating to set the address-out latch is used to start the 350 ns. delay. At the end of the delay period, the final gating is made to set the latch. The latch output blocks the initial set gating to stop the 350 ns. delay drive; the output then falls at the end of a 350 ns. delay period. The initial delay ensured that the address was on the bus-out lines for the required time before raising the address-out tag line.

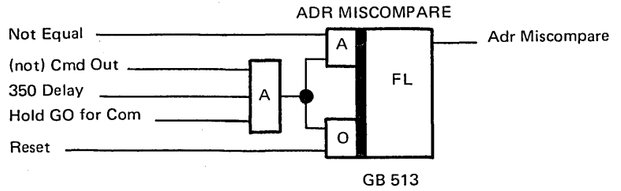




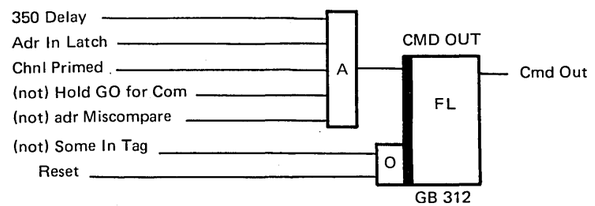
The recirculate latch is set when the 350 ns. delay signal falls. The latch output initiates a new drive for the 350 ns. delay. The prime purpose of this latch is to control the delay output. After the delay period, the output remains in the raised level until after the recirculate latch is reset. A combination of the recirculate latch output and the rise of the delay output gates the set of the select-out latch.

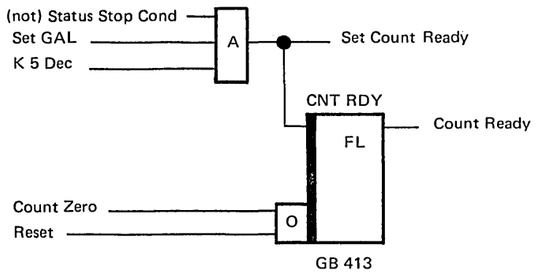


'Hold-out' enables the control unit to recognize 'select-out'. 'Select-out' on means that 'select-out' and 'hold-out' are both on.



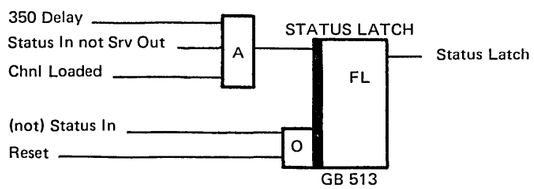
When the addressed control unit raises the address-in tag line, the recirculate latch is reset to start a 350 ns. dealy in the fall of the delay signal. During this period, the output of the address-compare circuits is allowed to settle and deskew bus-in. If the not-equal signal is developed, the address-miscompare latch is set. When the 350 ns. signal falls, the hold-GO-for-compare latch is reset to allow the command to be entered into the GO register. The latch set signal also initiates a new 350 ns. delay. With rise of the delay signal, the command-out latch is set if the addresses compared. The delay provides the necessary bus-out interface delay, and the latch output raises the command-out tag line. The output of the command-out latch also blocks the reset gating of the hold-GO-for-compare latch and the 350 ns. delay drive. The subsequent fall of the delay signal is not used in the sequence.





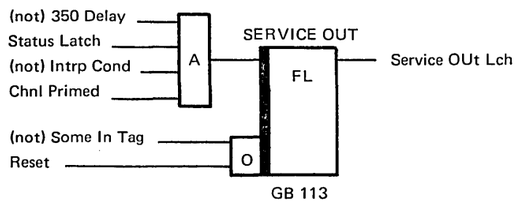
GB 413

At some point in time during the polling sequence, the microroutine has set the count-ready latch to indicate that the data-address and count information has been stored in local storage, and that the channel circuitry is conditioned for data transfer. The latch output is used to reset the poll-control latch.



GB 513

When the addressed control unit raises the status-in tag line, the status-in not service-out signal developed initiates a new drive for the 350 ns. delay. At the end of the delay period, the status latch sets. The latch output blocks the gating that developed the delay drive to start the fall delay. If the initial status is zero, the service-out latch is set at the end of the delay period. The output of this latch raises the service-out tag line on the interface to indicate acceptance of the status. The bus-out lines are forced to zero while the tag line is raised. The output of the service-out latch also resets the channel-primed latch, indicating the completion of the polling sequence to the microroutine.

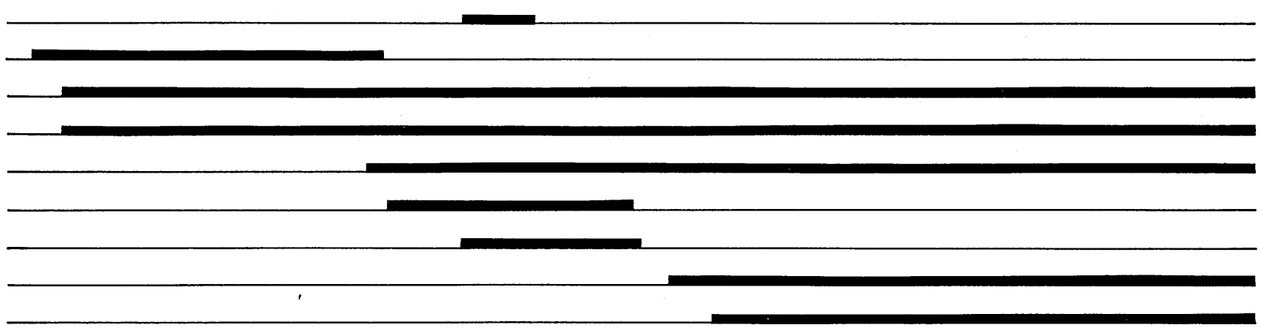


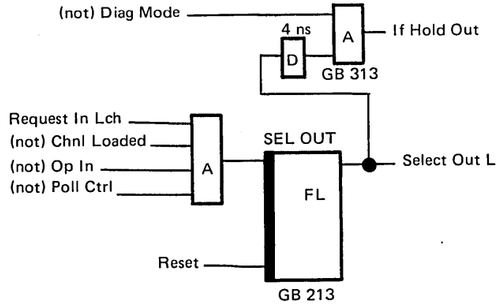
GB 113

If the status is not zero, the hardware does not automatically respond with 'service-out'. Instead, a request is made for the GSES trap, which will handle bad initial status and respond to 'status-in' with microcode.

When the addressed control unit drops the status-in tag line, the status latch and the service-out latch are reset, completing the channel connection. The next response is from the addressed control unit when ready for/with data.

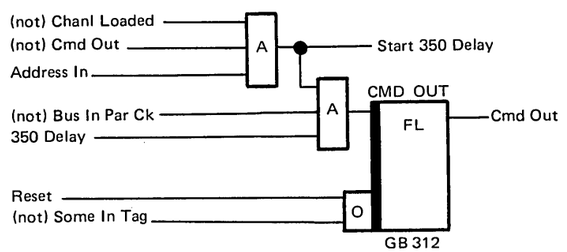
Device Initiated Polling Controls



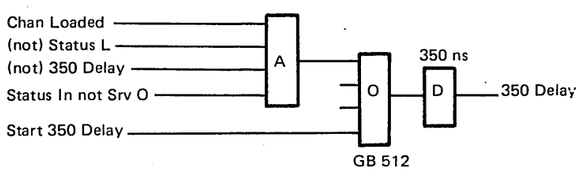


The select-out latch is set with the up-level of the request-in line at the interface. The output of this latch raises the hold-out line on the interface to initiate the polling. The interface select-out line is tied in the raised level. When the control unit receives the select-out and hold-out condition without the address-out tag, it tests its request-in control to determine whether it has placed the request. The control unit that placed the request sets its address on the bus-in lines and raises its address-in tag line.

When the channel is not connected with any of the devices for data transfer, a control unit may request service through the request-in line of the interface. Under this condition, the channel operational-in line is down, and the channel-loaded and the poll-control latches are reset. An automatic polling occurs without any microroutine. The operation ends with a trap request to handle the status. The controls stop with the device address in the GR-register and the status on the bus-in lines.



When the channel controls detect the address-in tag signal, the initial gating to set the command-out latch is completed. This signal drives the 350 ns. delay for the settling delay for the bus-out lines. The bus-out lines are forced to zero for the proceed indication. After the 350 ns. delay period, the command-out latch is set to raise the command-out tag line. At the same time the address on the bus-in lines is set into the GR register for later presentation to the CPU.

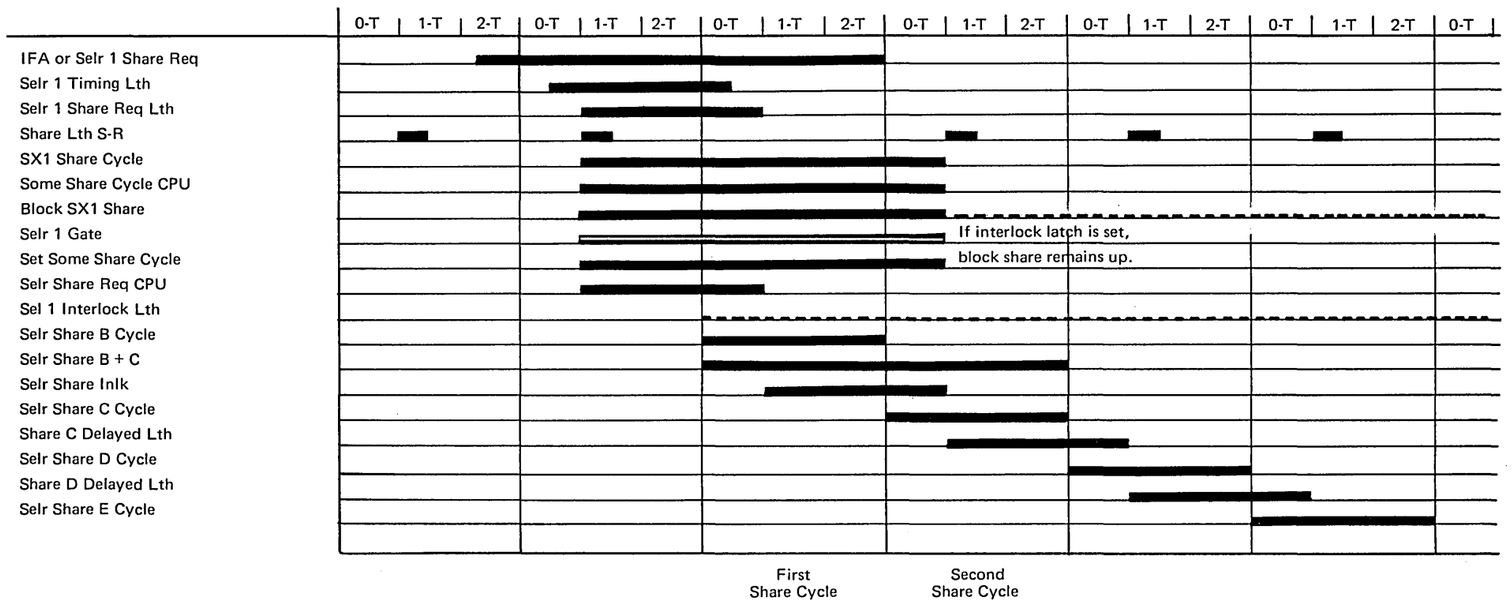


POLLING CLOCK

When the requesting control unit detects the proceed indication, it drops the address-in tag line. This causes the channel control to drop the command-out tag. When the control unit has placed its status on the bus-in lines, it raises the status-in tag line. When the status-in tag is detected, the request for a trap is developed to handle the information. The status latch and the service-out latch are not set. The status remains on the bus until the trap releases them by either accepting the status or calling for the status to be stacked.

Share-Cycle Controls

Share Cycle Timing Chart



All selector-channel data transfers are made using the share-cycle controls. The current microroutine is stopped for two control-word cycles (one memory word) for each data transfer. A data transfer is for one byte on a nonbuffered channel or up to a fullword on a buffered channel. The share-cycle transfer uses a storage word that is forced into the C-register by the controls. During the two cycles, the data is moved to or from storage, the address is incremented by the number of bytes in the transfer, and the count is decremented by the number of bytes in the transfer. The operating control address is held in the N2- and N3- registers and returned to the M2- and M3-registers during the storage 2 cycle of the share cycle.

A control unit requests data service by raising either the service-in or the data-in line on the interface. For the nonbuffered channel, this signal initiates the share-cycle request. The buffered input share-cycle request is developed when the four low-order positions of the buffer become full.

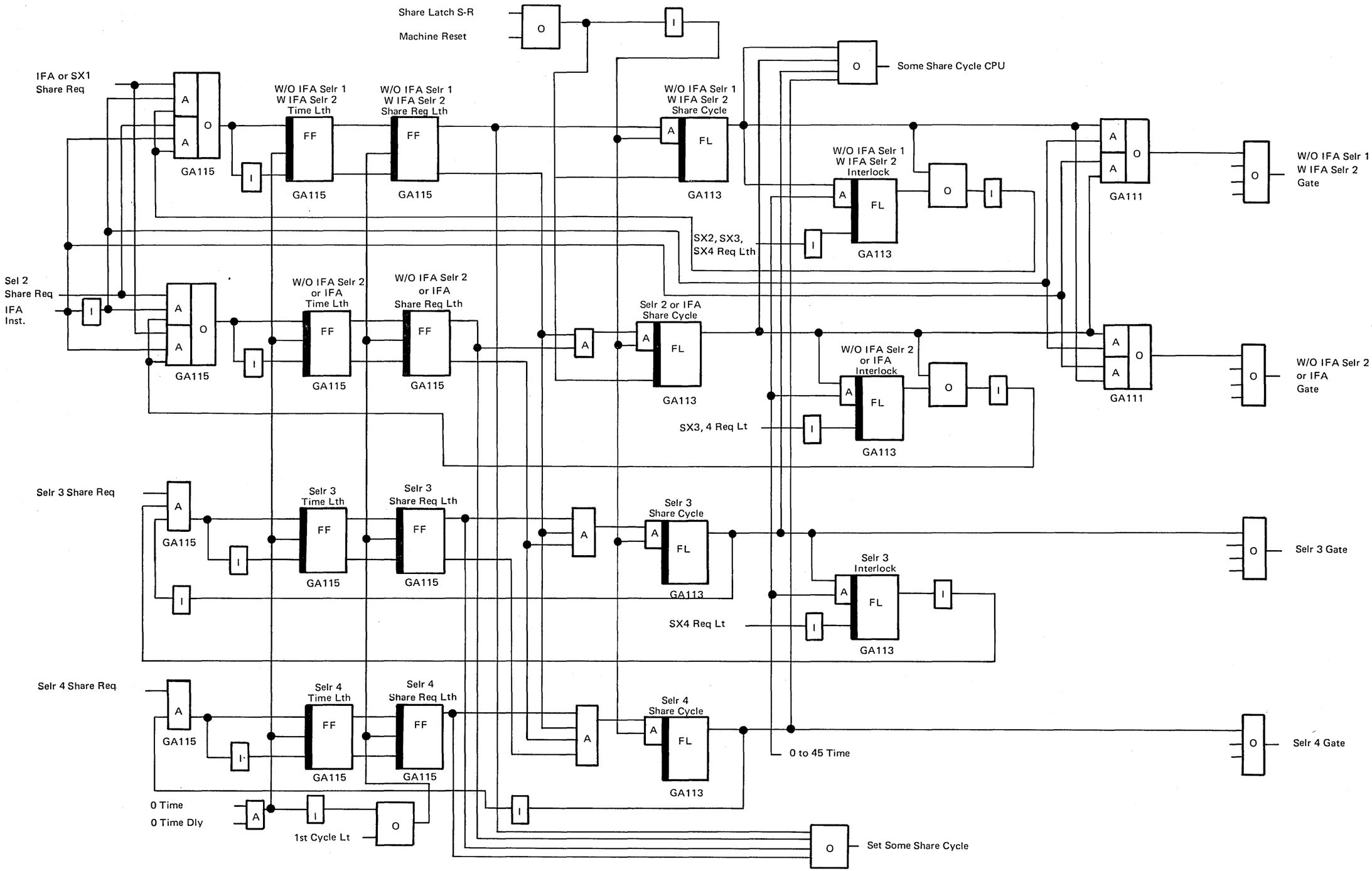
Only one share cycle can be in progress at a time. The channel obtaining service determines which area of local storage is accessed for protect key, count, and data-address information during the share cycle.

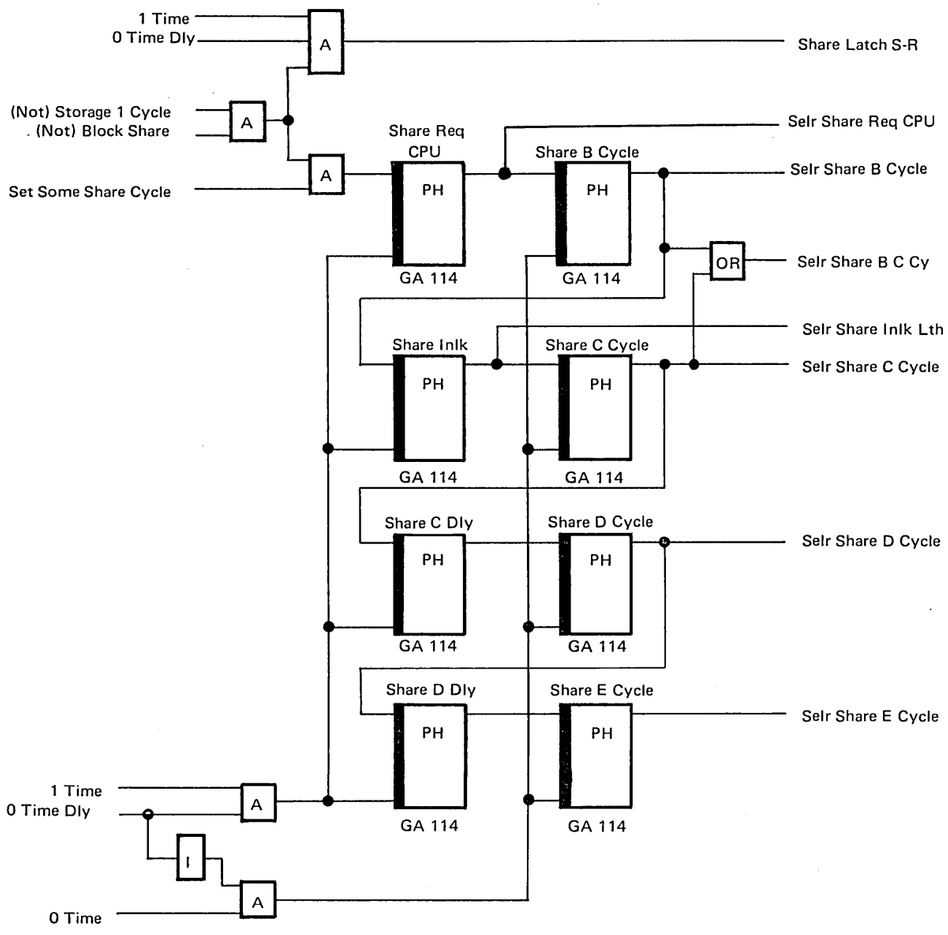
With the IFA option, channel 2 has the highest priority in case of simultaneous requests for share-cycle service; IFA is next in priority, and channel 3 is last. Without IFA, the priority is channels 1, 2, 3, and 4.

No channel can obtain service for a share cycle in the machine cycle immediately following a share cycle for that same channel. Also, each channel is guaranteed that no other channel may have more than one request serviced when a service request is outstanding for another channel. For example, suppose channels 2, 3, and 4 simultaneously request service. Channel 2 obtains service because it has the highest priority. Channel 3 obtains service next, and during the channel 3 share, channel 2 requests another share cycle. Channel 4 is serviced before channel 2 because the channel 4 request has been pending since the last channel 2 request.

When a channel has a share request, the early-share latch for that channel is set except when the channel is already taking share cycles. Any one of the early-share latches being set develops the set-some-share-cycle line to initiate the share-cycle sequence. The share-cycle clock is started at 1-time of the following cycle except when that cycle is the first cycle of a storage word. The clock is started by setting the share-request latch. At the following 0-time, the share-B latch is set to define the first of the share storage cycles. The timing chart shows the relations of the clock outputs and the major control signals.

Share Request Priority Controls





C-Register Share Words (With Buffer)

	C0		C1		C2		C3	
	0123	4567	1234	4567	0123	4567	0123	4567
Output	0100	0000	1011	1000	0000	1100	0000	1000
In Fwd	0100	1000	1000	1000	0000	1100	0000	1000
In Bwd	0100	1000	1000	1100	0000	1100	0000	1000
Skip	0100	1000	1000	0100	0000	1110	0000	1000

C-Register Share Words (Without Buffer)

	C0		C1		C2		C3	
	0123	4567	0123	4567	0123	4567	0123	4567
Output	0110	0000	1011	1000	0000	1100	0000	1000
In Fwd	0110	1000	1011	1000	0000	1100	0000	1000
In Bwd	0110	1000	1011	1100	0000	1100	0000	1000
Skip	0110	1000	1011	0100	0000	1110	0000	1000

With the rise of the share-request (share-A) signal, the highest priority channel having its early-share latch set has its share latch set. The output of the share latch gates the appropriate addresses and controls for the honored channel during the share cycle. The some-share-cycle signal developed from the share latch output blocks the previously addressed control word on the SDBO from entering the C-register and the early decodes. At the same time, a storage word for the share cycle is forced into the C-register. The local-storage and external addresses for the share cycle are forced through, gating with the channel share-cycle line and are not entered into the C-register. The channel command lines gate the appropriate bits into the C-register for the read/store control and the direction of the address update. The buffer memory flag bits control the address update and the count decrement.

During the first share cycle, the encoded local-storage address contains the data address of the specified channel. This data address is set into the M-register for the storage addressing. The address is then incremented or decremented by the amount equal to the sum of the buffer memory flag bits and destined back to the originating local-storage address. Depending on the direction of the data movement, the specified channel GBUF word is gated and the appropriate main-storage gates are set. The data follows the normal data paths either from the SDBO to the external bus or from the external bus to the SDBI. In the case of the read function, the data movement occurs at the start of the second share cycle.

During the second share cycle, the encoded local-storage address is incremented by one for the count of the specified channel. The count is decremented by an amount equal to the sum of the buffer memory flag bits and destined back to the originating local-storage address. During the latter portion of the second share cycle, the Z-register is sampled to set the GCL latches to indicate the remaining count to the buffer controls.

A share request for a second channel can be initiated during the storage 2 cycle of a share 2 cycle to follow immediately. No CPU cycle occurs between the share 2 cycle of the first share cycle and the share 1 cycle of the second share cycle. The interlock that ensures that the second channel is allowed a request allows one CPU cycle to occur between share cycles for the same channel. The third and fourth share cycle lines from the clock are used in conjunction with the second-cycle destining, and the retry controls. The operating channel share latch is reset at 1-time delay of the second share cycle after the local-storage access for the count.

OPERATIONS

The selector-channel data-flow paths follow those of the basic data flow shown previously in this manual. During the instruction setup and normal microroutine operation, the basic CPU paths are used in the normal manner. It is only during the share cycle used to move channel data that the flow is shown.

When the share-cycle request is honored, the CPU has already transmitted the address for the next control word to the storage unit. That address is also in the N-register, where it remains unchanged during the share cycle. The addressed data from control storage is blocked from entering the C-register. The selector-channel share-cycle controls force a storage word into the C-register to control the operation.

The data address for the main-storage access during the share cycle is read from the GDRL-register in expanded local storage into the B-register and the M-register. During the first share cycle, the address is updated (incremented or decremented) by the number of memory flag bits set for the transfer. The updated address returns to expanded local storage by way of the Z-register, the D-register, and the SDBO assembler. The new address is also available on the EBI to set the latches of the GDRL-register.

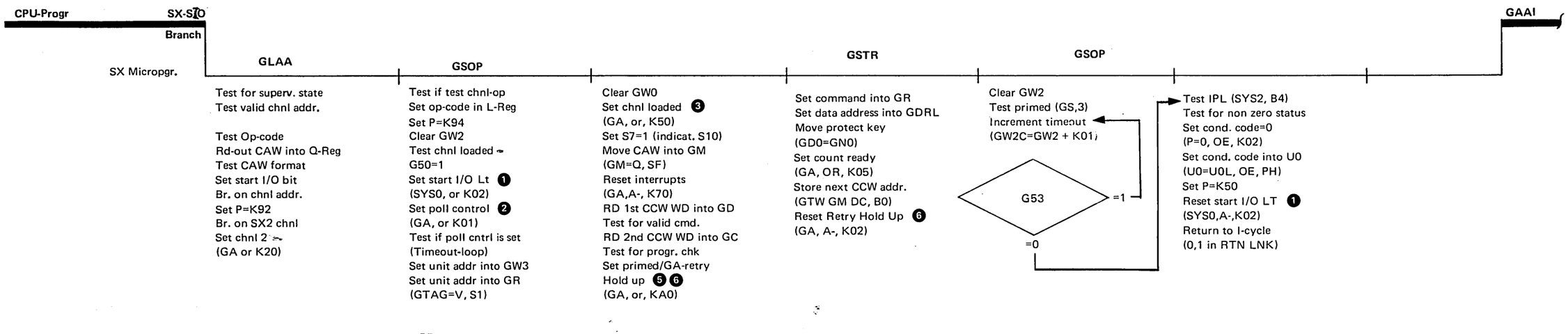
The share-cycle write data moves from main storage to the GBUF external by way of the SDBO, the SDBO preassembly latches, the byte-selection circuits, the SDBO assembler, and the EBI. The memory flag bits set determine which of the bytes read into the buffer. Read data moves from the GBUF external to main storage through the EDBO, the A-register, the A-byte assembler, and the SDBI. The memory flag bits set determine which of the bytes are stored.

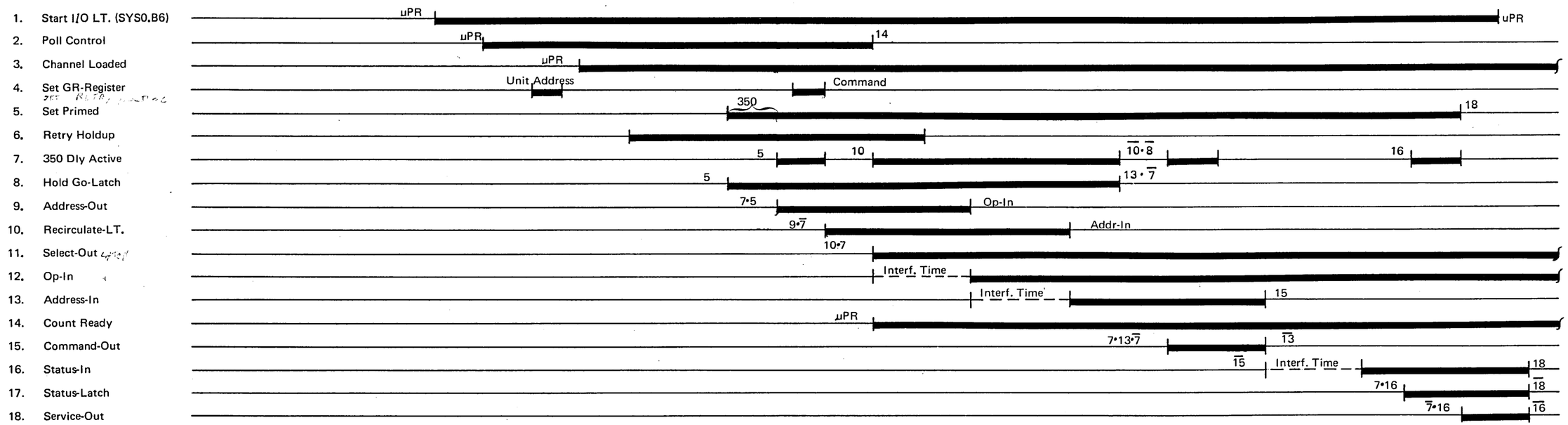
During the second share cycle, the count for the channel operation is read into the B-register from the GC-register in local storage. The count value is decremented by the number of memory flag bits set for the transfer. The updated count returns to local storage by way of the Z-register and the D-register. The new count is also available from the Z-register to set the latches of the GCL-register in the buffer externals. These latches are originally set from the EDBI when the CCW count was destined.

The control address left in the N-register at the start of the share cycles is moved to the M-register through the M-register assembler. The address remains in the N-register for use if another share-cycle request is honored immediately.

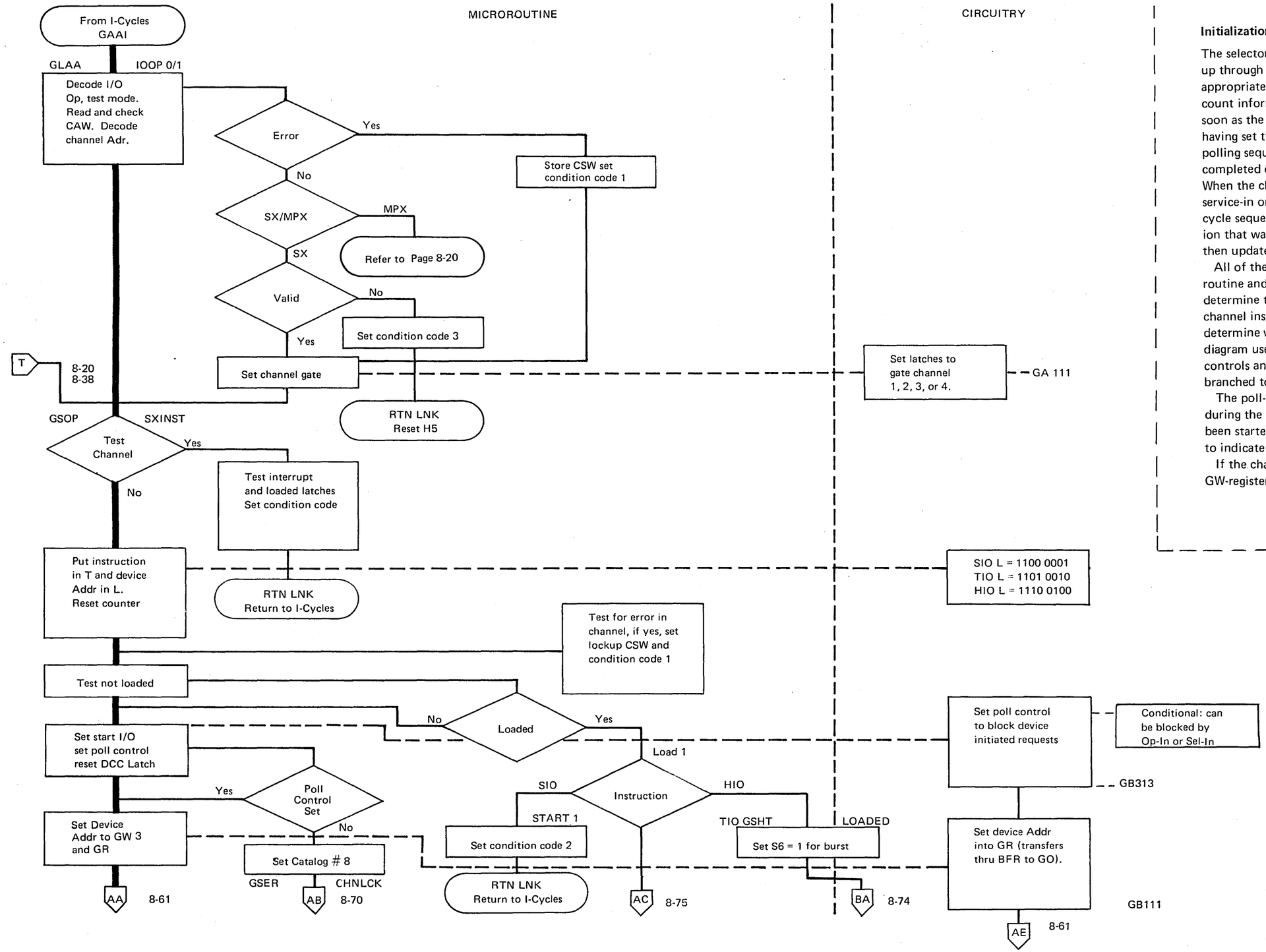
Start I/O

Operational Overview





Selector Channel Start I/O (Part 1 of 3)



Initialization

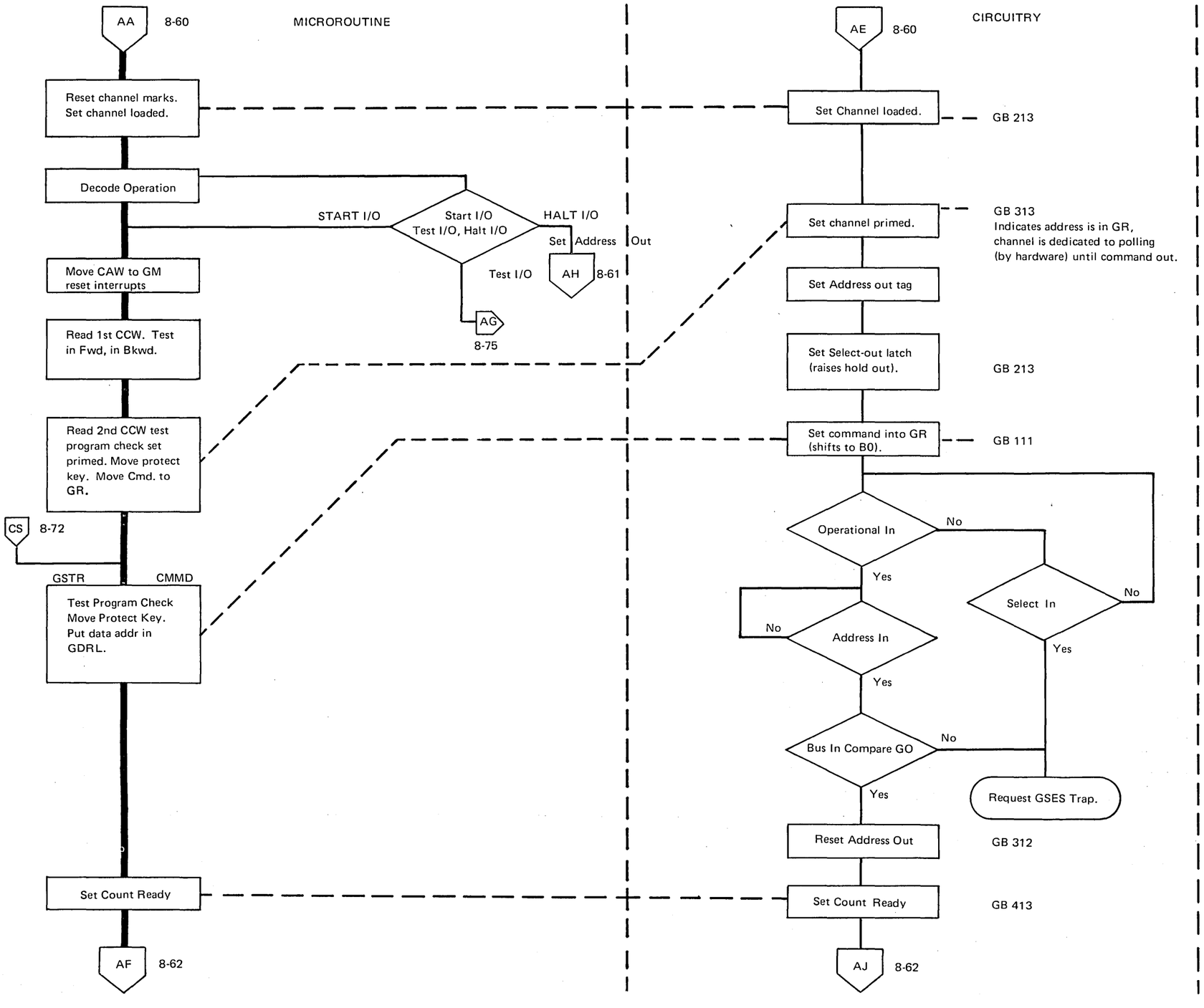
The selector-channel data-transfer operations are initiated and set up through a series of microroutines. During this period, the appropriate hardware controls are set and the data address and count information are set into assigned local-storage addresses. As soon as the start signal is given to the hardware controls after having set the device address, the controls initiate an automatic polling sequence. The microroutine waits until the polling is either completed or rejected before returning to the I-cycles routine. When the channel device is ready for data transfer, it raises the service-in or data-in line to signal. Data is handled with the share-cycle sequence as required. The data-address and count information that was stored in local storage is used for the operation and then updated.

All of the I/O instructions are decoded in the I-cycles (GAAI) routine and enter a common I/O decode (GLAA) routine to determine the channel and enter the CAW data. The selector-channel instruction is further decoded in the GSOP routine to determine which of the I/O instructions is being executed. This diagram uses the start I/O instructions to show the hardware controls and the polling sequence. The other instructions are branched to separate diagrams as indicated.

The poll-control latch is set to block channel-initiated action during the sequence. If a channel-initiated sequence has already been started, the latch cannot be set and the operation is branched to indicate the error condition before returning to I-cycles.

If the channel is available, the device address is set into the GW-register for later use in the channel operation.

Selector Channel Start I/O (Part 2 of 3)



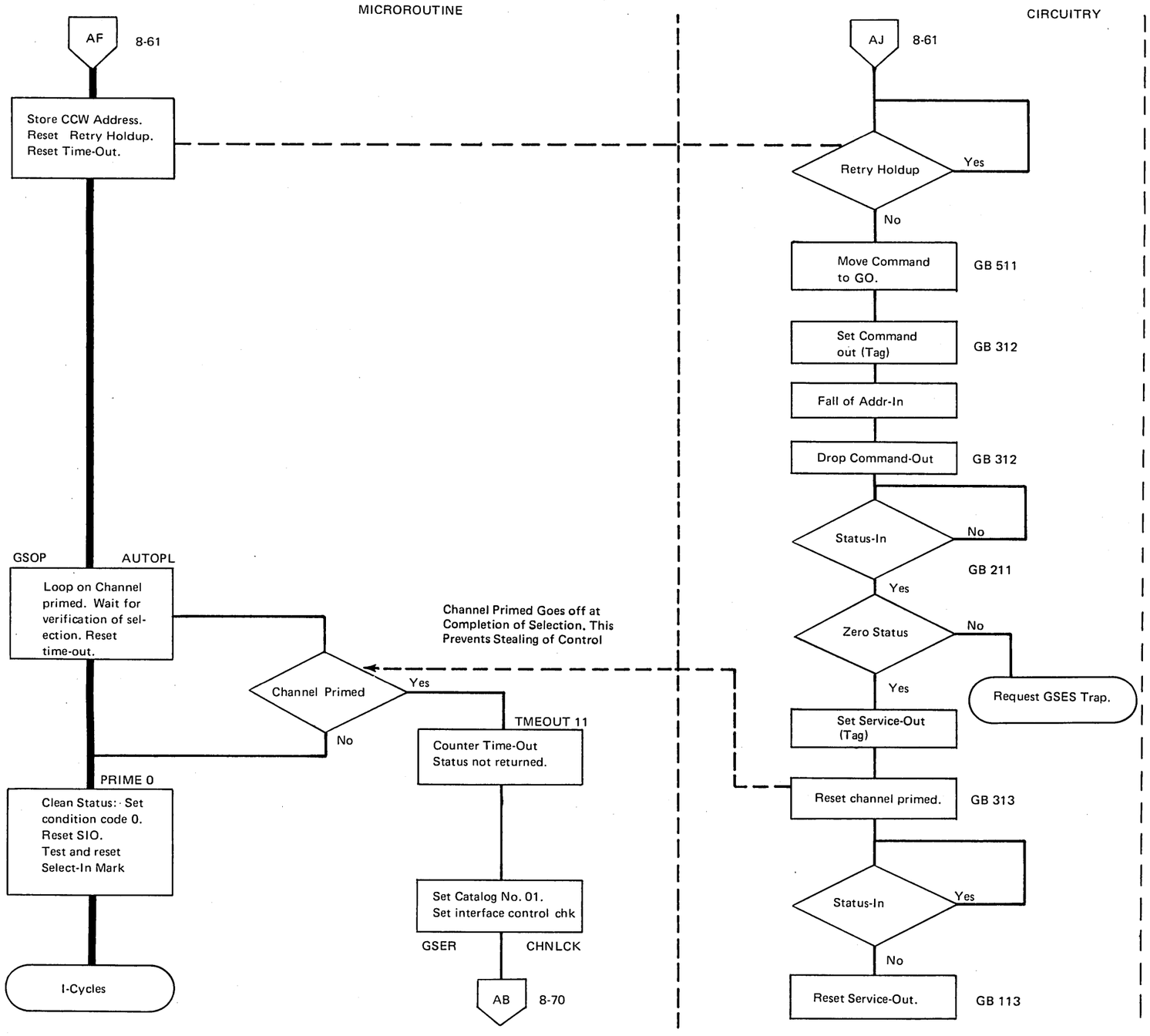
The hardware channel loaded latch is set to gate the controls for the initial sequence polling. The final test for the operating instruction is made, and those other than start-I/O are branched. The device address is set into the hardware GR-register, from which it is moved to the GO-register to set the bus-out lines. The CCW address and key from the CAW are set into the assigned GM-register.

The operation continues in the GSTR routine by reading the first CCW word containing the data address into the assigned GD-register and into the GDRL hardware latches. The channel-primed latch is then set to initiate the polling sequence. The retry holdup latch is set at the same time to serve as a later interlock between the hardware and the microroutine. The channel-primed signal raises first the address-out line and then sets the select-out latch to start the polling with the bus-out address. The select-out line is always raised, but the latch output raises the hold-out line, making the select-out line effective.

During the polling initiation, the microroutine continues by reading in the second CCW word. This information containing the count field is set into the assigned GC-register and the GCL hardware latches. The command is set into the GR-register ready to move into the GO-register when required.

The controls wait for a channel response. The responding control unit raises the operational-in line; and then after placing its address on the bus-in, raises the address-in line. The address-out line is dropped, and the incoming address is compared with that in the GO-register for a match. During this period, the microroutine sets the hardware count ready latch to indicate that the data-transfer information is set.

Selector Channel Start I/O (Part 3 of 3)



After storing the next CCW address in the GM-register, the microroutine resets the retry holdup latch to indicate that the command is ready. The microroutine waits in a time-out loop until the channel-primed latch falls, indicating that the polling is complete.

When the retry holdup latch is reset, the controls can continue the polling sequence following a matching address. The GO-register is reset, allowing the command to read in to set the bus-out lines. The command-out tag line is then raised to signal the control unit. The fall of address-in causes command-out to fall, and the GO-register output is forced to zero. The control unit normally responds with the initial status on the bus-in lines and raises the status-in tag line. If the status is zero, the polling was successful and the operation is allowed to continue. The service-out tag line is raised to indicate the acceptance. At this point, the channel-primed latch is reset to signal the microroutine. The controls wait for the fall of the status-in tag line before dropping the service-out line. If the status returned is not zero, the controls initiate a trap request to handle the condition. The next response is from the control unit on the channel when it is ready for data transfer.

Check Facilities

The selector/block-multiplexer channel check latches are located in the hardware as shown on second level diagram 51. These latches are set either by the microprogram word set GA function as a result of errors detected by the channel microprogram routines, or by hardware as a result of hardware-detected errors.

The output of the check latches goes to the GE (channel status) byte of the GSTAT external word facility. The GE byte also indicates program-controlled interrupt.

The channel control check latch can also be set by certain CPU share-cycle errors.

The handling of selector-channel checks varies depending on the cause of the check, the operation in process at the time of occurrence, etc. Some check conditions require use of the extended-logout capability while others do not. Also, there are three degrees of CSW store: Short CSW—Initial selection problems; Partial CSW—Device End status; and Full CSW—Channel End Status.

Channel Check During Selection:

1. Prepare log in control storage and store ECSW.
2. If log is allowed, store log in main-storage area designated by the IOEL pointer.
3. Set condition code 1 and store short CSW.

Channel Check During Data Transfer:

1. Prepare log in control storage.
2. Set interrupt latch.
3. During next interrupt, store ECSW and log in designated main-storage area.
4. Store partial or full CSW.

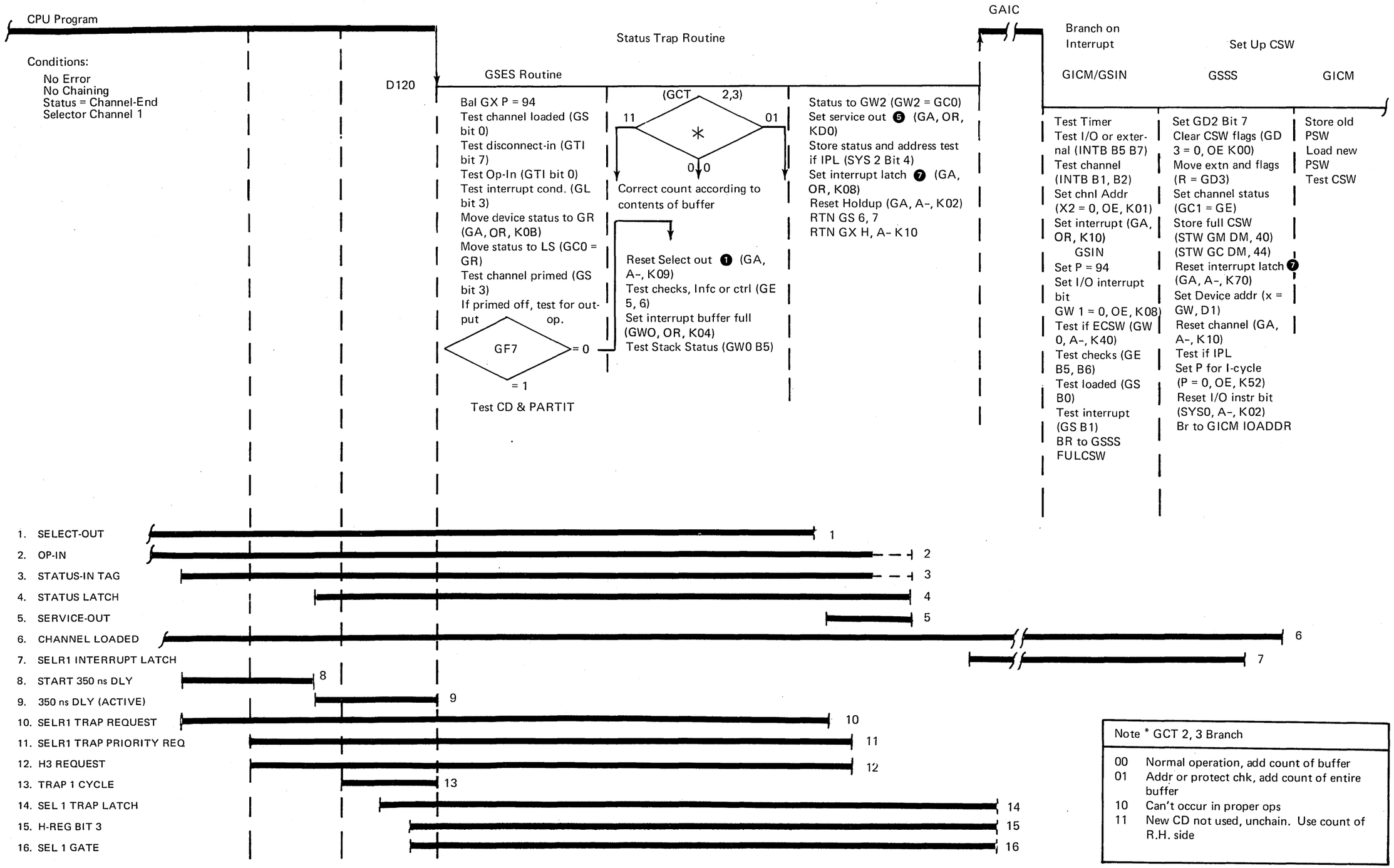
Address or Protection Check During Data Transfer:

1. Set program check or protection check bit.
2. Reset channel flags.
3. Normal channel-end handling: full CSW store with program check or protection check.

If the poll-control latch fails such that it cannot be set, the microcode cannot control the channel hardware, and therefore takes the entire system to a hard stop.

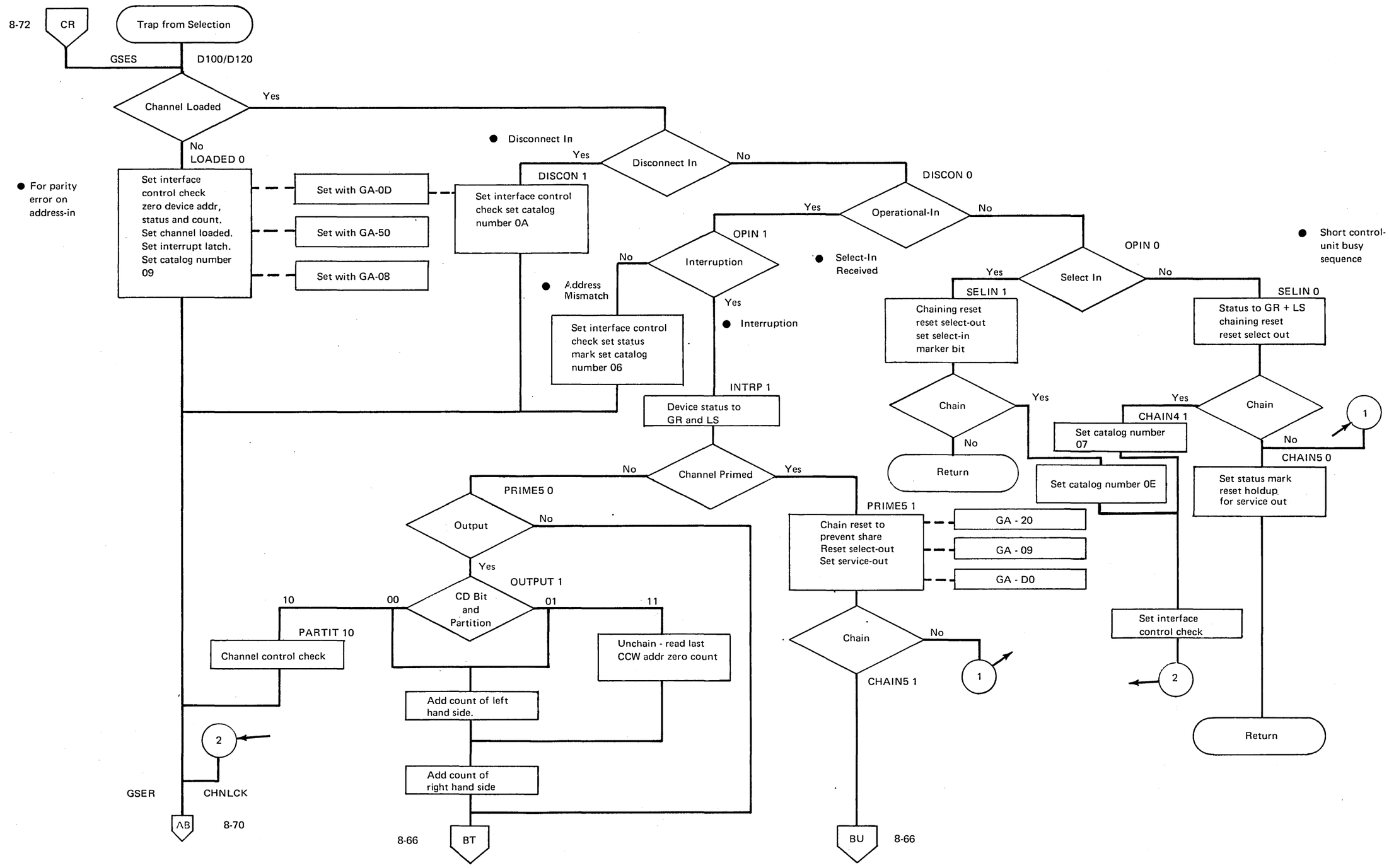
The selector- and block-multiplexer channels of the Model 145 monitor the interface for the expected tag sequences. If a control unit were to erroneously activate an unexpected tag in addition to maintaining a normal interface sequence, the channel *may not* cause a check due to the erroneous tag but might proceed as if the erroneous tag did not exist.

Selector-Channel Status Handling



Note * GCT 2, 3 Branch	
00	Normal operation, add count of buffer
01	Addr or protect chk, add count of entire buffer
10	Can't occur in proper ops
11	New CD not used, unchain. Use count of R.H. side

Selector-Channel Exceptional Status Trap (Part 1 of 2)



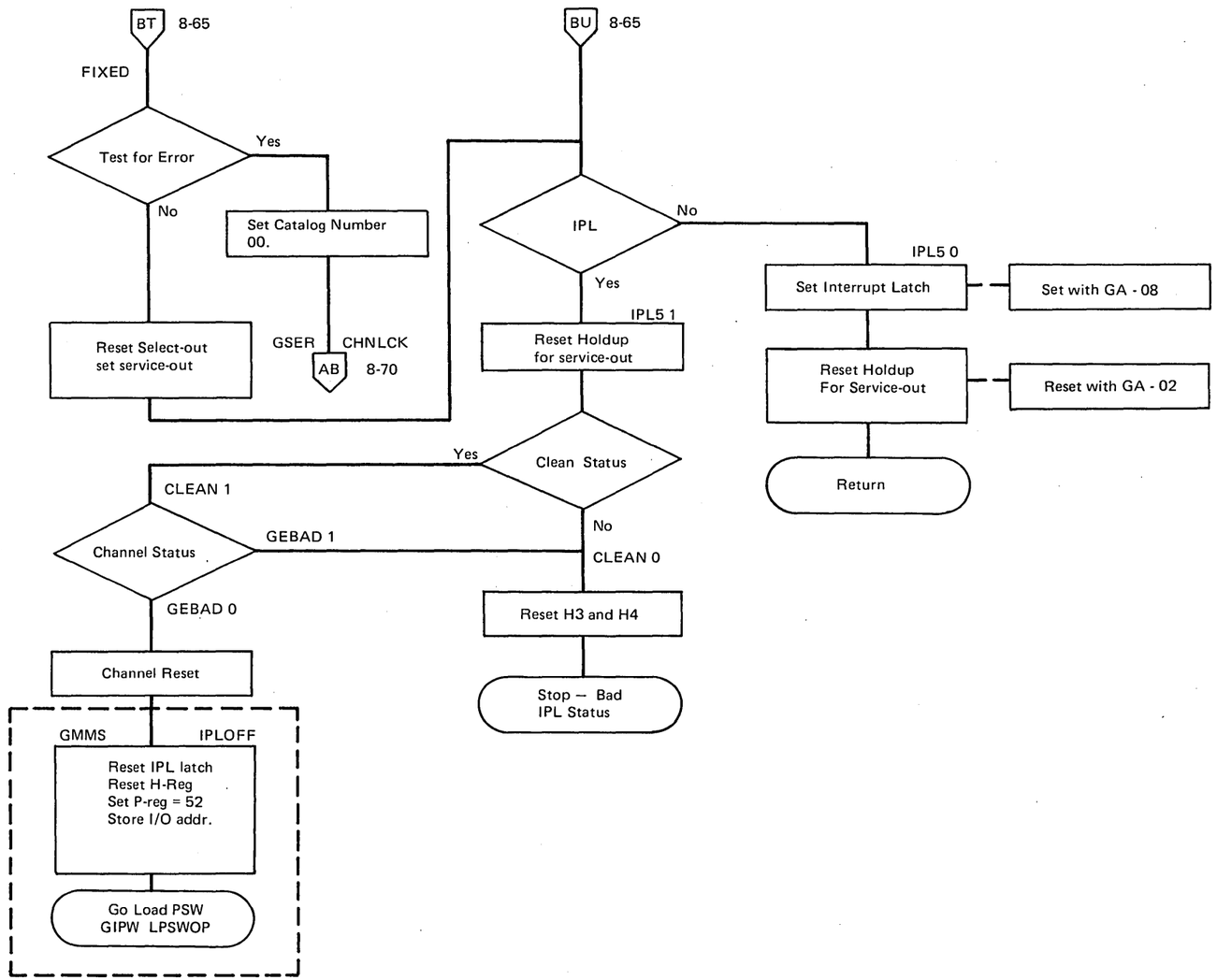
Selector-Channel Exceptional Status Trap (Part 2 of 2)

The Exceptional Status Trap (GSES routine) is used to handle exceptional conditions that occur during selection. The trap address is D120 for channels 1, 2, and 3 (without IFA) and D100 for channel 4, or for channels 2 and 3 with IFA.

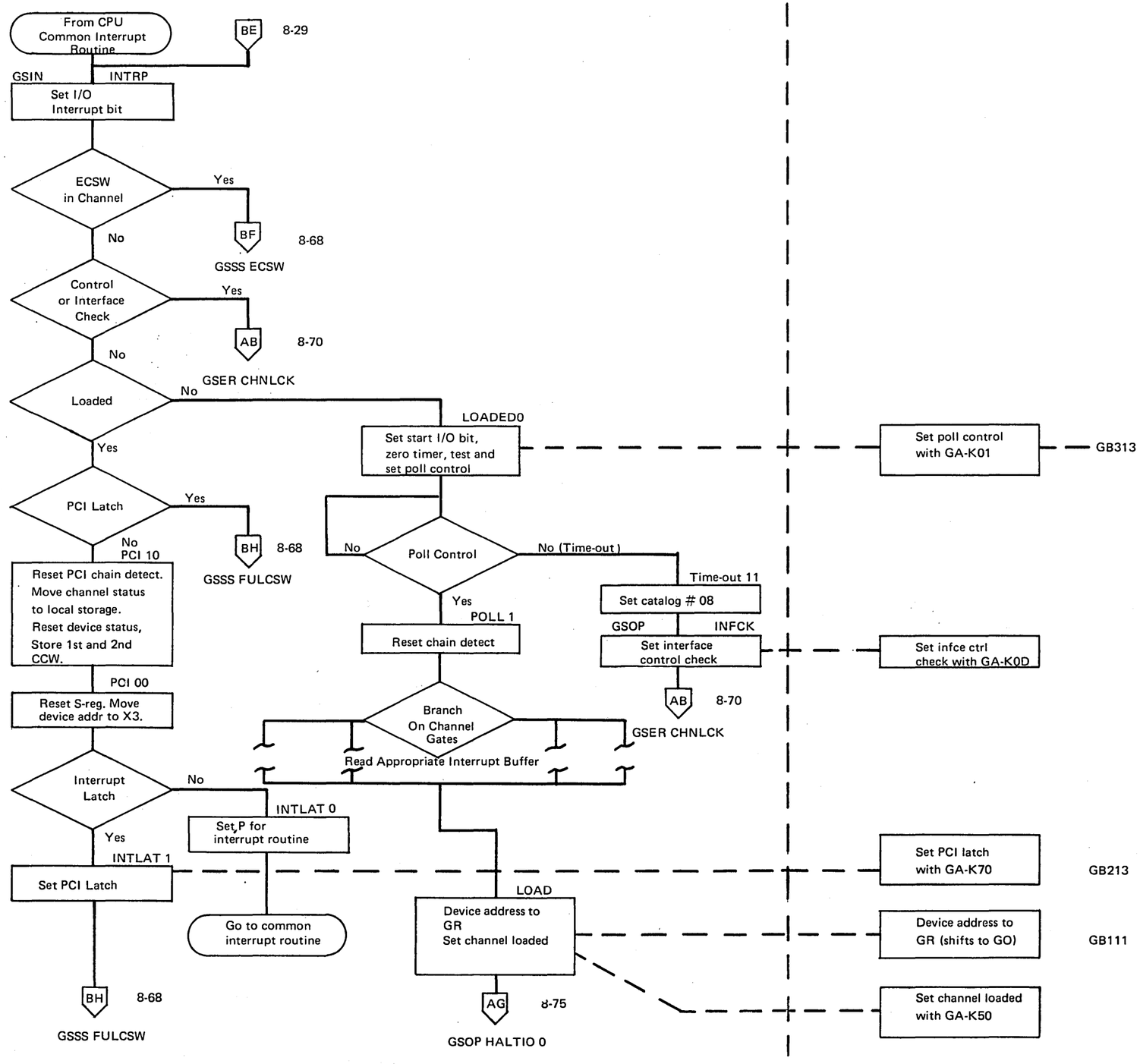
The exceptional status trap is also used to handle normal ending status. Refer to the chart Selector Channel Status Handling. This diagram shows a normal ending status handling sequence. The microprogram objectives are shown along with the state of pertinent latches and signal lines, tags, etc.

The interruption handling is also shown from the point of GAIC routine where Branch on Interruption occurs through the storing of the CSW and loading of a new PSW.

The IPL ending (IPLOFF branch to the GMMS routine) is duplicated on this chart because of its relationship to the status handling trap. Refer also to the IPL diagram. This particular microprogram sequence is the common ending routine for IPL operations on either byte-multiplexer channel, selector mode channel, block-multiplexer mode channel or integrated file adapter.



Selector-Channel Interruption Handling

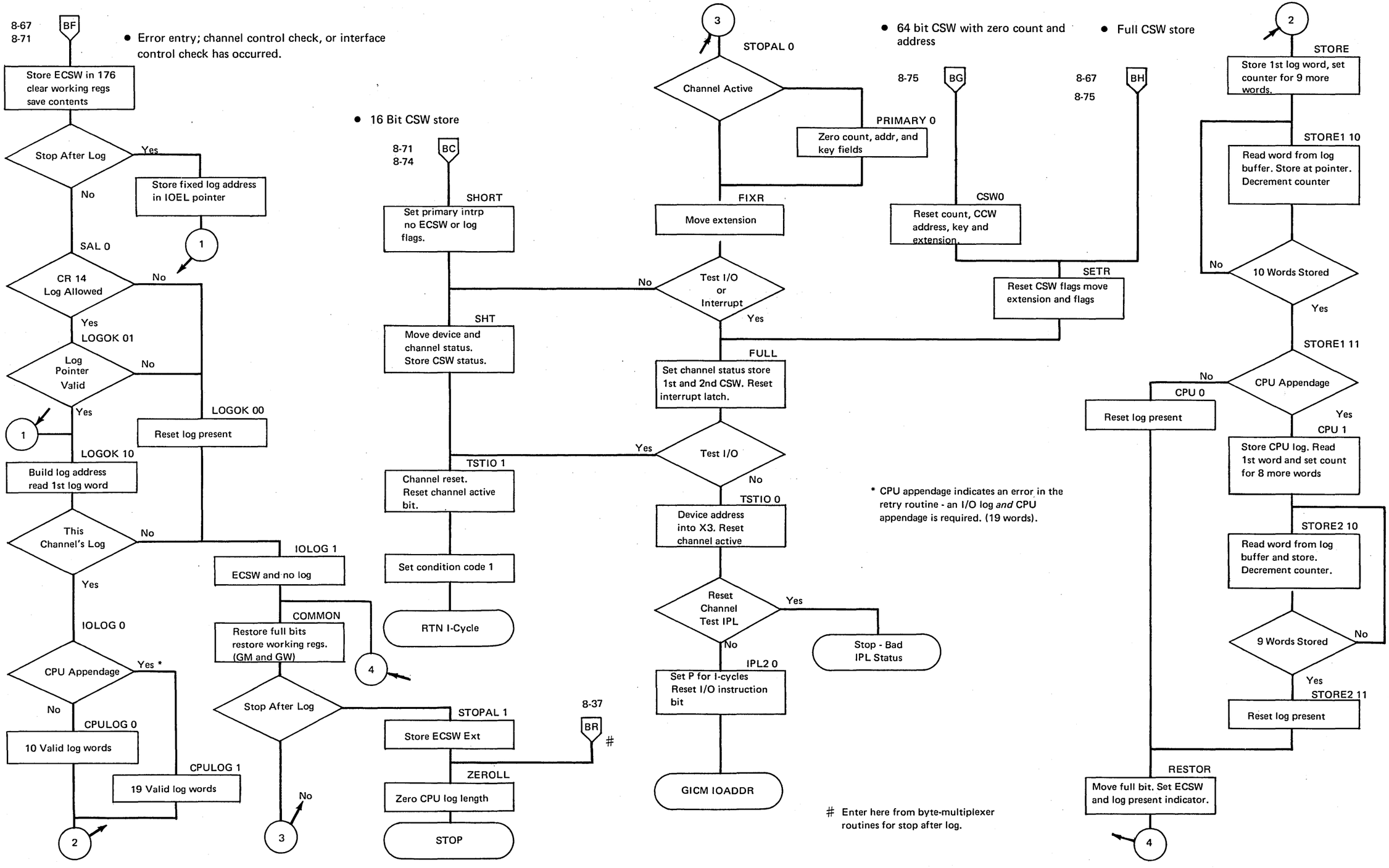


Entrance is from the CPU for handling selector channel I/O interrupts.

If the interrupt is taken, return to GICM. IOADDR with P = 52, S2 and S7 = 00. X2 = channel address and X3 = unit address.

If the interrupt is not wanted because of unavailability of the device, return to GIPW. CHWAIT 00 with P = 52 and S = XX.

Selector Channel Status Store



* CPU appendage indicates an error in the retry routine - an I/O log *and* CPU appendage is required. (19 words).

\# Enter here from byte-multiplexer routines for stop after log.

Channel Logout

Channel logout facilities provide a means of preserving for analysis a record of pertinent conditions that exist at the time a malfunction involving channels occurs. Logout is basically a "snapshot" of conditions that existed at the time the error occurred.

There are essentially two levels of channel logout, depending on the nature of the malfunction and on the setting of control register 14 bit 2. The bit determines, under control of the program, whether extended logouts can occur as part of the I/O interruption, or whether only limited channel logouts are allowed.

I/O Communications Area

The I/O Communications Area (IOCA), a permanently assigned area in main storage, is utilized by the logout operation for limited channel logout, to point to the address of the extended log, and for other logout control information. The IOCA is located in main storage 106-191 (A0-BF).

Channel ID: (Locations 168-171.) When stored during the execution of the Store Channel ID instruction, this area contains information that describes the addressed channel.

I/O Extended Logout Pointer: (Locations 173-175.) The I/O Extended Logout (IOEL) pointer field is program-set to designate an area to be used by channels to store the extended logout information. The low-order three bits of the pointer are ignored so that the extended logout always begins on a doubleword boundary. Logout information may be stored in the IOEL area, only when the IOEL mask bit (CR14 bit 2) is set to 1.

Limited Channel Logout: (Locations 176-179.) The limited channel logout field is also called the ECSW. This field contains model-independent information related to hardware errors detected by the channel. The ECSW is used to provide detailed machine status when errors have affected I/O operations. The field is accessible to the CPU program. The ECSW field may be stored only when the CSW or a portion of the CSW is stored. It may (or may not) be accompanied by the extended channel logout.

(Locations 180-184): Not used.

(Locations 185-187): Reserved for future use.

(Locations 188-191): Not used.

Refer to the sections "Byte-Multiplexer Channel" and "Selector Channels" for logout information as it applies to individual channel types.

Stop After Log

The stop after log switch on the console overrides the control register and allows extended channel logging when there is an interface control check or channel-control check.

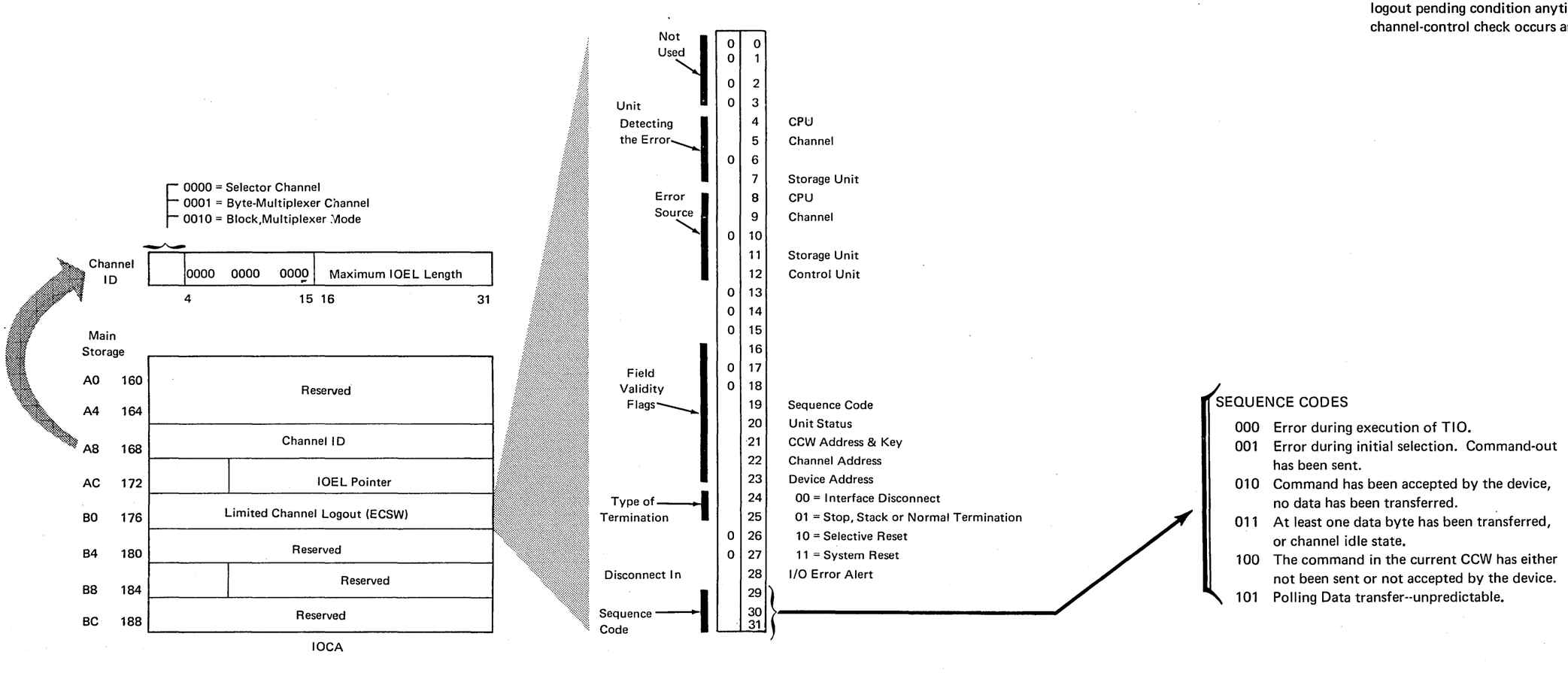
Stop after log causes the following action on a channel error:

- The IOEL pointer is set to 100 (hex).
- The ECSW is stored in the IOCA and the extended log is stored at 100.
- The CSW is not stored; the CSW information is extracted from the extended log by SEREP.

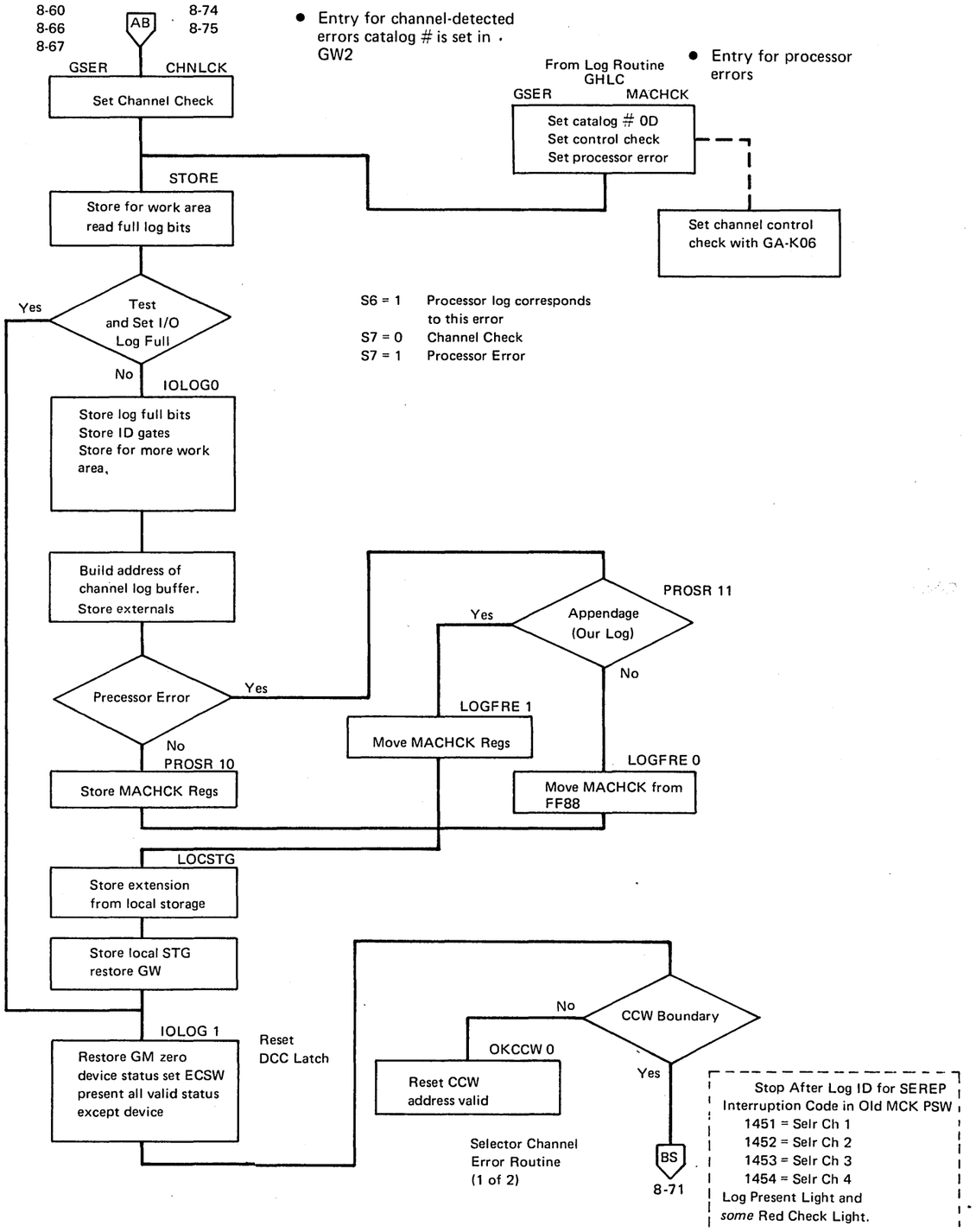
Both the ECSW and the extended channel log are always available when running in the stop after log mode.

The CPU places an identifier in main-storage locations 32 and 33 (machine check old PSW) to identify the failing channel. The identification is 1450-1454 for channels 0 through 4 respectively and 145F for the CPU. The system stops with at least one red light and the log present indication.

Note: The selector/block-multiplexer channel goes into the logout pending condition anytime an interface control check or channel-control check occurs and the sequence code is 011.



Selector Channel Error Routine (Part 1 of 2)

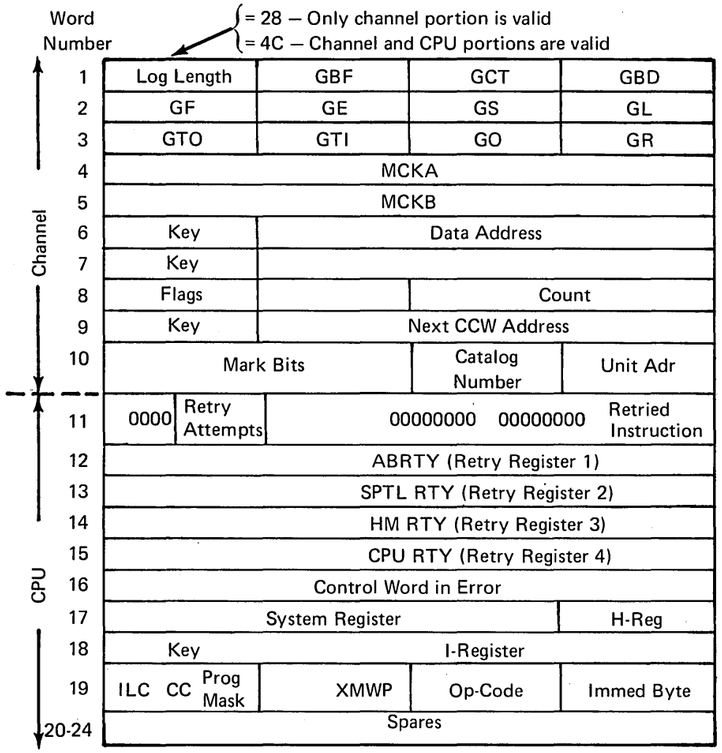


Selector/Block-Multiplexer Error Logout (GSER)

The extended logout routine is the same whether operating in selector mode or block-multiplexer mode. The channels attempt to logout any time a CSW is stored with either an interface control check or a channel control check. The logging is done both at I/O instruction under control of CR14 bit 2. A channel error does not cause a machine check interruption.

The channel log is located at the address designated by the IOEL pointer in location 172. The low three bits of the pointer are ignored, thus putting the log on a double-word boundary. If the pointer is set such that any part of the maximum length log would be out of storage, no part of the log is stored.

The contents and format of the log are as shown:



*General Working Registers; this usage only true when count ready latch is on.

The first byte indicates the number of bytes in the log. In the case of an interface error, the byte indicator designates that the first 40 bytes are valid. For channel control checks, a value of either 40 or 76 is stored, depending on the cause of the check.

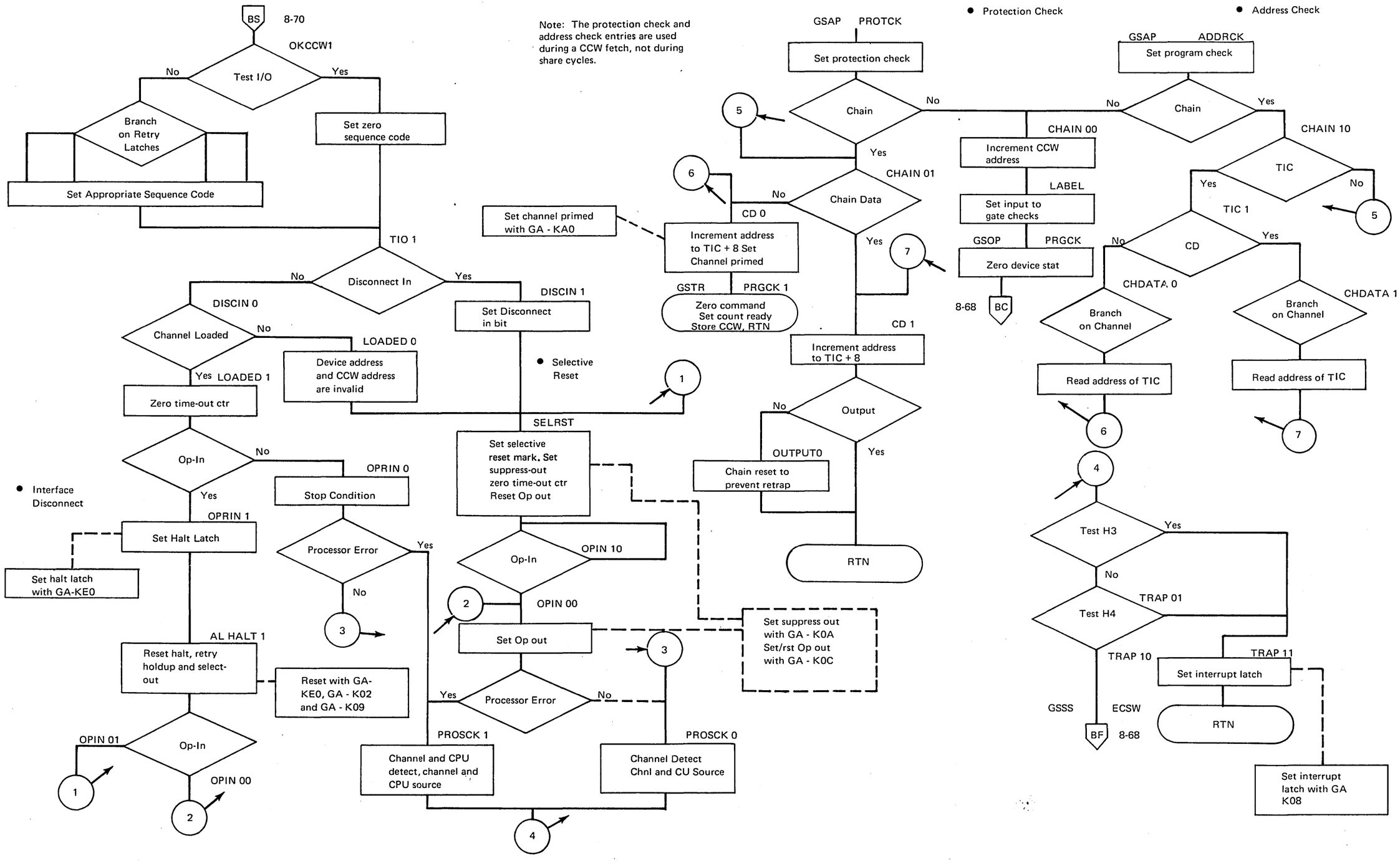
The next 11 bytes are the external registers of the chaining channel. Words 4 and 5 contain the machine check registers that give the status of the CPU error detection circuitry. Words 6 through 10 give the contents of local storage words 11 through 19 are the CPU registers that are stored in the case of unsuccessful microprogram retry in the channel.

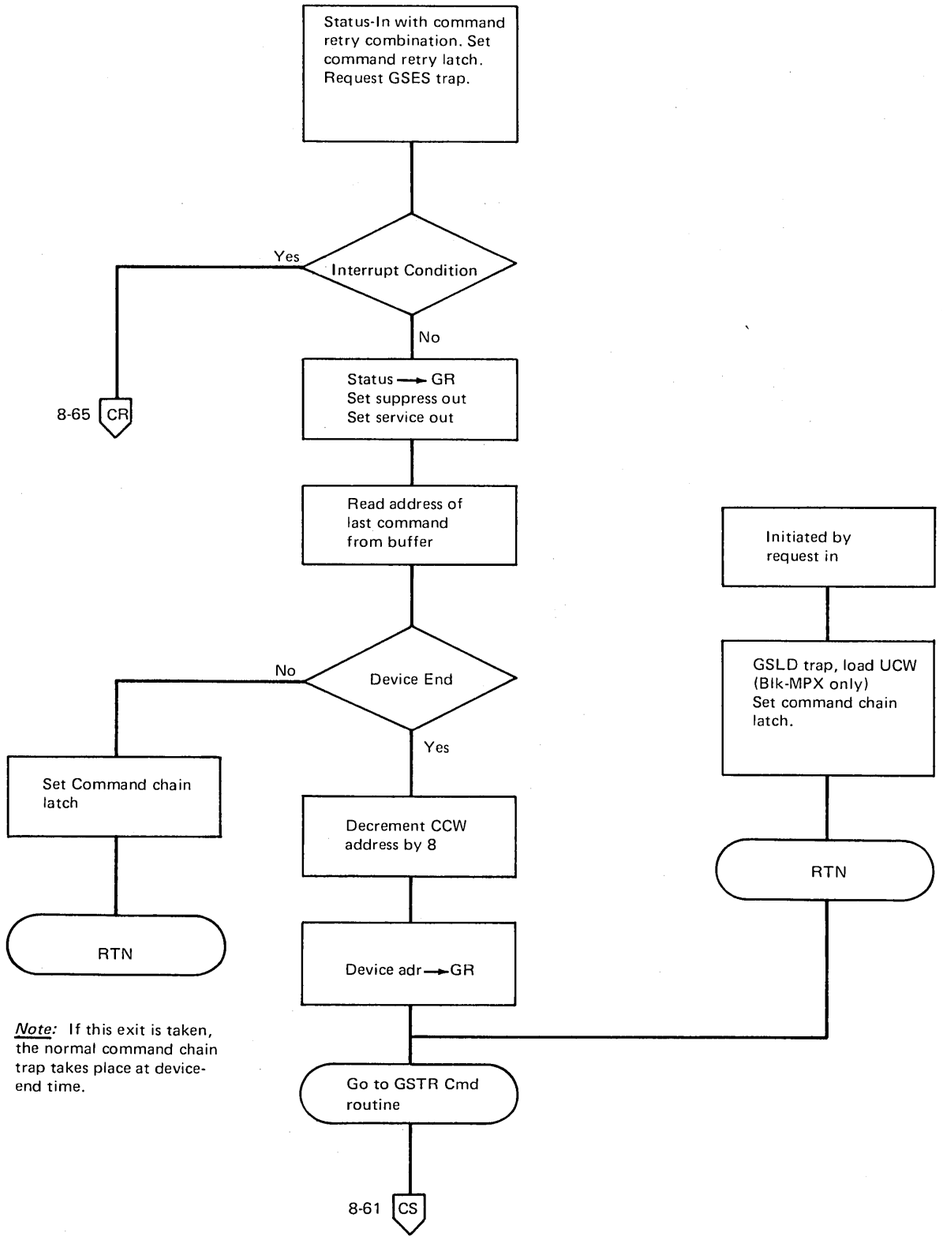
Catalog Numbers

00 — Indeterminate
01 — Automatic Selection Failed
02 — Halt Stop Will Not Set
03 — Microcode Select Failed on Halt
04 — Interface Disconnect Failed
05 — Selective Reset Failed
06 — Address Check on Channel-Initiated Selection
07 — Short Busy on Chaining
~~08 — Poll Control will not set (soft)~~ N/A
09 — Address Parity Error on Ctrl Unit Initial Selection
0A — Disconnect-In Received
0B — Hard Set of Poll Control Failed
0C — Unused OPS INTERRUPT
0D — Unseccessful Microprogram Retry
0E — Select-In on Chaining
0F — Status-In Parity Error

I/O Log Start Address:
Stop after Log: MS 100 (hex)
IOEL Pointer : MS AD-AF

Selector Channel Error Routine (Part 2 of 2)





Note: If this exit is taken, the normal command chain trap takes place at device-end time.

Command Retry Feature

Command retry is a channel-control unit procedure that can cause a command to be retried without requiring an I/O interruption. It is initiated by the control unit with a unique combination of status bits.

A control unit may request the retry of a command in order to recover from a transient error or because the state of the control unit or device prevented the execution of the command when it was previously issued.

A channel, upon accepting a request for command retry repeats the execution of the channel program, beginning at the last command executed.

I/O Interface Sequence

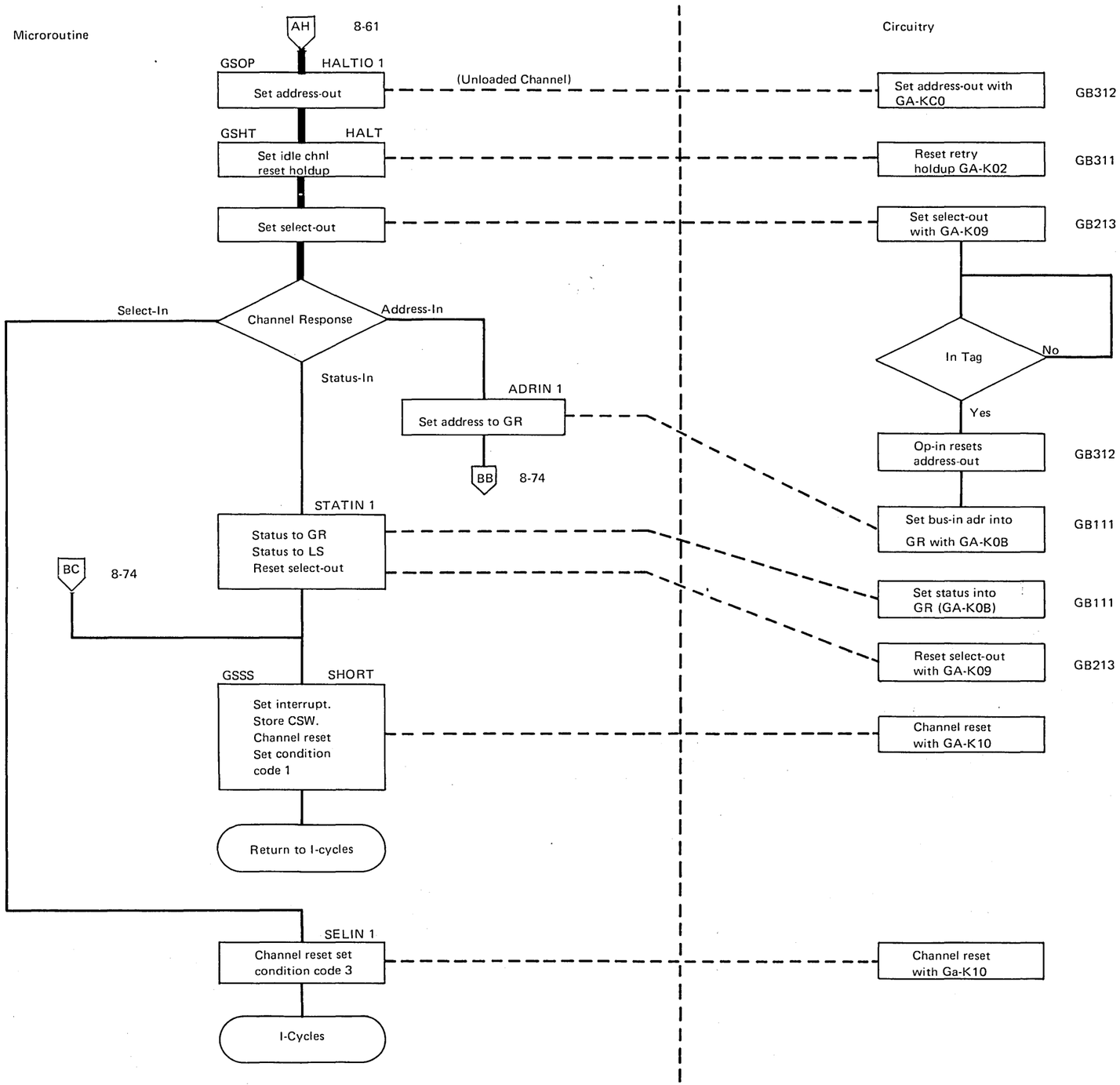
When the command being executed encounters a condition requiring retry, the control unit indicates it by raising 'mark 0 in' and 'status in' while presenting 'unit check' and 'status modifier' ('retry status') together with 'channel end' (meaning that the control unit or the device is not yet ready to retry the command), or with 'channel end' and 'device end' (meaning that the control unit and device are prepared for immediate retry of the command). 'Device end', if not presented with 'channel end', is presented later, when the control unit is ready to retry the command.

The channel acknowledges the occurrence of command retry by accepting the status byte containing retry status and indicating chaining. If 'device end' accompanies 'channel end', 'mark 0 in', and retry status, the channel immediately initiates a normal, chained initial-selection sequence, reissuing the previous command. If only 'channel end' and 'mark 0 in' accompany the retry status, the retry is not immediately performed.

Rather, when the 'device end' or 'device end' with 'status modifier' is presented to the channel, it is accepted with chaining indicated and a normal reselection occurs to reissue the previous command or; in the case of 'device end' with 'status modifier', the CCW following the previous command.

A channel indicates refusal to perform a command retry by accepting the status byte containing retry status without indicating chaining or by stacking the status byte. The stacked byte is treated as any stacked status.

Selector Channel Halt I/O, Halt Device (Part 1 of 2)



The halt-I/O instruction is used to stop a channel operation. The instruction could occur at any time. The action and the response reported in the condition code differ depending on the point of progress in the channel operation.

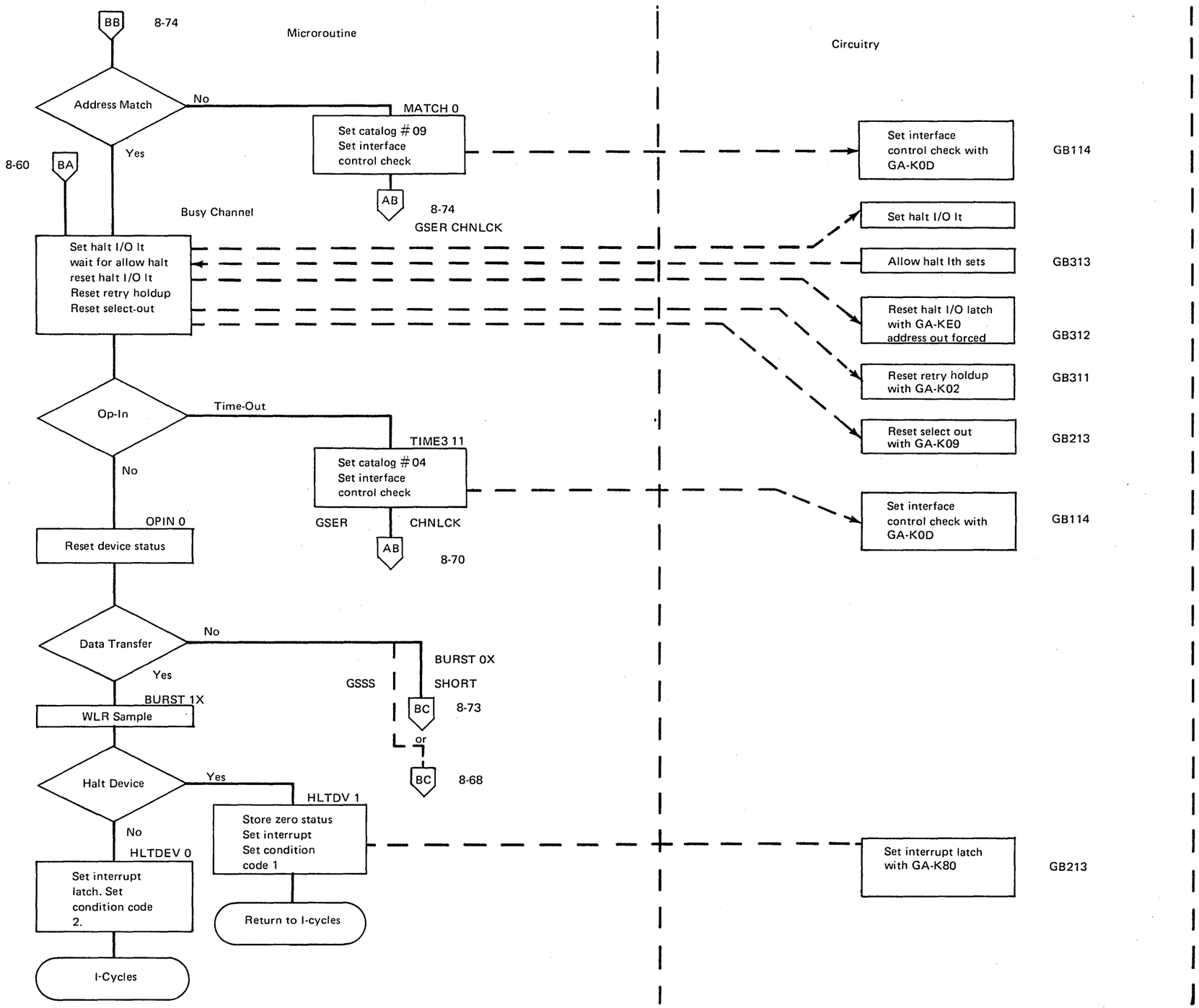
The instruction is initiated and decoded in the I/O initialization sequence (8-14). The first entry into this sequence is for a channel-loaded condition. In the case of an interrupt pending, the operation returns to I-cycles after setting the condition code to 0.

When the active bit is not present, the operation continues with a polling of the channel using the stated address. The automatic polling is not used because no command is to be presented. The microroutine sets the address into the GR-register and raises the address-out tag line after the address has had time to settle on the bus-out lines. After a delay of at least 400 ns, the select out latch is set to raise the hold-out line. The control unit should respond with status-in, select-in, or address-in depending on its condition.

A response of status-in indicates that the control unit is busy and cannot connect the addressed device. The status is read into the GR-register, and the channel controls are reset. The operation is returned to I-cycles after storing a CSW and setting condition code 1.

The select-in response indicates that the addressed device is either disconnected or has a power-off condition and cannot reply. The channel controls are reset and the operation is returned to I-cycles after setting condition code 3.

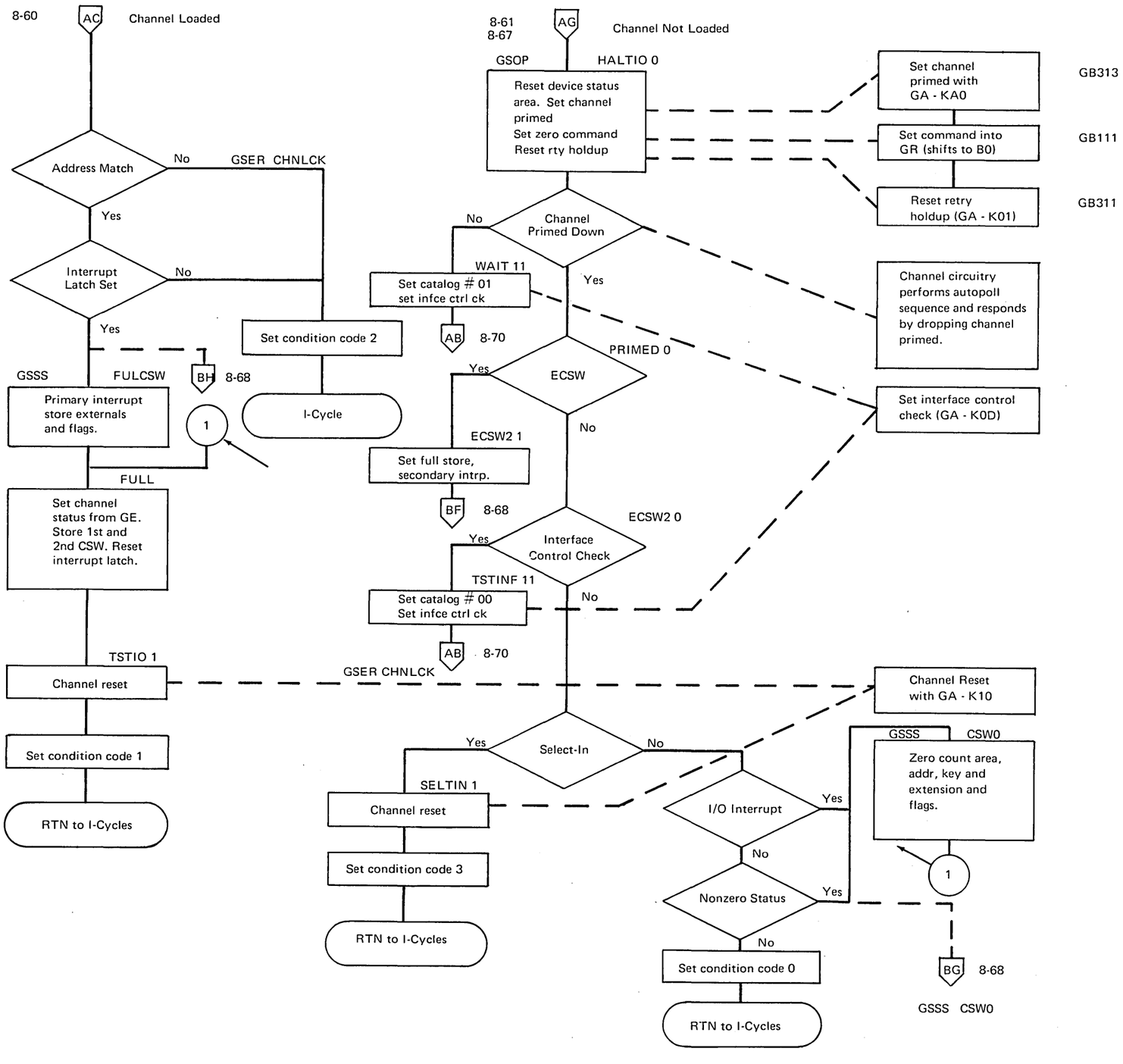
Selector Channel Halt I/O, Halt Device (Part 2 of 2)



The address-in response indicates that a connection was successful. The address on the bus-in lines is read into the GR-register and an address match is performed with the original address. If they do not match, the interface control check status is set with the appropriate error catalog number. With matching addresses, the command-out latch and the half I/O latch are set to indicate that the address has been taken. When the address-in line falls, the allow halt latch is set in the controls to indicate to the microroutine that a stop sequence can be initiated.

The halt-I/O latch (reset by the microroutine) forces the controls to raise the address-out tag line. The retry-holdup latch is reset to allow sensing the fall of the operational-in signal with the reset of the halt-stop latch. The select-out line is dropped with the address-out tag line raised to signal the channel control unit that it should disconnect. If the control unit fails to drop the operational-in line within the time-out period, the interface control-check status and the error catalog number are set, along with condition code 1.

If operational-in falls as requested, the interrupt latch is set to handle the operating ending conditions. The operation returns to I-cycles after setting the condition code to 2.



Selector Channel Test I/O

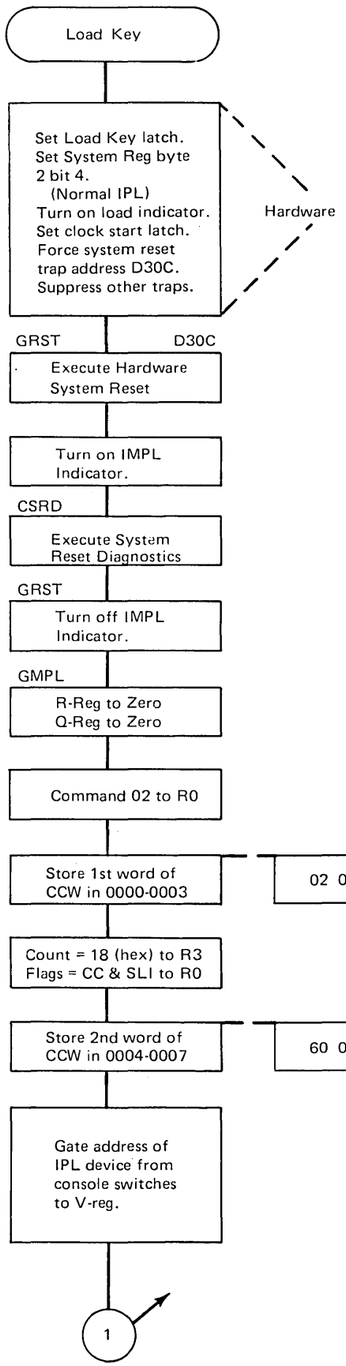
The test I/O instruction is used to obtain the operating state of the channel and device through the condition code set by its execution and, under certain conditions, by storing the CSW. The instruction can be used to selectively clear pending interruption conditions. The instruction may be executed at any time.

The instruction is initiated and decoded in the I/O initialization sequence. Two entries go into the test I/O sequence. The first is for a channel-loaded condition; the second is for nonloaded conditions. In the case of the nonloaded entry, the poll-control and channel-primed latches are set to interlock the channel for a polling sequence.

In the case of the channel previously loaded, the presented address is compared with the device address in storage. When the addresses match, an additional test is made of the interrupt latch. If the addresses do not match or if the interrupt latch is not set, the operation returns to I-cycles after setting condition code 2 (busy). For the case of address match and interrupt latch set, a CSW store operation is performed, the latch is reset, and the operation returns to I-cycles after setting condition code 1.

For the nonworking channel, the routine initiates automatic polling (refer to I/O initialization). The TIO command (0000 0000) is forced on the bus-out lines during the command-out portion. This causes the control unit to place its device status (even if stacked) on the bus-in lines. Clean status returns the operation to I-cycles with the condition code set to 0 (available). Any nonzero status results in storing the CSW and setting condition code 1 before returning to I-cycles.

Selector/Block-Multiplexer Channel IPL



- IPL is started by selecting an input device and pressing the load key on the console.
- IPL provides the facility for loading a program into main storage.
- IPL can be from devices on either byte-multiplexer channel, selector/block-multiplexer channel or IFA (if installed).

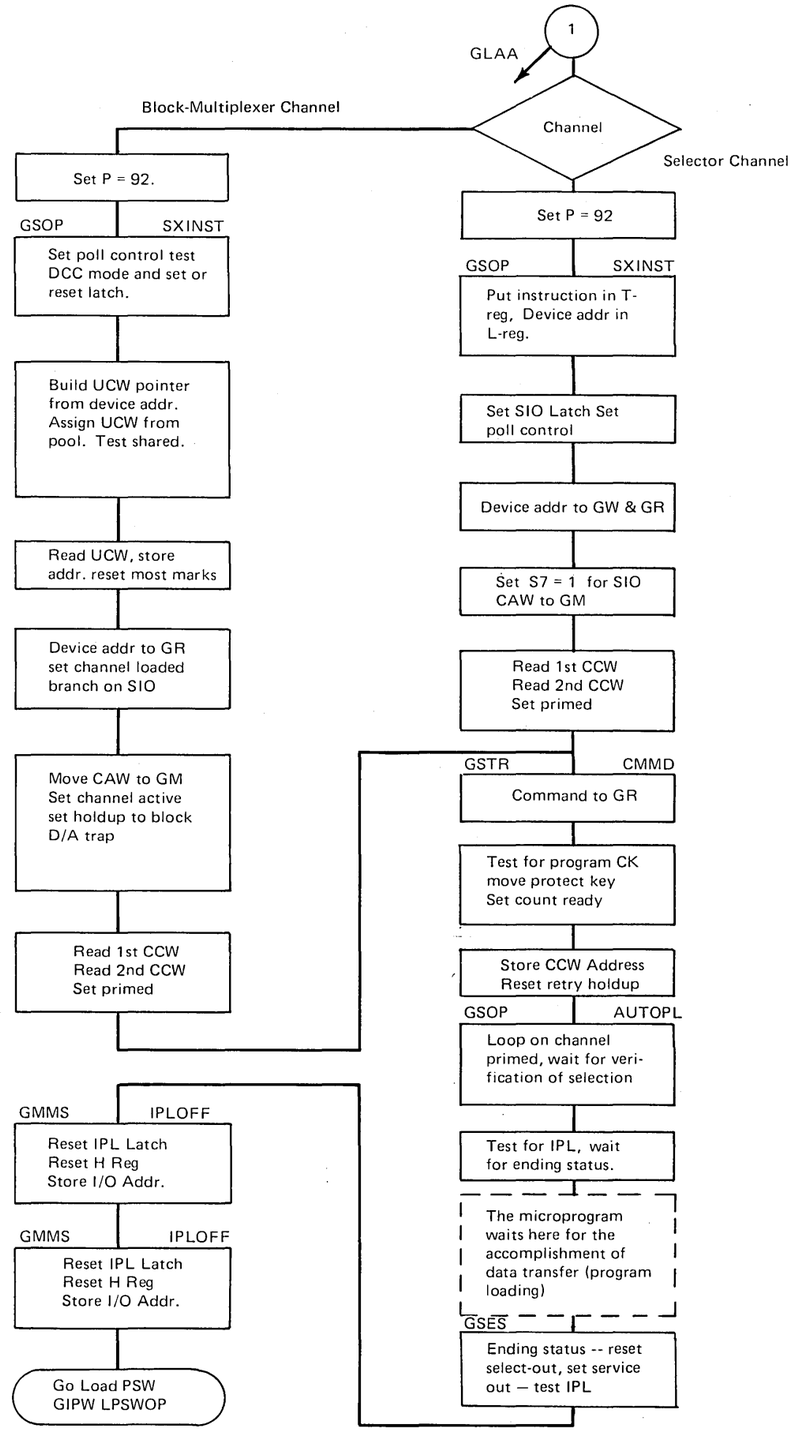
The IPL procedure resembles an SIO instruction in which the selected I/O device and a zero protection key is specified. The CCW for this instruction has a read command, zero data address, a byte count of 24, command chain flag on, SLI on, and command address 0.

The IPL operation reads new information into the first three doublewords of storage. The remainder of the program can be placed elsewhere.

The first doubleword contains the starting PSW for the entering program. The next two doublewords contain 2 CCWs that are command-chained. The operation continues with these and other chained CCWs to enter the full program.

Selection, handling of data and status, etc. follow the same general pattern as for SIO. This diagram is intended to give overall flow of the IPL procedure. Refer to other parts of this manual for details of selection, data handling, status handling, etc. Good status and correct programming are assumed.

At ending time (CC bit not on), all channels including IFA, branch to the GMMS (IPLOFF entry) to reset the IPL latch and end the IPL operation. The microprogram then branches to GIPW to load the PSW and return to I-cycles.



SELECTOR-CHANNEL WORD BUFFER

The Word Buffer Feature for the selector/block-multiplexer channels permits the assembly of a fullword (four bytes) before requiring a share cycle to store the data. When writing, fullword is read from storage into the buffer and the bytes are serially fed to the channel. This provides a 75% saving in the time required to service the channel for data. Each channel has its own buffer system; thus, the channel speeds and the CPU throughput are improved.

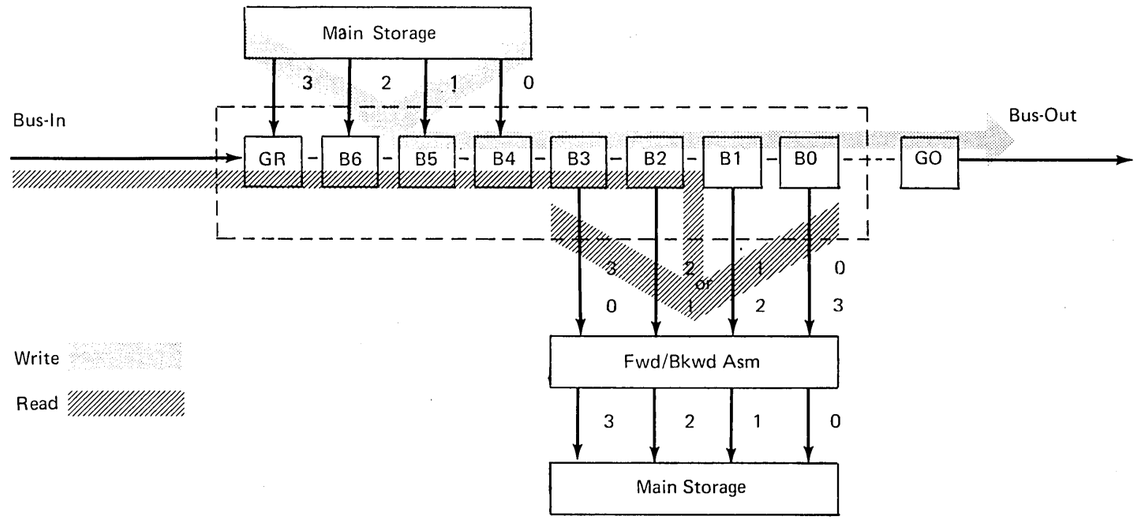
The buffer system adds seven byte registers between the normal channel GR-register and the GO-register. These are connected to allow shifting the bytes serially from the GR-register end toward the GO-register. This configuration allows space within the buffer for two full words. As soon as four input bytes are assembled and shifted into the B3, B2, B1, and B0 positions, a request is made for a share cycle to store the word. The request is made for a word during a write operation as soon as positions GR, B6, B5, and B4 are empty. The doubleword size of the buffer allows a delay in honoring a request equal to the time required to serially transfer four bytes on the channel.

Each buffer position including the GR-register has provisions for ten bits. These include a buffer-full bit in addition to the eight data bits and the parity bit. The buffer full bit, when on, indicates that the buffer position is in use. It may or may not contain data because the buffer-full bits are used (on an input operation) to position the bytes of data when less than a fullword is to be transferred. The bit indicates that the buffer position cannot have a byte shifted into it.

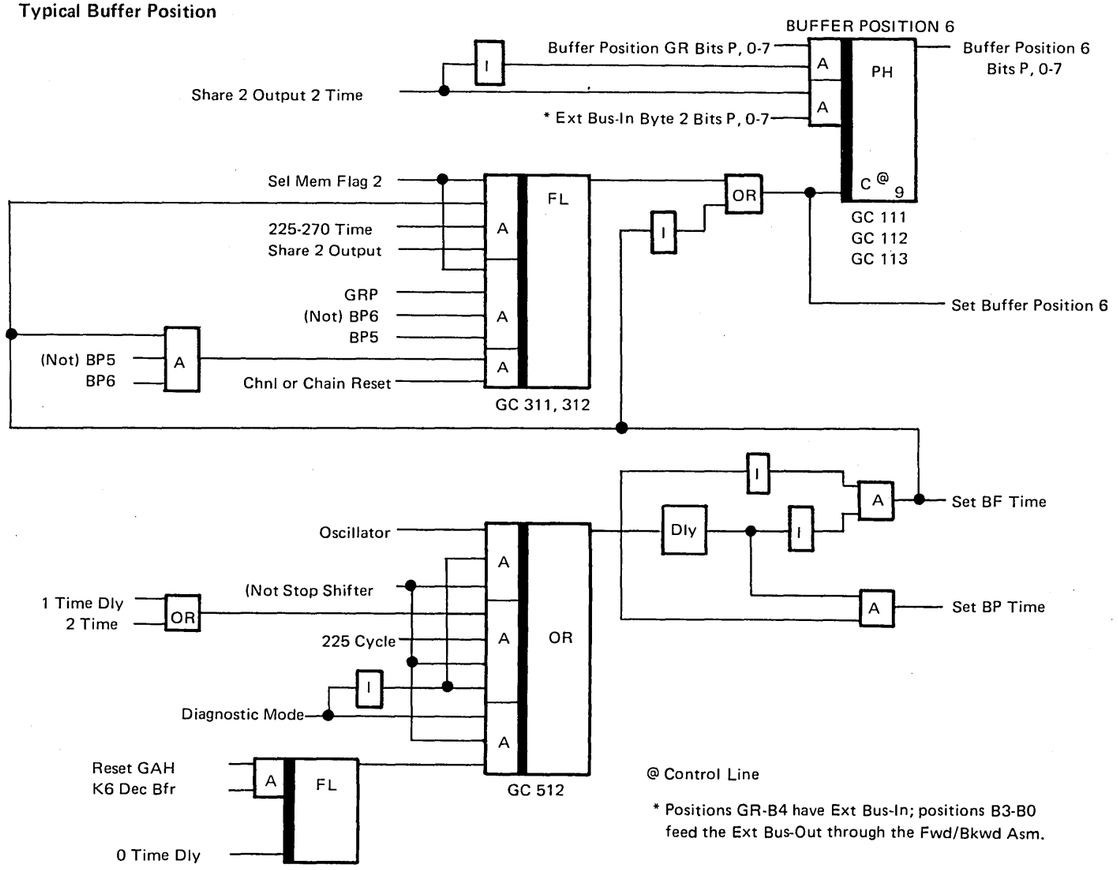
The buffer shifting is a ripple process in which a byte can move only when the position ahead of it is vacant. The byte may shift two positions if the buffer-full bits of both positions are off. A byte in the next position to the left cannot shift in the same cycle because the position to the right was not vacant at the start of the cycle. The example shows the shifting when a new word enters the buffer during a write operation. Two bytes of data still remained in the buffer from the previous word. During read operations, the shift is concerned only with moving a single byte entering the GR-register.

The buffer controls must be able to handle a data address that is not on word boundary for the first transfer. The first transfer may be for one, two, or three bytes depending on the two low-order bits of the data address. A transfer of less than a full word can also occur on the last transfer when the count remaining is less than four. The low-order address bits control the buffer positions to be used by forcing the buffer-full bits. These conditions also develop memory flag bits that control the data address and count updates. A forward/backward assembler reverses the order of the bytes for a read-backward operation.

Selector Channel Buffer Data Flow



Typical Buffer Position



Selector Channel Buffer

Buffer Shifting (Write)	Buffer Position GR	B6	B5	B4	B3	B2	B1	B0	G0
Request Honored	D	C	B	A	——	——	——	X	Y
1st Cycle	D	C	B	——	——	A	——	X	X
2nd Cycle Svc-Out	D	C	——	——	B	——	A	——	X
3rd Cycle	D	——	——	C	——	B	——	A	X
4th Cycle	——	——	D	——	C	——	B	A	X
5th Cycle	——	——	——	D	——	C	B	A	A
6th Cycle	——	——	——	——	D	C	B	A	A
Share Request									

DCBA = Four bytes of new word from storage.
XY = Two bytes remaining in buffer and GO.

GO has same data as B0 except when inbound tag lines are up, or Hold GO latch is on. B0 full is reset with Service-Out.

Buffer-Shift Controls

The buffer-shift control can move a byte of data either one position or two positions in the buffer, depending on the number of vacant positions ahead of the byte. The data-transfer circuitry between the buffer positions is in effect continuous across empty positions. The shift pulse repeats with each CPU clock oscillator pulse. More than one byte can be moving.

The buffer-full bits of the buffer positions control the shift and the final byte positioning. When the buffer-full bit for a position is not set, the set line for that position is developed. The data enters from the next higher-numbered buffer position and after a propagation delay is available at the output of the controlled buffer position. These outputs present the same data to the next lower-numbered buffer position. That position, if empty, has its set line developed to read the data into the latches for that position. The process would continue through to the last empty buffer position in the sequence except for the short time duration of the set control pulse (about 28 ns).

The buffer full (BF) bit latches are set during the buffer position set period. Each buffer-full latch has a follower BP latch that is set to the level of the buffer-full latch just before the buffer set period. It is the output of the BP latches that determines the distance to be shifted and the buffer-full latches to be set. Two gatings are provided for the set of the buffer-full latches as shown in the typical logic diagram. An advance of two is taken if the second position previous has its BP latch set and both the test position and the previous position have their BP latches reset. An advance of one is made if the position previous and the position after the test position both have their BP latches set. The previous position BF latch is reset when that position BP latch is set and the following position BP latch is reset, indicating that the data had a position to move.

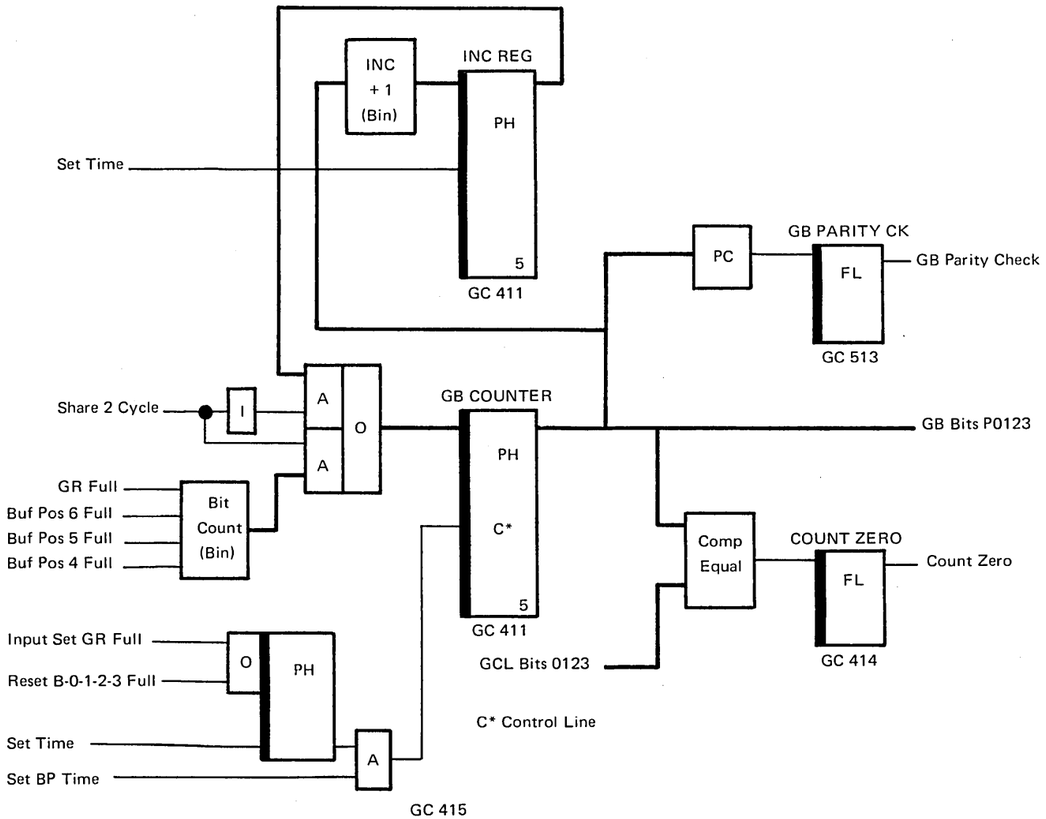
The buffer-full latches are set initially as the data bytes are set into the high-order buffer positions. For a read operation, the buffer-full latch for the GR-register is set with the service signal when the data byte is read in. For a write operation, the buffer-full bits for the four high-order buffer positions are controlled by the memory flag bits that indicate the number of bytes being transferred from storage. For read, the buffer-full latches of the four low-order buffer positions are forced from the two low-order address bits to indicate less than a fullword transfer. The forced buffer-full bits set in the low-order positions prevent data bytes from shifting into these positions. The actual transfer of bytes is controlled by the developed memory flag bits.

A test is made during each advance cycle to determine that the buffer-full bit count has not been altered. The test circuit contains provisions to accommodate new bytes entering the buffer and for bytes being removed from the buffer. The test is a simple odd/even count test. Any difference detected sets the buffer control check latch.

Buffer Byte Counter (GB)

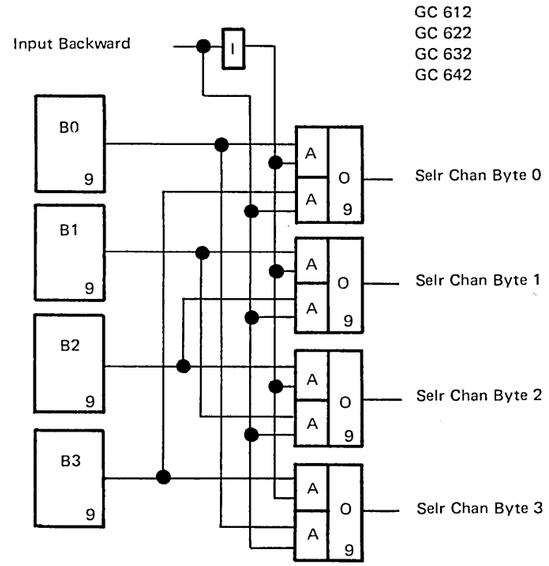
The buffer byte counter contains five latches that carry a binary indication of the number of bytes of data in the buffer. Four bits are used for the actual count, and the fifth is the parity bit (odd parity). The buffer count is used only for read operations, when the number of bytes in the buffer is a function of the count zero condition. The five latches do not in themselves function as a counter. The counter system contains a second set of five latches and a decode network. This second set of latches is set to a value one greater than the GB count through the logic decode. When a new byte of data enters the GR-register, the plus-1 value in the increment latches is set into the buffer byte count latches.

The buffer byte counter is initially set from the logic decode of the buffer-full bit set. With each transfer of data to storage, the four low-order buffer positions and their buffer-full bits are cleared. The buffer byte count is initialized again with the logical count of the buffer-full bits set.



Forward/Backward Assembler

The forward/backward assembler allows reversing the data bytes that form the word to be stored in the reverse order when the read-backward command is being processed. The byte in buffer position B0 that normally stores as byte 0 in storage is gated to store as byte 3 of the word. The assembler consists of two sets of gates. One, controlled by the in-forward signal, gates the buffer positions B0 through B3 into storage word bytes 0 through 3 respectively. The second gate, controlled by the in-backward signal, gates the same buffer positions into storage bytes 3 through 0 respectively. The actual bytes of the word to be stored are under control of the memory flag bits.



Share Cycle with Buffer

When the buffer is installed on the selector channel, the share request is made for a fullword instead of for each byte. The buffer-full bits determine the time for a share request. The GR, B6, B5, and B4 positions must be empty for the output request (write operation). For the input request (read operation), positions B3, B2, B1, and B0 must be full. In the input operation, part of the positions may have their buffer-full bits forced to transfer a partial word on the first transfer, but the positions appear to be full.

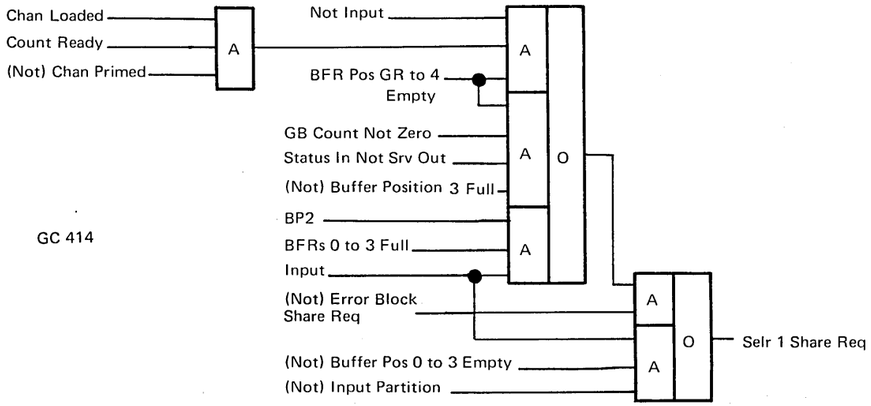
In all cases, the count ready latch must be set before share cycles, indicating that the count has been stored and that it has not been reduced to zero. For an output operation, the share request can be made when the first three positions are empty and the byte in position B0 has been transferred to the GO-register. By the time the request can be honored, the data byte in the B4 position will have transferred. During input operation, the request is made when the four low-order positions are full.

The request may also be forced with less than four bytes in the buffer when the status-in tag is received.

During the time that the reset signal is developed for the four low-order buffer positions, a block line is developed to prevent shifting data from the four high-order into the low-order positions. This block line is also developed during chain-data operations to hold data for the new address until the data address has been read in.

The full bits are turned on if the data address is not on a word boundary. If too many characters have been accepted, a chaining check occurs.

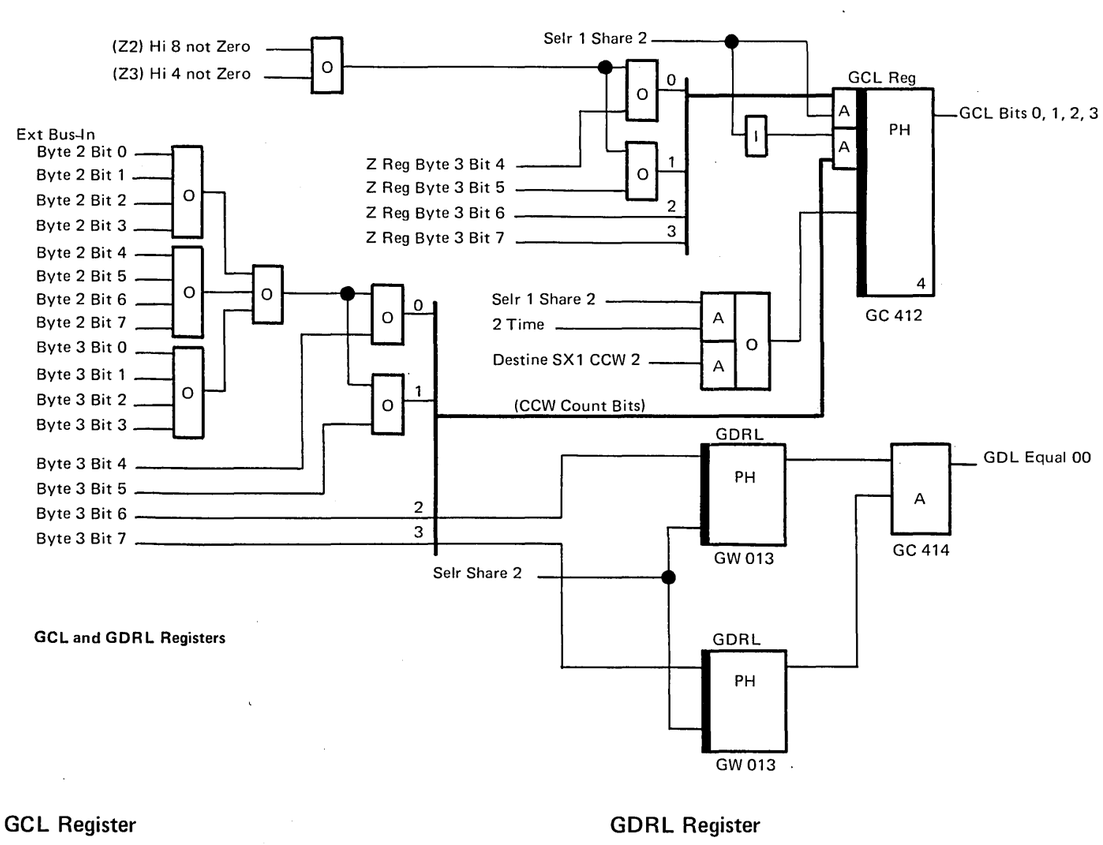
Note: The partition latch (used only with chain data) is used to block the buffer advance. The buffer must be empty before accepting new data when a chain data operation is done. The partition latch turning on causes the full latch for B4 to be held on and the partition latches for B3 and B2 to be held off. The data is held in B4 and transferred to B3. Because B3 cannot set its full latch, it does not recognize the data and cannot propagate the data further.



Buffer Share Request Generation

SHARE CYCLE INPUT OPERATION WITH BUFFER

Selector Channel Buffer



GCL and GDRL Registers

GCL Register

The GCL register contains four latches that represent the four low-order bits of the CCW count value set in the local-storage GC register. The latches are initially set from the external bus-in lines when the count value is destined. The latch entry is updated during each share cycle from the Z-register after the count is updated. The count value is not considered in the operation until its value is below nine. To ensure that the high-order bits of the count have been reduced before sampling, any high-order bit present in the GC-register (or Z-register) forces a count of twelve or more.

The output of the GCL register is compared with the output of the buffer byte counter during read operations to determine the actual count zero point. The two low-order bits of the GCL register are gated to the decode logic of the memory flag register on each share cycle. The memory flag bits control the bytes transferred and the count and data address updates.

GDRL Register

The GDRL register two low-order bits (30, 31) are gated to the decode logic of the memory flag register during share cycles. The flag bits are developed to control the bytes transferred and the count and data address updates.

After the first share cycle, bits 30 and 31 should be zero except for the read-backward operation when 11 indicates the word boundary.

Count GCL 1	Count GCL 2	Count GCL 3	Address GDRL Bit 30	Address GDRL Bit 31	Input Flag 0	Input Flag 1	FWD Latch 2	FWD Latch 3	Input Flag 0	Input Flag 1	BKWD Latch 2	BKWD Latch 3	Output Flag 0	Output Flag 1	Latch 2	Latch 3
0	0	1	0	0	1	0	0	0	1	0	0	0	1	0	0	0
0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0
0	0	1	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
0	1	0	0	0	1	1	0	0	1	0	0	0	1	1	0	0
0	1	0	0	1	0	1	1	0	1	1	0	0	0	1	1	0
0	1	0	1	0	0	0	1	1	0	1	1	0	0	0	1	1
0	1	0	1	1	0	0	0	1	0	0	1	1	0	0	0	1
0	1	1	0	0	1	1	1	0	1	0	0	0	1	1	1	0
0	1	1	0	1	0	1	1	1	1	1	0	0	0	1	1	1
0	1	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1
0	1	1	1	1	0	0	0	1	0	1	1	1	0	0	0	1
1	x	x	0	0	1	1	1	1	1	0	0	0	1	1	1	1
1	x	x	0	1	0	1	1	1	1	1	0	0	0	1	1	1
1	x	x	1	0	0	0	1	1	1	1	1	0	0	0	1	1
1	x	x	1	1	0	0	0	1	1	1	1	1	0	0	0	1

Memory Flag Bit Generation if Count is not Short

Memory Flag Register

The memory flag register contains four latches that are set to indicate the number of bytes in the storage transfer. One latch represents each byte of the word. A single register functions for all four of the possible selector channels. The information from the active channel is gated into the decode network to set the latches for the current share cycle. The flag bits control the buffer-full latch setting for a write operation and the storage entry byte gating for read operations. The flag bits also control the data address and count updates for both read and write operations.

A logic decode of the input bfr full bits, of the GCL register containing the two low-order count bits and the GDRL register containing the two low-order data address bits determine the flag latches to be set. The normal switching relies on the fact that these two registers do not have bits set in them at the same time. The exception occurs when the initial count value is less than four and is taken care of with special gating. The exception conditions may require the transfer of two bytes when the data address is three positions from the word boundary. This results in setting flag bits 1 and 2 to transfer the appropriate bytes and control the updates. The logic also takes into consideration the read-backward operation in which the boundary and the relative byte positions differ.

BLOCK-MULTIPLEXER FEATURE

- The block-multiplexer feature gives the additional capability of operating as block-multiplexer channels. It is largely implemented by microprogram routines.
- Channels with the block-multiplexer feature may operate as either selector channels or as block-multiplexer channels.
- Only block-multiplexing can use Disconnect Command Chaining (DCC).
- Any or all installed selector channels can have the block-multiplexer feature.
- As with the byte-multiplexer channel, block-multiplexer mode has multiple subchannels, each of which has an associated UCW (in control storage) and can support one I/O operation.

The block-multiplexer channel feature on the Model 145 permits any or all installed selector channels to operate as block-multiplexer channels. When a system is ordered, the user specifies those channels that are to operate only as selector channels and those that are to have the capability of operating in either selector or block-multiplexer mode.

The setting of a channel mode bit in a control register determines whether a channel with block-multiplexing capabilities operates as a block-multiplexer or selector channel. The mode bit is reset at IPL or on system reset and can be altered by programming at any time.

Better use of channel time can be achieved by operating the channel as a block-multiplexer channel. A single block-multiplexer channel can support interleaved, concurrent execution of multiple high-speed channel programs. The block-multiplexer channel can be shared by multiple high-speed I/O devices operating concurrently, just as the byte-multiplexer channel can be shared by multiple low-speed devices. Similarly, the block-multiplexer channel has multiple subchannels, each of which has an associated UCW in control storage and can support one I/O operation.

The number of UCWs required for the block-multiplexer channels installed on a Model 145 depends upon the number and type of I/O devices in the system configuration. The maximum number of block-multiplexer UCWs possible per system is 512 (assuming all are nonshared).

BLOCK-MULTIPLEXER CHANNEL OPERATION

A block-multiplexer channel functions differently from a selector channel in the way in which it handles command-chained channel programs. A selector channel executing a command-chained channel program is busy during the entire time the channel program is in operation, whether data transfer is occurring or not. A block-multiplexer channel executing a command-chained channel program has the ability to disconnect from the operational device during certain nondata transfer operations. Thus, a block-multiplexer channel can be freed during a nonproductive activity; for example, during disk seeking and most record positioning, thereby allowing more data to be transferred per unit of channel-busy time.

Refer to the diagram, Selector/Block-Multiplexer Channels. The two timing charts parallel the operation of a 3330 DASD on a selector channel and a 3330 DASD operating on a block-multiplexer channel. These are not necessarily typical configurations or operations but are presented to compare selector channel vs block-multiplexer channel. The objective on both devices is to read record 7.

For the selector channel, let us arbitrarily start the search at ID 5. The channel is busy from the start of the search until equal compare on record 7, and the completion of the read operation plus the unpredictable length of time until the CSW is stored. The average time for the record search is 12.5 ms.

For the block-multiplexer channel, the 3330 is sector-oriented (has rotational position sensing) and is able to alert the channel when the desired record is approaching the point of coming under the read head. The 3330 disk has 128 sectors. In the example, we perform SIO initial selection and Set Sector command to define the sector (100) that we wish to read. After setting the sector, the channel issues channel-end status, stores the UCW and releases the channel for other operations. This allows an average time of 8.3 ms that the channel is free for other operations.

As the sector approaches read time, the device indicates that it requires service by Request-In with Device-End status. This causes the UCW that was stored for this device to be loaded into the channel registers, and reselection occurs. Command chaining is active causing chaining to the next command—Search ID 7.

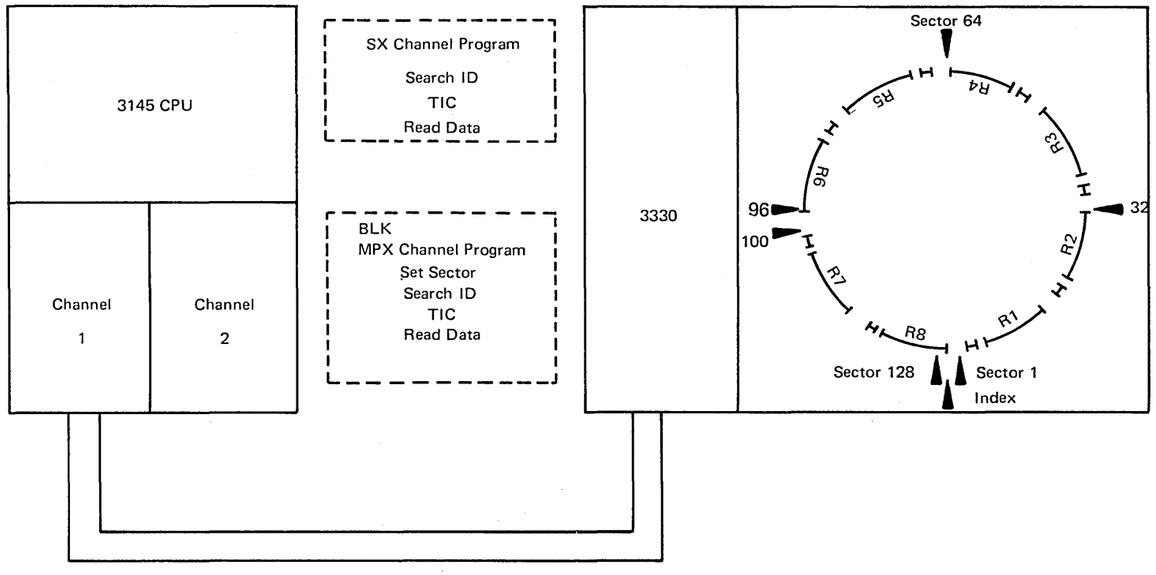
Reselection occurs again with CE-DE-Status Modifier. This means break the search loop ("jump" the TIC) and perform the next channel command—Read Data. Reselect again (each command causes selection), read data from record 7, then issue CE-DE, store the UCW and release the channel.

At this point, another difference between a selector channel and a block-multiplexer channel becomes apparent. The selector channel remains busy after CE-DE status until the CSW is stored. The channel has to wait for either a TIO instruction, or interrupts allowed. On the channel with block-multiplexing capabilities, even if a device is incapable of disconnecting on command chaining, the channel stores the UCW and releases the channel for other operations (upon receipt of CE-DE status).

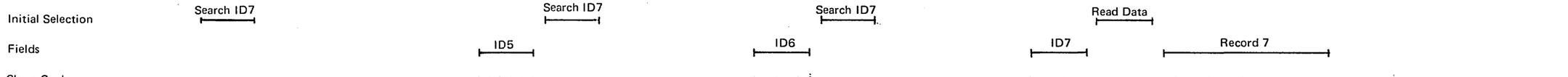
A basic principle of block-multiplexer operations is that channel-end status *allows* devices to DCC; the option of doing it is dependent on the capability of the device.

CHANNEL AVAILABLE INTERRUPTION

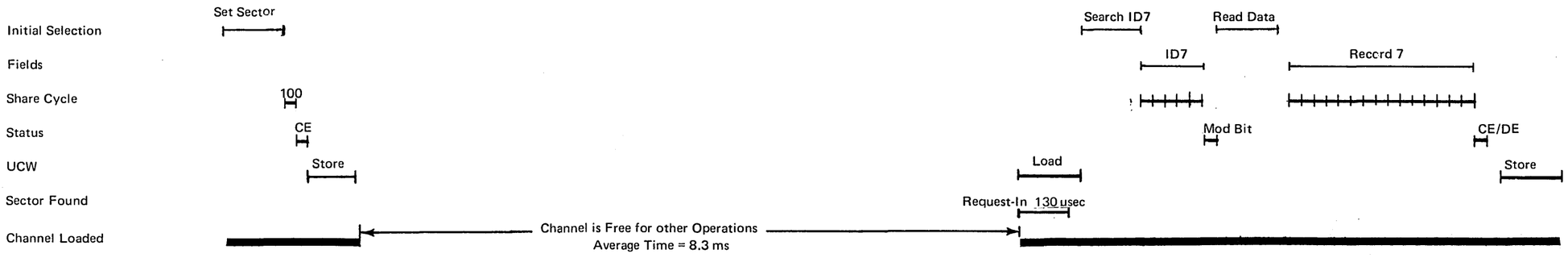
The channel available interruption facility adds efficiency to block-multiplexer channel operations. With a typical operation of several devices operating on a block-multiplexer channel, the channel is alternately busy and free. Unless the free time between CE and DE is utilized fairly consistently, it is obviously of little advantage. It is desirable for the CPU program to issue instructions during the free time.



SELECTOR CHANNEL



BLOCK-MULTIPLEXER CHANNEL



The channel available interruption capability is the means of informing the program that the channel is available. The operation is: The program tries for a channel operation. If the channel is busy, the response will be to set condition code 2 and the channel tried latch. When the channel becomes free, a channel available interruption is taken. Thus the desired operation may be performed at the first opportunity. This avoids repeated busy's that would result from a program loop.

BLOCK-MULTIPLEXER CHANNEL UCW ASSIGNMENT

On the Model 145, with the block-multiplexer option, a block of control storage is reserved for UCWs. This group of UCWs constitutes a 'pool'; that is, it is available to all installed block-multiplexer channels, with no set number for any given channel. Any multiple of 16 UCWs up to 512 maximum can be specified.

Up to 16 per channel (64 maximum) of the UCWs available to the block-multiplexer channels can be reserved for shared subchannels; the remaining UCWs are used for nonshared subchannels. Four plugcards are provided to assign the channel number and the device address set of up to 16 shared subchannels for each channel. The service representative wires these plugcards to the user's specification at installation time or when changes are required. (Refer to Shared Subchannel Assignment.) Each shared subchannel is accessed from a block of 16 contiguous device addresses of the form X0 through XF, and no more than one control unit should be attached to each shared subchannel.

The 256 possible device addresses on a block-multiplexer channel are divided into 32 device address groups of eight. Each block-multiplexer channel has a UCW address pointer table with a halfword entry for each device address group. The halfword entry in the table translates the actual UCW addresses in control storage from the device address as needed.

Each device address group is one of three types according to the UCW requirements of the devices that are to be attached. All devices in an address group must have the same UCW requirements. The three types are as follows.

Type 1: Each device of the group requires an unshared UCW,

Type 2: All the devices of the group use the same shared UCW, or

Type 3: A UCW is not to be assigned for any device address of the group, and the devices are not allowed to operate in block-multiplexer mode.

All the devices on a shared control unit must use the same UCW and may require several Type 2 device address groups. The type of each device address group and the listing of the shared address groups that are to use the same UCW must be specified at order time so that the UCW address tables can be correctly initialized from the plugcards. Also, an estimate of the number of UCWs required for unshared (Type 1) operation is necessary so that the size of the UCW pool can be set realistically.

When the first start I/O is executed for a nonshared device, the channel determines whether a block of eight UCWs has been assigned to the range of eight addresses in which the device address falls. If a UCW is not assigned and the device is successfully selected, the channel assigns nonshared UCWs to a block of eight contiguous device addresses of the form X0 through X7 or X8 through XF. These UCWs remain assigned until a system reset occurs.

For example, the assignment of a nonshared UCW to device 1A3 (channel 1, device A3) causes the assignment of UCWs to I/O addresses 1A0 through 1A7.

When the first start I/O is initiated to another device on the same channel and in that same block of addresses, the channel determines that a block of UCWs has already been assigned, and the correct UCW is addressed.

If a start I/O with a Type 1 device address that has not had a UCW assigned is attempted on an available channel and no UCWs are available, the result is a condition code 3 setting.

The method used for semidynamic UCW allocation of Type 1 devices allows flexibility in reassignment of a device from one channel to another.

Shared UCW Assignment

A plugcard is used for each channel with block-multiplexer channel feature for assigning initial UCW type and related UCW pointer and control register values. The plugcards provide two bits, A and B, for each group of 16 addresses. This is double the number of addresses in a group of addresses in the address pointer tables. Thus, Type 2 UCW assignment is restricted to groups of 16 addresses (two address groups in the table). The bits are plugged to indicate 11 at assembly time. The significance of the jumpers is as follows:

A	B	
1	1	Selector channel mode *only,* DCC not allowed.
1	0	Unshared UCW/DCC allowed.
0	1	Assigned to shared UCW.
0	0	Continuation of addresses for a shared UCW.

During IMPL, the A and B bits are read from the plugcards, and the UCW Address Pointer tables are initialized as required. For example, suppose the customer has requested that address groups 80-87 and C0-C7 be assigned a shared UCW and that devices in group 90-97 be restricted to selector mode only. Assume that we are discussing a system with a single block-multiplexer channel. Group 80-8F and C0-CF would have the A and B jumpers wired to 01. Group 90-9F would have the jumpers wired to 11. All other groups would be wired to 10. You cannot wire groups of eight, but the plug wiring controls groups of 16 in the range X0-XF. In the preceding example, we were forced to dedicate groups of 16 to satisfy the customer's requirements.

Each shared UCW requires a UCW and one additional byte. Because of this, a group of eight UCWs provides seven shared UCWs. The eighth UCW in a group provides the extra seven bytes required for the seven shared UCWs. Thus, in reality, the group of 16 UCWs that are available for shared control units provide facilities for 14 shared UCWs on each selector/block-multiplexer channel.

Semidynamic UCW Address Assignment

The general theory of semidynamic UCW assignment has been previously discussed. This section is devoted to the details of how this operation is accomplished. First consider the functional units involved in UCW assignment. In addition to the plugcards and associated circuitry, there are tables, registers, and the UCW pool in control storage.

UCW Address Pointer Tables

Each block-multiplexer channel has a 16-word UCW Address Pointer table in control storage. These tables start at locations B000, B100, B200, and B300 for channels 1, 2, 3, and 4, respectively. The table is initialized according to plugcard wiring for shared UCWs (specific address in the UCW pool), and with the value 0002 for UCW address groups that require unshared UCWs. If selector mode of operation is required for devices on the selector channel with block-multiplexer mode feature, the UCW pointer table for the appropriate address groups is initialized to 0003.

The two low-order bits of the UCW pointer (bits 6 and 7) define the restrictions that are initially imposed on the related device address group. In subsequent operations, these bits indicate whether or not UCWs are assigned. The other bits are initialized to 0's if the group is unassigned, or to a location in the UCW pool starting at 9000 if shared UCWs are assigned. As groups of addresses are dynamically assigned UCWs, the pointer is set to appropriate addresses in the UCW pool.

UCW Assignment Registers

Four halfword registers in control storage are used by the microprogram to control semidynamic UCW assignment. These registers are located in control storage FFD4-FFDB. Bit 0 in the first register, Master DCC Control, is set from a control register and controls whether or not Disconnect Command Chaining (multiplex-mode) is permitted on the addressed channel. The register at FFD6, Start UCW Pool, is initialized to the first available unshared UCW in the pool as determined by shared UCW assignment. The halfword at FFD8 is initialized at IMPL time to the address of the end of the UCW pool +1 to indicate End UCW Pool. The fourth halfword indicates Next Available Group, and is also initialized at IMPL time. This value is initially the same as Start UCW Pool, and is modified upward as Unshared UCW groups are assigned. Before each group is assigned, a comparison is made of End UCW Pool and Next Available Group to determine whether UCWs are available for assignment.

Pointer Adr →	B000	B008	B010	B018	B020	B028	B030	B038
Device Adr →	00-07	08-0F	10-17	18-1F	20-27	28-2F	30-37	38-3F
Initial Value →	0002	0002	0002	0002	0002	0002	0002	0002
Assigned →	9088	9100	9080		9108			
	B002	B00A	B012	B01A	B022	B02A	B032	B03A
	40-47	48-4F	50-57	58-5F	60-67	68-6F	70-77	78-7F
	0002	0002	0002	0002				
			9100					
	B004	B00C	B014	B01C	B024	B02C	B034	B03C
	80-87	88-8F	90-97	98-9F	A0-A7	A8-AF	B0-B7	B8-BF
	9001	9001	0003	0003				
	B006	B00E	B016	B01E	B026	B02E	B036	B03E
	C0-C7	C8-CF	D0-D7	D8-DF	E0-E7	E8-EF	F0-F7	F8-FF
	9009	9009						

UCW Pointer Table Channel 1

B100		
00-07		
0002		
B102		
40-47		
0002		
		B13C
		B8-BF
		0002
		B13E
		F8-FF
		0002

Channel 2

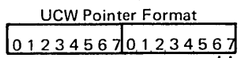
B200		
00-07		
0002		
B202		
40-47		
0002		
		B23C
		B8-BF
		0002
		B23E
		F8-FF
		0002

Channel 3

B300		
00-07		
0002		
B302		
40-47		
0002		
		B33C
		B8-BF
		0002
		B33E
		F8-FF
		0002

Channel 4

UCW Pointer Format



00 = Unshared/Assigned/DCC Allowed
01 = Shared
10 = Unassigned
11 = DCC Not Allowed

Control Storage — Control Reg 0 Bit 0 = DCC Mode

FFD4	0123----		Start UCW Poll
FFD8	End UCW Poll +1		Next/Current Group

Example	8	—	—	—	9	0	8	0
	9	2	C	0	9	0	8	0

UCW Pool

The pool of UCWs is located in control storage starting at address 9000. Each UCW is eight bytes, and a group of UCWs is eight UCWs that relate directly to device address groups. UCWs are not contiguous in the pool. A block of 16 consecutive UCWs are alternately assigned to two groups of device addresses. The *odd* and *even* groups require equal numbers of UCWs from the block of 16 (one row on the pool chart). This is the reason UCW groups are provided in multiples of two; therefore, the number of UCWs available is any multiple of 16 up to 512 maximum. The maximum pool would require the block of control storage from 9000 to 9FFF inclusive. The term *even* refers to addresses with a low-order hex 0 (9000, 9010, 9020, etc.). The term *odd* designates addresses with a low-order hex 8 (9008, 9018, 9028, etc.).

Operation

Semidynamic assignment of unshared UCWs is accomplished by the GSOP routine previous to a SIO selection. Consider a sequence of examples:

Example 1: Assume a single selector channel and 96 UCWs. Shared UCWs have been assigned by plugcard for addresses 80-8F and C0-CF on channel 1. Devices in address groups 90-9F have been wired for restriction to selector-mode operation. All other groups of UCWs are wired to permit being dynamically assigned as unshared UCWs.

With the conditions described, the first SIO is to device address 110. From channel/device address 110, the routine builds pointer address B010. The value 0002 that is in the pointer indicates that UCWs are unassigned; and therefore need to be assigned for the group of addresses 10-17. The End UCW Pool +1 (value 92C0 and the Current Pointer value 9080) are read and compared; inequality indicates that pool space is available. The Current Pointer is moved into B010, thereby establishing UCW assignment for the addresses 10-17 to the UCWs in locations 9080, 9090, 90A0, . . . 90E0, 90F0. The Current Pointer is updated by +8 to 9088, becoming the Next Available Group pointer.

Example 2: SIO on device 111. Build UCW Pointer address B010. The pointer was set to 9080 in example 1; thus bit 6 = 0, meaning that UCW is assigned, proceed with channel operation.

Example 3: SIO on device 101. This gives Pointer address B000. 0002 (B6=1) indicates that UCWs have not been assigned to this group. Repeat test to see whether UCWs are available and then read current pointer (9088). Store 9088 in channel 1 UCW pointer B000. This establishes UCW assignment for devices 00-07 in locations 9088, 9098, . . . 90E8, 90F8. The current pointer is now updated by +78 to 9100, the Next Available Group.

Example 4: SIO on device 153. Pointer address = B012. Unassigned, available. Read Current Pointer (9100) and store in table at B012. This assigns UCWs to the even doublewords in the group (9100, 9110, 9120, . . . 9160, 9170). The Current Pointer is updated to 9108 (Next Available Group).

Example 5: SIO on device 123. Pointer address = B020. Store Current Pointer (9108) in B020, this assigns UCWs to the odd doublewords; that is, 9108, 9118, 9128, + . . . 9168, 9178. The current pointer is updated to 9180, next available group.

Example 6: SIO on device 10C. Pointer address = B008. Read Current Pointer—change it from 9180 to 9100 (because device address bit 4 = 1). Set pointer value 9100 in B008, and update pointer to 9188.

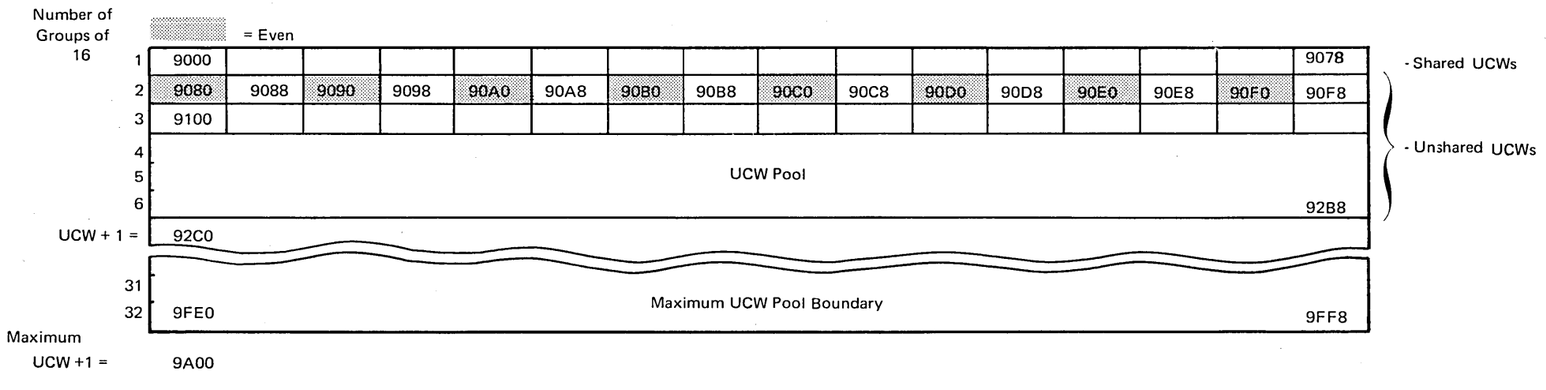
We have assigned two pointer table entries the same value: 9100. This does *not* cause duplicate assignment of areas in control storage with the same UCW. The method of interpreting device addresses prevents this from happening. Consider the two addresses 53 and 0C that apparently are being pointed to the same UCW area.

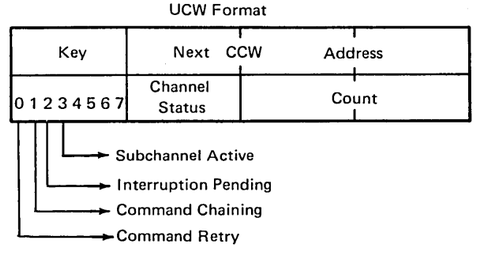
Case 1: Device address 53. From the GSOP microroutine;

```
GD=              1001  0001  0000  0000    GW3=Device Address
GD3=GW3XH,OE,GD3       0011  0000          = 0101  0011
                 1001  0001  0011  0000    XH  = 0011  0000
                   9     1     3       0 = Correct UCW address
```

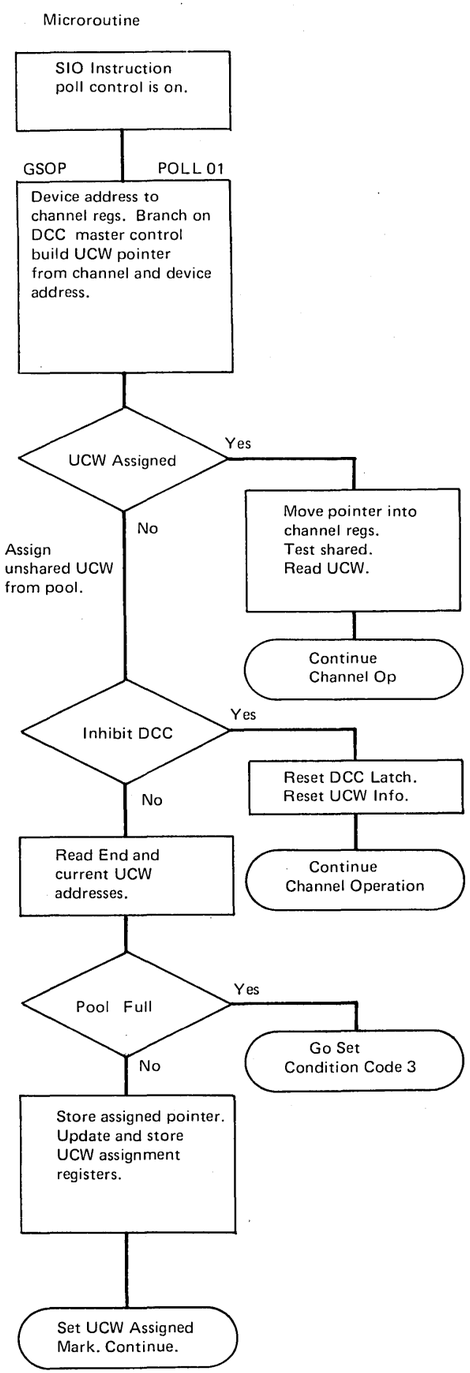
Case 2: Operation with device address 0C. Refer again to the GSOP microroutine;

```
GD=              1001  0001  0000  0000    GW3= 0000  1100
GD3=GW3XH,OE,GD3       1100  0000          XH  = 1100  0000
                 1001  0001  1100  0000
                   9     1     C       0 = Correct UCW address
```





Semidynamic Unshared UCW Assignment



Example

UCW Assignment for device 10 on channel 1

Device Address = 0 0 0 1 0 0 0 0
X3, OR, K07 0 0 0 0 0 1 1 1

0 0 0 1 0 1 1 1
Bits 45 = 00 0 0 1 1 1 0 0 0
X3, A, K38

Read Channel Adr 0 0 0 1 0 0 0 0
Build X2 and X3 – X2 = 0, OE, KB0

Pointer Address =

X2 X3
1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0
B 0 1 0

Read Pointer
RDH R CS, X R2 R3
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
B6 Unassigned
B7 DCC

Read End and Current Pointer
RDW R DC, D8 (FFD8)

R0 R1 R2 R3
1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0
9 4 0 0 9 0 8 0

Compare
R0, OE, R2
R1, OE, R3
Z Bus ≠ 0, Therefore S2 = 1 Move R To Y

Device Address V3
0 0 0 1 0 0 0 0
B4

Store pointer value (Y = 9080) in address at X (B010)
R3 B4 = 0 UCW set is even - perform update
of value in R3 R3
1 0 0 0 0 0 0 0
R3 = R3+K08 0 0 0 0 1 0 0 0
1 0 0 0 1 0 0 0 = Next available group

Store address of next available group in FFDA.
FFDA = 9088

Assignment of unshared UCWs is performed for a complete device address group by the GSOP routine at selection time for the first successful SIO to any device of the group.

When the microroutine determines that the channel is unloaded, it set poll control and begins the UCW initialization procedure. The master DCC control byte is read first to determine whether DCC is permitted; if not, the DCC is allowed, the DCC mark bit is set and the DCC latch is set.

The device address bits are manipulated to build the low-order byte of the UCW pointer address; the channel address determines the high-order byte. The pointer is read and tested to determine whether a UCW has been assigned to this group. If so, the channel operation proceeds. If not assigned, and DCC is allowed, the address group requires UCW assignment.

The End UCW Pool, and Current UCW Pointer values are read and compared to determine whether UCWs are available for assignment; if not, the routine ends and sets condition code 3.

If UCWs are available, the assignment continues. The Current/Next Available UCW pointer address in FFDA is moved and stored in the pointer table at BXXX (the address that was generated from the channel/device address). The current UCW address is updated as required and becomes the next available UCW address for the next group assignment. A mark bit is set to indicate UCW assigned, and the channel operation continues.

STORE/LOAD UCW TRAPS

A significant factor in block-multiplexer channel operation is the storing and loading of the UCW. This is the heart of Disconnect Command Chaining. The general philosophy is similar to byte-multiplexer channel operation. The store UCW and load UCW operations are performed by microprogram trap in the GSLD routine.

After trapping to the GSLD routine, the state of operational-in determines whether to take the store or the load routine. The store UCW routine is taken when the DCC latch is on, the channel is loaded, and operational-in drops. This may be between channel-end and device-end on a command chaining channel, or after ending status.

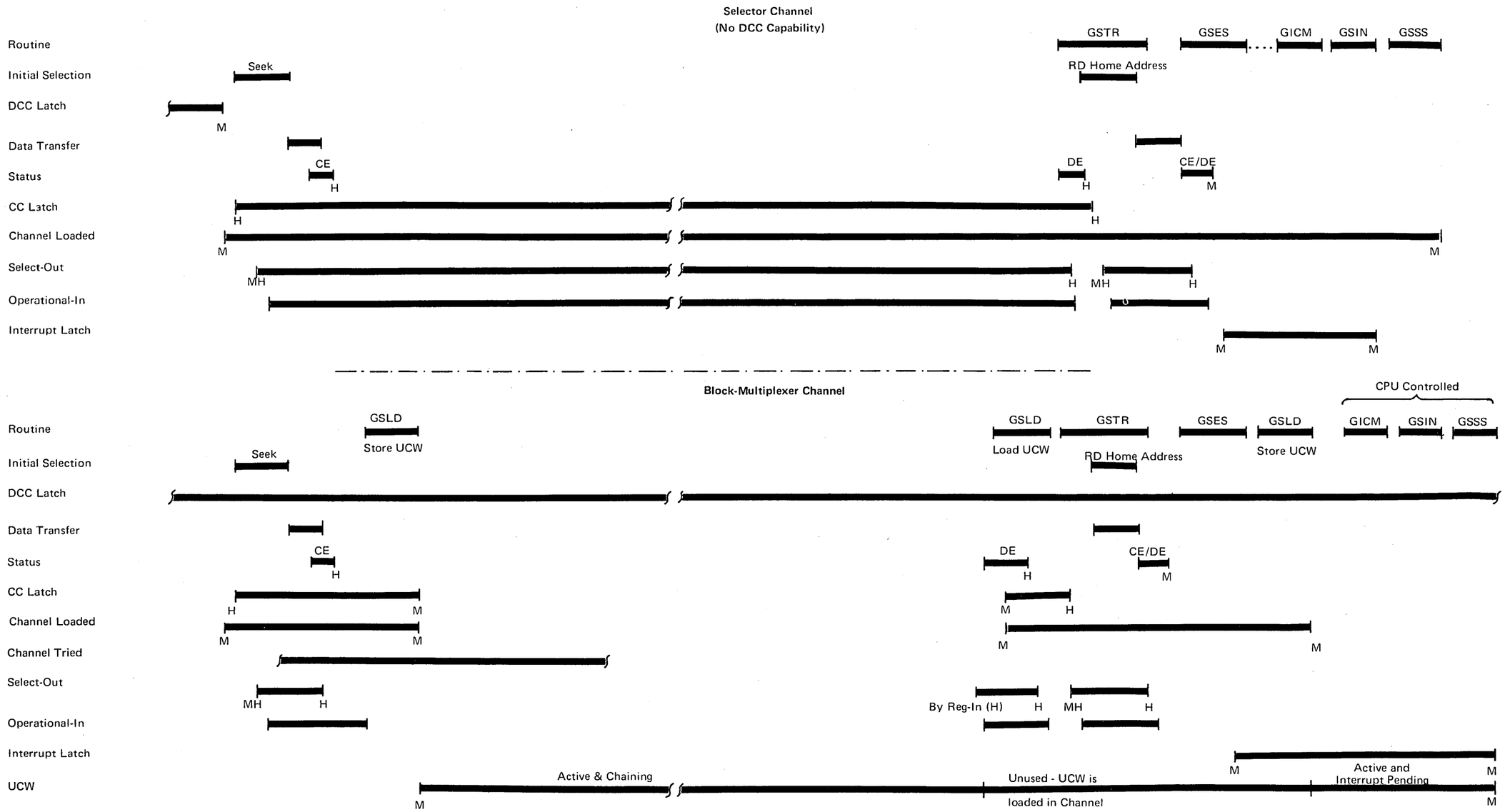
The UCW is stored in the UCW pool at the address that was assigned during the SIO of this device or another device in the same address group. The channel is active during the store routine. The store UCW routine stores all pertinent information for the channel operation in the control-storage area assigned. The channel is then freed for other operations. If the channel-tried latch was previously set, a channel available interrupt is requested at this time.

When the device requires service, request-in is raised. If the channel is busy, request-in is ignored and will be repeated at the next rotation of the disk. If not busy, a control-unit-initiated selection is performed. Select-out is set, address-in occurs, and the address is latched in the GR register. The channel responds with command-out. The control unit responds with status-in, and a load UCW trap is requested. The UCW pointer address is developed from the channel/device address. Because operational-in is up, the load UCW routine is performed. The UCW is loaded into local storage. The command chaining latch is set, also channel loaded and channel active, and the routine returns from the trap.

The CC latch requests a chaining trap in GSTR to handle command chaining. From this point, operation is essentially the same as for selector channel until ending time.

At ending status time (assume CE status, no chaining), the routine traps to GSES to handle status and put it in the interrupt buffer (if available) and turn on the interrupt latch. DCC latch, channel loaded, and Op-In down requests GSLD trap again to store the UCW. The UCW is stored with CC reset and interruption pending on. With status and address in the interrupt buffer, the channel is unloaded and is available for other operations. Under CPU control, GICM, GSIN and GSSS handle the interrupt and store the CSW.

Store/Load UCW

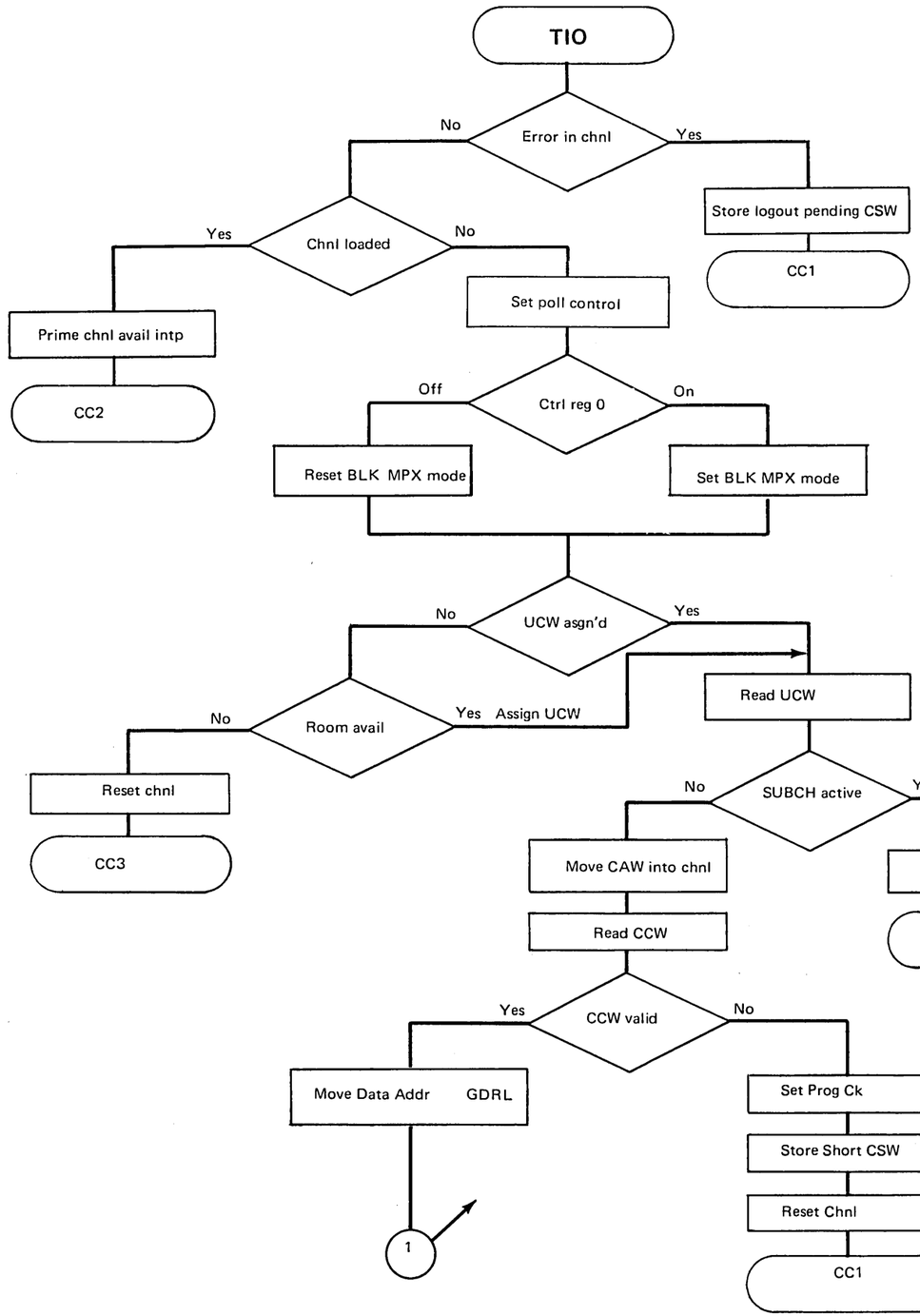


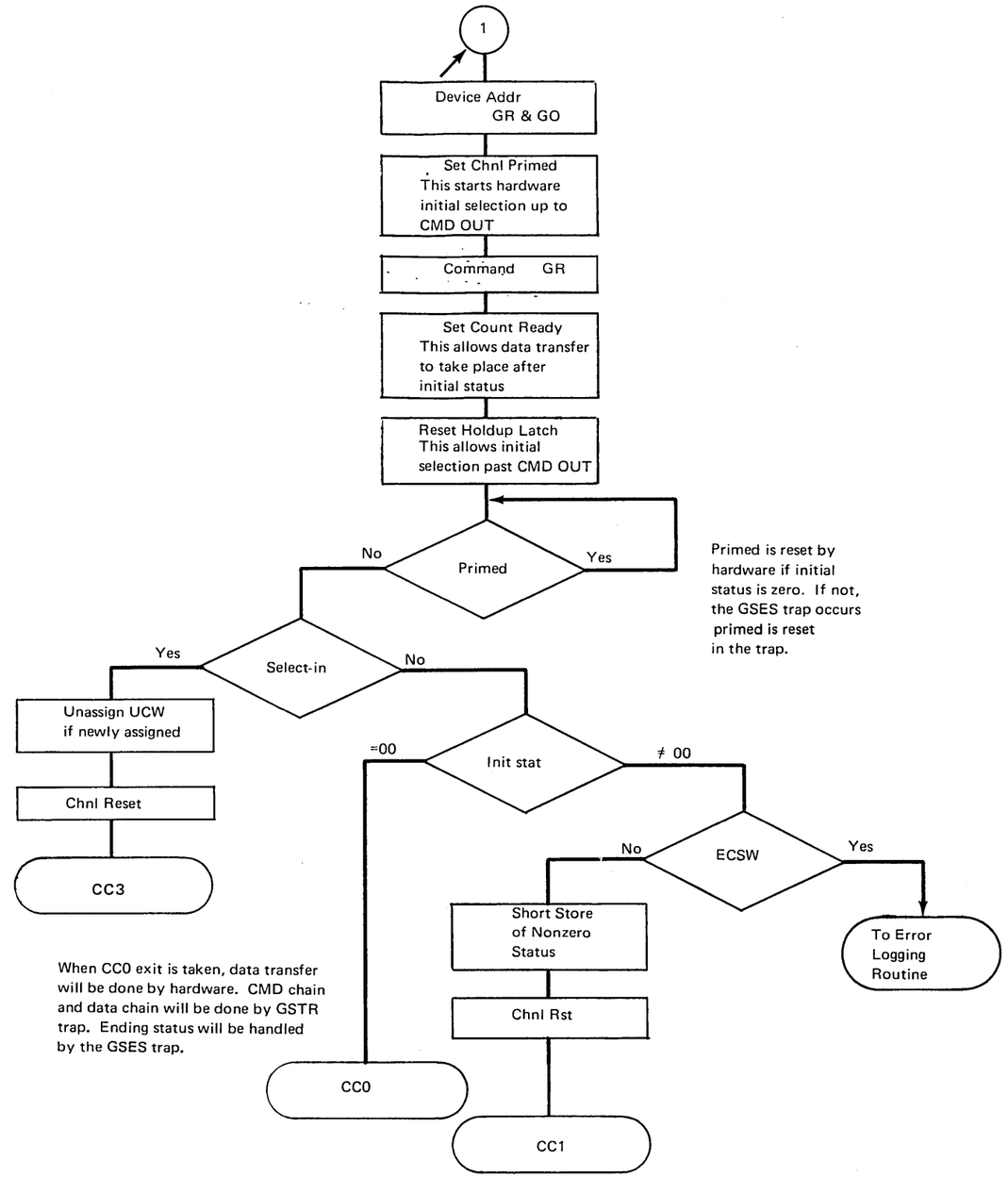
H = Set/Reset by Hardware

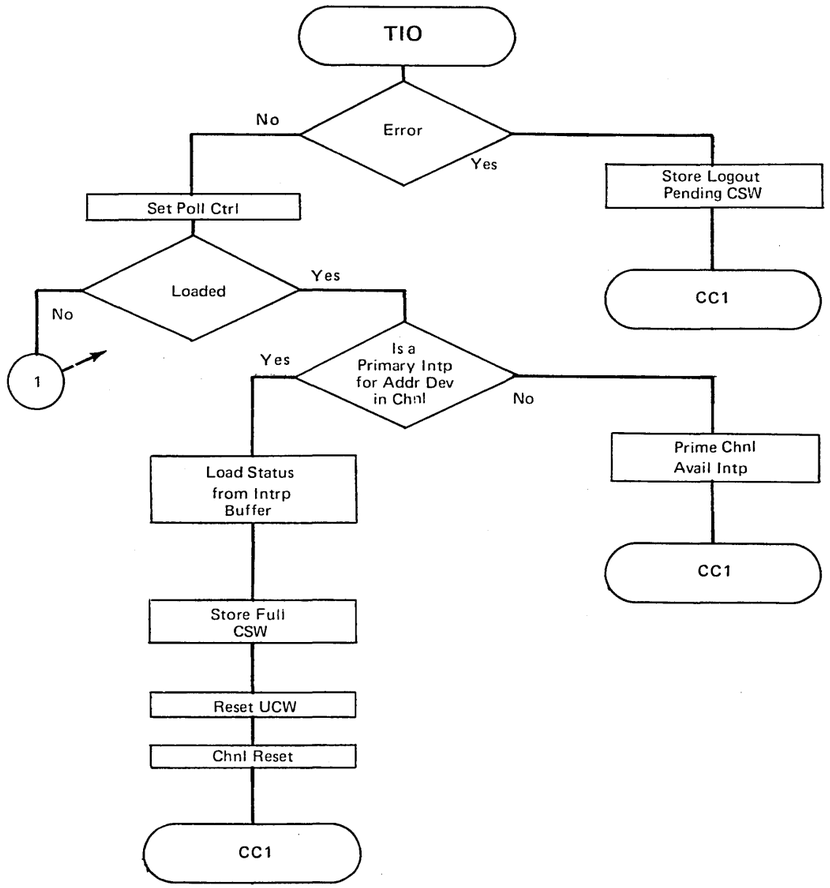
M = Set/Reset by Microprogram

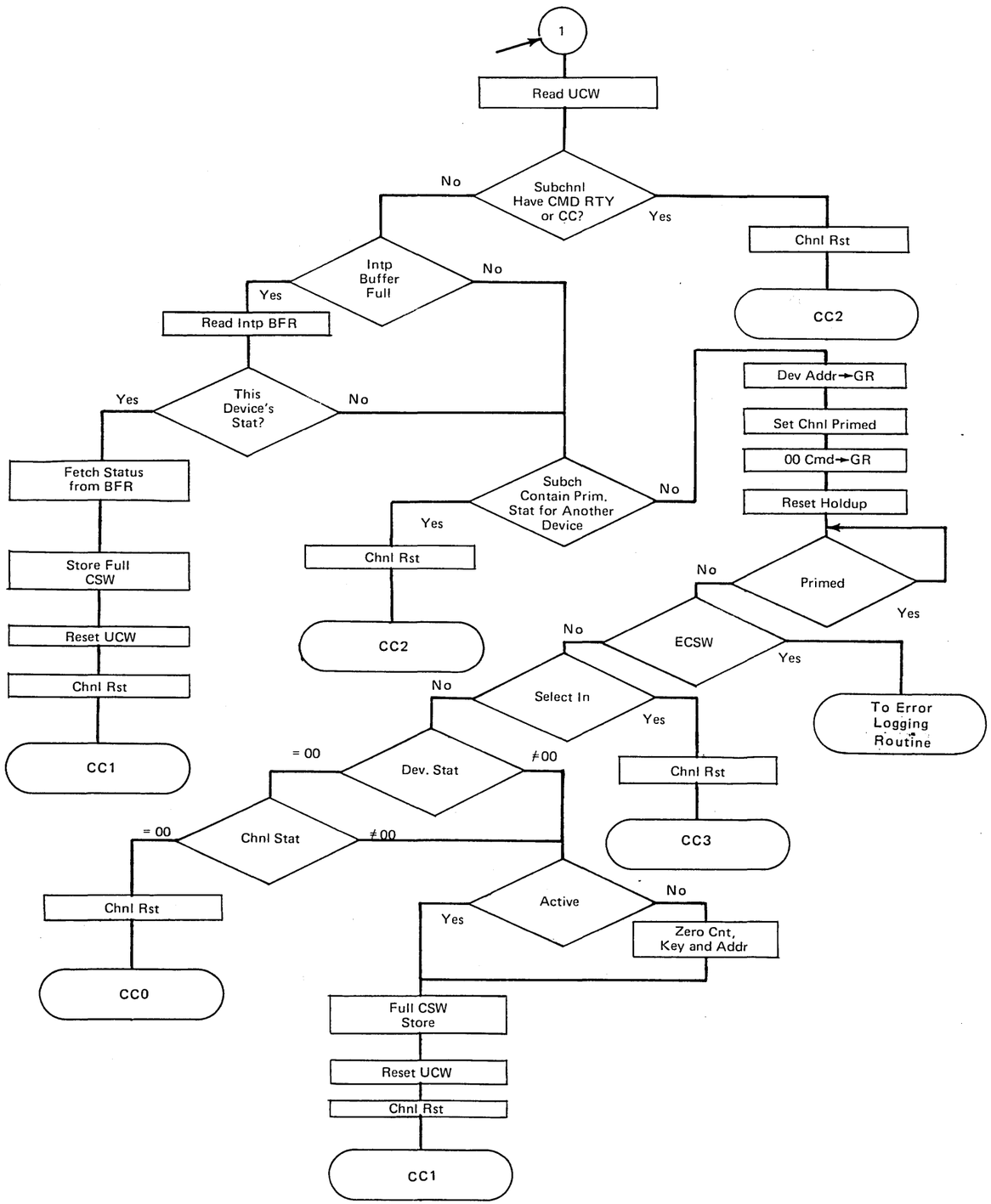
MH = Function of Microprogram and Hardware

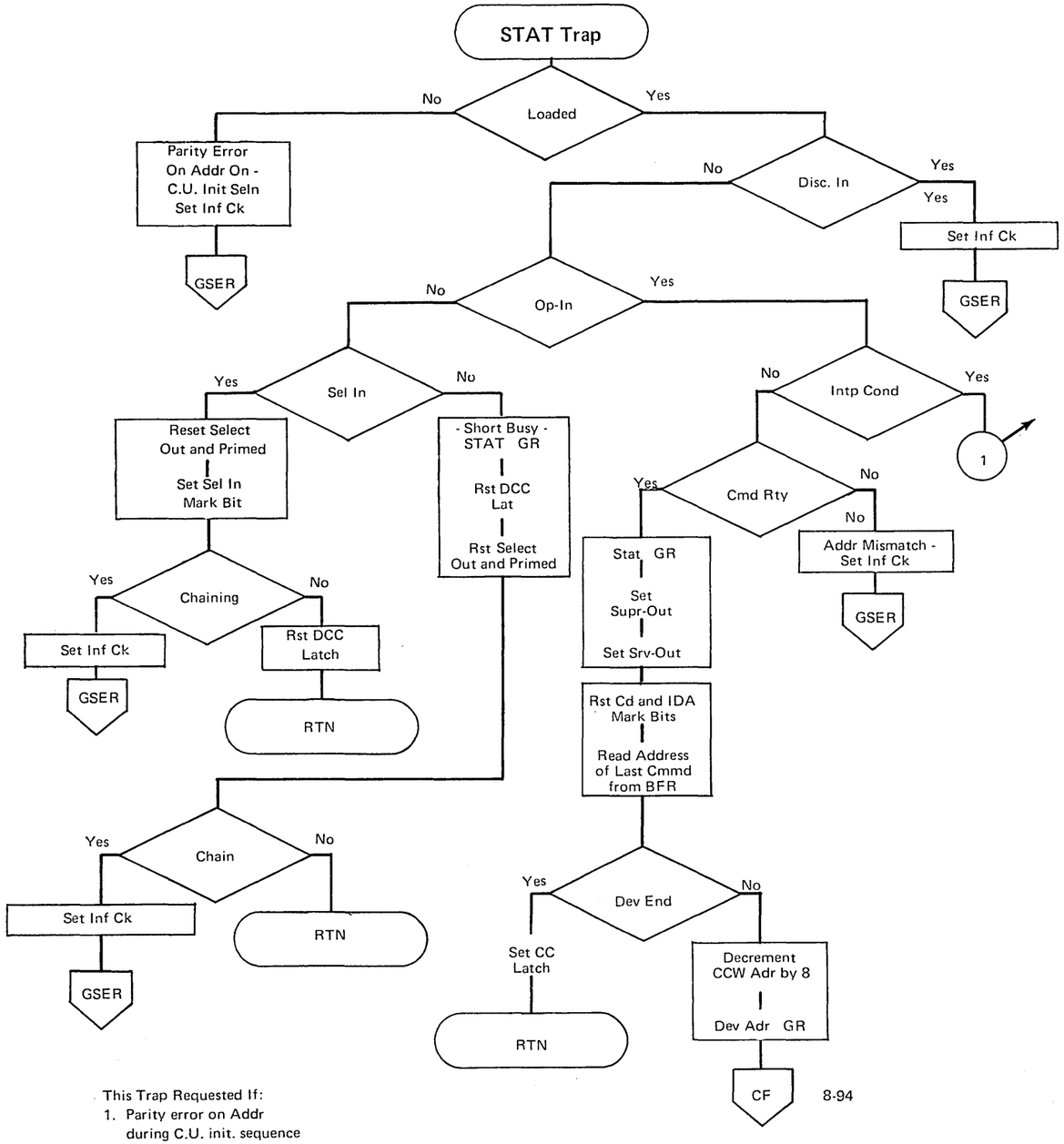
BLOCK-MULTIPLEXER OBJECTIVES






TIO
Error
No
Yes
Store Logout Pending CSW
CC1
Set Poll Ctrl
Loaded
Yes
No
1
Yes
Is a Primary Intp for Addr Dev in Chnl
No
Prime Chnl Avail Intp
CC1
Load Status from Intrp Buffer
Store Full CSW
Reset UCW
Chnl Reset
CC1

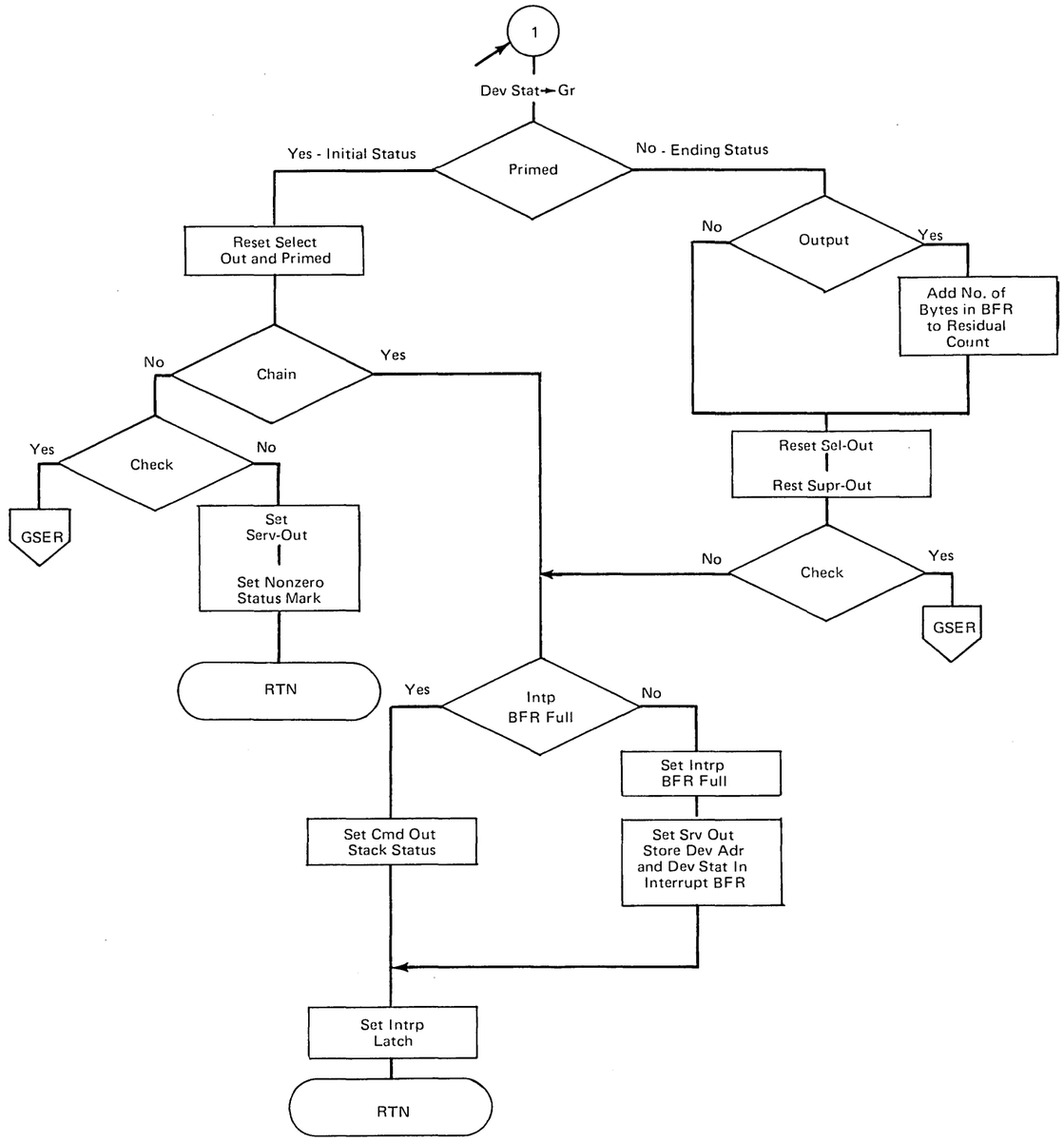

1
Read UCW
Subchnl Have CMD RTY or CC?
No
Yes
Chnl Rst
CC2
Intp Buffer Full
Yes
No
Read Intp BFR
This Device's Stat?
Yes
No
Fetch Status from BFR
Store Full CSW
Reset UCW
Chnl Rst
CC1
Subch Contain Prim. Stat for Another Device
No
Yes
Chnl Rst
CC2
Dev Addr→GR
Set Chnl Primed
00 Cmd→GR
Reset Holdup
Primed
No
Yes
ECSW
No
Yes
To Error Logging Routine
Select In
No
Yes
Chnl Rst
CC3
Dev. Stat
= 00
≠00
Chnl Stat
= 00
≠00
Chnl Rst
CC0
Active
Yes
No
Zero Cnt, Key and Addr
Full CSW Store
Reset UCW
Chnl Rst
CC1

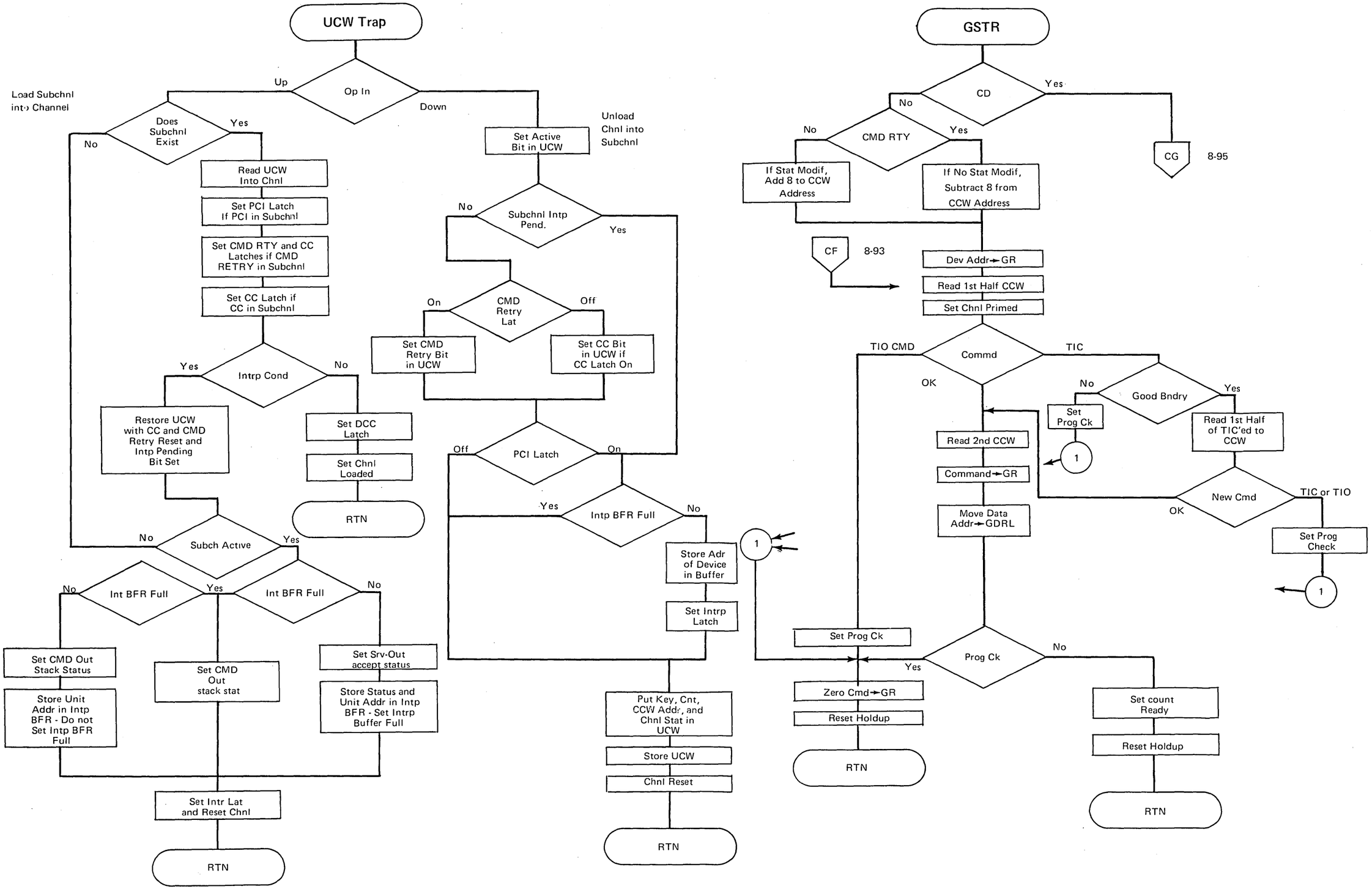


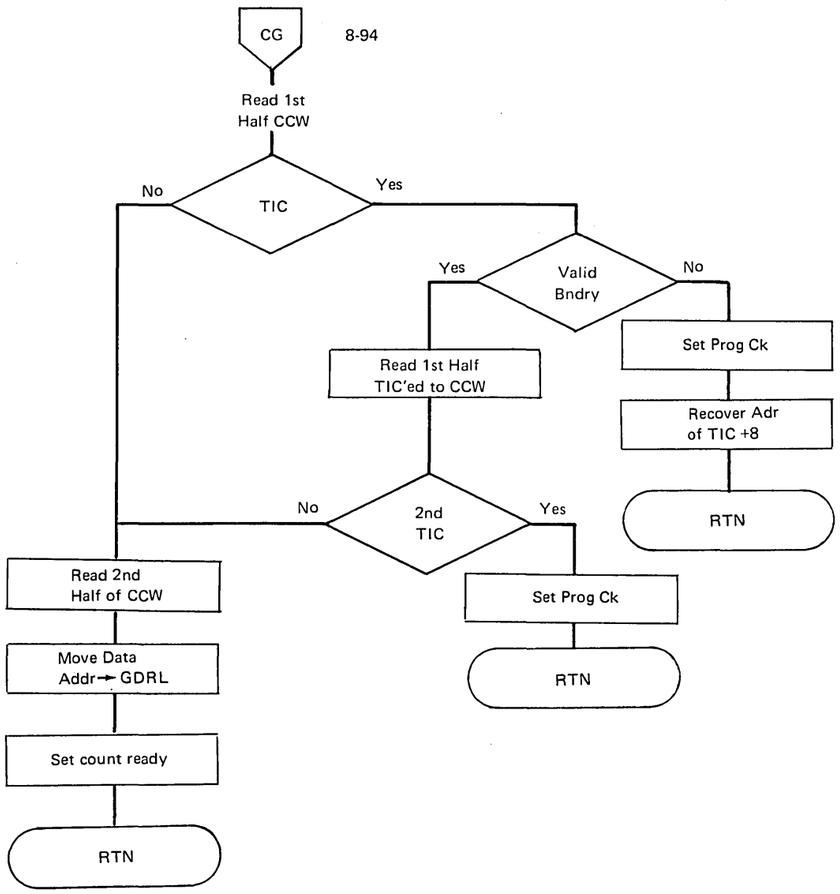
8-94

This Trap Requested If:

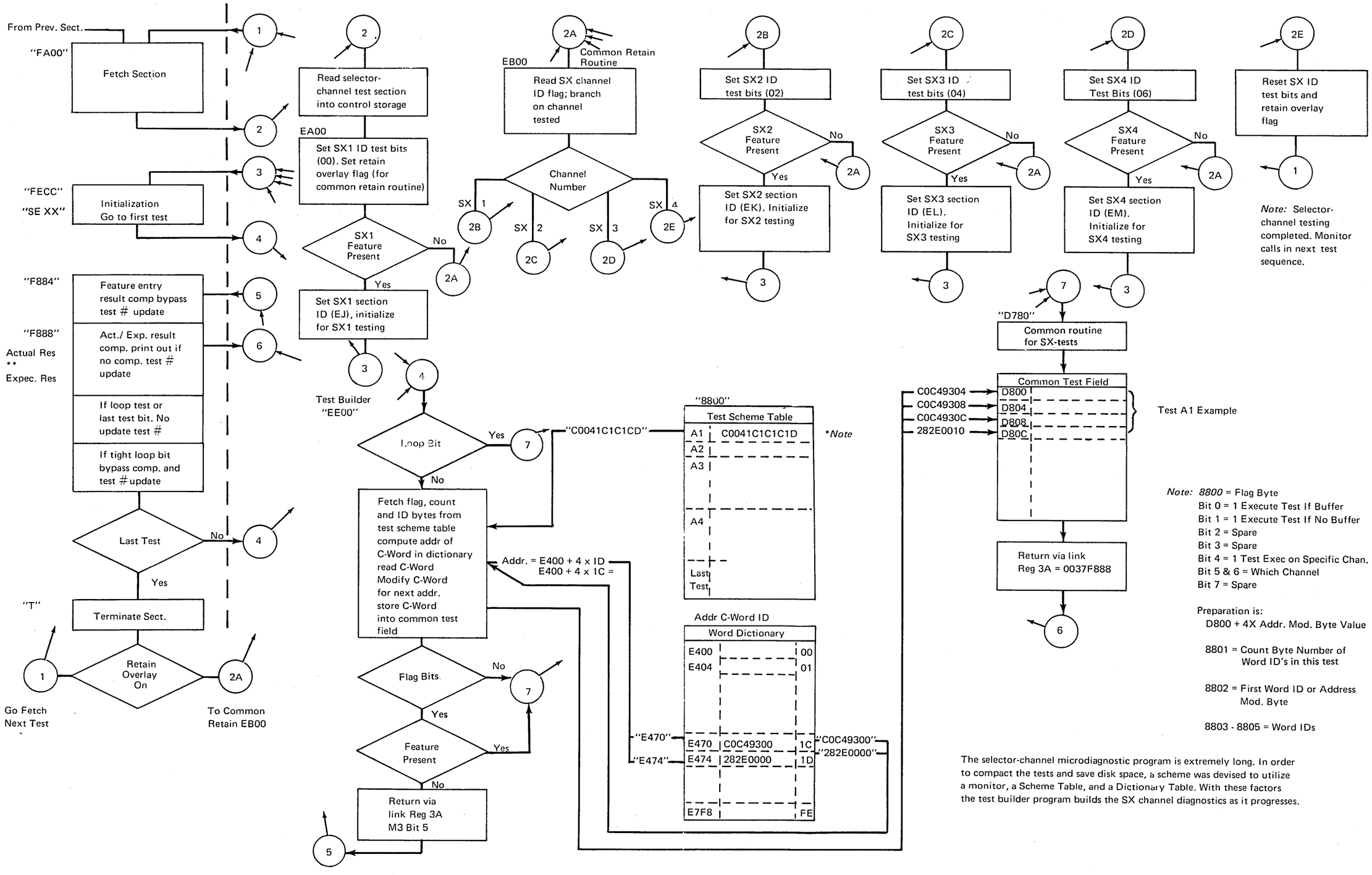
1. Parity error on Addr during C.U. init. sequence
2. Adr in does not match adr out.
3. Select-in on initial seln.
4. Short Ctrl-unit busy seq.
5. Status-in with intrp. condition.
6. Disconnect-in received
7. Command retry signal




UCW Trap
Load Subchnl into Channel
Up
Op In
Down
Unload Chnl into Subchnl
Does Subchnl Exist
No
Yes
Read UCW Into Chnl
Set PCI Latch If PCI in Subchnl
Set CMD RTY and CC Latches if CMD RETRY in Subchnl
Set CC Latch if CC in Subchnl
Intrp Cond
Yes
No
Restore UCW with CC and CMD Retry Reset and Intp Pending Bit Set
Set DCC Latch
Set Chnl Loaded
RTN
Subch Active
No
Yes
Int BFR Full
No
Yes
Set CMD Out Stack Status
Store Unit Addr in Intp BFR - Do not Set Intp BFR Full
Set CMD Out stack stat
Int BFR Full
Yes
No
Set Srv-Out accept status
Store Status and Unit Addr in Intp BFR - Set Intrp Buffer Full
Set Intr Lat and Reset Chnl
RTN
Set Active Bit in UCW
Subchnl Intp Pend.
No
Yes
CMD Retry Lat
On
Off
Set CMD Retry Bit in UCW
Set CC Bit in UCW if CC Latch On
PCI Latch
Off
On
Intp BFR Full
Yes
No
Store Adr of Device in Buffer
Set Intrp Latch
Put Key, Cnt, CCW Addr, and Chnl Stat in UCW
Store UCW
Chnl Reset
RTN
1
GSTR
CD
No
Yes
CG 8-95
CMD RTY
No
Yes
If Stat Modif, Add 8 to CCW Address
If No Stat Modif, Subtract 8 from CCW Address
CF 8-93
Dev Addr→GR
Read 1st Half CCW
Set Chnl Primed
Commd
TIO CMD
TIC
OK
Read 2nd CCW
Command→GR
Move Data Addr→GDRL
Set Prog Ck
Zero Cmd→GR
Reset Holdup
RTN
Prog Ck
Yes
No
Set count Ready
Reset Holdup
RTN
Good Bndry
No
Yes
Set Prog Ck
1
Read 1st Half of TIC'ed to CCW
New Cmd
OK
TIC or TIO
Set Prog Check
1


CG
8-94
Read 1st Half CCW
TIC
No
Yes
Valid Bndry
Yes
No
Read 1st Half TIC'ed to CCW
Set Prog Ck
Recover Adr of TIC +8
RTN
2nd TIC
No
Yes
Read 2nd Half of CCW
Set Prog Ck
Move Data Addr→ GDRL
RTN
Set count ready
RTN

SELECTOR-CHANNEL TEST EXAMPLE



Note: *8800* = Flag Byte
Bit 0 = 1 Execute Test If Buffer
Bit 1 = 1 Execute Test If No Buffer
Bit 2 = Spare
Bit 3 = Spare
Bit 4 = 1 Test Exec on Specific Chan.
Bit 5 & 6 = Which Channel
Bit 7 = Spare

Preparation is:
D800 + 4X Addr. Mod. Byte Value

8801 = Count Byte Number of Word ID's in this test

8802 = First Word ID or Address Mod. Byte

8803 - 8805 = Word IDs

The selector-channel microdiagnostic program is extremely long. In order to compact the tests and save disk space, a scheme was devised to utilize a monitor, a Scheme Table, and a Dictionary Table. With these factors the test builder program builds the SX channel diagnostics as it progresses.

Chapter 9. Optional Features

IBM 1401/1440/1460 AND 1410/7010 COMPATIBILITY

The 1401/1440/1460 and 1410/7010 Compatibility Features for the Model 145 consist entirely of microprogram instructions. Emulator programs provided for the Model 145 use these Compatibility Feature microroutines in conjunction with simulation routines, the Model 145 instruction set, and the OS supervisor and data-management routines to emulate the 1400/7010 series operations.

Two compatibility features are available:

- 1401/1440/1460 Compatibility. This feature permits the system to execute 1401/1460 and 1440 instructions.
- 1401/1440/1460, 1410/1710 Compatibility. This feature executes 1401/1440/1460 *and* 1410/7010 instructions.

This section describes the characteristics of the 1401/1440/1460, 1410/7010 feature, because it is capable of performing the functions of both features, depending on the emulator program that is in use at the time. Two emulator programs are provided for the two unique compatibility operations. Differences in storage requirements, common region assignment, etc. are indicated as necessary for the two features.

For ease in discussing the various combinations of features, the term 1400 refers to 1401/1440/1460 operations, and the term 1410 refers to 1410/7010 operations. 1400/1410 refers to the combined feature.

The operations performed by this feature (via microprogram routines are:

- 1400/1410 Op-code fetching and decoding, address translation, indexing, and error checking.
- 1400/1410 arithmetic operations, including Modify Address for 1400 but excluding Multiply and Divide.
- Comparisons.
- 1400/1410 Branch if Character Equal (BCE), Branch if Bit Equal (BBE), and Branch if Wordmark or Zone Equal (BWE). The 1400 also handles unconditional branches and store address registers.
- CPU data moves.
- Data handling on input/output operations.
- 1400/1410 address verification (BIFLAG).

The compatibility feature fetches and analyzes each 1400 instruction. It executes some instructions directly; for others, it passes control to the emulator program, which emulates the instruction and returns control to the compatibility feature.

The Integrated Emulator Program uses a special emulator instruction for performing the operations of the 1401/1410 compatibility features. This special instruction has a six-byte format. The first byte is always EA. The second byte designates the operation to be performed. The remaining bytes are used as needed for address and general register indicators. Refer to the Emulator Instruction section for details.

1400-series characters (program and data) are stored in the simulated 1400/1410 main storage by the Integrated Emulator Program as a special 8-bit internal code. The bits of the internal code are weighted for 1400/1410 representation as shown.

Internal Code bit	0	1	2	3	4	5	6	7
1400/1410 Weight	8	4	2	1	B	A	WM	0*

*Bit 7 on is an error.

The exchange of information between the integrated emulator program and the 1400/1410 Compatibility feature is accomplished through the defined use of general registers and a section of main storage referred to as the *common region.*

STORAGE ALLOCATIONS

The emulator program designates certain required Model 145 facilities. Segments of main storage are assigned, but they have address flexibility that can be under control of the operating system. General registers are given assignment by the emulator program. The microprogram routines reside in control storage.

Main Storage

Segments of main storage are assigned for:

- The emulator routines, consisting of simulation routines and the emulator Op codes DIL, BDIL, ANUM, COMP, MCPU, MIO, and BIFLAG. The amount of main storage required depends on the options chosen when the emulator is generated. The common region area varies depending on whether the 1400 or the 1400/1410 feature is present.
- Data-management routines. The control program and the number and type of input/output units that are in use affect the storage requirement.
- Buffers: Main storage required for buffers for unit-record equipment, tape units, and disk units depends on the number and types of units being emulated.

1401/1440/1460	PARTITION OR REGION	1410/7010
2K to 10K	Data-Management Routines	2K to 10K
	Buffers and Control Blocks	
	Available Storage	
2K to 16K 1401 Systems 2K to 16K 1440 Systems 8K to 16K 1460 Systems	Simulated 1400/7010 Storage (size of system being emulated)	10K to 80K 1410 Systems 40K to 100K 7010 Systems
Approximately 20K to 34K	Emulator Routines (including common region)	Approximately 22.5K to 44K

Local Storage

General registers 2 through 7 are used to simulate 1400 registers; as base addresses that point to emulator facilities; and as work registers, control byte, and other control information for the operation of the feature.

Control Storage

Control storage contains the compatibility sequences HAUM, HBVW, HCOM, HICY, HIND, HIOA, HMCP, HMIO, HMOD, HOPD, HOUT, and HZAS. The number of words required is approximately 1200 for the 1400, and 1450 for the 1410.

		GENERAL REGISTERS BYTE USAGE		
GR	BYTE 0	BYTE 1	BYTE 2	BYTE 3
2	MICROPROGRAM WORK REGISTER			
3	ZEROS	COMMON REGION Base Address B_2ddd		
4	VALID ADDRESS*	BAR	B-ADDRESS REGISTER	
5	VALID ADDRESS*	AAR	A-ADDRESS REGISTER	
6	DIL COUNT†	IAR	INSTRUCTION ADDRESS REGISTER	
7	CONTROL BYTE	LINK REGISTER		

* BYTE ZERO = VALID ADDRESS
BYTE NONZERO = INVALID ADDRESS

† 1400 – BIT 7 SHOULD ALWAYS BE 0
1410 – BIT 7 INDICATES INTERRUPT PENDING

Common Region

The elements of the common region are listed by *word/byte* in the following chart. The *Hex Addr* column is the displacement of its word from the DIL pointer (B^1ddd or B^2ddd). The pointer addresses must be located on a 256-byte boundary.

Number of Words	Hex Addr Byte(s)	Use/Contents
1A. 8	-20	8 words of working storage for micro-program use
1. 1	B^2ddd 0 1, 2, 3	D-modifier Current IAR if an I/O Op having instruction length of greater than 4. If a CPU Op, bytes 1, 2 = 0; byte 3 = instruction length/switches.
2. 1	04 0 1, 2, 3	Zeros Minimum boundary of target CPU
3. 1	08 0 1, 2, 3	Zeros Maximum boundary of target CPU
4. 1	0C 0 1, 2, 3	Zeros Storage wrap address
5. 1	10 0 1, 2, 3	CPU indicator DIL Interrupt address
6. 1	14 0 1, 2, 3	Zeros DIL restart address
7. 1	18 0 1, 2, 3	Zeros BDIL restart address
8. 1	1C 0 1, 2, 3	Zeros Address error address
9. 1	20 0 1, 2, 3	Zeros 1400 U I/O Op-code address (Not used in 1410)
10. 1	24 0 1, 2, 3	Zeros 1410 Control Address Register (CAR)
11. 1	28 0 1, 2, 3	Zeros 1400 unusual instruction format address or 1410 invalid instruction format address
12. 1	2C 0 1, 2, 3	Zeros 1400 move or load I/O Op-code address or 1410 Interrupt handling address
13. 1	30 0 1, 2, 3	Zeros Address pointer to a collating sequence table
14. 64	x00	Internal code to EBCDIC conversion table
15. 64	x00	EBCDIC to internal code conversion table
16. 64	B^1ddd	1400/1410 Op-code table
17. 64	100	1400 hundreds address table or 1410 ten thousands address table
18. 64	200	1400 tens address table or 1410 thousands address table
19. 64	300	1400 units address table or 1410 hundreds address table
20. 64	400	1410 tens address table
21. 64	500	1410 units address table
22. 4	400	1400 index table. Note displacement or
16	600	1410 index table. Note displacement

1A. Eight words of working storage are reserved for micro-programming. These words are accessed by the microprogram through a negative displacement from the B^2ddd operand of the DIL instruction. The purpose of the eight words is for hardware retry and the interruptibility of long operations. These words provide the microprogram with storage space for re-entrant hardware routines.

1. *Byte 0 – DMOD:* The d-modifier associated with the current 1400/1410 instruction.

   Bytes 1, 2, 3 – CURRENT IAR: When the microprogram detects a valid-length I/O operation with an X-control field, these three bytes contain the address of the I/O instruction.

When I/O operations are not encountered, bytes 1 and 2 are set to zeros, and byte 3 is used as an instruction-length counter, and switches as follows.

For 1400

Byte Bits:	Length 0	1	2	3	4	Indicator 5	6	7	Instruction Length
	0	0	0	1	0	0	0	0	1
	0	0	1	0	0	0	0	1	2
	0	0	1	1	0	0	0	0	3
	0	1	0	0	0	1	0	0	4
	0	1	0	1	0	1	0	1	5
	0	1	1	0	0	1	0	0	6
	0	1	1	1	0	1	1	0	7
	1	0	0	0	0	1	1	1	8

Note: If instruction length is greater than 8, that byte becomes FF.

Indicator Bits:
- 5 on = Complete A-address
- 6 on = Complete B-address
- 7 on = d-modifier for instruction lengths of 2, 5, and 8

For 1410

Byte Bits:	Length 0	1	2	3	4	Indicator 5	6	7	Instruction Length
	0	0	0	1	0	0	0	0	1
	0	0	1	0	0	0	0	1	2
	0	0	1	1	0	0	0	0	3
	0	1	0	0	0	0	0	0	4
	0	1	0	1	0	0	0	0	5
	0	1	1	0	0	1	0	0	6
	0	1	1	1	0	1	0	1	7
	1	0	0	0	0	1	0	0	8
	1	0	0	1	0	1	0	0	9
	1	0	1	0	0	1	0	0	10
	1	0	1	1	0	1	1	0	11
	1	1	0	0	0	1	1	1	12

Indicator Bits:
- 5 on = Complete A-address
- 6 on = Complete B-address
- 7 on = d-modifier for instruction lengths of 2, 7, and 12

Note: The dual use of these bytes is stressed and should be kept in mind. The length and indicator bits are always set except for the following conditions.

- An I/O Op-code with an I/O X-control field, and having an instruction length greater than four, is detected. In this case, bytes 1, 2, and 3 are an absolute binary address of the I/O Op-code.
- When the control byte of an Op-code table entry is X'00'.
- When an unusual condition is encountered while decoding the operation code.

2. *Byte 0:* Zeros.

   Bytes 1, 2, and 3: — Minimum Boundary of Target CPU: This is the address of simulated 1400/1410 storage location 0. It is also the relocation factor for target storage.

3. *Byte 0:* Zeros.

   Bytes 1, 2, and 3: — Maximum Boundary of Target CPU: This is the address of the last effective storage location of the target CPU.

4. *Byte 0:* Zeros.

   Bytes 1, 2, and 3 — Storage Wrap Address: This is the storage wrap address for 1400/1410 indexing. It contains up to 16,000 in binary plus the relocation factor for the 1400, and up to 100,000 in binary plus the relocation factor for the 1410.

5. *Byte 0 — CPU Indicators:* This byte is used as a CPU compare indicator, as follows.

Bit 0	Zero balance (1410 only).
Bit 1	Low
Bit 2	High
Bit 3	Equal
Bit 4	Divide overflow (1410 only).
Bit 5	Arithmetic overflow
Bit 6	Not used
Bit 7	Not used.

   Bytes 1, 2, 3 — DIL Interrupt Address: A DIL count is kept in bits 0-6 of byte 0 in GPR 6. The DIL count is decremented by one for each DIL instruction issued by the emulator program. When the DIL count goes to zero, the address kept here is loaded into the current PSW to branch to an emulator-program subroutine that handles time dependencies. If the DIL count is initially zero, it is assumed that no interrupt is contemplated.

6. *Byte 0:* Zeros.

   Bytes 1, 2, and 3 — DIL Restart Address: Contains the address of a single DIL instruction. This is used by the microprogram to restart DIL when a system interrupt is taken after a 1400/1410 B, V, or W Op-code execution, or after a 1400 Q or H Op-code execution.

7. *Byte 0:* Zeros.

   Bytes 1, 2, and 3 — BDIL Restart Address: Contains the address of a single branch DIL.

8. *Byte 0:* Zeros.

   Bytes 1, 2, and 3 — Address Error: Contains the binary address of an emulator program subroutine to handle addresses that constitute an address reference outside of target storage.

9. *Byte 0:* Zeros.

   Bytes 1, 2, and 3 — 1400 U I/O Op-Codes: Contains an address of an emulator program to be inserted in the current PSW when the 1400 instruction contains an X-control field, that is, a 1400 U%xxd instructions. (Not used for 1410.)

10. *Byte 0:* Zeros.

    Bytes 1, 2, and 3: Contains the binary address of the 1410 control address register. (Not used for 1400.)

11. *Byte 0:* Zeros.

    Bytes 1, 2, and 3 — Invalid Format Address: Contains the address of an emulator-program subroutine to which the compatibility feature links when:
    1. A wordmark is missing from an Op-code, or
    2. An invalid instruction format is detected.

12. *Byte 0:* Zeros.

    Bytes 1, 2, 3 (1400) — 1400 Move or Load I/O Op-Code: These bytes contain the subroutine address that simulates the 1400 move I/O or load I/O operation.

    Bytes 1, 2, 3 (1410) — 1410 Interrupt: These bytes contain the subroutine address that handles the 1410 interruptions, if permitted within 1410 DIL operations.

13. *Byte 0:* Zeros.

    Bytes 1, 2, and 3: Contains the binary address of a collating sequence table (1410 only).

14. *64-Word Internal Code to EBCDIC Conversion Table:* This table is used by the emulator program to convert internal code to EBCDIC (see Table on page 9-7).

15. *64-Word EBCDIC to Internal Code Conversion Table:* Used by the emulator program to convert EBCDIC to Internal Code (see Table 1).

16. *64-Word Op-Code Table:* The first word of this table is at the address specified by B^{1}ddd of the DIL or BDIL instruction. It is arranged so that the 1400/1410 Op-code in Internal Code minus 2 (resets the wordmark bit) plus the B^{1}ddd portion of DIL equals the correct table entry for that Op-code. The table must be on a 256-byte boundary (last eight bits of address are zeros). Each word of the table is:

    Byte 0 = control byte.
    Bytes 1, 2, 3 = subroutine address to simulate Op code with the exceptions of 1400 V, W, Q, and H Op codes and 1410 B, V, and W Op codes.

17.,
18.,
19.,
20.,

21. *64-Word Address Conversion Tables:* These tables are used to translate 1400/1410 addresses (in internal code) to a Model 145 binary address that includes the relocation factor. See "Conversion of 1400/1410 Addresses" within the DIL description.

22. *4-Word Index Table (1400):* When 1400 is specified, this table is displaced x'400' from B^{1}ddd. The last three words correspond to the 1400 index registers. The contents of each word represent the address of the units position of a 1400 index register with the relocation factor included. The first word contains the minimum 1400 address (in binary) plus the relocation factor.

    16-Word Index Table (1410): When 1410 is specified, this table is displaced x'600' from B^{1}ddd. The last fifteen words correspond to the 1410 index registers. The contents of each word represent the address of the units position of a 1410 index register with the relocation factor included. The first word contains the minimum 1410 address (in binary) plus the relocation factor.

1400 Char	BCD Code	WITHOUT WORDMARKS Internal Code	I/C Hex	EBCDIC	Hex	WITH WORDMARKS Internal Code	I/C Hex	EBCDIC	Hex
Blank	000000	0000 0000	00	0100 0000	40	0000 0010	02	0000 0000	00
.	111011	1011 1100	BC	0100 1011	4B	1011 1110	BE	0000 1011	0B
) or ¤	111100	1100 1100	CC	0100 1100	4C	1100 1110	CE	0000 1100	0C
[	111101	1101 1100	DC	0100 1101	4D	1101 1110	DE	0000 1101	0D
<	111110	1110 1100	EC	0100 1110	4E	1110 1110	EE	0000 1110	0E
ǂ (GM)	111111	1111 1100	FC	0100 1111	4F	1111 1110	FE	0000 1111	0F
& or +	110000	0000 1100	0C	0101 0000	50	0000 1110	0E	0001 0000	10
$	101011	1011 1000	B8	0101 1011	5B	1011 1010	BA	0001 1011	1B
*	101100	1100 1000	C8	0101 1100	5C	1100 1010	CA	0001 1100	1C
]	101101	1101 1000	D8	0101 1101	5D	1101 1010	DA	0001 1101	1D
;	101110	1110 1000	E8	0101 1110	5E	1110 1010	EA	0001 1110	1E
△	101111	1111 1000	F8	0101 1111	5F	1111 1010	FA	0001 1111	1F
-	100000	0000 1000	08	0110 0000	60	0000 1010	0A	0010 0000	20
/	010001	0001 0100	14	0110 0001	61	0001 0110	16	0010 0001	21
,	011011	1011 0100	B4	0110 1011	6B	1011 0110	B6	0010 1011	2B
% or (	011100	1100 0100	C4	0110 1100	6C	1100 0110	C6	0010 1100	2C
V (WS)	011101	1101 0100	D4	0110 1101	6D	1101 0110	D6	0010 1101	2D
\	011110	1110 0100	E4	0110 1110	6E	1110 0110	E6	0010 1110	2E
≠	011111	1111 0100	F4	0110 1111	6F	1111 0110	F6	0010 1111	2F
b	010000	0000 0100	04	0111 1010	7A	0000 0110	06	0011 1010	3A
#	001011	1011 0000	B0	0111 1011	7B	1011 0010	B2	0011 1011	3B
@ or '	001100	1100 0000	C0	0111 1100	7C	1100 0010	C2	0011 1100	3C
:	001101	1101 0000	D0	0111 1101	7D	1101 0010	D2	0011 1101	3D
>	001110	1110 0000	E0	0111 1110	7E	1110 0010	E2	0011 1110	3E
√	001111	1111 0000	F0	0111 1111	7F	1111 0010	F2	0011 1111	3F
?	111010	1010 1100	AC	1100 0000	C0	1010 1110	AE	1000 0000	80
A	110001	0001 1100	1C	1100 0001	C1	0001 1110	1E	1000 0001	81
B	110010	0010 1100	2C	1100 0010	C2	0010 1110	2E	1000 0010	82
C	110011	0011 1100	3C	1100 0011	C3	0011 1110	3E	1000 0011	83
D	110100	0100 1100	4C	1100 0100	C4	0100 1110	4E	1000 0100	84
E	110101	0101 1100	5C	1100 0101	C5	0101 1110	5E	1000 0101	85
F	110110	0110 1100	6C	1100 0110	C6	0110 1110	6E	1000 0110	86
G	110111	0111 1100	7C	1100 0111	C7	0111 1110	7E	1000 0111	87
H	111000	1000 1100	8C	1100 1000	C8	1000 1110	8E	1000 1000	88
I	111001	1001 1100	9C	1100 1001	C9	1001 1110	9E	1000 1001	89
!	101010	1010 1000	A8	1101 0000	D0	1010 1010	AA	1001 0000	90
J	100001	0001 1000	18	1101 0001	D1	0001 1010	1A	1001 0001	91
K	100010	0010 1000	28	1101 0010	D2	0010 1010	2A	1001 0010	92
L	100011	0011 1000	38	1101 0011	D3	0011 1010	3A	1001 0011	93
M	100100	0100 1000	48	1101 0100	D4	0100 1010	4A	1001 0100	94
N	100101	0101 1000	58	1101 0101	D5	0101 1010	5A	1001 0101	95
O	100110	0110 1000	68	1101 0110	D6	0110 1010	6A	1001 0110	96
P	100111	0111 1000	78	1101 0111	D7	0111 1010	7A	1001 0111	97
Q	101000	1000 1000	88	1101 1000	D8	1000 1010	8A	1001 1000	98
R	101001	1001 1000	98	1101 1001	D9	1001 1010	9A	1001 1001	99
ǂ (RM)	011010	1010 0100	A4	1110 0000	E0	1010 0110	A6	1010 0000	A0
S	010010	0010 0100	24	1110 0010	E2	0010 0110	26	1010 0010	A2
T	010011	0011 0100	34	1110 0011	E3	0011 0110	36	1010 0011	A3
U	010100	0100 0100	44	1110 0100	E4	0100 0110	46	1010 0100	A4
V	010101	0101 0100	54	1110 0101	E5	0101 0110	56	1010 0101	A5
W	010110	0110 0100	64	1110 0110	E6	0110 0110	66	1010 0110	A6
X	010111	0111 0100	74	1110 0111	E7	0111 0110	76	1010 0111	A7
Y	011000	1000 0100	84	1110 0111	E8	1000 0110	86	1010 1000	A8
Z	011001	1001 0100	94	1110 1001	E9	1001 0110	96	1010 1001	A9
0	001010	1010 0000	A0	1111 0000	F0	1010 0010	A2	1011 0000	B0
1	000001	0001 0000	10	1111 0001	F1	0001 0010	12	1011 0001	B1
2	000010	0010 0000	20	1111 0010	F2	0010 0010	22	1011 0010	B2
3	000011	0011 0000	30	1111 0011	F3	0011 0010	32	1011 0011	B3
4	000100	0100 0000	40	1111 0100	F4	0100 0010	42	1011 0100	B4
5	000101	0101 0000	50	1111 0101	F5	0101 0010	52	1011 0101	B5
6	000110	0110 0000	60	1111 0110	F6	0110 0010	62	1011 0110	B6
7	000111	0111 0000	70	1111 0111	F7	0111 0010	72	1011 0111	B7
8	001000	1000 0000	80	1111 1000	F8	1000 0010	82	1011 1000	B8
9	001001	1001 0000	90	1111 1001	F9	1001 0010	92	1011 1001	B9

GM = Groupmark
WS = Word Separator
RM = Recordmark
WM = Wordmark

The BCD code bit weights are: B A 8 4 2 1

The Internal Code Bit weights are: 8 4 2 1 B A WM X
where X being 0 = standard operation
1 = error

The EBCDIC code bit weights are: X $\overline{WM}$ B A 8 4 2 1
where X being 0 = special character
1 = standard character
where $\overline{WM}$ being 1 = no word-mark
0 = word-mark

1410/7010 Op Codes

OP CODE/ INSTRUCTION	FUNCTION	MNEMONIC	REMARKS BCD CODE	REMARKS CARD CODE
ARITHMETIC CODES				
A	Add	A	BA1	12-1
S	Subtract	S	CA2	0-2
?	Zero and Add	ZA	CBA82	12-0
!	Zero and Subtract	ZS	B82	11-0
@	Multiply	M	C84	4-8
%	Divide	D	A84	0-4-8
LOGIC OPERATION CODES				
J(I) Blank	Branch Unconditional	B	CB1	11-1
J(I)d	Test and Branch (Conditional)	See Table I for d-character		
R(I)d	Branch if I/O Channel Status Indicator On	See Table II for d-character (X Op Code specifies CH 2 operation)		
B(I)(B)d	Branch if Character Equal	BCE	Contents of B compared to d	
W(I)(B)d	Branch if Bit Equal	BBE	Contents of B compared to d	
V(I)(B)d	Branch on Word Mark or Zone Equal	See Table III for d-character		

OP CODE/ INSTRUCTION	FUNCTION	MNEMONIC	REMARKS BCD CODE	REMARKS CARD CODE
GENERAL DATA OPERATION CODES				
D(A)(B)d	Move Data	d is modifier (See chart)		
Z(A)(B)	Move Characters and Suppress Zeros	MCS	A81	0-9
E(A)(B)	Move Characters and Edit	MCE	CBA41	12-5
MISCELLANEOUS OPERATION CODES				
C(A)(B)	Compare	C	CBA21	12-3
T(A)(B)d	Table Lookup	See Table IV for d-character		
G(C)d	Store Address Register	S—R	d-character specifies register (A, B, E, F)	
, (A)(B)	Set Word Mark	SW	CA821	0-3-8
□(A)(B)	Clear Word Mark	CW	CBA84	12-4-8
/(B)	Clear Storage	CS	CA1	0-1
/(I)(B)	Clear Storage and Branch	CS	CA1	0-1
.	Halt	H	BA821	12-3-8
. (I)	Halt and Branch	H	BA821	12-3-8
N	No Operation	NOP	B41	11-5

DATA MOVE, d-CHARACTERS AND MNEMONICS

DIRECTION OF MOVE	CONDITION WHICH ENDS OPERATION	NO PORTION MOVED	MOVE NUMERIC PORTION OF A-FIELD TO B-FIELD	MOVE ZONE PORTION OF A-FIELD To B-FIELD	MOVE NUMERIC AND ZONE FROM A-FIELD TO B-FIELD	MOVE WM IN A-FIELD TO B-FIELD	MOVE NUMERIC AND WM FROM A-FIELD TO B-FIELD	MOVE ZONE AND WM FROM A-FIELD TO B-FIELD	MOVE NUMERIC, ZONE, AND WM FROM A- FIELD TO B-FIELD	BCD CODING (BITS)
RIGHT TO LEFT	Move Data One Position	BLANK SCNLS	1 MLNS	2 MLZS	3 MLCS	4 MLWS	5 MLNWS	6 MLZWS	7 MLCWS	NONE
	Move Data to 1st WM in A-Field	¢ SCNLA	/ MLNA	S MLZA	T MLCA	U MLWA	V MLNWA	W MLZWA	X MLCWA	A
	Move Data to 1st WM in B-Field	— SCNLB	J MLNB	K MLZB	L MLCB	M MLWB	N MLNWB	O MLZWB	P MLCWB	B
	Move Data to 1st WM in either A- or B-Field	& SCNL	A MLN	B MLZ	C MLC	D MLW	E MLNW	F MLZW	G MLCW	A B
LEFT TO RIGHT	Move Record to 1st WM in either A- or B-Field	8 SCNR	9 MRN	0 MRZ	# MRC	@ MRW	: MRNW	> MRZW	T√M MRCW	8
	Move Record to 1st RM in A-Field	Y SCNRR	Z MRNR	≠ MRZR	, (comma) MRCR	% MRWR	= WS MRNWR	, (apostrophe) MRZWR	" MRCWR	A 8
	Move Record to 1st GM-WM in A-Field	Q SCNRG	R MRNG	! MRZG	$ MRCG	* MRWG	) MRNWG	; MRZWG	Δ MRCWG	B 8
	Move Record to 1st RM or GM-WM in A-Field	H SCNRM	I MRNM	? MRZM	. MRCM	□ MRWM	(MRNWM	? MRZWM	≠ MRCWM	A B 8
	BCD CODING (BITS)	NONE	1	2	1, 2	4	1, 4	2, 4	1, 2, 4	

OP CODE/ INSTRUCTION	FUNCTION	MNEMONIC	REMARKS BCD CODE	REMARKS CARD CODE
INPUT-OUTPUT OPERATION CODES				
M/L(%Un)(B)R	Read a Card without/with Word Marks	R/RW	Units Position of X — Control Field Specifies Pocket Selection, if any	
M/L(%20)(B)W	Write a Line without/with Word Separator Characters	W/WW		
M/L(%21)(B)W	Write Word Marks without/ with Word Separator Characters	WM/WMW		
M/L(%4n)(B)W	Punch Card without/with word marks	P/PW	Units Position of X — Ctrl Field Specifies Pocket Sel	
M/L(%T0)(B)R	Read Console Printer/Load-Read Console Printer	RCP/RCPW		
M/L(%T0)(B)W	Write Console Printer/Load-Write Console Printer	WCP/WCPW		
U(%Un)d	Unit Control		See Table V for d-char	
M/L(%Un)(B)R	Read Tape without/with Word Marks	RT/RTW	See d-character Notes	
M/L(%Un)(B)W	Write Tape without/with Word Marks	WT/WTW		
M/L(%FO)(B)R/W	Seek Disk	SD		
M/L(%F1)(B)R	Read Disk Single Record without/with word marks	RD/RDW		
M/L(%F1)(B)W	Write Disk Single Record without/with word marks	WD/WDW		
M/L(%F2)(B)R	Read Disk Full Track without/with word marks	RDT/RDTW		
M/L(%F2)(B)W	Write Disk Full Track without/with word marks	WDT/WDTW		
M/L(%F3)(B)W	Write Disk Check without/with word marks	WDC/WDCW		
M(%F4)W	Write Address	— —		
Kd	Stacker Select and Feed	SSF	See Table VI for d-character	
Fd	Control Carriage	CC	See Table VII for d-character	

	d	OPERATION	MNEMONIC
CHARACTER AT d FOR			
TABLE I	**Blank**	Unconditional	B
	9	Carriage Channel 9	BC9
	@	Carriage Overflow (Channel 12)	BCV
	/	Compare Unequal	BU
	S	Compare Equal (B = A)	BE
	T	Compare Low (B < A)	BL
	U	Compare High (B > A)	BH
	V	Zero Balance	BZ
	W	Divide Overflow	BDV
	Z	Arithmetic Overflow	BAV
	Q	Branch Inquiry	BNQ
	1	Overlap In Process on Channel 1	BOL 1
	2	Overlap In Process on Channel 2	BOL 2
	R	Printer Carriage Busy	BPB
TABLE II	**1-Bit**	I/O Not Ready	BNR 1 or 2
	2-Bit	I/O Channel Busy	BCB 1 or 2
	4-Bit	I/O Data Check	BER 1 or 2
	8-Bit	I/O Condition	BEF 1 or 2
	A-Bit	I/O No Transfer	BNT 1 or 2
	B-Bit	I/O Wrong Length Record	BWL 1 or 2
	≠	Any I/O Channel Status Indicator ON	BA 1 or 2
	@	Read Back Check (write disk check)	BRC 1 or 2
TABLE III	**1**	Word Mark	BW
	2,B,K,S	Zone	BZN
	3,C,L,T	Word Mark or Zone	BWZ

	d	OPERATION	MNEMONIC
CHARACTER AT d FOR			
TABLE IV	**Blank**	Lookup to End	--
	1	Lookup Low	LL
	2	Lookup Equal	LE
	3	Lookup Low or Equal	LLE
	4	Lookup High	LH
	5	Lookup Low or High	LLH
	6	Lookup Equal or High	LEH
	7	Lookup to Any	--
TABLE V	**B**	Backspace Tape one record	BSP
	E	Skip and Blank Tape	SKP
	M	Write Tape Mark	WTM
	R	Rewind Tape	RWD
	U	Rewind and Unload Tape	RWU
TABLE VI	**0**	Normal Read Pocket	
	1	1 Pocket	
	2	8/2 Pocket	
TABLE VII	**1**	Immediate Skip to Channel 1	
	2	Immediate Skip to Channel 2	
	3	Immediate Skip to Channel 3	
	4	Immediate Skip to Channel 4	
	5	Immediate Skip to Channel 5	
	6	Immediate Skip to Channel 6	
	7	Immediate Skip to Channel 7	
	8	Immediate Skip to Channel 8	
	9	Immediate Skip to Channel 9	
	0	Immediate Skip to Channel 10	
	#	Immediate Skip to Channel 11	
	@	Immediate Skip to Channel 12	
	A	Skip after Print to Channel 1	
	B	Skip after Print to Channel 2	
	C	Skip after Print to Channel 3	
	D	Skip after Print to Channel 4	
	E	Skip after Print to Channel 5	
	F	Skip after Print to Channel 6	
	G	Skip after Print to Channel 7	
	H	Skip after Print to Channel 8	
	I	Skip after Print to Channel 9	
	?	Skip after Print to Channel 10	
	.	Skip after Print to Channel 11	
	□	Skip after Print to Channel 12	
	J	1 Immediate Space	
	K	2 Immediate Spaces	
	L	3 Immediate Spaces	
	/	1 Space After Print	
	S	2 Spaces After Print	
	T	3 Spaces After Print	

X-CONTROL FIELD NOTES:

Channel and overlap—mode by char in hundreds position of X-ctrl field. %—Ch 1, non-overlap; @—Ch 1, overlap; □—Ch 2, non-overlap; *—Ch 2, overlap.

Read/Write tape in odd-parity mode accomplished by substituting B for U in tens position of X-control field. (Mnemonic character of B follows standard mnemonic.)

d-CHARACTER NOTES:

d-character for Write Tape to end of storage is X; mnemonic is WTE.

d-character for Read Tape to end of storage or IRG is $; mnemonic is RTG. (These operations *cannot* be overlapped.)

MNEMONIC NOTES:

Overlap mnemonic for all applicable input-output codes is alphabetic O which follows the standard mnemonic.

OPERATION CODES

1	Read
2	Print
3	Print-Read
4	Punch
5	Read-Punch
6	Print-Punch
7	Print-Read-Punch
8	Read Release
9	Punch Release
A	Add
B	Branch
C	Compare
D	Move Digit
E	Edit
F	Form Control
H	Store B Star
K	Stacker Select
L	Load
M	Move
N	No Op
P	Move Record
Q	Store A Star
S	Subtract
U	Unit Control
V	Branch — WM or Zone
W	Branch — Bit Equal
X	Move — Insert Zeros
Y	Move Zone
Z	Move Zero Suppress
.	Stop
□	Clear Wordmark
/	Clear Storage
,	Set Wordmark
%	Divide
#	Modify Address
@	Multiply
?	Zero and Add
!	Zero and Subtract

d MODIFIERS

d CHARACTERS FOR BRANCH

BIIId

d-Character	Branch On
b	Unconditional
9	Carriage Channel #9
@	Carriage Channel #12
A	"Last Card" Switch (Sense Switch A)
B	Sense Switch B*
C	Sense Switch C*
D	Sense Switch D*
E	Sense Switch E*
F	Sense Switch F*
G	Sense Switch G*
K	End of Reel * **
L	Tape Transmission Error*
N	Access Inoperable*
?	Reader Error if I/O Check Stop Switch is off**
!	Punch Error if I/O Check Stop Switch is off**
P	Printer Busy (print storage feature)*
≠	Print Error if I/O Check Stop Switch is off**
/	Unequal Compare (B ≠ A)
*	Inquiry Clear*
Q	Inquiry Request*
R	Printer Carriage Busy (print storage feature)*
S	Equal Compare (B=A)*
T	Low Compare (B < A)*
U	High Compare (B > A)*
V	Read-Write Parity Check or Read-Back Check Error*
W	Wrong-Length Record*
X	Unequal-Address Compare*
Y	Any Disk-Unit Error Condition*
Z	Overflow**
%	Processing Check with Process Check Switch off**

**Special feature.*

***Conditions tested are reset by a BRANCH IF INDICATOR ON instruction.*

d CHARACTERS FOR BRANCH IF WORDMARK OR ZONE

VIIBBBd

d-Character	Condition
1	Wordmark
2	No zone (No-A, No-B-bit)
B	12-zone (AB-bits)
K	11-zone (B, No-A-bit)
S	Zero-zone (A, No-B-bit)
3	Either a wordmark, or no zone
C	Either a wordmark, or 12-zone
L	Either a wordmark, or 11-zone
T	Either a wordmark, or zero-zone

d CHARACTERS FOR FORM CONTROL

Fd

d	Immediate skip to	d	Skip after print to
1	Channel 1	A	Channel 1
2	Channel 2	B	Channel 2
3	Channel 3	C	Channel 3
4	Channel 4	D	Channel 4
5	Channel 5	E	Channel 5
6	Channel 6	F	Channel 6
7	Channel 7	G	Channel 7
8	Channel 8	H	Channel 8
9	Channel 9	I	Channel 9
0	Channel 10	?	Channel 10
#	Channel 11	.	Channel 11
@	Channel 12	□	Channel 12

d	Immediate space	d	After print-space
J	1 space	/	1 space
K	2 spaces	S	2 spaces
L	3 spaces	T	3 spaces

1440 Op Codes

Because of the complexity of the 1440 Op-code and modifier structure, a brief summarization in this manual is impractical. Refer to the System Operation Manual, IBM 1440 Data Processing System, A24-3116, for detailed 1440 Op-code information.

1400/1410 COMPATIBILITY FEATURE INSTRUCTION EA

The 1400/1410 compatibility feature instruction is six bytes long. The first byte is the Op-code EA. The second byte designates the function to be performed. The remaining four bytes are used as described under each function.

The functions are:

1st Byte	2nd Byte	Mnem	
EA	00	DIL	Do Interpretive Loop Function: 1400/1410 Op-code fetching, indexing, address conversion, and some branching.
EA	01	BDIL	Branch DIL
EA	04	ANUM	Add Numeric Function: 1400/1410 arithmetic operations except multiply and divide.
EA	05	COMP	Compare Function: 1400/1410 compares, low order to high.
EA	08	MCPU	Move Data in CPU Function: 1400/1410 data moves that are internal to the target storage.
EA	09	MIO	Move Data for Input/Output Function: Data is moved to simulated 1400/1410 storage from Model 145 buffer area, and vice versa.
EA	0C	BIFLAG	Branch Invalid Flag Function: Branches to an error routine if an invalid address is indicated.

DIL And BDIL — Do Interpretive Loop

Instruction formats are:

DIL

EA 00	B^1ddd	B^2ddd

BDIL

FA 01	B^1ddd	B^2ddd

B^1ddd contains the base address of the Op code Table.
B^2ddd contains the base address of the first word in the common region.
B^1 and B^2 are GPR 3, the common region base register.

A. *Branch DIL* takes the address in the IAR (GPR 6) and stores it in the BAR (GPR 4) in case the branched-to instruction is a Store B-Address Register; then it takes the branched-to address in the AAR (GPR 5) and puts it in the 1400/1410 IAR (GPR 6). BDIL then branches to the DIL microprogram entry.

B. *Normal DIL*

1. DIL count (Byte 0-GPR 6): When decremented to zero, the DIL interrupt address (word 5 of the common region) is put into the current PSW. This causes routing to a software subroutine for handling time dependencies. If the initial DIL count is zero, it is assumed that no interruptions are contemplated and that DIL count is not being used.
2. Op-code verification: Checks the 1400/1410 operation code for a wordmark bit (bit 6 = 1). If the wordmark is missing, the 1400 unusual-instruction-format address or the 1410 invalid-format address (word 11 of the common region) is used to branch to an invalid-format routine.
3. Routing: The Op-code table is arranged so that the proper index into the table is obtained through the following procedure: (1) subtract 2 from the fetched 1400/1410 Op-code (reset the wordmark bit); (2) add the result to the B^1ddd operand of the DIL instruction. The word at this calculated address contains the address of a software simulation routine for the 1400/1410 Op code, in addition to a control byte for the 1400/1410 Op code.

   This provides the transition between the 1400/1410 program and the software routine.

   The format of the word fetched from the Op-code table is:

CTL	XXX

CTL: Control Byte

The Control Byte bit significance is shown in the Control Byte Chart.

XXX: 24-bit address of the subroutine that is to simulate a given operation code. (*Note:* V, W, Q, and H Op-codes do not contain a subroutine address.)

CONTROL BYTE CHART (1400 and 1410)

1400 Control Byte Chart

Bits	0, 1	0 0	Not M, L, Q, U, operation codes
		0 1	Q operation code
		1 0	M or L operation code
		1 1	U operation code
Bit	2	1	Address double operation code
Bit	3	1	B operation code
Bit	4	1	Clear storage or set wordmark operation code
Bit	5	0	Exit immediate to software routine
Bit	6	1	High-speed execution of current operation code
Bit	7	0	1410 (error)
		1	1400

1410 Control Byte Chart

Bits	0, 1	0 0	Normal CPU op
		0 1	G or $ Op code
		1 0	I/O operation without X-control field
		1 1	M, L, U operation
Bit	2	1	Address double operation code
Bit	3	1	Op code can have a B-address
Bit	4	1	Op code can have a B-address
Bit	5	0	Exit immediate to software routine
Bit	6	1	High-speed execution of current operation code
Bit	7	0	1410
		1	1400 (error)

4. Conversion of Addresses: Conversion of address characters from internal code to an equivalent binary value is accomplished with address-decoding tables. There is a separate table for each 1400/1410 address position (hundreds, tens, and units for 1400 address) (ten thousands, thousands, hundreds, tens, and units for 1410). Each table consists of 64 words (256 bytes) starting on a 256-byte boundary, that is, the binary address of the first byte in the table has zeros in the last eight bits. Therefore, each byte in the table has an address in which the last eight bits correspond to a possible internal-code configuration if all bits of the internal code were used (eight bits equal 256 combinations). Because bit 6 of the internal code is always zero for 1400/1410 address fields (no wordmarks) and bit 7 is always zero, any combination of internal-code bits is located in the table on a word boundary (last two bits of the address are zero) and offers 64 combinations (six bits equal 64 combinations). Each word in the table has the following format.

   Byte 0: Value byte — this byte contains information to determine:

   a. Which index word to select (tens table only for 1400; hundreds and tens tables for 1410).
   b. When an invalid character is used to index the table (all tables).
   c. When the indexing character has a zone combination that will cause an invalid address (hundreds and units tables for 1400; ten thousands, thousands, and units tables for 1410).

   Bytes 1-3: These bytes contain the binary value of the 1400/1410 address character according to its function (ten thousands, thousands, hundreds, tens, or units). The relocation factor is included in bytes 1-3 of the ten thousands table for 1410.

   To select the correct corresponding binary value, the microprogram inserts the internal-coded 1400/1410 address character as the last eight bits of the base address of the proper address-decoding table. The high-order address-decoding table (ten thousands for 1410, hundreds for 1400) is located at B^1ddd plus 100 hex (256 bytes from op-code table). Each successive table is located at multiples of 100 hex. These tables are generated by the Integrated Emulator Program according to the system being emulated. When 1400 is specified, only the hundreds, tens, and units tables are used. When 1410 is specified, all five tables are used.

   The final 1400/1410 address is the summation of the binary values extracted from the address-decoding tables.

5. 1400 I/O Detection: DIL performs X-control field decoding on the A-field of the 1400 instruction as defined in conjunction with the CTL byte from the Op-code table.
   a. For an M or L operation code with an X-control field, hex 2C is added to the B^2 operand of the DIL instruction and the word at that location fetched. The address in that word is placed in the current PSW.
   b. For a U operation code with an X-control field, hex 20 is added to the B^2 operand of the DIL instruction and the word at that location fetched. The address in that word is placed in the current PSW.
   c. For an M, L, or U operation code without an X-control field, the address fetched from the Op-code table for M, L, or U entries is placed in the current PSW.

5A. A valid 1400/1410 I/O instruction with X-control field places the address of the I/O instruction into bytes 1, 2, and 3 of the first word of the common region (B^2ddd + 0 of the DIL instruction).

6. D-Modifier Character: If the 1400/1410 instruction has a d-modifier, place the modifier in the common region. Store it in the B^2ddd field + 0 of the DIL instruction.

7. Exit:
   a. Normal Exit: set up the DMOD and the CURRENT IAR or INSTR LENGTH/SWITCHES in the common region (first word) as per the definition.
   b. No Exit: high-speed implementation for the following 1400/1410 instructions.

      - Branch Character Equal
      - Branch Bit Equal
      - Branch if Wordmark or Zone

- Branch Unconditional (1400 only)
- Store A-Register (1400 only)
- Store B-Register (1400 only)

DIL in these cases performs the appropriate test for a microcode branch to Branch DIL microcode if the result of the test is true. Otherwise, it branches to the DIL microcode entry point in microcode. Store A- or Store B-Register operations branch to the DIL microcode entry.

8. Interruptibility: DIL is interruptible at the start of DIL. DIL is also interruptible at the end of a high-speed execution. Interruptibility is periodically checked during microcode execution.

Condition Code: The condition code remains unchanged.

ANUM — Add Numeric

This instruction does the 1400/1410 decimal-add or subtract operations. It uses a control byte that assists the microprogram in the implementation of its several running states. This instruction is interruptible.

ANUM

EA 04	B^1ddd	B^2ddd

The low eight bits of the B^1ddd effective address are interpreted to be R^1R^2.
R^1 contains the units address of the destination field.
R^2 contains the units address of the source field.
B^2ddd is the address of the control byte, which is defined as follows:

X0 = 1410 Add
X1 = 1400 Add
X2 = 1410 Subtract
X3 = 1400 Subtract
X4 = 1410 Zero and Add
X5 = 1400 Zero and Add
X6 = 1410 Zero and Subtract
X7 = 1400 Zero and Subtract
X9 = 1400 Modify Address
XBDF (bit 7 on) Default to X9

ANUM, upon completion, set/resets the 1410 zero balance status indicator and sets the 1400/1410 overflow indicator, if appropriate.

ANUM executes from arithmetic low order to high order, (high memory address to low address). Certain cases of subtract may require a recomplement cycle. This must start at the initial R^1 address.

Interruptibility is maintained for all running states. Retry is a function of the microprogram technique. Upon completion, the R^1, R^2 GRs will be decremented by the number of bytes processed.

All arithmetic functions are performed under complete algebraic sign control. The sign of a factor is determined by the combination of zone bits in the units position of the fields specified by the instruction being executed.

Condition Code: The condition code remains unchanged.

COMP — Compare

This instruction performs the 1400/1410 compare function.

COMP

EA 05	B^1ddd	B^2ddd

The low eight bits of the B^1ddd effective address are interpreted to be R^1R^2.
$R^1$and R^2 contain the units addresses of the fields to be compared.

Execution:

1. Execution is from high address to low address (low-order data position to high-order data position).
2. It stops on the first wordmark encountered in either field (bit 6 = 1 of the current byte).
3. It sets the bits of the CPU indicator (word 5 of the common region) according to the compare results.
4. Interruptible as on long move.

Registers at end:

R^1 and R^2 are decremented by the field length.

Condition Code: The condition code setting at the end of the compare is one of the following.

01 = R^2 wordmark stopped the compare.
10 = R^1 wordmark stopped the compare.
11 = R^1 and R^2 wordmarks occurred simultaneously and stopped the compare.

MCPU — Move Data in CPU

This instruction does the 1400/1410 CPU data moves. The value of the control byte pointed to by the MCPU instruction causes proper execution of the current move operation.

MCPU

EA 08	B^1ddd	B^2ddd

The low eight bits of the B^1ddd effective address are interpreted to be R^1R^2.
R^1 contains the address of the destination field.
R^2 contains the address of the source field.
B^2ddd is the address of the control byte that controls the move.

Execution

1. The bits of the MCPU control byte are weighted the same as the internal code (8 4 2 1 B A 0 0). The status of the control-byte bits controls the operation as follows.
   A. Move direction:
      - 8 off — Move is right to left
      - 8 on — Move is left to right

   B. Data Transfer:
      - 4, 2, 1 off — No data transferred; scan for wordmarks, recordmarks, or groupmark-wordmarks.
      - 1 on — Transfer the numeric portion of the data field.
      - 2 on — Transfer the zone portion of the data field.
      - 4 on — Transfer the wordmarks from the A-field to the B-field.

   C. Stop conditions on a right-to-left move (8-bit off):
      - B and A off — Transfer or scan only one storage position.
      - B off, A on — Transfer or scan to A-field wordmark.
      - B on, A off — Transfer or scan to B-field wordmark.
      - B and A on — Stop transfer or scan at either A-field or B-field wordmark.

   D. Stop conditions on a left-to-right move (8-bit on):
      - B and A off — Stop transfer or scan at first wordmark sensed in either field.
      - B off, A on — Stop transfer or scan at A-field recordmark.
      - B on, A off — Transfer or scan to A-field groupmark-wordmark.
      - B and A on — Stop transfer or scan on either an A-field record mark or A-field groupmark-wordmark.

   E. Bits 6 and 7 should be 01 for 1400, and 00 for 1410, for the Model 145 system.
2. Minimum Model 145 implementation includes the MCPU commands for the MCPU control-byte settings shown as follows.

Control Byte

Bit 0 = 0 Must be zero.
Bit 1 = 0 Must be zero.
Bit 2: 0 for move; 1 for load.
Bit 3: 0 for tape or unit record; 1 for disk.
Bit 4: 0 for no translation; 1 for translation, or 0 for odd parity; 1 for even parity.

as defined in the following chart.

Bit 2 3 4	Significance
0 0 0	A nine-track, odd-parity EBCDIC tape-move operation, or a unit-record move operation. (The odd-parity word-separator character is 2D in the buffer.) See bit 4 parity input/output significance below.
0 0 1	A nine-track, even-parity EBCDIC tape-move operation, a seven-track tape-move operation, or a unit-record move operation. (The even-parity word-separator character is 6D in the buffer.) See bit 4 parity input/output significance below.
0 1 x	Specification check.
1 0 0	A nine-track, odd-parity EBCDIC load-mode tape operation, or a unit-record load-mode operation. See bit 4 parity input/output significance below.
1 0 1	A nine-track, even-parity EBCDIC, load-mode tape operation, or a seven-track load-mode tape operation, or a load-mode unit-record operation. See bit 4 parity input/output significance below.

Bit 4 Parity Input/Output Significance (for the above conditions only).

When bit 4 = 0 (odd parity):

On Input: If buffer character bit 1 = 1, substitute an asterisk (in internal code) and set invalid-data condition code.

If buffer character bit 1 = 0, data is translated directly to internal code.

Bits

0 (8	1 4	2 2	3 1	4 B	5 A	6 0	7 0)	Meanings
0	0	0	1	0	0	0	1	Move Numeric
0	0	1	0	0	0	0	1	Move Zone
0	0	1	1	1	1	0	1	Move
1	0	1	1	1	1	0	1	Move Record
0	1	1	1	0	1	0	1	Load
0	0	0	0	1	0	0	1	Scan Left to B-field wormark
1	0	0	0	1	0	0	1	Scan Right to GMWM

Note: Control-byte bit 7 is always on for 1400 operation.

3. MCPU is interruptible.

Registers at end: R^1 and R^2 are incremented/decremented by this field length.

Condition Code: The condition code is unchanged.

MIO — Move Data for Input/Output

This instruction assists in the simulation of 1400/1410 I/O instructions. Data is moved from the 1400/1410 main-storage image to the Model 145 buffer area, or vice versa.

MIO

EA 09	B^1ddd	B^2ddd

Registers

The low eight bits of the B^1ddd effective address are interpreted to be R^1R^2.

R^1 is the register containing the Model 145 buffer address.

R^2 is the register containing the 1400/1410 data-field address.

Rc is the count register that monitors the length of execution, and is located by subtracting 1 from the register number specified by R^1. It contains the absolute binary number using one origin counting, which represents the number of bytes to be transferred. No more than two bytes for 1400 or three bytes for 1410 of the GPR represented by Rc should contain a count. The count should never specify a buffer overrun into a protected area, nor an invalid System/370 address.

B^2ddd points to a control word structured such that byte 0 is a control byte and bytes 1, 2, and 3 are the addresses of a translation table that is located on a 256-byte boundary.

Note: Output with translation may have the translate table located on a 256 + 1 boundary.

On Output: The translated character has bit 1 forced to 0 before it is moved into the buffer.

When bit 4 = 1 (even parity):

On Input: If buffer character bit 1 = 0, substitute an asterisk (in internal code) and set invalid-data condition code.
If buffer character bit 1 = 1, data is translated directly to internal code.

On Output: No change to characters — data bit 1 should be 1 after translation.

1 1 0 — A disk load-mode operation. No translation is required.

1 1 1 — A disk load-mode operation, but

On Input: Translate from EBCDIC to internal code.

On Output: Translate from internal code to EBCDIC.

Bit 5 = 0 — Output to the Model 145 buffer area from 1400/1410 main storage.

Bit 5 = 1 — Input to 1400/1410 main storage from the Model 145 buffer area.

Bit 6 = 0 — Do not stop on groupmark-wordmark. Always stop at end of 1400/1410 storage or if count is zero.

Bit 6 = 1 — Stop on groupmark-wordmark. Always stop at end of 1400/1410 storage or if count is zero.

Bit 7 — Must be on for 1400, must be off for 1410.

Validity Checking Data

Invalid data must be sent to the 1400/1410 CPU as an asterisk (in internal code) by hardware. It will be handled at the logic level.

Word Separators

During load-mode tape handling, because movement is under the byte-count control of Rc, it is clearly possible to be working on a wordmarked character in 1400/1410 storage or a word separator in the buffer, and have the count go to zero after transmitting a word-separator character to the buffer. In this case, the microcode sets the condition code as described in the *Condition Code* section, and the problem of handling this situation is delegated to software.

Rc After a MIO

At the start of an MIO, Rc is used as a counting register; in 1400 operations, the register has bytes 0 and 1 set to zero and bytes 2 and 3 are used for a count. In 1410 operations, the register has byte 0 set to zero and bytes 1, 2, and 3 are used for a count. This is a *requirement* of the Op-code function. At the conclusion of the operation, byte 0 of Rc has the following bit settings.

Byte 0 = 0 0 0 0 0 0 0 0 — Count went to zero; no stop condition satisfied.

Byte 0 = 0 0 1 0 0 0 0 0 — Groupmark-wordmark encountered and caused the stop.

Byte 0 = 0 0 0 1 0 0 0 0 — End of 1400/1410 storage was encountered and caused the stop.

The above stop conditions may be tested with the LTR instruction.

Bytes 2 and 3 contain the residual count (1400). Bytes 1, 2, and 3 contain the residual count (1410).

Note: Invalid lengths are not checked in the microcode.

Condition Codes

CC = 00 — Normal ending of MIO.

CC = 01 — Word-separator or wordmark problem encountered on last character of the count.

CC = 10 — Invalid data encountered (cannot occur on output).

CC = 11 — Word-separator or wordmark problem and invalid data detected during the MIO operation.

BIFLAG — Branch on Invalid Flag

This instruction tests R^1 and R^2 to see whether an addressing flag (invalid 1400/1410 address) exists in either register specified by R^1 or R^2. The instruction form is:

BIFLAG

EA 0C	B^1ddd	B^2ddd

The low eight bits of the B^1ddd effective address are interpreted to be R^1R^2.

R^1 any general register. Normally it is the 1400/1410 BAR.

R^2 any general register. Normally it is the 1400/1410 AAR.

B^2ddd is the base displacement address of the 1400/1410 common region.

Execution

R^1 and R^2 are checked to see whether any bits are on in byte 0 of the registers. If any are on, a branch to the address-error routine is taken through the 1400/1410 common-region word. If no bits are on in byte 0 of either R^1 or R^2, processing continues at the next sequential System/370 address.

Condition Code

Unchanged.

PROGRAM DEBUGGING INFORMATION

1. System/370 Interruptions
   a. *System/370 Op-Code Check* caused if the EA Op code is not installed on the system, or if an invalid secondary decode byte is encountered.
   b. *System/370 Addressing or Protection Check* caused by B^1ddd having a value other than zero for B^1, or by having an invalid B^2ddd address for the system. This check is also caused by a violation of the protection key, or if R^1 and/or R^2 contains an invalid address with respect to the system or its protection key.
   c. *System/370 Specification Check* caused when invalid control bytes are encountered.
2. Miscellaneous Interruptions
   a. Register 2 must be zero upon issuance of this operation, or unpredictable results can occur.
   b. Register 3 must point to the hardware common region, or unpredictable results can occur.
   c. *1400/1410 Address Error or Storage Wrap* can be caused by a flag set on in byte 0 of either R^1 or R^2. This indicates an error in either the 1400/1410 program or the address-conversion table. Storage wrap is caused by encountering an X'03' at either the high or low end of the target storage area. R^1 and R^2 are unchanged. GP register 2 may be non-zero.
   d. *Operation Exception* if EA is encountered in RI Mode.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

DIRECT CONTROL

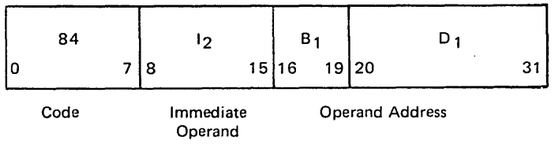
OPERATION

The Direct Control feature in System/370 provides a means of communicating between two CPUs, or between a CPU and external devices. It is intended primarily for transmission of control information. A CPU communicates with external devices by using the external-interruption mechanism and the write direct and read direct instructions. A byte of information and control signals are exchanged over the direct control interface lines.

Write Direct

The write direct instruction is used to place information on the direct control bus-out (dir-out) lines, and read direct is used to take information from the direct control bus-in (dir-in) lines. The write direct instruction causes the byte of information (8 bits) at the location designated by its operand address to be placed as static signals on the dir-out lines.

Write Direct



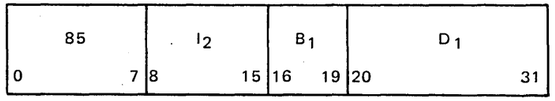
These signals may be changed at varying intervals by repeating the write direct instruction or they may be allowed to remain an indefinite period. No parity is presented with these eight bits of control information. The write direct instruciton also causes the eight bits, contained in bit positions 8-15 of the instruction, to be sent out as eight timing pulses on the timing signal bus-out (sig-out) lines. Again no parity is presented. At the same time, a ninth and similar pulse is sent out on the write-out line. The leading edge of these timing pulses coincides (within skew limitations) with the leading edge of the write-out pulse, and the write-out pulse overlaps the change of the signal on the dir-out lines.

The timing signals and the write-out signal may be used to alert equipment to which data is to be sent. When communicating with another CPU, the timing pulses are used to cause an external signal interruption at the receiving CPU; the write-out pulse is used to ensure the validity of the control information.

Read Direct

The read direct instruction causes the information appearing on the eight dir-in lines to be placed as eight bits in the location in storage designated by the operand address (provided the hold-in signal is absent).

Read Direct



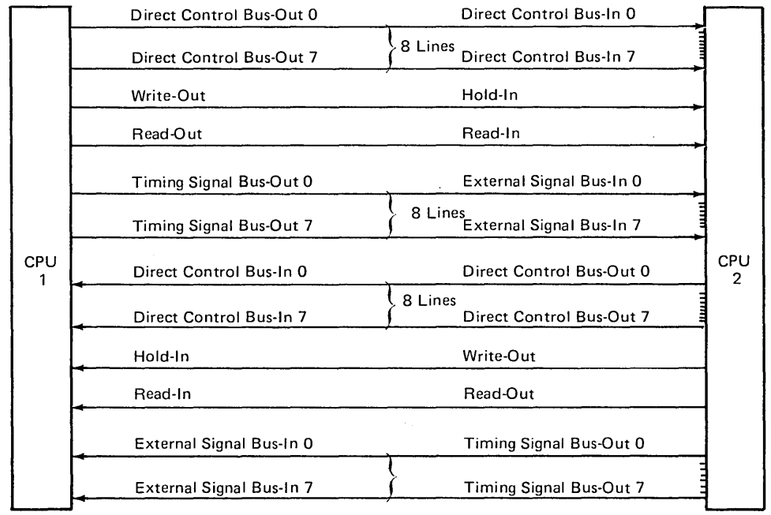
Information on the dir-in lines may not be valid while the hold-in signal is active. No parity is transmitted with the control information, however, a parity bit is generated by the CPU as the data is placed in storage.

Prior to accepting the control information, the read direct instruction causes the eight bits in positions 8-15 of the instruction to be sent out as eight timing pulses on the timing signal bus-out (sig-out) lines. At the same time, a ninth and similar pulse is sent out on the read-out line. The leading edges of the timing pulses and the read-out pulse must coincide within the skew limitations.

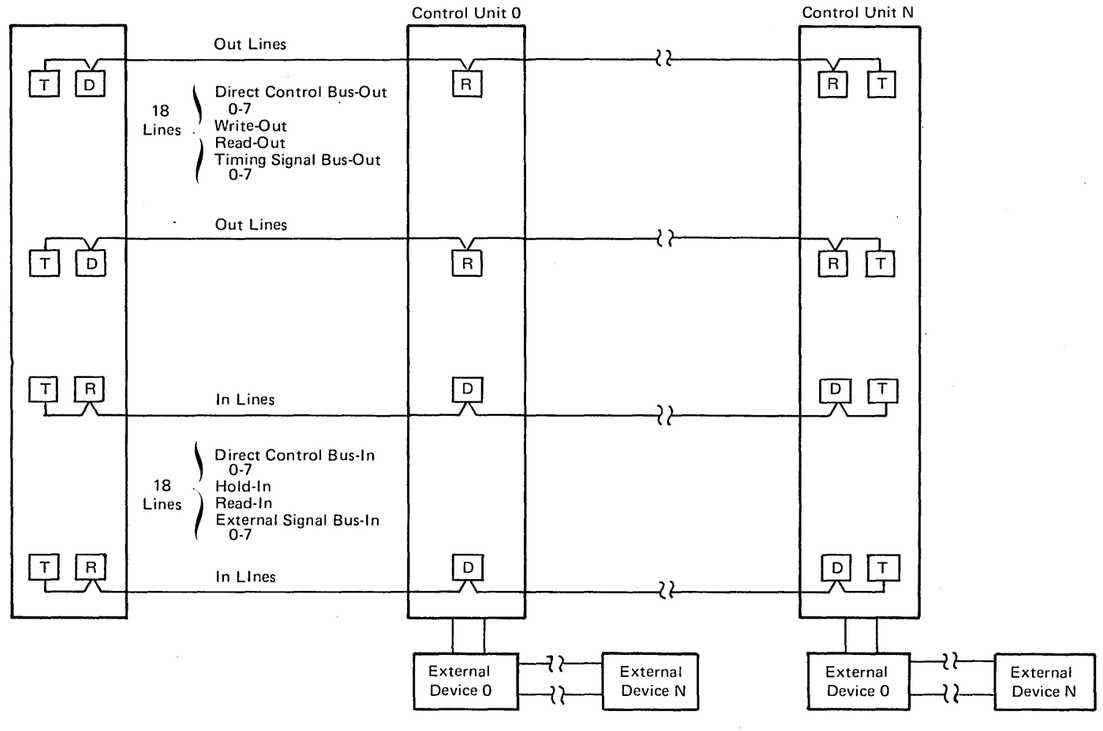
The function of the hold-in signal is to allow the external device to inhibit (hold up) the read operation until current data has been placed on the dir-in bus. Hold-in also prevents a read operation while information on the dir-in lines is changing and therefore invalid. When communicating between CPUs, the write-out pulse of the sending CPU is received as the hold-in signal at the receiving CPU, and thereby prevents the reading of invalid information by the receiving CPU.

Devices connected to the CPU should be designed to respond quickly to the CPUs read-out signal by dropping (deactivating) the hold-in line. Note that hold-in overlaps the period when information is changing on the dir-in lines. Refer to the timing chart (signals originating outside the CPU), page 9-20. Therefore, time is allowed to complete a data-sending operation should the external device have one in progress.

Direct Control Interface, CPU to CPU



Direct Control Interface, CPU to External Device



DEFINITIONS OF INTERFACE LINES

In the following signal description, the up-level is the active level; the down-level is the inactive level.

Direct Control Bus-Out

The direct control bus-out is a set of eight lines from a CPU to the external equipment. The external equipment could be another CPU; in which case, direct control bus-out is connected to direct control bus-in of the other CPU.

Data on the direct control bus-out is placed only during the execution of the write direct instruction. The data on the lines represents the byte at the location designated by the operand address of the last write direct instruction. The data placed on the direct control bus-out remains valid until intentionally changed, as for example, at the execution of the next write direct. The write-out pulse overlaps a change on the direct control bus-out by 100 nanoseconds, that is, data already on the dir-out lines is valid for at least 100 nanoseconds after the rise of the write-out pulse to its up-level and new data is valid at least 100 nanoseconds before the fall of the write-out pulse below its up-level.

Write-Out

Write-out is a line from the CPU to external equipment. The external equipment could be another CPU; in which case, the write-out line is connected to the hold-in line of the other CPU.

The function of the write-out line is to signal the external equipment when the CPU is placing data on the dir-out lines, and to indicate that the data is, therefore, presently invalid. The down-level of write-out indicates that the data on the dir-out lines is valid. The up-level of the write-out pulse overlaps the transition of any signal on direct control bus-out by a minimum of 100 nanoseconds, and the leading edge of the write-out signal must coincide with the leading edge of the pulses on the timing signal bus-out (within skew limitations).

Read-Out

Read-out is a line that connects the CPU to the external equipment. The external equipment could be another CPU; in which case, the read-out line is terminated, but serves no function.

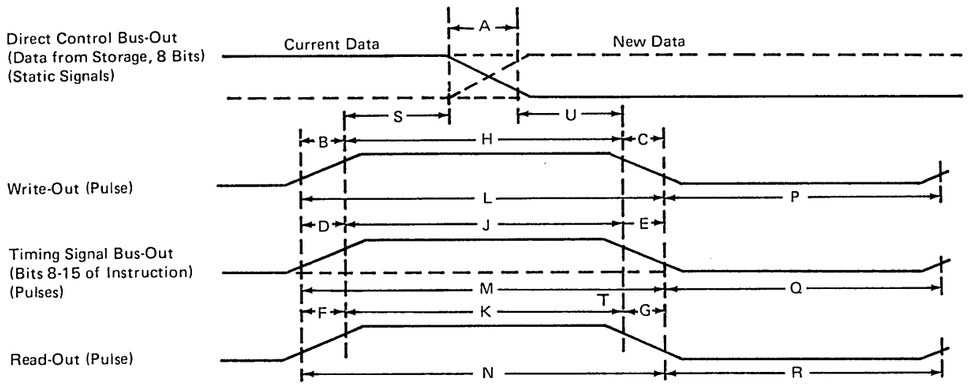
The purpose of the read-out line is to provide a means of signaling the external equipment that a read direct is being executed and that the external equipment must provide valid data on the direct control bus-in, as indicated by the down-level of the hold-in signal.

Within skew limitations, the leading edge of the read-out signal must coincide with the leading edge of the pulses on the timing signal bus-out.

Timing Signal Bus-Out

Timing signal bus-out is a set of eight lines from the CPU to the external equipment. The external equipment could be another CPU; in which case, the timing signal bus-out is connected to the external signal bus-in of the other CPU. The sig-out-0 and sig-out-1 lines are terminated, but serve no purpose. That is, the sig-out-0 and sig-out-1 lines of a CPU are usable with external devices but are terminated as sig-in-0 and sig-in-1 in a receiving CPU.

Signals Originating Within The CPU



A, B, C, D, E, F, G	Maximum transition time is 200 ns.
H, J, K	Minimum duration is 500 ns.
L, M, N	Maximum, including transition, 1000 ns.
B, D	Leading edges coincidental within skew tolerances.
F, D	Leading edges coincidental within skew tolerances.
S	Overlap of write-out to start of change A, 100 ns (min)
U	Overlap of write-out to end of change A, 100 ns (min)
T	Earliest time to sample hold line during read direct
P, Q, R	Minimum down time between pulses is 500 ns.

During a read direct or a write direct, the eight bits contained in the instruction (positions 8-15) are sent out as eight timing pulses on the eight sig-out lines. The leading edge of the timing pulses must coincide, within skew limitations, with the leading edge of either the write-out or the read-out signal.

When the timing signal bus-out is connected to external signal bus-in of another CPU, the timing pulses on positions 2-7 cause an external signal interruption at the receiving CPU. (See "External Signal Bus-In".)

Direct Control Bus-In

The direct control bus-in is a set of eight lines from the external equipment to the CPU. The external equipment could be another CPU, in which case, the direct control bus-in connects to the direct control bus-out of the other CPU.

The data appearing on the direct control bus-in are read by the CPU only during the execution of read direct. The data is stored in the location designated by the operand address of the read direct instruction. The CPU reads the direct control bus-in only when the direct control bus-in information is valid and after the read-out pulse occurs. The data already on the bus is valid for at least 100 nanoseconds after hold-in rises to its up-level; new data is valid at least 100 nanoseconds before hold-in falls below its up-level. When executing read direct, sampling of the hold line to determine validity of the data shall not start until completion of the read-out pulse. Sampling of the direct control bus-in shall be completed within 100 nanoseconds of the time that both the read-out and hold-in lines are at a down-level.

Hold-In

Hold-in is a line from the external equipment to the CPU. The external equipment could be another CPU, in which case, the hold-in line is connected to the write-out line of the other CPU. The purpose of the hold-in signal is to prevent the CPU from reading the data from the direct control bus-in until such data is valid, or until the external device has replaced the information on direct control bus-in with current data.

The hold-in signal shall be in the hold position (up-level) for at least 100 nanoseconds on either side of any signal transition on direct control bus-in; i.e., hold-in must be at an up-level for at least 100 nanoseconds before data is invalid, and must remain up at least 100 nanoseconds after new data is valid on the direct control bus-in.

The hold-in signal must have a minimum up-level duration of 500 nanoseconds and, when at the down level, must remain for a minimum duration of 500 nanoseconds. Refer to the timing chart (signals originating outside the CPU), page 9-20.

After the read-out pulse is generated during the execution of read direct, the CPU senses for a down-level of the hold-in line so that reading of the direct control bus-in can be made, completing the instruction. Because the CPU will hang-up waiting for hold-in to drop, devices connected to the CPU should be designed to respond quickly to the CPUs read-out signal by dropping (deactivating) the hold-in line. If the delay between the termination of read-out and the termination of hold-in is relatively long, serious interference with the computer program can occur.

The hold-in signal can occur at any time; it does not have to be synchronized with the read-out pulse. (It may occur before, during, after, or even as a result of either transition of the read-out pulse.)

Read-In

Read-in provides no function except as a termination for the read-out line in the CPU to CPU configuration.

External Signal Bus-In

Eight lines, sig-in-0 through sig-in-7, make up the external signal bus-in. Six of these lines provide access to the computer's external interruption mechanism. Two lines, sig-in-0 and sig-in-1, are terminated in the CPU but serve no other function. The external equipment could be another CPU; in which case, the external signal bus-in connects to the timing signal bus-out of the other CPU.

The purpose of the external signals bus is to provide a path to the external-interruption mechanism of the CPU. The external interruption can occur only after the current instruction is completed and when system mask bit 7 is a one. The interruption causes the external old PSW to be stored at location 24 and an external new PSW to be fetched from location 88. As a result of an external interruption, the external signals are placed in bit locations 26-31 of the external old PSW.

The external signal requests (pulses) may occur at any time and have no relation to the timing of other signals on the direct control interface.

The requests are preserved until honored by the CPU. All pending requests are presented simultaneously when an external interruption occurs. Each request is presented only once. When several requests from one source are made before the interruption is taken, only one interruption occurs. *(Engineering Note:* If a constant up-level, for any reason, appears on the external signal bus-in, it must no result in CPU hang-up.)

Because of possible skew between pulses, the CPU cannot guarantee that simultaneous pulses (requests) will be recognized as such. Skew may cause simultaneous requests to appear as separate requests and result in more than one interrupt.

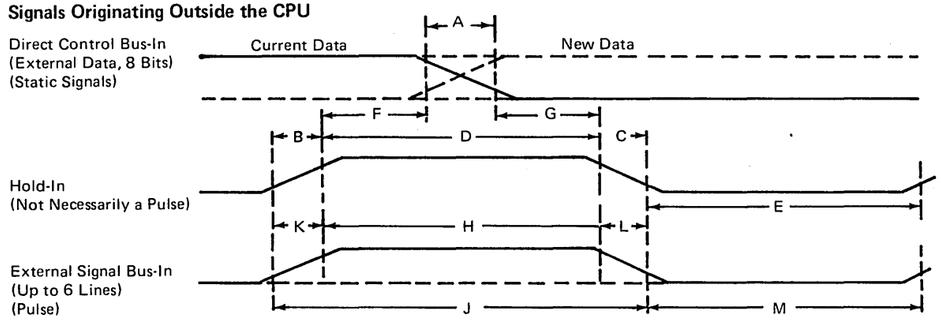
Signal Duration

NAME	DURATION (IN NS) MIN. SIGNAL*	MAX. SIGNAL**	MIN. DOWN LEVEL
Write-Out	500	1,000	500
Read-Out	500	1,000	500
Timing Signal (Sig-Out)	500	1,000	500
External Signal (Sig-In)	500	1,000	500
Hold-In	500	None	500

*Measured at up-level.
**Including transition time: from and to down-level.

Note: Transition of any signal originating in the CPU may not exceed 200 nanoseconds. Refer to timing chart (Signal Originating Within the CPU), page 9-19.

Signals Originating Outside the CPU



A, B, C, K, L	No minimum transition duration specified
D, E, M	Minimum is 500 ns, no maximum specified
F	Overlap of hold-in to start of change A, 100 ns (min) No maximum specified
G	Overlap of hold-in to end of change A, 100 ns (min) No maximum specified
H	Minimum duration 500 ns
J	Maximum, including transition, 1000 ns

DIRECT CONTROL EXTERNAL WORD

The direct control external word (DC at address 17) provides the communications line between the microprogram and the direct control feature.

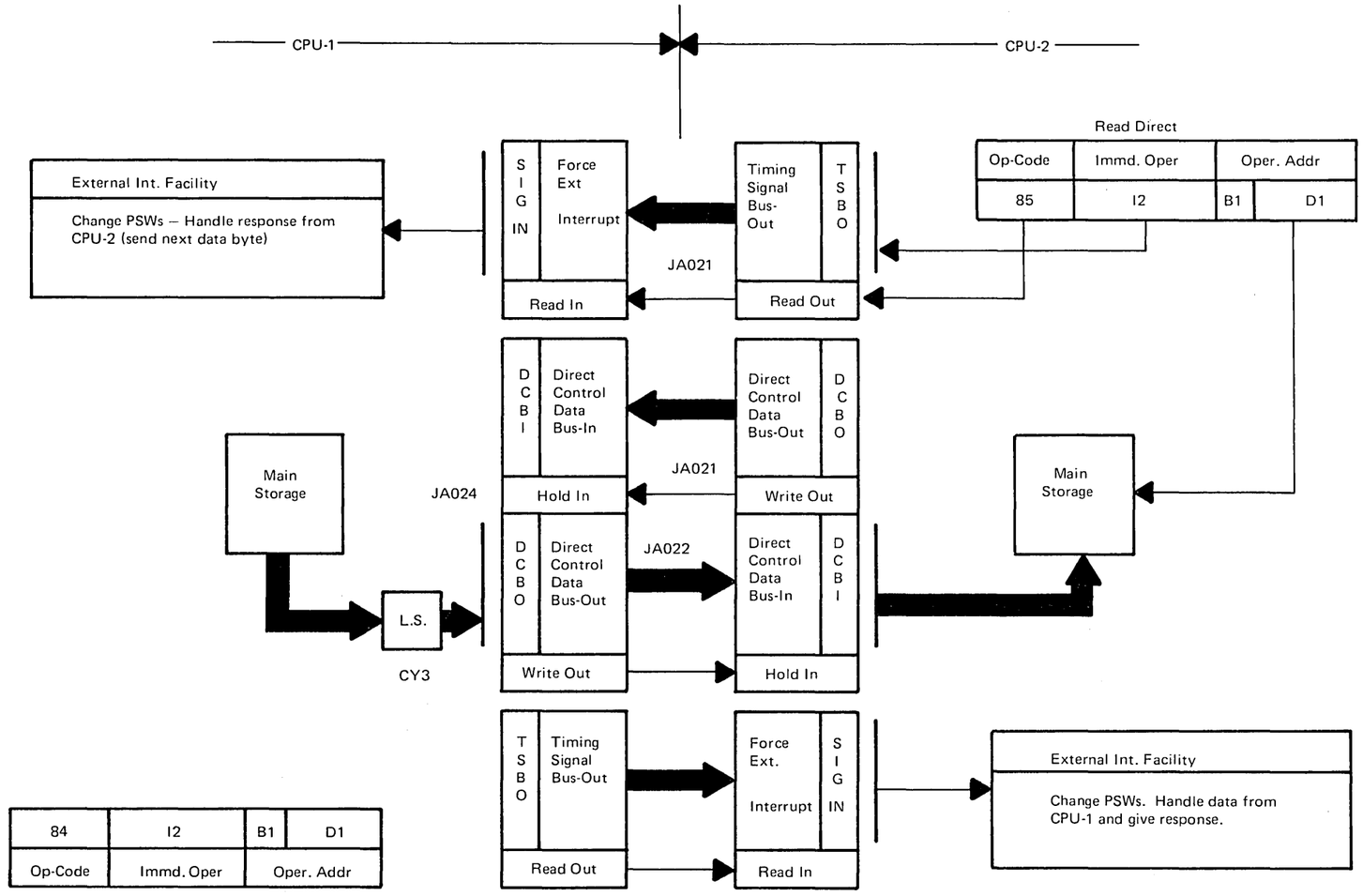
This word contains the direct control bus-out register (DCBO), the direct control bus-in register (DCBI), the direct control timing-signal bus-out register (TSBO), and the direct control hold-in register (DCHI).

Bits	Name
Byte 0 DCBO	
0-7	Direct-control bus-out
Byte 1 DCBI	
0-7	Direct-control bus-in (may not be used as a destination)
Byte 2 TSBO	
0-7	Time-signal bus-out
Byte 3 DCHI	
0	Direct-control hold-in not used (may not be used as a destination)
1-7	

Example: CPU-1 sends one byte to CPU-2. CPU-1 was directed by the write direct instruction to move a byte of data from main storage to the DCBO and signal CPU-2 by an external interrupt that the data was available.

CPU-2 was interrupted by an external interrupt, and by analyzing the cause of the interrupt determined that it must do a read direct. The read direct instruction took the data from CPU-2s DCBI and placed it in main storage of CPU-2.

DATA FLOW AND CONTROLS FOR DIRECT CONTROL



Refer to Diagram 1-38 in the 3145 Maintenance Diagrams Manual Form SY24-3580

TIMINGS FOR DIRECT CONTROL

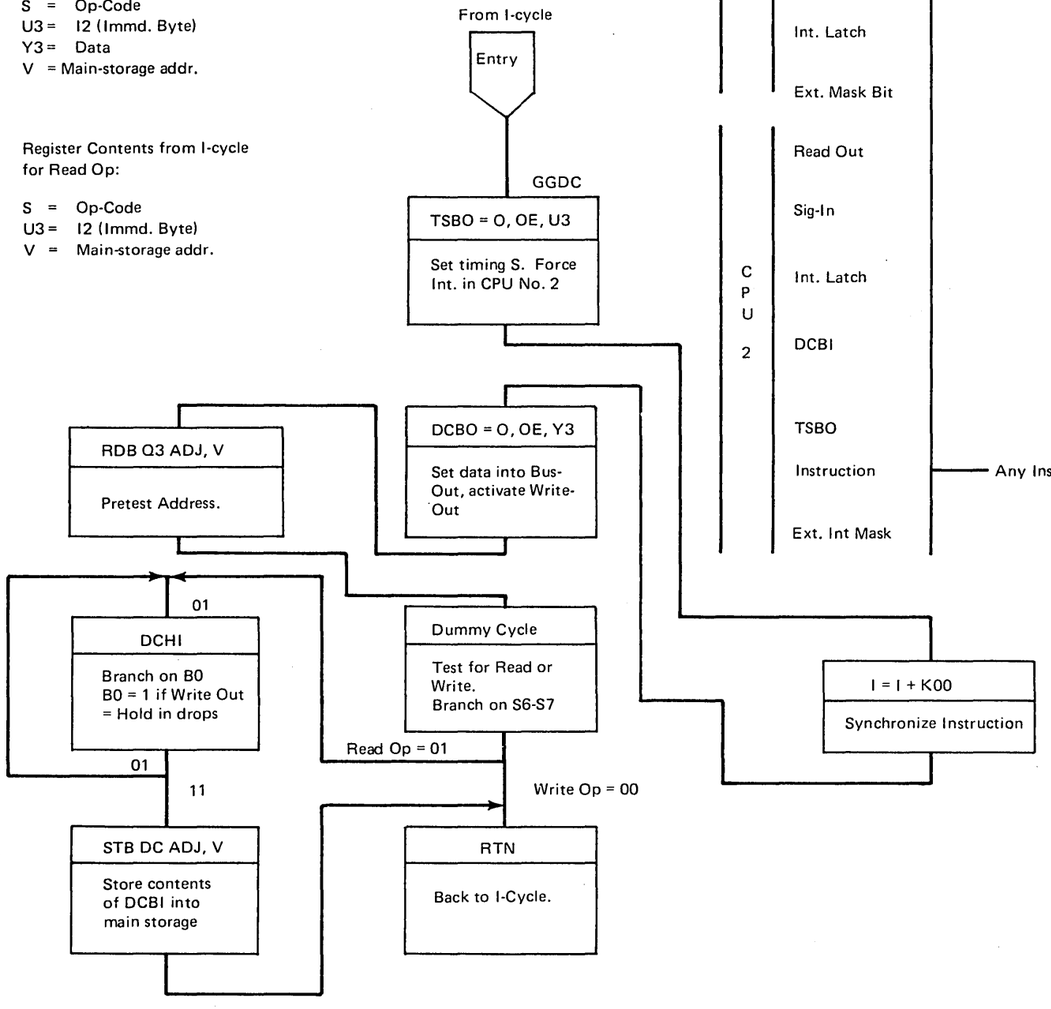
Microcode Flow Chart for Write and Read Direct

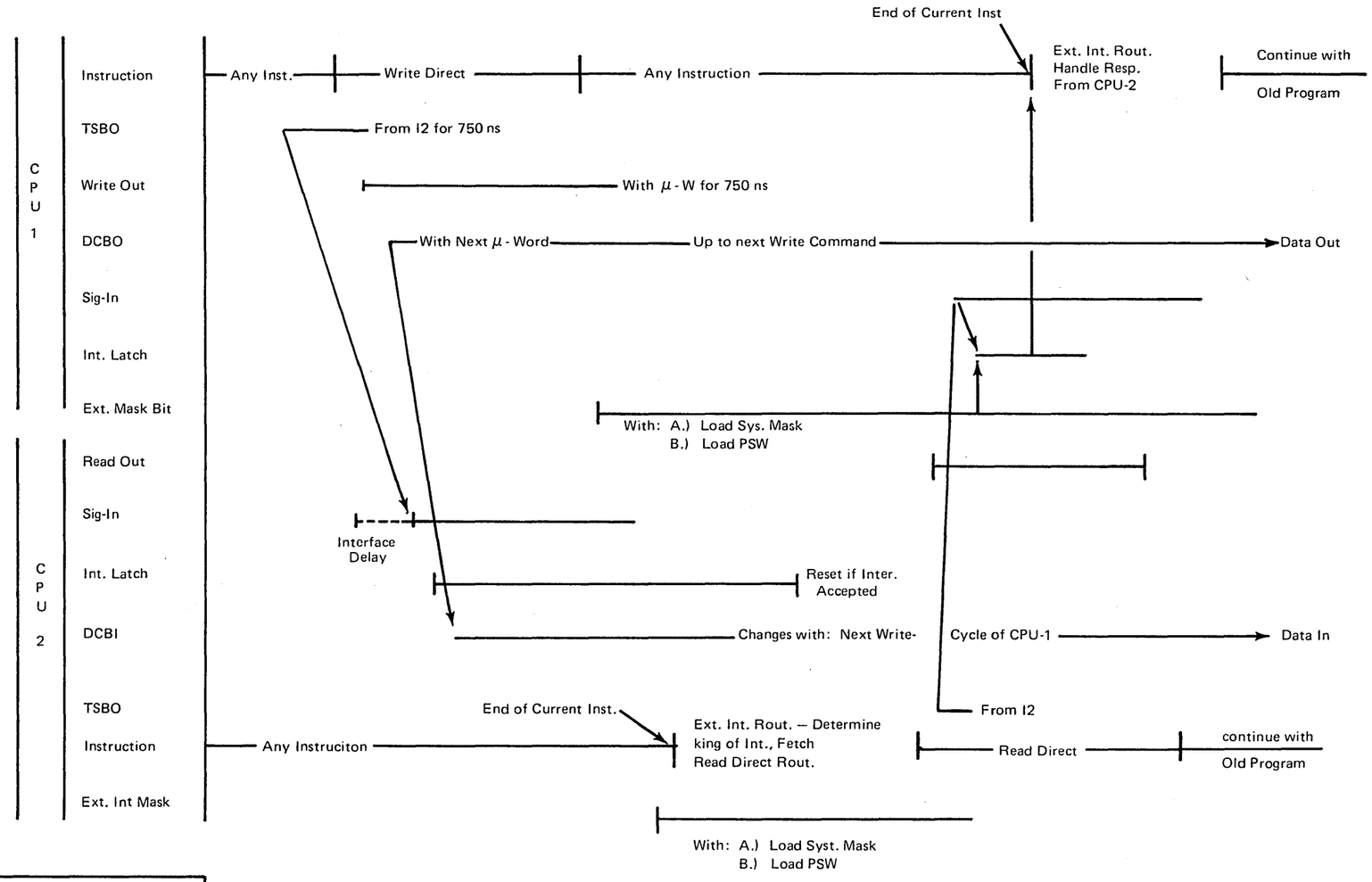
Register Contents from I-cycle for Write Op:

S = Op-Code
U3 = I2 (Immd. Byte)
Y3 = Data
V = Main-storage addr.

Register Contents from I-cycle for Read Op:

S = Op-Code
U3 = I2 (Immd. Byte)
V = Main-storage addr.





REMEMBER

There is a Reader's Comment Form at the back of this publication.

CHANNEL–TO–CHANNEL ADAPTER

INTRODUCTION

The channel-to-channel adapter (CTCA) transfers data between a System/370 Model 145 channel and another channel in either System/370 or System/360. The connected channels are referred to as channel X and channel Y. The CTCA resides in and is powered by the channel X system.

The primary application of the CTCA is to transfer data between main storage of two processing systems. The adapter may also be used to connect two channels of the same system for relocation of blocks of main-storage data.

The adapter has an X-side and a Y-side that connect as control units to channel X and channel Y. Each side communicates with its channel on tag lines in the same manner as any other I/O control unit.

DATA FLOW

The data path through the adapter is nine bits wide; an eight-bit byte plus parity. Entry is from the channel bus-out lines, and exit is to the channel bus-in lines. Notice that the Y-side of the adapter is identical to the X-side. The logic and data path are the same if X and Y are reversed. For uniformity, the descriptions and diagrams are in terms of channel X.

Input

Channel X places three kinds of information on its bus-out lines during the interface sequence;

1. At address-out time, the address of the X-side of the adapter goes to the address compare X circuit.
2. At command-out time, the command byte goes to the command register.
3. At service-out time, the data byte goes to data buffer Y.

Output

D ata buffer Y is the output buffer for the data byte transferred from channel X to channel Y. The buffer also sends four other kinds of information to channel Y on the bus-in Y lines:

1. Address X, the address plugged on the card, is sent at address-in time.
2. The command register output is sent in response to 'sense command byte'.
3. The status byte is sent at status-in time.
4. The sense byte is sent in response to 'sense adapter status'.

OPERATIONAL CHARACTERISTICS

Two channels communicate through the CTCA using pairs of operations. For example, a read command from channel X will be handled by a write command from channel Y. The adapter responds to the channels with status information that allows continuation or indicates the condition that prevents continuation.

COMMANDS

The CTCA decodes and uses eleven commands. Bit positions 4-7 indicate the basic operation, and positions 0-3 indicate the modification code. This code expands the basic operation and provides modification bits for the programmer to use. In the commands listed below, X = don't care, and M = modifier.

Command	Bus-Out Bit Positions 0	1	2	3	4	5	6	7
Test I/O	X	X	X	X	0	0	0	0
Write	0	M	M	M	M	M	0	1
Write end-of-file	1	X	X	X	X	X	0	1
Read	M	M	M	M	M	M	1	0
No-operation	0	X	X	X	X	0	1	1
Control	M	M	M	M	M	1	1	1
Sense adapter status	X	X	X	0	0	1	0	0
Modified No-Op								
(Disable Compt)	1	1	X	X	X	0	1	1
(Enable Compt)	0	1	X	X	X	0	1	1
Sense command byte	X	X	X	1	0	1	0	0
Read Backward	M	M	M	M	1	1	0	0

Control Command

The control command as used in the CTCA is always an immediate command. This means that channel-end status will be sent to the channel during the initial selection sequence, thus freeing the channel during the Start I/O operation. A Sense Command byte must be issued by the non-initiating channel to free the adapter of the control and return it to the idle state. A control from channel X may be busy-rejected because Y had previously issued a command.

Control Issued to an Idle Adapter

Channel X initiated the control. The complete command byte will be set into the X-command register, and Channel-End will be sent with initial status, thus freeing channel X. A control immediate latch will be set in the X-side and an Attention-Interrupt will be presented to channel Y. Channel Y, when free, can accept the attention as defined by the IBM System/360 Principles of Operation.

To free the adapter, a Sense Command Byte command must be issued by channel Y. At the termination of the Sense Command Byte from channel Y, a Device-End for channel X is generated.

Control to a Busy Adapter

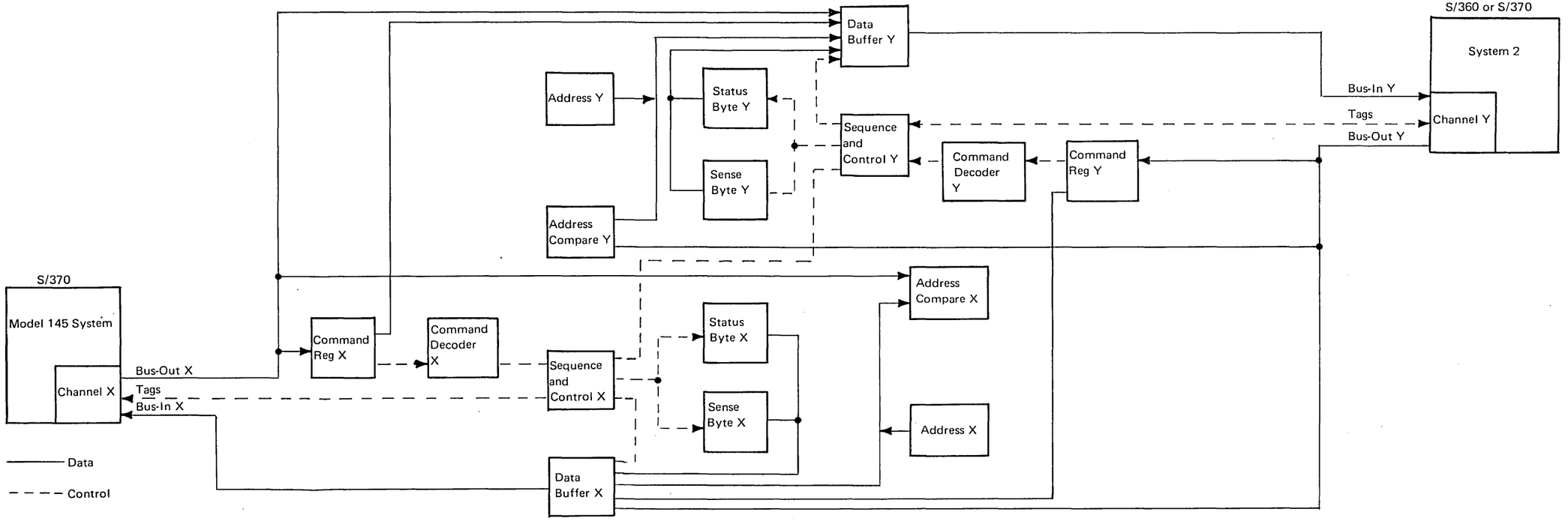
A control issued by channel X will receive busy status for the following reasons:

1. A previously issued control by X has not been cleared. This response will be the busy bit alone.
2. A previously issued control by channel X was cleared with a channel Y Sense Command Byte, but before the Device-End had been accepted by X. The response will be Busy and Device-End. This will clear the Device-End and leave the adapter idle.
3. The control is issued by X after Y has issued a command. The response will be Busy and Attention. The Attention after being presented will no longer bring up the Request-In tag and try to interrupt the CPU; but if any other command such as control were issued, the response will still be Busy and Attention.

Control to a "Not Ready" Adapter

A control from channel X to the adapter with the Y-side not ready will receive Unit Check.

CTCA Data Flow and Operation



Sense Adapter Status

This command is used to interrogate the condition of the adapter to determine the cause of a unit check. No change is made at the adapter by the execution of this command. The Unit-Check status bit is ignored, and the initial status will be all zeros; then the Sense Byte is sent as data to the channel. Channel-End and Device-End are presented as the final status. Any outstanding status other than Unit Check is presented to the channel as initial status, and the adapter disconnects after initial selection. This command receives a Busy response if there was an outstanding control at the adapter.

Sense Adapter Status Bits

The following is an explanation of what the combination of bits indicates.

1. All bits OFF indicate that the other side is ready. This is the normal condition.
2. Bit 1 ON (Intervention Required) and bit 7 OFF (Halt I/O or Sel Rst) indicate that the other side is not ready due to a system reset on that side.
3. Bits 1 and 7 both ON indicate that the other side is not ready due to a Halt I/O or Selective Reset.
4. Bit 3 ON (equipment check) indicates that a hardware failure has occurred on this side of the adapter. Bits 0, 4, 5, and 6 must be interrogated to determine whether the other side of the adapter was also in error. If they are all zero, this indicates that the side doing the sense had the only error.
5. Bit 2 ON (Buffer Data Check) indicates that the data buffer used on this side of the adapter has detected bad parity. The operation is not stopped, but a unit check is presented with the final status.

Sense Command Byte

The sense command byte command is used by one system to examine the command byte of the other system. If the adapter is idle when the sense is performed, the data byte will be all zeros. For the data byte to contain the command byte of the other system, the following commands were issued by the other channel.

Channel Y Previously Issued	Sense Data							
	0	1	2	3	4	5	6	7
Adapter Idle	0	0	0	0	0	0	0	0
Control	X	X	X	X	X	1	1	1
Read Backward	X	X	X	X	1	1	0	0
Read	X	X	X	X	X	X	1	0
Write	0	X	X	X	X	X	0	1

The only exception is when channel X issues a sense command byte to the adapter before a previous control from channel X has been cleared. If the control has not been answered by a sense from channel Y, then the sense would receive a busy response. If the X control had been answered, but the Device-End had not been accepted or had been stacked by channel X, then the sense would receive both Busy and Device-End in the initial status byte. This clears the Device End.

Read or Read Backward

The CTCA does not recognize the difference between a read and a read-backward command. In both cases, the primary function of the adapter is to transmit data bytes to the initiating channel. It is the function of the channel to place the data bytes in main storage in the correct order.

Read to an Idle Adapter

When a Read or Read Backward command is issued to an idle adapter, the issuing channel receives an all zero initial status, and then is held up until the other channel responds with a Write. An Attention is immediately set up to signal the nonissuing channel that an operation is waiting. The complete Read command is stored in the command register and is available to the Y-side by a Sense Command Byte command.

If a previous Write command had been issued by the Y-channel before the Read, the Attention on the X-side is reset and both operations are performed. A Service-In requests a byte of data from the writing (Y) channel, and it is passed through the adapter to the X-channel. The operation continues until either channel responds with a Stop (Cmd-0 to Srv-I). Upon receiving the Stop, a status containing channel-end is presented to both channels. If neither channel is chaining, the acceptance of the status by both channels frees the adapter.

Read to a Busy Adapter

There are three responses to a Read issued by a channel X.

1. If a previously issued X-control command has not been cleared (the X-control command still valid), the adapter responds to initial selection with a Busy status.
2. If there is outstanding status information stored in the X-side of the adapter, the response is Busy plus the status. This status could be Channel-End and Device-End from a previous operation that was stopped with a Halt I/O, or Device-End from a previous (X) sense command byte that had not been accepted by the channel; or a Device-End produced by the Y-side going from the not-ready to ready state.

Busy and Attention status is the response to an X-read command, if a channel Y had previously issued a Read, Read Backward, or Control.

If the Attention had not been previously accepted by channel X, the Read command clears it as an interruption condition, although it still appears as a response to another Read, until the previously issued command by channel Y is satisfied. A (Y) Write is satisfied only by an X-read, and a Y-control is satisfied only by an X-Sense command.

Read When EOF was Issued

If the X-adapter side End of File has been set by a previously issued Y-Write End of File command, the X-Read is rejected and the Unit-Exception bit is presented in the initial status byte.

Write

The Write command in the CTCA accepts data from the Writing (initiating) channel and transmits the data to the Reading (non-initiating) channel. A Write command might be issued to either an idle or a busy adapter. In either case, the response is the same as for a read command.

If Y-Read command has been issued before the X-Write, both operations are performed. Channel X receives a zero status in response to its command whether attention is accepted or not. Service-In requests a data byte from channel X and transmits it to channel Y. This operation continues until either channel signals Stop. If the operation is stopped by the reading channel, one more byte of data than is required is sent to the adapter by the writing channel.

Write End-of-File

The Write EOF command is used to signal a Read or Read Backward on the other side of the adapter when no more data is to be sent. This command is especially useful in trying to stop an operation where individual writes are being issued to chained reads. The Write EOF sets an EOF latch in the other side of the adapter that indicates that no more data will be given and that the Read chaining is broken.

When channel X issues a Write EOF, a pending Read on channel Y receives Channel-End, Device-End, and Unit Exception, ending status. This occurs before any data has been transferred but after zero initial status is sent to the channel. If the Write EOF were issued before the Read, and initial status was presented, the Unit-Exception bit would be presented alone and the Read command would be rejected.

The Write EOF is treated on the writing side as a No-Op, in that Channel-End and Device-End are returned in the initial selection status and the adapter is available for more commands.

Because a Write EOF is only meant to terminate a Read (or Read Backward), the EOF condition is reset if any command other than Read is issued on the other side. In this case, the EOF condition is never indicated and is lost.

If after a Write EOF is issued, another command such as control is given on the same channel and before a Read is issued on the other channel, the attention interrupt condition will be kept pending until after the Unit Exception is sent in response to a Read command.

Write EOF commands are not stackable; that is, if three EOF commands are issued by channel X, then channel Y issues the Read Command only. The first Read command receives a Unit Exception.

A Halt I/O, or a Selective or System Reset on either side of the adapter resets the EOF condition.

No Operation

No operation command in the CTCA is always a command immediate. The contents of the adapter latches are not affected by this command. The response that isssent to the channel if this command is issued to an adapter that is busy or has an outstanding status is the same as that stated for the read command

Test I/O

A Test I/O command in the CTCA may be used by the programmer to determine the status of the adapter any time the channel is free. The status received indicated the condition of the adapter as follows.

Zero status indicates that the adapter is idle.

A Busy bit indicates that the control unit is busy with an X-control command that has not been cleared.

If the Y-channel had previously issued a Read, Read Backward, Write, or Control that was still in the adapter, and an X-test I/O command is issued, the adapter responds with Attention status.

A Device-End status response to an X-Test I/O indicates that the previously issued X-control has been cleared or that the other side went from Not-Ready to Ready state. This clears the Device-End status.

A Channel-End and Device-End response indicates that a previously issued command was stopped by a Halt I/O. Any of the previously discussed status conditions presented to the channel as Interruption have the same meaning.

Halt I/O

The Halt I/O condition (Interface Disconnect) stops the adapter data transfer and makes the adapter not-ready. When the CTCA recognizes the Halt I/O condition, its response to the halting channel is immediate. The adapter drops all In-Tags lines, sets Channel-End and Device-End (if the reset occurs after initial status time), and waits for a chance to send the status to the channel. If the channel not initiating the halt is operating with the adapter, it will receive Channel-End, Device-End and Unit Check. If this channel does a Sense Adapter Status command to ascertain the cause of the Unit Check, it will receive a data byte with bits 1 and 7 on. Bit 1 ON indicates Intervention Required (adapter is not ready). Bit 7 ON indicates that a Halt I/O or Selective Reset caused the not-ready condition.

System or Selective Reset

A System or Selective reset is handled like a Halt I/O except that no Channel-End or Device-End is sent to the issuing channel. When the adapter is powered up, both the X- and Y-side receive an automatic reset.

A System or Selective Reset causes the adapter to become not ready. In order to make the adapter ready again, it is necessary to issue a Read, Read Backward, Write, Write End-of-File, Control, or No-Operation to the adapter. For example, if channel X issues a System Reset, the X-side is not ready until channel X issues one of the commands previously stated. The Y-channel has all of its commands except Sense Adapter Status rejected with Unit Check as long as the X-side remains not ready. When an X-command makes the X-side ready, a Device-End interruption occurs on the Y-channel, indicating that the adapter is ready.

When a Device-End is pending or stacked and the adapter is made ready on the other side, the ready Device-End is saved. After the pending Device-End is cleared, the adapter interrupts with the ready Device-End. When the adapter has a pending or stacked Device-End, a saved ready Device-End interrupt, and the Adapter is made not-ready, the saved Device-End is reset and only the pending or stacked Device-End interrupt occurs. The state of the X-adapter side ready latch has no effect on commands issued by channel X. The X-ready latch affects only channel Y-commands.

Status Byte

Each side of the adapter has a status-byte register where six of the eight available status bits are stored. The status bits appear on bus-in to the channel as follows:

Bus-in Bit Position	Status
0	Attention
1	Not Used
2	Not Used
3	Busy
4	Channel-End
5	Device-End
6	Unit Check
7	Unit Exception

Sense Byte

The CTCA uses seven sense bits to indicate to the channel the cause of a unit check presented in the status byte. Each side of the adapter stores its own sense byte. The bits appear on bus-in to the channel as follows:

Bus-in Bit Position	Sense Indication
0	Unused
1	Intervention required
2	Buffer data check (other side of adapter)
3	Equipment check
4	Selection check (other side of adapter)
5	Control sequence check (other side of adapter)
6	Status generation check (other side of adapter)
7	Halt I/O or selective reset

Intervention Required: The intervention-required bit is turned on if the other side of the adapter is not-ready because of one of the following three conditions:

1. System Reset
2. Selective Reset
3. Interface Disconnect (Halt I/O)

Buffer Data Check: This bit indicates that bad parity was detected at the output of the data buffer that feeds the bus-in interface for the other side of the adapter. This indicator is operative only during data service.

Equipment Check: This bit indicates that this side of the adapter has experienced a hardware-error condition. There are five error conditions that can occur on either side of the adapter:

1. Command register parity check
2. Command decode error
3. Selection check
4. Invalid sequence check
5. Status generation check

These error indications are available to the field engineer via the console display on the host CPU.

Selection Check: Selection Check is used to indicate that there has been a failure in the adapter selection logic. A check is indicated if the adapter is selected and the select-out tag has been propagated at the same time.

Invalid Sequence Check: This bit will be on if the Control Sequence Latches do not come on or do not go off in the correct order. These latches are used to control the interface In-Tags and other adapter timings.

Status Generation Check: This bit is turned on when an invalid status is generated. The fact that the status was incorrect is determined with the use of parity-predict logic for the status byte.

Halt I/O or Selective Reset: This bit is turned on when a Halt I/O (Interface Disconnect) or System or Selective Reset has been issued on the other side of the adapter.

SPEED

Data bytes are transferred in burst mode at the speed of the slower channel. The operational-in line of the adapter is held up from the time of selection until end status, forcing burst-mode operation.

ADDRESSING

The CTCA has two eight-bit addresses. One address for each side of the adapter is plugged by the customer engineer at the time of installation.

SELECTION

To select the adapter, channel X places an address byte on bus-out at address-out time. The address byte must match the X-side plugged address. The plugged address is then returned to the channel at address-in time.

COMPATIBILITY

The System/370 CTCA has expanded functions that are not available on the System/360 channel-to-channel adapter. The new adapter can use System/360 adapter programs by inhibiting or modifying the expanded functions.

The following are affected.

1. The Write EOF command becomes a Write command.
2. The Sense Adapter Status command becomes the Sense Command Byte command. Sense bits are not presented to the channel.
3. The Unit Check and Unit Exception status bits are inhibited.
4. The Ready latches for both sides of the adapter are forced on and are not allowed to reset. The mode of operation is selected by a Modified No-Op command issued by either channel as follows.

   To enable compatibility, the command is 01XXX011; for expanded function, the command is 11XXX011.

   X-aide power-on reset places the adapter in compatibility mode.

ONLINE OFFLINE MODES

The CTCA has an I/O interface switch located on the operator's console of the X (host) system. This switch allows CTCA to be logically removed from the X- and Y-systems. When the adapter has been removed (disabled), it will be unavailable to both the X- and Y-sides. An unavailable adapter will not respond to its address if selection is attempted (the adapter propagates select-out).

An indicator labeled I/O INFC DSBLD indicates that the adapter is offline. This indicator is lit only if the following conditions are satisfied on both sides of the adapter at the same time.

1. Neither interface is operating with a channel (both Op-In tags are down).
2. Neither interface is chaining commands.
3. Neither interface has pending or stacked status of any type.
4. Neither adapter side has a control command outstanding.

With the I/O interface switch in the DISABLE position and all of the four conditions (items 1-4) satisfied simultaneously, the indicator is active, indicating that the adapter is then unavailable to the X- and Y-systems. The adapter remains in this condition until the switch is moved to the ENABLE RESET position.

I/O INTERFACE ISOLATION

The CTCA has added circuits to allow the power to be turned off and on without disturbing the operation of the Y-channel. This allows maintenance of the host (X) system and the CTCA without disturbing the operation of the (Y) system.

Select Out Bypass

Because the adapter receives its power from channel X, it is unnecessary to propagate the channel X select-out tag when power is off. A select-out bypass circuit is provided to enable channel Y to operate with other control units when power is off on channel X.

Some degree of flexibility in assigning priority to both sides of the adapter is accomplished by allowing the select-out and select-in tag lines to be interçhanged. A jumper arrangement is provided to allow the service representative to accomplish the interchange.

The X-side of the adapter goes to tailgates and therefore can be placed in any position on the interface, but it must always be connected to one of the channels of the host system.

Power On-Off Sequence

In order to ensure proper isolation, the host (X) system must supply the following hardware and controls when powering up or down:

1. An I/O interface switch.
2. An I/O INFC DSBLD indicator.
3. Automatic control of the power sequencing of the + 6 volts' supply used for the interface driver circuits. This is to eliminate transients on the signal lines.
4. A relay-controlled pick for the ground side of the 6-volt circuit for the select-out bypass relay. The sequencing of the functions in items 3 and 4 is accomplished as follows.

To turn off power, proceed as follows.

1. Send operator messages to both systems, specifying removal of the adapter from use.
2. Move the I/O interface switch to the DISABLE position.
3. Wait for the I/O INFC DSBLD indicator (green) to light.
4. Press the power-off push button.

The reverse of the previous procedure is used for power-on.

1. Press the power-on push button.
2. I/O INFC DSBLD indicator should come on.
3. Move the I/O interface switch to the ENABLE position. The I/O INFC DSBLD indicator should go out.
4. Adapter is not ready, but it is available to the program.

DISPLAYS

The CTCA uses two external registers, one for each side of the adapter for display in the host CPU. The format of these words is shown here:

Bit Position

0	–	Command check
1	–	Intervention required
2	–	Buffer check
3	–	Equipment check
4	–	Selection check
5	–	Sequence check
6	–	Status Generation check
7	–	HIO or selective reset
8	–	Ready latch
9	–	Select propagate
10	–	Side selected
11	–	Sequence counter 1
12	–	Sequence counter 2
13	–	Sequence counter 3
14	–	Sequence counter 4
15	–	Sequence counter 5
16	–	Disconnect
17	–	Status parity bit predict
18	–	Attention latch
19	–	Busy bit
20	–	Channel-End
21	–	Device-End
22	–	Unit check
23	–	Unit exception
24	–	Stack status latch
25	–	Write command
26	–	Read command
27	–	Control command
28	–	Sense command
29	–	End-of-File command
30	–	Spare
31	–	Enable compatibility (X side)

The X-side uses external register 11 (CTCAX). The Y-side uses external register 13 (CTCAY).

PROGRAMMING NOTES

The following information explains operations that are peculiar to the adapter.

Initial Program Load (IPL)

During the initial program loading operation, a system reset pulse precedes the selection of the adapter. This causes the adapter to become not-ready, and any outstanding operation on the other side of the adapter is terminated with Channel-End and Device-End sent to a Read or Write, and Device-End sent to a Control command. Another Device-End will be presented because the Adapter on the IPL side went from not-ready to ready, and then Attention status will be presented because of the IPL Read Command. The Y-side of the adapter must respond with a Write command after the Ready Device-End is accepted (in order to complete the IPL).

Command Chaining

Command chaining may be performed by channel X and channel Y and/or both simultaneously; however, it must be remembered that all operations performed by the CTCA occur in pairs. Therefore, an illogical sequence of commands may send a status of Busy and Attention to the last channel issuing a command. For example, if both channels are chaining and a read command is issued by channel Y followed by a control from channel X, channel X will receive an Attention that will cause channel X to disconnect from the adapter. Channel Y remains connected to the adapter until a correct response is received from channel X, in this case a Write or a Write End-of-File command. Because the sense commands do not require a reply, they should not be chained in answer to a chain of control commands. When channel X issues a Write, Read, or chained control command to the adapter, the Busy condition remains active until a correct response is received from channel Y or until either channel issues a Halt I/O or Reset.

Halt I/O

The Halt I/O instruction always causes a Channel-End and Device-End to be sent to the channel that issued the halt, *only* if initial status was accepted before the adapter was halted. The channel- and Device-End status will be sent to the channel as an interruption or in response to a Test or Start I/O instruction.

Attention

If the adapter has a pending attention interruption, a Test I/O will clear the attention as an interruption condition although it will not reset the attention status bit. Any subsequent Test I/O will receive attention until a match between the channel command occurs; for example, a Write on one channel to a Read on the other. When a device has the attention bit on in the status byte and receives Start I/O, the attention bit will be reset if a match occurs, and the adapter will execute the operation. Under the same condition, if a mismatch occurs, the Start I/O will receive a condition code of 01 with the Busy and Attention bits stored in the unit status position of the CSW.

Wrong Length Record

A Wrong Length Record indication can occur under the following four conditions at the end of the operation:

Read on Channel X to Write on Channel Y	Wrong Length Record Indicated on Channel X	Channel Y
1. X Count equal to Y count.	No	Yes
2. X count greater than Y	Yes	Yes
3. X count less than Y	Yes	Yes
4. X count equal to Y count less 1	Yes	No

SUPPRESS DATA

The Suppress-Out tag is used in the CTCA to keep the channel from being overrun during data-chaining operations. It is also used at any time a particular channel needs to stop the operation to allow servicing of one of its other channels. The CTCA will not request service (Service-In tag) from a channel while its Suppress Out tag is up.

FUNCTIONAL UNITS

The CTCA functional packaging allows a complete logical function to be performed by one MST card. Five of the cards are for the X-side of the adapter, and each has an exact counterpart on the Y-side. Interchanging X and Y on the line and block names adapt these diagrams to the Y-side.

Address Compare and Data Flow X

- The card location is B-B2R2. The ALDs are XX001, XX011, XX021-051.

Five functions shown on the diagram (CTCA Data Flow and Operation) page 9-25 are packaged on this card. They are address X, address compare X, data buffer Y, command register X, and command decoder X.

For channel X to select the adapter, the bus-out X at address-out time must compare with the plugged address X. Plugged address X is sent to the data buffer X at address-in time.

Bus-out has three kinds of information; each is identified and selected at the correct time by the channel out-tags.

The address compare circuit is enabled by address-out.

The command register accepts bus-out at command-out time.

Data buffer Y accepts bus-out at service-out time.

The three-way selector gates the command-register output, the data byte, and the Y-side address. The two-way selector gates the output of the three-way selector and the input B bus-in lines. Input B is the sense byte or the status byte.

The eleven commands from channel X are decoded from the X-command register output into nine adapter commands. Read Backward is interpreted as the read command.

The output of the command register and command decoder are checked for parity; either one in error turns on X command check.

The command register output generates a new parity bit that goes to the Y-data buffer with bits 0-7. The output of the data buffer is parity-checked; an error causes a buffer data check.

Selection and Reset X

- The card location is B-B1T4; the ALDs are XX601-621.

Selection: The channel must address the adapter, and then send select-out and receive operational-in to establish correct selection.

The channel X hold-out and select-out tags set the select-out latch.

Successful address compare allows 'address matched X' to reset the select propagate latch. The select-out latch may then set the X-side selected latch.

X-side selected sets sequence 1 X which generates the operational-in tag.

Reset: Power-on reset causes both sides of the adapter to receive an automatic reset.

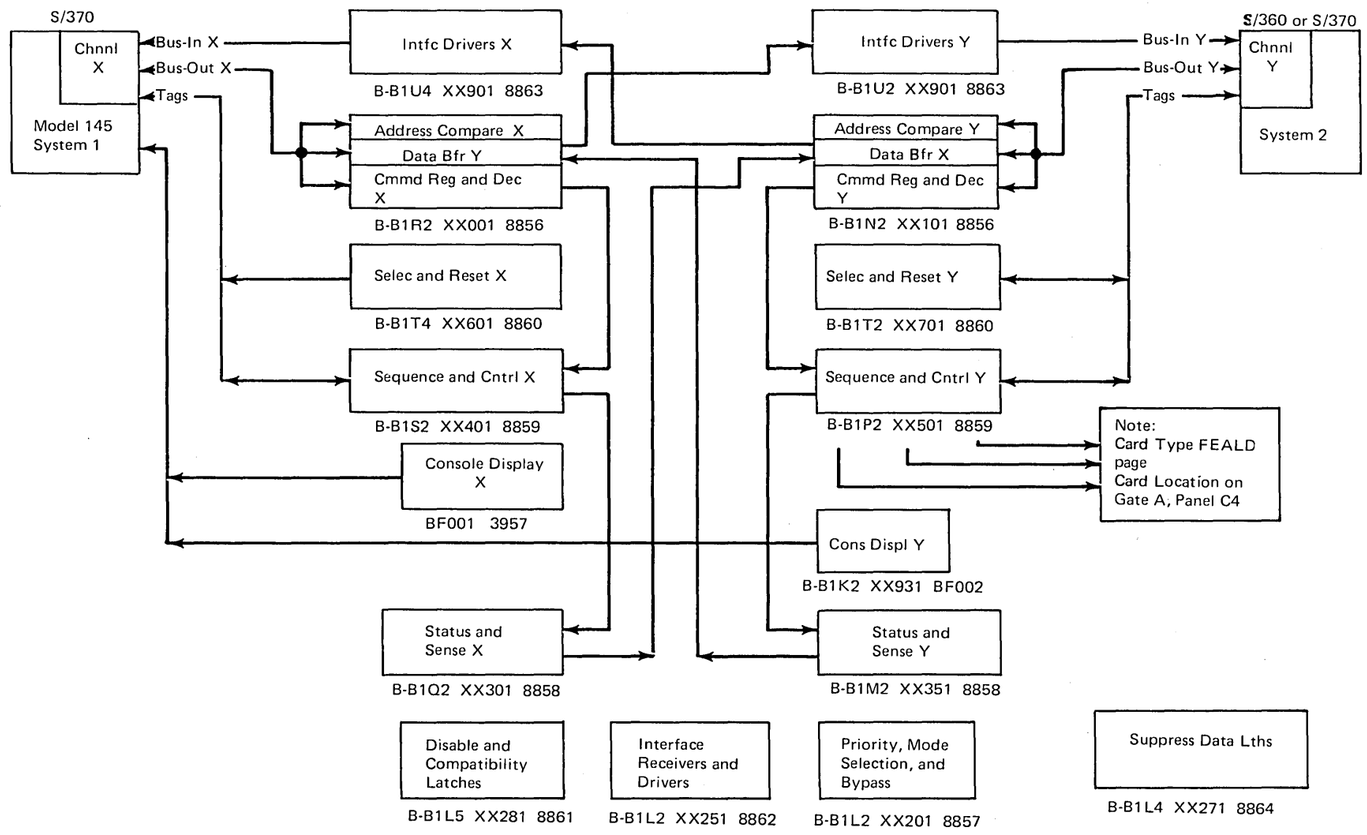
The suppress-out tag and the operational-out tag control system-reset and selective-reset latches.

Sequence and Control X

- The card location is B-B1S2, and the ALDs are XX401-451.

Interface tags and adapter commands enter this card to generate the gates and controls needed to transfer data through the adapter.

Locations



Sequence: A series of latches from sequence 1 to sequence 5 allows the adapter to keep in step with the channels.

Control: The decoded commands, sequence latches, and interface tags set a series of latches that control data transfer.

The gate selector selects the sense byte, data byte, command byte, and status byte.

'Start data service sequence' or 'sense command or status' sets the first byte latch before the go latch is set.

'Bus-in set' is activated for each kind of byte set into the data buffer. Notice that control lines from both sides of the adapter generate the 'bus-in set' pulse.

The adapter check register is set by buffer check, sequence check, and status check errors.

Disable and Compatibility

- The card location is B-B1L5, The ALD is XX281.

Disable: The I/O interface switch on the console lets channel Y operate when channel X is powered up or down. The 'set disable latch' line is controlled by the switch. Other input lines prevent setting the latch in the middle of an operation.

Compatibility: System/360 channel-to-channel programs can be used if the CTCA is in compatibility mode. The compatibility latch must be reset to allow compatibility. Bit 0 of the modified No-Op command provides the set and reset of the latch.

Select Priority, Mode Selection, and Bypass

- The card location is B-B1L2; the ALDs are XX201-211.

Select Priority: Each side of the adapter may be logically on select-out or select-in. The chart shows plugging for high or low priority for both sides.

	High Priority	Low Priority
Y-side	BC-BB	BC-CC
	BA-CA	BA-BB
	CB-CC	CB-CA
X-side	CG-CH	CG-BG
	AG-BG	AG-AH
	BH-AH	BH-CH

Mode Selection: Data-in mode is selected by plugging DE to DD.

Select-out Bypass: With power off, the relay is normal, allowing the channel Y select-out line to take the following path: pin BC to pin BB, through the normally closed relay point, through the Dot OR to pin CB, to pin CC and select-out tag Y.

Status, Sense, and Input B Bus-in X

- Card location B-B1Q2.
- ALD XX301-321, XX325-345.

This card analyzes the condition of the X-side of the adapter to generate the six status bits and three of the eight sense bits. The other five sense bits are generated with the function they represent on the other side of the adapter.

The circuits shown on ALD XX301-321 include:

Status Bits

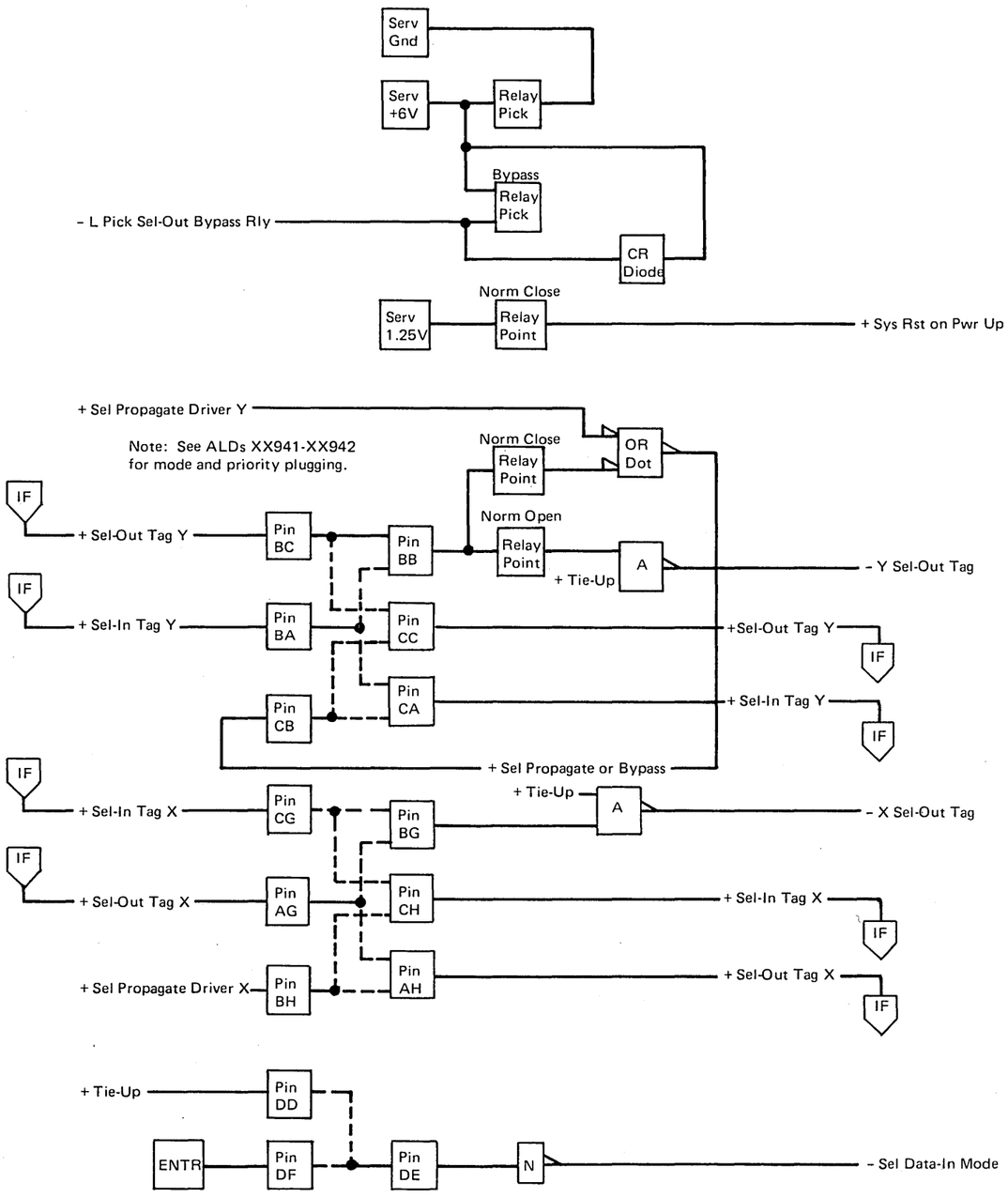
0	Attention
3	Busy
4	Channel-end
5	Device-end
6	Unit check
7	Unit exception

Sense Bits

1	Intervention required
3	Equipment check
7	Halt I/O or selective reset

Input B Bus-in: Status and sense bits 1, 3-6, and parity are selected by 'gate status X' and 'gate sense byte X' to become part of the 'input B bus-in X' byte that goes to the Y-data buffer. The remaining three bits of the 'input B bus-in X' are selected on the address compare and data flow card.

Select Priority, Mode Selection, and Bypass



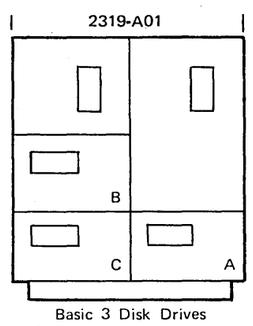
CHAPTER 10. INTEGRATED FILE ADAPTER

CONTENTS

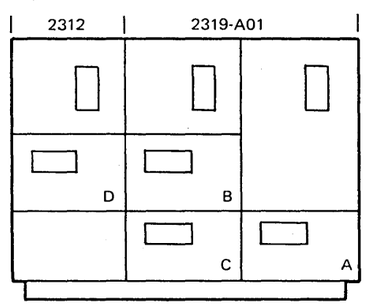
ABBREVIATION LIST

Adv	Advance
ALU	Arithmetic Logic Unit
AM	Address Mark
Amp	Amplifier
Appndg	Appendage
Attn	Attention
BCA	Bit Count Appendage
BR	Bit Ring
C	Count
CAR	Cylinder Address Register
CAW	Channel Address Word
CC	Condition Code
CC	Chained Command
CC	Cyclic Code
CCW	Channel Command Word
CD	Chained Data
CE	Channel End
CE	Customer Engineer
Chk	Check
Chnl	Channel
CKD	Count, Key, and Data
Clk	Clock
Cmd	Command
Cnt	Count
Coax	Coaxial cable
Con-con	Contingent connection
Cond	Condition
CPU	Central Processing Unit
Ctrl	Control
CS	Control Storage
CSW	Channel Status Word
CU	Control Unit
CUA	Control Unit Address
CUB	Control Unit Busy
CUE	Control Unit End
Cyc	Cycle
Cyl	Cylinder
D	Data
DE	Device End
Dec	Decode
Dest	Destination
Det	Detect
DF	Disk File
Diag	Diagnostic
Diff	Difference
DL	Data Length
Dly	Delay
EBI	External Bus In
EBO	External Bus Out
Enc	Encode
Eq	Equal
Ext	External
FBO	File Bus Out
FDR	File Data Register
FM	File Mask
Gen	Generate
HA	Home Address
Hi	High
HIO	Halt Input/Output
Hwd	Hardware
ID	Identifier
IFA	Integrated File Adapter
IL	Incorrect Length
Ind	Indicator
Inh	Inhibit
Inlk	Interlock
Intp	Interrupt
IPL	Initial Program Load
K	Key
KD	Key and Data
KL	Key Length
L	Length
Lch	Latch
Lo	Low
Lth	Latch
Mod	Module
ms	Milliseconds
MS	Main Storage
ns	Nanoseconds
Op	Operation
Osc	Oscillator
PCI	Program Controlled Interrupt
Pos	Position
Prog	Program
PSW	Program Status Word
Rd	Read
Rdy	Ready
Recal	Recalibrate
Req	Request
Rst	Reset
R0	Record 0
R1	Record 1
Sch	Search
SDBI	Storage Data Bus In
SDBO	Storage Data Bus Out
Sel	Select
Sel'd	Selected
Selr	Selector
SERDES	Serialize/Deserialize
SIO	Start Input/Output
SLI	Suppress Length Indication
SS	Single-shot
Std	Standard
Supp	Suppress
Sw	Switch
SX	Selector Channel
Sync	Synchronize
TIC	Transfer in Channel
TIO	Test Input/Output
Trig	Trigger
us.	Microseconds
VFO	Variable Frequence Oscillator
WLR	Wrong Length Record
Wr	Write
Xfer	Transfer

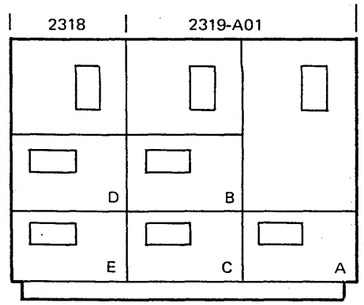
IFA/2319-A01 Disk Storage Facility



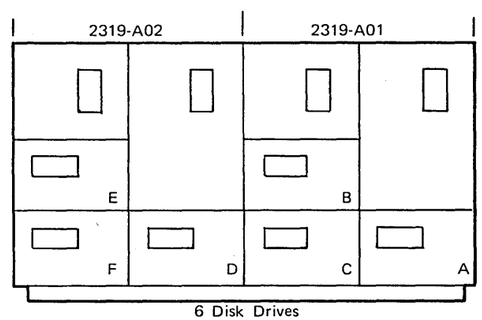
Basic 3 Disk Drives



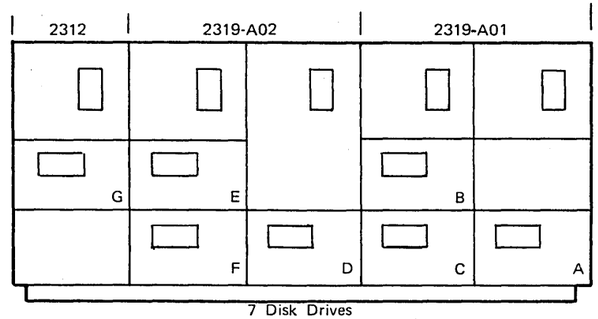
4 Disk Drives



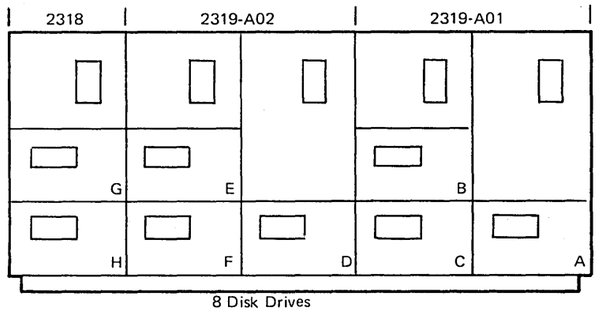
5 Disk Drives



6 Disk Drives



7 Disk Drives



8 Disk Drives

The IFA must always use the 2319-A01 for its first three drives. The basic three-drive system can be expanded to eight drives. The diagrams show the normal configuration for each combination between three and eight drives. Alternate combinations can be made for the last five drives using the 2312, the 2313, and the 2318.

DIAGRAM LIST

INTRODUCTION

GENERAL DESCRIPTION

- The Integrated File Adapter (IFA) feature connects from three to eight IBM 2319-type disk drives to the System/370 Model 145.
- The IFA feature is assigned exclusive use of the channel-1 address and functions as both channel and control unit for the files.
- Data transfer takes place one byte at a time on a share-cycle basis the same as with selector channels (non buffered).
- The initiation of operations and the initiation of each step of the file sequence require the use of the CPU controls and microprogramming.

The Integrated File Adapter (IFA) feature for the System/370 Model 145 provides the means to attach and control three to eight disk storage drives of the IBM 2314 type. The IFA takes the place of the normal channel control unit. The adapter operates with an IBM 2319-A1 containing the first three disk storage drives.

The primary control for the IFA is contained in the CPU, where it can make use of the CPU hardware and microprogram for operation. The IFA adds hardware for control during periods of reading and writing the file records and for other periods of control when the microprogram is not required. The 2319-A1 contains the read clocking circuits, the write oscillator, and the storage module switching for up to eight files. The disk storage drives operate in the same manner as the 2314 system connected to a selector channel. The record format is identical, and operation requires the same programming systems.

The interface to the disk drives is identical to that used between the disk drives and the 2314-A. When more than the three disk drives contained in the 2319-A1 are required for the system, the additional drives can be made up from any combination of the following, not to exceed a total of eight:

1. IBM 2312	1 Drive
2. IBM 2318	2 Drives
3. IBM 2319-A2	3 Drives
4. IBM 2313	4 Drives

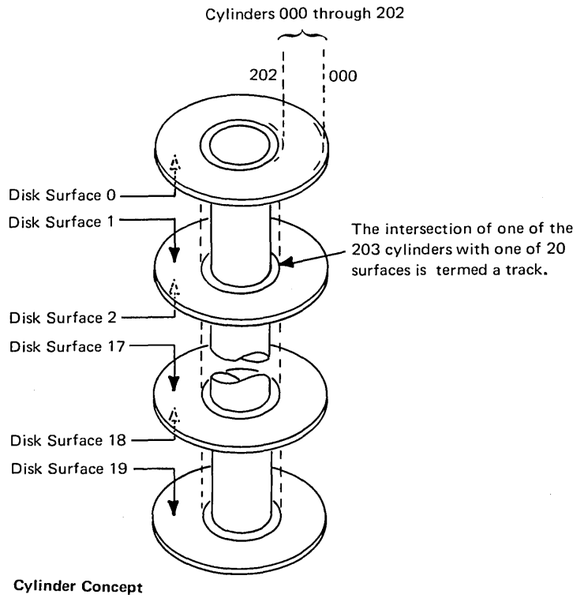
The IFA feature functions both as the disk storage control unit and the channel control in the system. This combination eliminates the need for channel interface controls, and thus, provides rapid access to the files. When the IFA is included in the 145, the selector channel 1 addressing and its space is assigned to the IFA. The IFA controls also make use of the space assigned to selector channel 4. A 145 system with the IFA feature can only have selector channels 2 and 3.

The IFA feature posts a Channel ID in main storage word 168 of 00 0A 00 60. The first half byte indicates a selector channel. The 00A identifies the IFA to the program. The low-order byte (60) indicates a maximum log of 96 bytes.

The IFA control unit provides the necessary controls for reading and writing the file records. In addition, it provides the controls for the seek function that places the heads into position to read or write the desired track on the file. The IFA can handle all of the basic control commands used by the 2314 without two-channel switch.

The IFA control unit consists of one board of MST logic located in the normal channel-1 position and an SLT board in the 2319-A1. The IFA does not use the channel-4 logic board but does use its external and local storage addresses.

The IFA control-unit operation is initiated as a channel operation using the I/O instructions and channel commands. Primary control information for the file operation is stored in the CPU. Operating commands are processed by microprograms stored in the CPU control storage. The microprogram starts the operation by developing the appropriate information for a portion of the sequence and issues a mini-op to the control-unit hardware. While the hardware is performing the mini-op, the microprogram stores a link address and returns to CPU operation. When the hardware finishes that portion of the sequence, it requests a trap to return to the microprogram link address to continue the operation.



Cylinder Concept

An operation may require several of these transfers between the microprogram and hardware to complete a command.

Data movement during the hardware control periods is performed by requesting a selector channel share-cycle for each byte. The CPU or other channel operations have use of the CPU hardware and microprogram for other operations when time is not required by the IFA controls either for setup or data transfer. The file operation should never overrun during normal operation because of the assigned priorities.

DATA STORAGE

- Data is written on the surfaces of an IBM 2316 Disk Pack with specific address locations.
- Data is written serially by bit on the file tracks with the clock bits written alternately.
- The CPU byte parity bits are dropped, and a four-byte cyclic-check is developed and written with each record field.

The 2319-A1 Disk Storage Facility uses the 2316 Disk Pack for its storage media. Each pack has twenty recording surfaces on which data records can be written. Each disk surface has 203 concentric tracks on which data can be recorded. (Only 200 tracks are normally active.) Twenty read/write heads are arranged on each disk drive access to simultaneously align with the same numbered track of their respective disk surface. These twenty tracks are referred to as a cylinder for addressing (seek location). The cylinder number and the head number define a specific track. For multi-track operations, all twenty tracks of the cylinder are available with sequential head switching.

Data is recorded on the disk surface serially by byte and serially by bit starting with bit 0 of the first byte of the record. The clock bits used to write the data are also written on the track alternating with the data bits. This allows a synchronized clock to be developed during subsequent read operations. The clock bits form a continuous pattern that defines *data cells* on the track. The data 1-bits are written midway between associated clock bits as discrete spots. Nothing is written in the *data cell* for the data 0-bits. The eight bits of the successive bytes are written as a continuous pattern without separation or extra bits for markers.

The parity bits contained in the CPU data bytes being written are dropped in favor of a two-byte cyclic-code, a byte of file identification, and byte of bit count for the field. These four bytes are written following each field on the track. When subsequently reading the data from the track, the code and count are again developed and compared with those previously written to verify the data. A parity bit generator develops a parity bit for each byte being read to send with the byte when entering the CPU.

BASIC IFA DATA FLOW

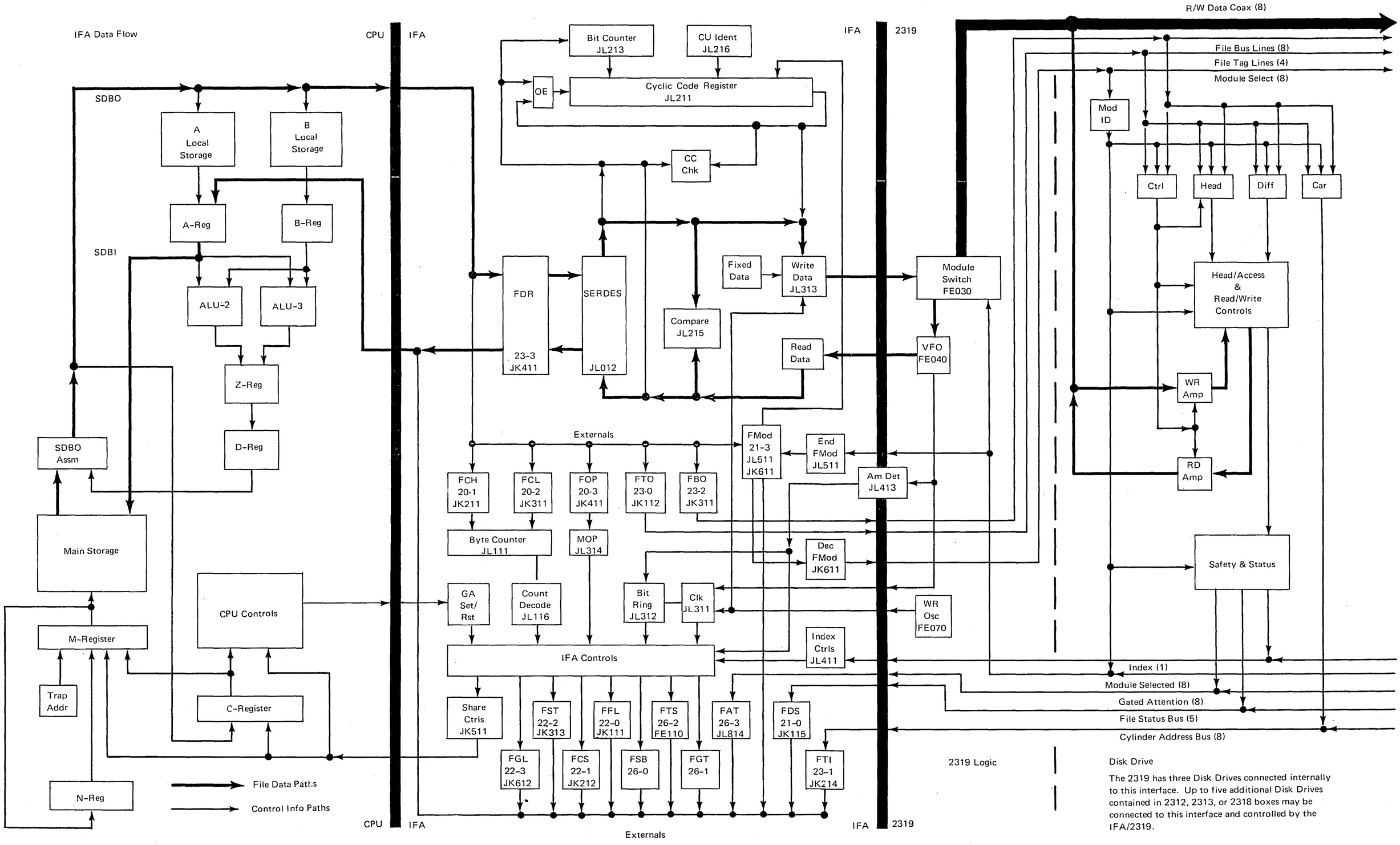
- Data from main storage is buffered by byte and then serialized by bit to write on the selected file.
- Data read from the selected file is deserialized and buffered by byte before entering the CPU and main storage.
- Control and data information are transferred from the CPU to the IFA feature through external registers.
- Control and data information from the IFA feature to the CPU use external registers and some direct control lines.
- For search operations, data from main storage is buffered by byte and then serialized to compare with the data reading from the selected file.

The data flow diagram shows the data paths and the major control elements of the control unit of the IFA feature. The external bus-in (EBI) lines are used to transfer both data and control information from the CPU to the IFA external registers. The information comes from either main/control storage or from the ALU output. These bytes are set into appropriate external registers for use by the IFA control unit.

Both data and control information entering the CPU from the IFA external registers use the external bus-out (EBO). The information enters the CPU A-register, from which it is either stored in the main/control storage or is used within the CPU processing controls. These flow paths are the normal transfer paths between the CPU and the external registers.

Within the IFA feature, information entering the external registers is transferred to its respective operating registers at the appropriate time. Data moves from the file-data register (FDR) to the SERDES. The high and low count registers (FCH, FCL) transfer to the byte counter, and the file-op register (FOP) transfers to the mini-op register (MOP).

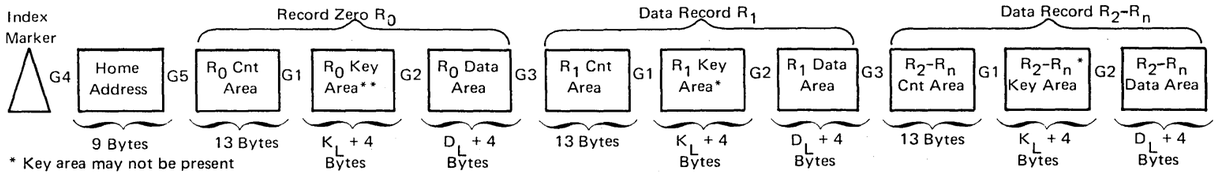
The data byte in the SERDES is serialized and gated to the selected file, the compare circuits, and the cyclic-check circuits depending on the operation to be performed. This path is used by the write and search operations. For read operations, the data from the selected file enters the SERDES, where it is deserialized. The completed byte transfers to the file-data register (FDR) from which it is transferrred to CPU storage. For search operations, the data being read from the selected file feeds the compare circuits and is not deserialized. This sequence repeats for each byte until all of the data is transferred. The request to move data makes use of the selector-channel share-cycle controls.



Disk Drive

The 2319 has three Disk Drives connected internally to this interface. Up to five additional Disk Drives contained in 2312, 2313, or 2318 boxes may be connected to this interface and controlled by the IFA/2319.

TRACK FORMAT



- Data is written in tracks on the disk surface, and the track is formatted into one or more records.
- The track formatting is done with a group of initial write or formatting commands.
- Gaps are written between records and fields of the track to provide time for control-unit operation.

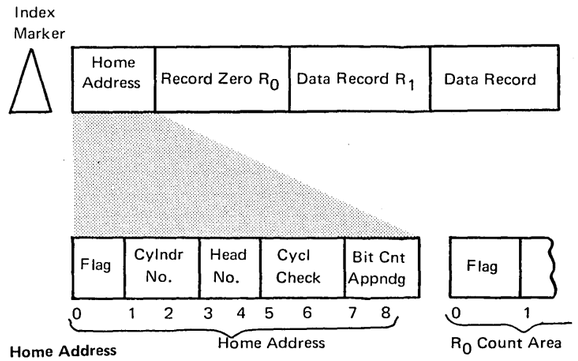
The track is the smallest addressable unit on the file. The track may, however, have several records that can be read out selectively through programming. The starting point for all tracks on the file is from a fixed index point. The first recorded area on each track following the index point is the home-address field that defines the address of the track. This is followed by record-0 that normally contains a track descriptor record used in IBM programming systems. Following record-0, the track can have one or more data records (R1-Rn) depending on the length of the records. The identifiers, the data, and the gaps must total no more than 7294 bytes.

Track formatting refers to the initial writing of the track record structure for a specific job. When recorded data and key fields are rewritten, they are written within the structure of the track formatting. The gaps between the record fields may vary slightly because of subsequent speed variations of either the write oscillator or the disk drive speed. This is one of the reasons for using gaps between records and between record fields. The second reason for the gaps is to allow time for the control unit to change the controls between fields. What is initially defined as a simple gap is actually made up of two or three sections adjoined.

Home-Address Field

- The address of the track is written as the home-address field of each track.
- The home-address field contains a flag byte that defines the track conditions.

Each track has a written field at the beginning that defines the track address and the track condition. This area is available to the program with the write- HA, the read-HA, and the search-HA commands. The home address is normally written after a 73-byte gap from the index point. In cases of a defect at the start of the track, this gap is increased to 778 bytes. The basic gap allows for differences in index-point-to-head relationship in different drives and time to advance and reselect the head when required. All but the last eight bytes of the gap are written with FF-bytes. The remaining eight bytes of the gap are written with synchronizing and identification information for the following field. The area includes four 00-bytes and one FF-byte for VFO clock synchronization followed by two address-mark bytes and the sync byte to synchronize the bit ring. The sync byte of 0D identifies the home-address field.



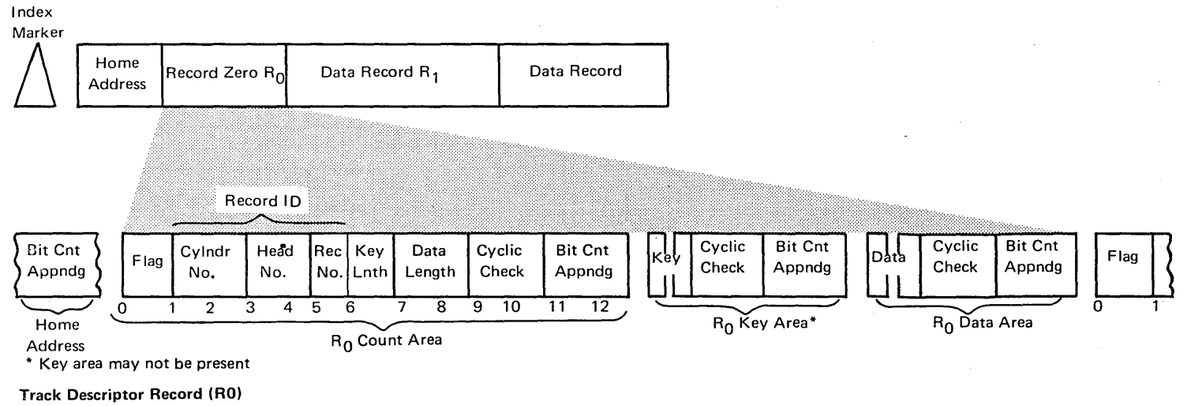
Home Address

The record portion of the home-address field contains five bytes of data followed by the 4-byte cyclic-check information. The data bytes are identified as follows:

Flag	1 byte	This byte indicates the track condition. Bits 0 through 5 are zeros (not used). Bit 6 indicates a defective track. Bit 7 indicates an alternate track.
Cylinder	2 bytes	Contain the cylinder address of the track. (First byte always zero.)
Head	2 bytes	Contain the head address of the track. (First byte always zero.)

Record Zero (R0)

- Record zero is used in the IBM programming systems as a descriptor defining the recorded area.
- R0 can be used to store application data in other systems.
- R0 contains provisions for count, key, and data fields the same as other records. (The key field is normally omitted.)
- The record length is defined by the length values formatted in the count field.



Track Descriptor Record (R0)

Record zero (R0) is the first information record on the track. It is normally used as a track descriptor record in IBM programming systems. In this case this record contains the identifier of the last record on the track and the number of bytes remaining unused on the track. If the track is defective, R0 contains the address of the alternate track. When the programming system does not require R0 for this purpose, it can be used as the first data record on the track. R0 contains the same three fields as the subsequent records on the track. These fields are designated as count, key, and data.

Count Field: The interrecord gap between the home-address field and the count field of R0 contains 43 bytes. This gap is made up of the post-record gap of the HA field, the interrecord skew compensation, and the VFO/AM sync area. The first two portions of the gap (35 bytes) are written with FF-bytes. The VFO/AM area contains four 00-bytes, one FF-byte, two AM-bytes, and the 0B sync byte to identify the R0 count field.

The count field contains nine bytes of data followed by the four-byte cyclic-check information. The data bytes are identified as follows:

Flag	1 byte	This byte is written by the control unit and is the same as the home-address field flag byte.
Cylinder	2 bytes	Contain the cylinder address of the track.
Head	2 bytes	Contain the head address of the track.
Record	1 byte	Contains the track record number (0 for R0).
Key L	1 byte	Contains the length of the key field as a byte count.
Data L	2 bytes	Contain the length of the data field as a byte count.

The cylinder, head, and record number portion of the count field compose the identifier (ID) used to locate a specific record with the search-ID command.

Key Field: The key field contains the number of bytes specified by the key length (KL) byte in the count field (255 bytes maximum) followed by the four-byte cyclic-check information. If the key length in the count field is zero, no key field is written and no space is allowed for the field or its associated gap.

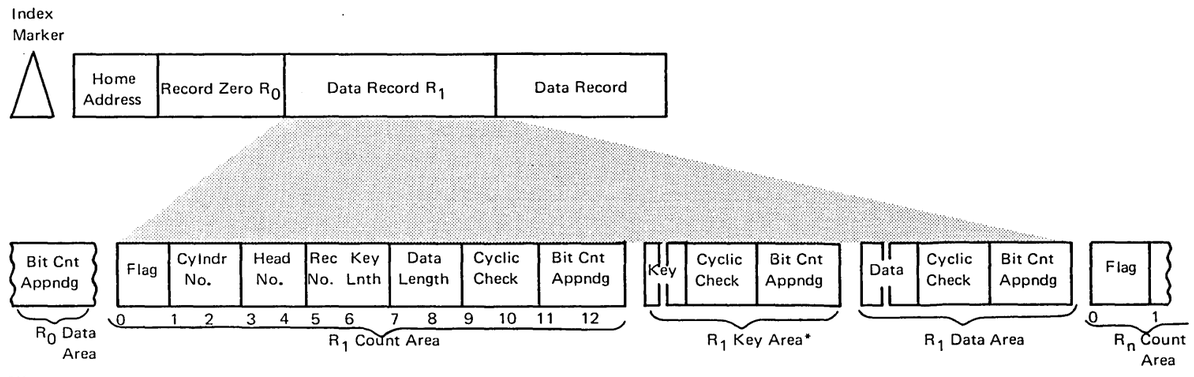
When the key field is written, a gap of 41 bytes is written between the cyclic-check of the count field and the key field. This gap is made up of the post-record gap of the count field, interfield skew compensation, and the VFO/AM sync area. The first two portions of the gap (33 bytes) are written with FF-bytes. The VFO/AM area contains four 00-bytes, one FF-byte, two AM-bytes and the 0A sync byte to identify the key field.

The key-field information comes from the system storage during a write operation. This information may be an identifier portion of the data field, a search argument, or a store catalog for the record. A search on the key field may be used to identify or locate the data-field information.

Data Field: The data field contains the number of bytes specified by the two data length (DL) bytes of the count field followed by the four-byte cyclic-check information. If the data length in the count field is zero, this record is the 'end of file' indicator and no data or its associated gap is written.

A gap of 41 bytes is written between the previous field and the data field. This gap is made up of the post-record gap of the previous field, the interfield skew compensation, and the VFO/AM sync area. The first two portions of the gap (33 bytes) are written with FF-bytes. The VFO/AM area contains four 00-bytes, one FF-byte, two AM-bytes, and the 09 sync byte to identify the data field.

The data information comes from the system storage during a write operation. This area contains the actual record being stored, and its content is based on the application. Information read from the data field is normally stored directly into the system storage. The information may be fed to the compare circuits as part of a search function, and the information not stored.



*Key area may not be present

Data Records

Records R1 Through Rn

- Records R1 through Rn on a track are numbered in ascending sequence and separated by interrecord gaps.
- Interrecord gaps are formatted with greater length, as the records become longer, to accommodate greater variables during rewrite.
- The count field contains the record number and track addresses along with the field lengths.
- The key fields and data fields of all records have the same specifications as R0.

Records R1 through Rn on a track contain the application information. Their format except for the count-field sync byte and flag are identical to that of R0. The number of records that can be written on a track is a function of the size of the records. The track accommodates a total of 7294 bytes following the normal R0 record. This byte count includes the count fields and all gaps in addition to the key information and data information to be written. Each pair of records is separated by an interrecord gap that is variable in length to allow for skew in rewriting due to speed variations. The option for the key field is the same as for R0.

Count Field: The interrecord gap written between records has a minimum of 43 bytes. The gap is increased when the combined key-field and data-field length of the previous record exceeds eight bytes. The longer gap is to allow for the greater skew from speed variations that can occur with the longer record. The gap is increased by a factor of .043 times the KL and DL values of the previous record. The total gap is made up of the post-record gap of the previous record, the interrecord skew compensation, and the VFO/AM sync area. The first two portions of the gap (35 + variable bytes) are written with FF-bytes. The VFO/AM area contains four 00-bytes, one FF-byte, two AM-bytes, and the 0E sync byte to identify a normal count field (not R0).

The count field of records R1 through Rn differs from the count field of R0 in the extension of the use of the flag bits. This byte is written by the control unit as follows.

Bit-0	This bit is set to 1 for all odd-numbered records, and to 0 for all even-numbered records, to aid in missing record detection.
Bit-1	This bit is set to 1 for all segments of overflow records except the last. The bit is set to 0 for the last segment and for all non-overflow records.
Bits 2-5	Not used (set to 0).
Bit-6	Written the same as in the home-address flag.
Bit-7	Written the same as in the home-address flag.

The remainder of the count field is identifical to that of R0.

Key and Data Fields: The key-field and the data-field formats and gaps for R1 through Rn are the same as those described for R0. The key field may be used or not at the user's option. When the KL byte of the count field is set to zero, the key field and the associated gaps are not written. When the DL bytes of the count field are set to zero, the record signals the end-of-file and no data field is written.

IFA CHANNEL COMMANDS

- All IFA operations are initiated with channel instructions.
- The basic channel commands are expanded with modifier bits to define the control or fields.
- Channel commands are chained to perform the desired operation.
- The command codes are executed by developing a sequence of mini-ops for control-unit operation.

The IFA control unit is similar to the conventional file control units that interface with a channel. The same programming systems that are used to operate a 2314 on a selector channel can be used with the IFA (except two-channel switch). The channel instructions are used to initiate all IFA operations. The control, sense, and data movement operations are initiated with the start-I/O instruction and a CAW to define the location of the command sequence.

The IFA uses five of the six channel command words chained into a sequence to execute an operation. (The read-backward channel command is not used.)

TIC	xxxx	1000	Command sequence branch.
Sense	mmmm	0100	Read out sense information.
Control	mmmm	mm11	Seek and non-data operations.
Write	mmmm	mm01	Write and search operations.
Read	mmmm	mm10	Read operations.

The 'm' in positions of the command bytes indicate modifiers for the basic command. These modifiers are used in file operations to define the control or fields that the command is to execute.

The following modifier bit assignments are used to define the file read and write commands to the control unit:

Bit 0	Multi-track operation.
Bit 1	Search high.
Bit 2	Search equal.
Bit 3	Count field.
Bit 4	Key field.
Bit 5	Data field.

Home-address field is defined with bit 3 and bit 4 without bit 5.

Any command that is not defined in the *IFA Commands* chart is invalid and results in posting a command-reject indicator. The command reject is also given along with invalid-sequence indicator when a command is not preceded in a chain by the correct command. The command reject is posted with the file-protect indicator when the command is not allowed by the file mask.

Most of these commands are defined to the file control unit with a sequence of file mini-ops that are developed by the microprogram. These mini-ops and their use are defined under the heading of "Mini-Op Control." The commands are further defined under the group title of "Sense Command," "Control Commands," Write Commands," Read Commands," and "Search Commands."

Command Codes	Single Track Hex	Single Track Binary	Multi-Track Hex	Multi-Track Binary
Transfer-in-Chan	x8	xxxx 1000	--	---- ----
*Sense	04	0000 0100	--	---- ----
Control,				
No Operation	03	0000 0011	--	---- ----
*Set File Mask	1F	0001 1111	--	---- ----
Restore (2321 Cmd)	17	0001 0111	--	---- ----
*Recalibrate	13	0001 0011	--	---- ----
*Seek	07	0000 0111	--	---- ----
Seek Cylinder	0B	0000 1011	--	---- ----
Seek Head	1B	0001 1011	--	---- ----
*Space Count	0F	0000 1111	--	---- ----
Write,				
*Home Address	19	0001 1001	--	---- ----
Record Zero	15	0001 0101	--	---- ----
*Count/Key/Data	1D	0001 1101	--	---- ----
Special CKD	01	0000 0001	--	---- ----
Erase	11	0001 0001	--	---- ----
*Data	05	0000 0101	--	---- ----
Key and Data	0D	0000 1101	--	---- ----
Read,				
Home Address	1A	0001 1010	9A	1001 1010
Count	12	0001 0010	92	1001 0010
*Data	06	0000 0110	86	1000 0110
Key and Data	0E	0000 1110	8E	1000 1110
Count/Key/Data	1E	0001 1110	9E	1001 1110
Record Zero	16	0001 Q110	96	1001 0110
IPL	02	0000 0010	--	---- ----
Search,				
Home Addr Equal	39	0011 1001	B9	1011 1001
*Identifier Equal	31	0011 0001	B1	1011 0001
Identifier High	51	0101 0001	D1	1101 0001
Ident Equal/High	71	0111 0001	F1	1111 0001
*Key Equal	29	0010 1001	A9	1010 1001
Key High	49	0100 1001	C9	1100 1001
Key Equal/High	69	0110 1001	E9	1110 1001
Search (Scan),				
*Key/Data Equal	2D	0010 1101	AD	1010 1101
Key/Data High	4D	0100 1101	CD	1100 1101
Key/Data Eq/High	6D	0110 1101	ED	1110 1101
Continue Scan,				
Search Equal	25	0010 0101	A5	1010 0101
Search High	45	0100 0101	C5	1100 0101
Search Equal/High	65	0110 0101	E5	1110 0101
Not Status Mod	55	0101 0101	D5	1101 0101
Set Status Mod	35	0011 0101	B5	1011 0101
or	75	0111 0101	F5	1111 0101

*These commands are more completely detailed on the following pages.

Transfer-in-Channel (TIC) Command

- The transfer-in-channel command branches the command-chain sequence to a non-sequential address.
- When used with the status-modifier indicator from a successful search, the command is skipped.

The transfer-in-channel (TIC) command provides a means of branching in the command chain. The command is not functional in the file control or in the file but is in effect interpreted by the channel controls. The CCWs are normally taken from sequential addresses starting with the address contained in the CAW. When the TIC command is read in, the data address specified in the CCW replaces the current CCW address and is used to enter the next CCW immediately. The new address can be a previous address to cause one or more of the commands to be repeated. It can also be used to unconditionally branch to a disconnected group of CCWs.

In the file sequence, the TIC command is used to repeat a search sequence until the search argument in main storage is satisfied. A successful search results in the setting of the status-modifier bit that in turn causes the skip of one CCW address. In this case the TIC command is skipped to allow advance to the following read or write command.

The first command of a chained command sequence cannot be the TIC command. The TIC command cannot branch to another TIC command in sequence. In either case the chained command sequence is ended with a command sequence error posted.

MINI-OP CONTROL

- The CCW commands are converted into one or more mini-op codes that execute the operation.
- Mini-ops are sent to the file-control hardware along with a count value and a sync byte as required.
- The value of the count is always twenty greater than the length of the field including the cyclic check to allow trap time.
- The mini-op byte provides the basic operation along with modifiers to effect the desired conditions.

The CCW commands are interpreted by the microroutines to determine the operation to be performed. The data commands that involve reading or writing data on the disk file are encoded into one or more mini-op codes that are sent one at a time to the file control unit. The mini-ops are developed one at a time based on the current command and the previous-op algorithm. When the file control completes a mini-op, a trap is requested to obtain the next mini-op.

When the series of mini-ops to execute one command are completed, the microroutine obtains the next CCW and the operation is continued with the mini-ops developed for the new command. The complete sequence of mini-ops to perform the command-chain sequence may include no-op mini-ops to effect time-out conditions in the sequence.

Each mini-op, when sent to the file control, has a count value to define the number of bytes to be counted. The read-data and write-data mini-ops are also supplied with an appropriate sync byte for the operation. In a read operation the sync byte is compared with the byte being read to identify the field. For the write sequence, the sync byte is written just ahead of the new data. With the exception of a no-op mini-op with a count of zero used to effect a file-control reset, the count value sent to the file control is always greater than twenty. In the cases of the read-data and write-data mini-ops, the count value is the sum of the field length, the four-byte cyclic-check, and the twenty-byte post-field gap. The write-gap mini-op has a count equal to the number of bytes to be written.

The mini-op is executed as specified by the code and the modifiers while the count value is being decremented with each cycle of the bit ring. The ring may be driven by either the read clock or the write clock. When the count has decremented to twenty-four for a data operation, the controls are changed to handle the four-byte cyclic-check. At 'decode 20', a trap is requested for the next mini-op, and the post-field gap is either clocked-through or written, depending on the operation. The microroutine must develop the new mini-op and count value before the counter decrements to one; otherwise an overrun is signaled.

The mini-op codes for IFA operation are set into the FOP external from the external bus-in during the microprogram trap routine. When the execution of the previous operation reaches the 'count 0 gate' condition, the new mini-op is transferred into the MOP register for execution. This sequence is repeated until all of the commands in the chain have been executed.

The bits of the operation registers are defined as follows:

Bit 0 Op Code (Read)
Bit 1 Op Code (Write)
Bit 2 Address Mark (Omit)
Bit 3 Search
Bit 4 Scan
Bit 5 Index Start
Bit 6 Format
Bit 7 Skip

MINI-OP CODES

The two high-order bits of the mini-op define the operation code to the file hardware.

(00) No-Op: This mini-op is issued with a count to effect a time-out. It is issued without a count to effect a file-control reset.

(01) Write Data: This mini-op is issued along with a count to effect the writing of a field. The sync byte supplied defines the field.

(10) Read Data: This mini-op is issued along with a count to effect the reading of a field. The sync byte supplied defines the field.

(11) Write Gap: This mini-op is issued along with a count to write the gap preceding a write-data operation. The sync byte ending the gap is supplied with the following write-data mini-op.

MINI-OP MODIFIERS

The low-order six bits of the mini-op are used as modifiers to the basic mini-op code.

Bit 2 Address Mark (Omit): This bit is used with the write-gap mini-op to block the address mark when the erase command is executed. The bit is also set with the read-data mini-op to block the data-check controls when the space-count command is executed.

Bit 3 Search: This bit is used with the read-data mini-op to execute the search-key and search-KD commands. The search bit gates the compare latches for serial data comparing. The bit is not set for search-HA and search-ID commands because the comparing is performed in the CPU.

Bit 4 Scan: This bit is used with the read-data mini-op to perform the search-KD (scan) commands. The scan bit gates the scan mask byte (FF) that blocks the compare for that byte. The search bit must also be set to gate the comparing.

Bit 5 Index Start: This bit is used to instruct the hardware to wait for the index pulse before starting this mini-op. For example: When the write-HA command is executed, the 'index start' bit in the write-gap mini-op starts the writing following the index point. In the case of a read-HA command, the bit is set with the no-op mini-op to initiate the reading at the appropriate point.

Bit 6 Format: This bit is used with the write-gap and write-data mini-ops to indicate the formatting sequence. The presence of the bit causes the write controls to remain set at the end of the field because the entire track is being written.

Bit 7 Skip: This bit is used with the read-data mini-op to allow clocking over the field without transferring the data. For example: When a write-data command after a search-ID command is performed, the key field must be accurately clocked-through to find the starting point for the data operation.

STATUS AND SENSE INDICATIONS

- Conditions that occur during the execution of an instruction command-sequence are reported as channel status and sense information.
- The status information is set into the CSW at the end of the operation.
- The sense information remains stored in the IFA and requires a sense command to store it in main storage.

All status conditions in the IFA are associated with a specific device address except the control unit end (CUE). The CUE applies to the control-unit base address. A pending status is cleared when any of the attached disk drives are addressed except when a contingent connection exists. The final status for an operation appears as two eight-bit bytes in the channel status word (CSW). When chaining commands, each command in effect returns a status, but it does not enter the CSW unless the conditions require a termination of the operation. A normal channel-end (CE) and device-end (DE) indication without any check indications allows the chaining to continue. Any check indications that appear in the CSW bytes occurred on the last command that was attempted.

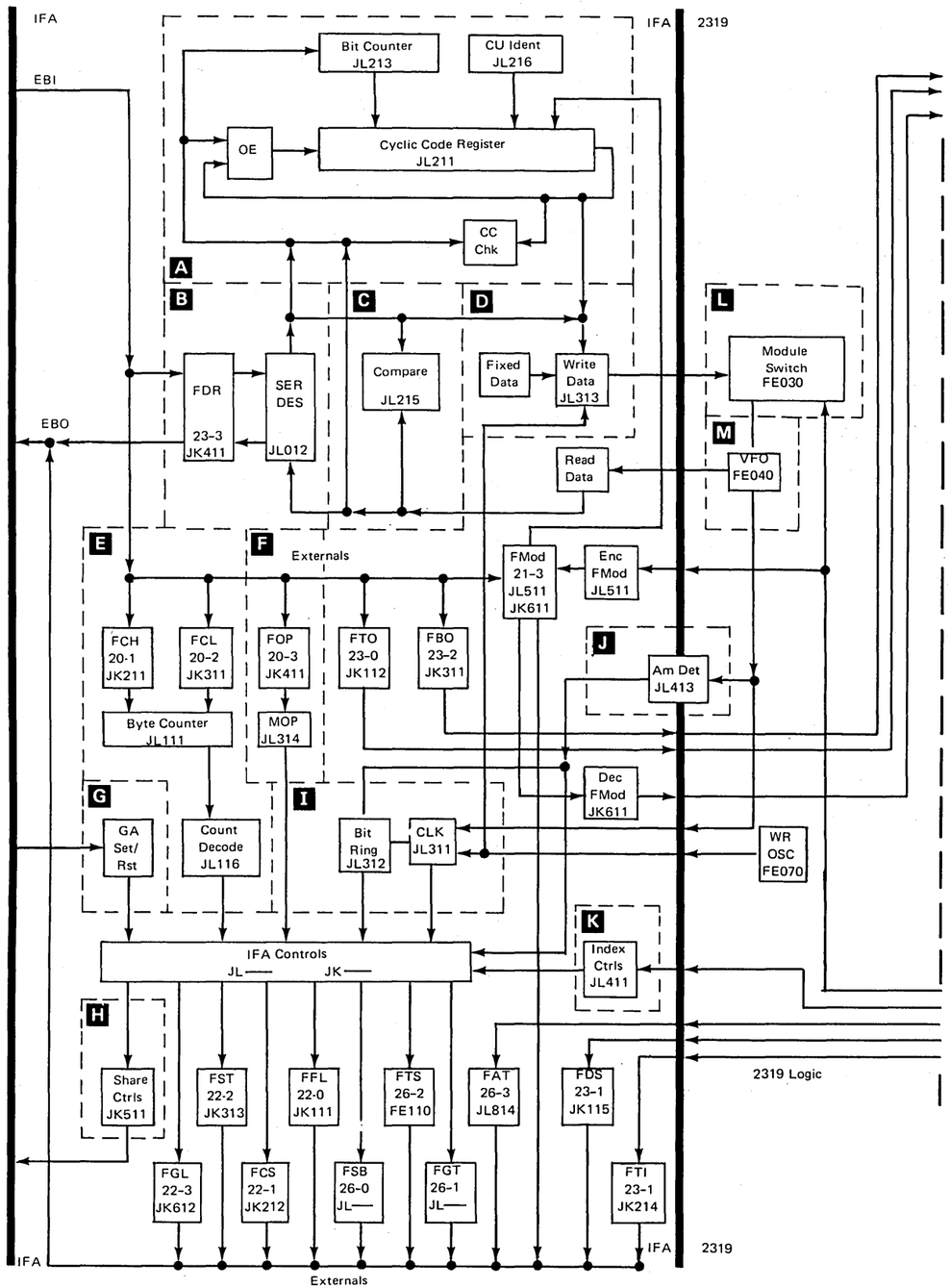
FUNCTIONAL UNITS

IFA DATA FLOW INDEX

The IFA control unit contained in part in the 3145 and in part in the attached 2319-A01 is divided into functional units in this section of the manual to aid in understanding. The control-unit section of the IFA composite data flow has been subdivided to show these areas. There are a few areas that do not lend themselves to a functional area discussion. In most cases the parts associated with other areas are included in that area. The externals and other major storage assignments are included as an extension of this heading to allow easy association in the following functional-unit discussions.

The following reference list defines the heading names for the indicated functional units:

- A Cyclic-Check Controls
- B Data-Transfer Controls
- C Compare Circuits
- D Write-Clock/Data Generation
- E Field-Count Controls
- F Operation Registers
- G IFA Set/Reset Functions
- H Share-Request Controls
  Share-Cycle Controls
- I IFA Clock and Bit Ring
- J Address-Mark Detection
- K Index Controls
- L Data Module Switch
- M Variable Frequency Oscillator



Data Flow Index

IFA EXTERNAL ASSIGNMENTS

The IFA external registers are used to transfer information between the IFA hardware and the CPU. The transfer of information to and from these registers follows the same paths as for other externals. Entry to the registers is from EBI lines from the CPU. The registers are read out over the EBO bus to the CPU.

Four of these external words are the normal assignments for channel 1, and the fifth word is normally assigned to channel 4. Their mnemonic names have been changed slightly to reflect the IFA.

Word 26 byte 3 (FAT external) has in addition to its normal gated attention assignment, a multiple use in diagnostic mode. The normal FAT information is blocked, and seven different bytes can be gated to read out as address 26-3. These diagnostic bytes are gated by setting the binary address of the desired byte in bits 5, 6, and 7 of the FBO external while in diagnostic mode.

Diagnostic Address 0	
0	No-Op
1	Read Data Op
2	Write Data Op
3	Write Gap Op
4	Compare Read Data
5	Read Buffer 0
6	Read Buffer 7
7	Serial Data

Diagnostic Address 1	
0	Counter Pos 128
1	Counter Pos 64
2	Counter Pos 32
3	Counter Pos 16
4	Counter Pos 8
5	Counter Pos 4
6	Counter Pos 2
7	Counter Pos 1

Diagnostic Address 2	
0	Count 0 Gate
1	Count Decode 1
2	Count Decode 2
3	Count Decode 3
4	Count Decode 7
5	BCA Time
6	Data Gate
7	Orientation Latch

Diagnostic Address 3	
0	Count Decode 20
1	Count Decode 21
2	Count Decode 22
3	Count Decode 23
4	Count Decode 24
5	Count Decode 25
6	Count Decode 26
7	Count Decode 15

Diagnostic Address 4	
0	Read Gate
1	Write Gate
2	Wr Clock Gate
3	Standard Index
4	Block Clock Bits
5	Serialized Data
6	Write Sync Gate
7	Write Zero

Diagnostic Address 5	
0	Bit Ring 3
1	CC Register Pos 0
2	CC Register Pos 15
3	BCA Position 1
4	BCA Position 128
5	Address Mark 1
6	Read Sync Gate
7	Head Condition

Diagnostic Address 6	
0	IFA High Trap Req
1	IFA Low Trap Req
2	Force Trap Bit-4
3	Force Trap Bit-5
4	Error Timeout
5	Control Tag A
6	Spare
7	Spare

Diagnostic Address FAT External Bit Assignments

Word	Byte-0	Byte-1	Byte-2	Byte-3
20 FBAK	FWB	FCH	FCL	FOP
Display Lower Roller Pos 8	0 Spare 1 Spare 2 Spare 3 Spare 4 Entry 5 (Not Used) 6 7	0 32K 1 16K 2 8K 3 4K File 4 2K Counter 5 1048 High 6 512 7 256	0 128 1 64 2 32 3 16 File 4 8 Counter 5 4 Low 6 2 7 1	0 *Op Code (Read) 1 *Op Code (Write) 2 Address Mark (Omit) 3 Search 4 Scan 5 Index Hold 6 Format 7 Skip
21 FCND	FDS	FHC	FED	FMOD
Display Lower Roller Pos 7	0 Busy 1 On-Line 2 Un-Safe 3 Spare 4 Pack Change 5 End of Cylinder 6 Multi-Module Seltd 7 Seek Incomplete	0 1 2 3 Diagnostic, 4 Head/Cylinder 5 (CE Panel) 6 7	0 1 2 3 Diagnostic 4 Error Display 5 (CE Panel) 6 7	0 Module Select Gate 1 2 Module 3 Select 4 Module not Selected 5 Physical 6 Module 7 Selected
22 FSTAT	FFL	FCS	FST	FGL
Display Lower Roller Pos 6	0 Chain Data 1 Command Chain 2 Supp Length Ind 3 Skip 4 Spare 5 Input 6 Cntl Store Cnt Rdy 7 Output	0 Prog Ctrld Intrpt 1 Incorrect Length 2 Program Check 3 Protection Check 4 Channel Data Check 5 Channel Ctrl Check 6 Interface Ctrl Check 7 Spare	0 Channel Busy 1 Interrupt Latch 2 Control Unit End 3 Control Unit Busy 4 Block CE Mode 5 Spare 6 Spare 7 Spare	0 Main Store Cnt Rdy 1 Count Zero 2 FDR Full 3 Interrupt Condition 4 Share Cycle Error 5 Retry Code 0 6 Retry Code 1 7 Retry Code 2
23 FTAG	FTO	FTI	FBO	FDR
Display Lower Roller Pos 5	0 Set Cylinder Tag 1 Set Head Tag 2 Set Difference Tag 3 Control Tag 4 CUA Load 5 Spare-A 6 Spare-B 7 Spare-C	0 128 1 64 2 32 3 16 Cylinder 4 8 Address 5 4 Register 6 2 7 1	0 128; (Write Gate)* 1 64; (Read Gate)* 2 32; Seek Start 3 16; Rst Head Req 4 8; (Erase Gate)* 5 4; Select Head 6 2; Return to 000 7 1; Head Advance	0 1 2 3 File 4 Data 5 Register 6 7
26 FERR	FSB	FGT	FTS	# FAT
Display with Address Switches	0 CC Hardware Error 1 Track Overrun (Wr) 2 Bus-Out Parity Ck 3 Ser-Des Check 4 Data Check 5 Data/Cmd Overrun 6 Missing Adr Mark 7 Write Current Err	0 Command Overrun 1 Erase Gate 2 High Compare 3 Low Compare 4 Error Timeout 5 Selected Gated Attn 6 Contingent Connect 7 Spare	0 Test Sel Sw 0 1 Test Sel Sw 1 2 Test Sel Sw 2 3 Test Sel Sw 3 4 CE Error Disable 5 CE Mode Latch 6 Allow CE Mode Sw 7 Gated Attn Spare	0 Gated Attention 7 1 Gated Attention 6 2 Gated Attention 5 3 Gated Attention 4 4 Gated Attention 3 5 Gated Attention 2 6 Gated Attention 1 7 Gated Attention 0

External Word And Bit Assignments For IFA

\# Diagnostic Address bytes are also gated to the FAT external.

* The file head control gates are transmitted in this byte but are not displayed.

* Mini-Op Codes: 00=No-Op 01=Write Data 10=Read Data 11=Write Gap

IFA Local-Storage Assignments

The IFA has eight local-storage words assigned for its use. These are the four words normally assigned to channel 1 and the four words normally assigned to channel 4. These words function as the operating UCW area for operating with main storage and for operating with control storage. The remainder of the words are used for work areas. Four bytes of the work area have bit assignments. The local-storage assignment chart defines the word use and the bit assignments of the four bytes.

Word Addr	Word Name	Word Assignments Byte-0	Byte-1	Byte-2	Byte-3
28	FD	Protect Key	Main Storage Data Address		
29	FC	Flag	CCW Op	Main Storage Count	
2A	FM	Protect Key	CCW Address		
2B	FM	Unit Address	Prev Op Algm	File Mask Algm	Byte Read Area
2C	FA	Cylinder No	Head Number	Control Storage Addres	
2D	FB			Control Storage Count	
2E	FS	Work Area (R)	Work Area (KL)	Work Area (DL)	Work Area (DL)
2F	FL	Mini-Op Link Word			

BYTE DETAIL

FC0	Flag	FC1	CCW Op	FW1	Prev Op Algm	FW2	File Mask Algm
0	Chain Data	0	Multi-Track	0	Rd or Sch HA	0	Inh Set FM
1	Com Chain	1	Search Hi	1	Wr or Sch HA	1	Allow Wr HA, R0
2	SLI	2	Search Eq	2	Allow Wr Data	2	Inh Wr Count
3	Skip	3	Count	3	Allow Wr KD	3	Inh Wr K and D
4	PCI	4	Key	4	Allow Wr CKD	4	Inh Seek, Recal
5	Zero	5	Data	5	Search ID	5	Inh Seek Cyl
6	Cyl Overflow	6	Read	6	Search Key	6	Inh Seek Head
7	Zero	7	Write/Search	7	Rd C or Sch ID	7	Index Passed

IFA Local Storage Assignments

IFA Control-Storage Assignments

The IFA makes use of twelve words of control storage to retain operating information. Four words starting with address F900 are used to save logout and operating information for the interrupt operation (interrupt buffer). Four words starting with address FFA0 are used for work area, CAW backup, IFA identification, and the first four sense bytes. Four words starting with address FFF0 are used for the read/write area for the home address and the count fields. The last word is used as an overflow link word for return information in record-overflow operations. The IFA has exclusive use of these words.

Word Addr	Word Assignments Byte-0	Byte-1	Byte-2	Byte-3
F900	FTAG Register Save Area			
F904	Interrupt Buffer			
F908	Interrupt Buffer			
F90C	Interrupt Buffer			
FFA0			Index Trap Link for FL	
FFA4	I-Cycles CAW Backup			
FFA8		FF85 Save Area	CU # - Drive #	CU # -Max Dr #
FFAC	Sense-0	Sense-1	Sense-2	Sense-3
FFF0	Bit 0=Seek Dir		Head Save Area	Flag
FFF4	Cylinder No.	Cylinder No.	Head No.	Head No.
FFF8	Record No.	Key Length	Data Length	Data Length
FFFC	Record Overflow Link Word			

IFA Control Storage Assignments

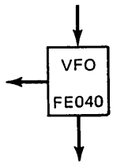
IFA GA Set/Reset Functions

The IFA has a group of set/reset functions under control of the microroutine GA-address. These controls work the same as the set/reset controls for selector channel and with the same addressing. The selection of the control group is made by setting the IFA channel gate (GA, OR, 10). With any other channel gate, the selector-channel functions are set. The terms *set* and *reset* define the OR and A- logic functions used in their control. Either function can be used to set or reset a control function.

h	Set GAL	Reset GAL	Set GAH	Reset GAH
K Field	GA, OR, 0h	GA, A-, 0h	GA, OR, h0	GA, A-, h0
1	Set Inc Length	Reset FCS	Set IFA Chan Gate	Rst Command Overrun
2	Set Prog Check	Reset PCI	Set Channel 2 Gate	
3	Set Prot Check	Rst Trap Req	Set Channel 3 Gate	Machine Reset
4	Set Chan Ctrl Chk	Rst CCW 0 & WLR		
5	Set Allow Restart	Rst Low Prior Req	Set Write Clk Gate	Rst Orientation Lch
6	Set Trap Taken	Chain End Reset	Set CS-CR, In Lchs	Rst Cnt Rdy, In, Out
7	Set Contingent Con	Rst Contingent Con	Set CS-CR, Out Lchs	Set Halt I/O
8	Set Allow D/A		Set MS-CR, In Lchs	Set CE End-Op SS
9	Set Chan Busy	Reset Chan Busy	Set MS-CR, Out Lchs	Diag Index
A	Set Intrp Latch	Rst Intrp Latch	Set Control Pulse	Diag Raw Data Pulse
B	Set CUB	Reset CUB	Diag Read Data	Set Diag Read Gate
C	Set Block CE Mode	Rst Block CE Mode	Diag Clk Gap Sense	Bit Ring Advance
D	Set Low Prior Req	Rst H/L Comp, CC Er	Diag Data Gap Sen	Set Diag Mode Latch
E	Set IFA Inh Traps	Rst IFA Inh Traps	Set Data Field Lch	Rst Diag Mode Latch
F				

Set/Reset Controls For IFA

VARIABLE-FREQUENCY OSCILLATOR (VFO)

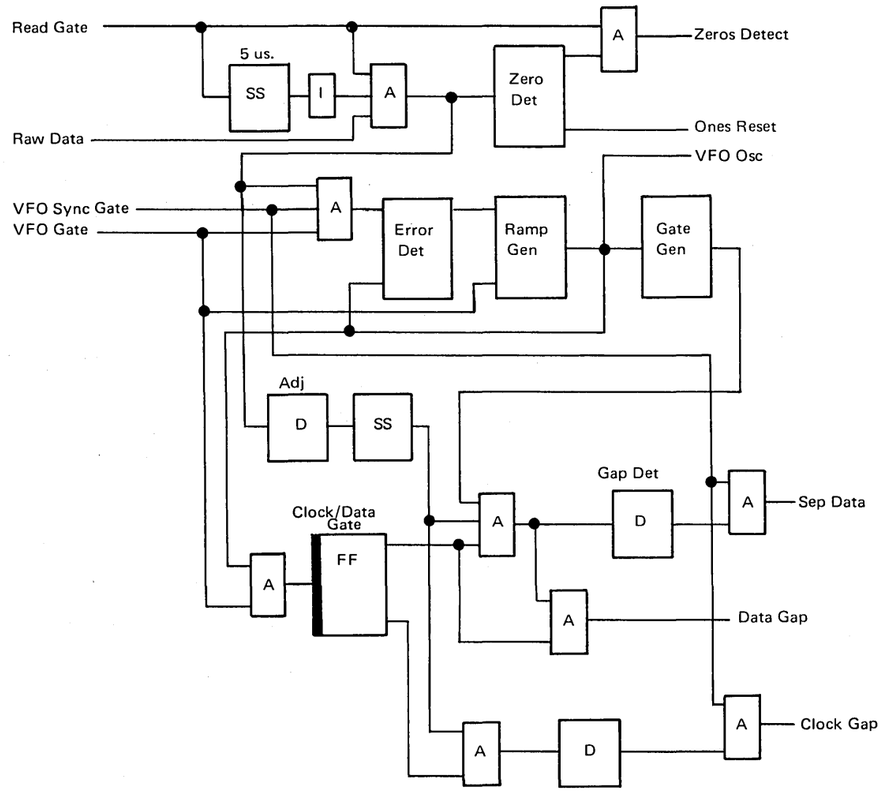


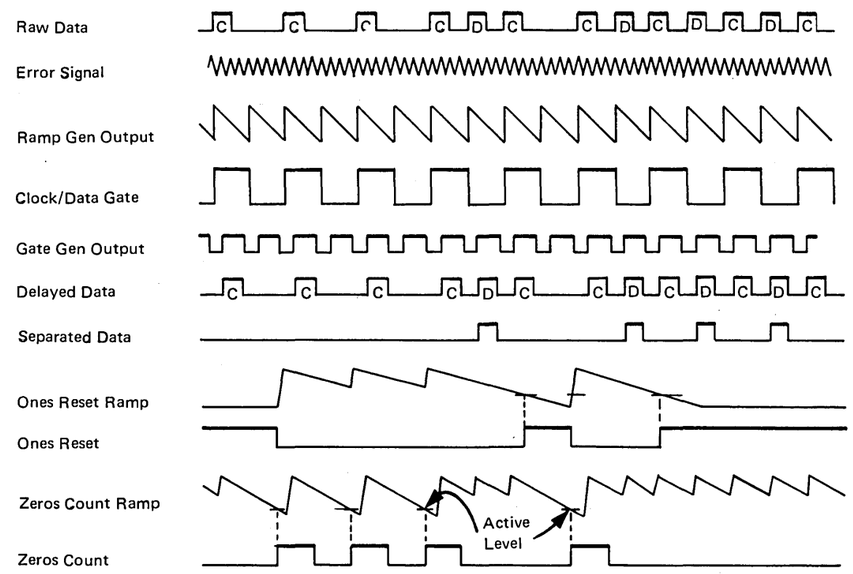
- The IFA uses a VFO synchronized with the file read data to separate the written clock and data bits.
- The VFO control circuits function with the search-AM routine to synchronize the bit ring.
- The VFO control circuits recognize the address mark and sync byte during the search-AM routine.

The IFA control unit makes use of a variable frequency oscillator (VFO) for its read oscillator to drive the clock. The VFO offers more consistent data reading conditions than a reading system using the separated read-clock bits because a missing bit condition is filled by the VFO pulse. The oscillator has a nominal free-running frequency equal to the write frequency (5 MHz).

An error-detection circuit compares the VFO timing and the 'raw read data' entering from the file to determine the relative phase of the pulses and develop an error signal to correct the VFO frequency. The error correction is damped to prevent correction for individual missing or displaced pulses from the file.

The VFO circuits also develop signals used in the search-address-mark sequence to start the VFO and the bit ring. The search routine and the synchronization of the VFO are discussed in greater detail under the heading of "Address-Mark Detection."





VFO Circuits: The VFO oscillator is a ramp generator that develops a decaying sawtooth output. When the error bias line is held at the reference level, the output has a nominal frequency of 5 MHz. Changing the error bias above or below the reference raises or lowers the output frequency. The ramp output is fed to the 'error detector', the 'VFO trigger', and the 'gate generator'.

Error Detector: The error-detect circuit compares the difference in phase and frequency between the output of the VFO and the incoming 'raw read data' from the file. Any variation in phase between the two signals develops an error signal that feeds the VFO. The difference may be either positive or negative with the error signal shifting in respect to a dc reference line. The final error signal must cause the VFO to operate at the approximate frequency of the incoming raw data. The phase angle between the two signals entering the error detector must remain great enough to develop the error signal. The error signal is allowed to hunt slightly to provide a faster correction time. It is dampened sufficiently to prevent changing the level for missing bit conditions.

VFO Trigger: After the VFO has been synchronized and has been operating in a single-frequency area, the output of the VFO is gated to drive a binary trigger. Starting the trigger from the reset position with a pulse that has been defined as a clock bit defines the set output of the trigger as clock-bit time, and the reset output of the trigger as data-bit time. These outputs are used to gate the bits of the 'raw read data' line into separated clock and data bits.

Gate Generator: To provide a definite gating period for the data bits being sampled from the raw-read-data line, a gate signal is developed that lies centered within the VFO trigger data gate. This sample gate is developed from the ramp output of the VFO with appropriate delay and singleshot circuits.

Read Data Delay and Singleshot: The gating developed to sample and separate the bits of the raw-read-data input have been delayed in their development. In order to sample a clock or data bit with the developed gates, the raw data must be delayed. An adjustable delay line provides the means to delay the bits until they fall within the established gates. The delayed input is fed through a singleshot to shape the incoming bits to be gated.

Separated Clock/Data: The separated-clock bits are obtained from the delayed raw-data input by gating with the clock-gate output of the VFO trigger. The separated-data bits are obtained by gating with both the data-gate output of the VFO trigger and the output of the gate generator.

Gap Sensor: Two signals are developed in the gap sensor that are used to determine the presence of the address-mark bytes and the sync byte during the search-AM routine. The separated-clock bit line is sampled to find a period of greater than 1.6 us. without a clock bit. A time-out defines the five missing clock bits of the address-mark byte and develops the clock-gap-sense line to signal the search routine. The separated-data bit line is sampled to find a period of greater than 1.4 us. without a data bit. A time-out defines the four missing data bits of the sync byte and develops the data-gap-sense line to signal the search routine. This condition causes the bit ring to start advancing with the next clock bit.

Zeros Detector: The raw-read-data line is sampled separately by the zeros detector to develop two lines that are needed in the early part of the search-AM routine to determine when to sync the VFO. The zero detector can differentiate between single- and double-frequency recording. An output is developed on the 'zeros count' line for each clock bit that is not followed by a data bit (single frequency). The 'zeros count' pulses are counted for a period of nine bits to allow the VFO time to synchronize before setting the VFO trigger. If a data (one) bit is detected (double frequency), the ones-reset line is developed to stop the count. The reset line is held with a trigger that remains set until the next zero is detected.

VFO Adjustment Procedures

1. This procedure uses the write oscillator for adjusting the VFO.
2. The 2319 dc voltages should be checked before these adjustments are made. See the "2319 Installation Instructions," Section E.
3. All oscilloscope probes must be X10 and the scope grounded.
4. All adjustments and test points are in the 2319 frame A1.
5. Use IFA microdiagnostic routine SBA0 to operate the system.

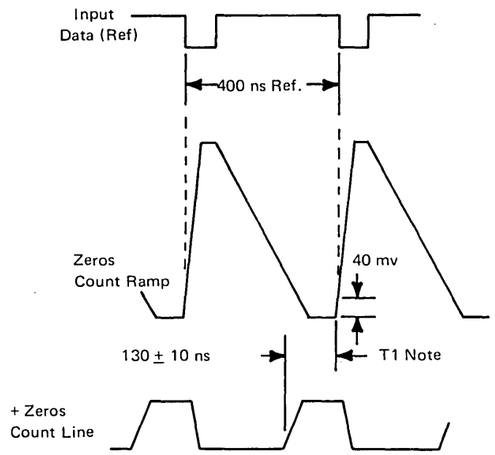
Write Oscillator: The write oscillator, located in the 2319 frame, is used as a reference for adjusting the VFO. Measured at 01A-A3C6B02, the plus pulse width at the 50% level must be 100 ± 15 nanoseconds. The pulse cycle must measure 200 ± 10 nanoseconds.

Read Gate Singleshot: Externally sync scope on '-read gate A' A3D7D12 and observe A3D7D04. Adjust the potentiometer on the A3D7 card so that the plus level sweep occurs within 5 ± 0.2 microseconds.

Error Detector: With the file control in the reset condition, place the negative or common lead of a dc voltmeter (20,000 ohms/volt with an isolated ground) on pin A3J4J07. With the positive voltmeter lead on pin A3J4J04, adjust the 200-ohm potentiometer on the A3J4 card for 1 ± 0.05 volts.

Zero Detector:

1. Add jumper between pins A3L4B02 and A3K6D12. Add jumper from pin A3H6D10 to pin A3D6D02.
2. Externally sync the scope on '+ zeros count' at pin A3H4B04 and set sweep to 50 ns/cm and vertical sensitivity to 2 v/cm.
3. Observe the scope in alternate sweep mode with the 'zeros count' line at pin A3G6B03 and the 'zeros count ramp' at pin A3H4B05.
4. Adjust the 10K-ohm potentiometer on the A3H4 card to obtain the relationship shown in the diagram.

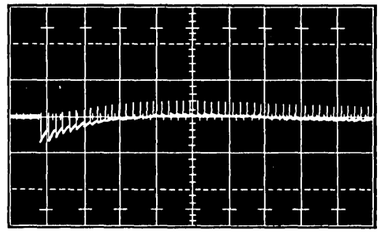


Note: Measured from the 10% point of the Zeros Count line rise time to 40 mv from the 0% point of the Zeros Count Ramp fly-back time.

Zeros Detector Adjustment

Ramp Generator:

1. Before this adjustment is made, the error-detector and the zeros-detector adjustments must be correct.
2. Internally sync the scope and monitor the '+VFO trigger' at pin A3F4B09. Adjust the 1000-ohm potentiometer on the A3F4 card for a pulse width of 200 ± 25 nanoseconds.
3. Externally sync the scope on the 'VFO gate' at pin A3F4B03, set the scope main sweep at 2 us/cm, and set the vertical sensitivity to 0.1 V/cm.
4. Monitor the 'error signal' at pin A3J4B08 and adjust the 1000-ohm potentiometer on the A3F4 card for minimum error signal.



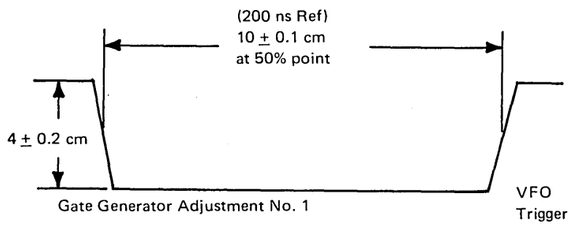
Minimum Error Signal

5. When this is done, the VFO nominal frequency is equal to the system nominal frequency. Verify the adjustment by rechecking step 2 and if incorrect, repeat steps 2 through 5.

Data Separator: The data separator is a function of the proper timing between the output of the 'gate generator' and the 'raw data' delayed by the delay line. After the initial adjustment, the adjustment of the delay line and/or the gate generator is made only when a VFO card is changed. Before adjustments are made, the ramp-generator and error-detector cards must be properly adjusted.

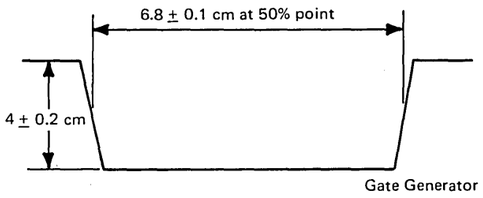
Gate Generator:

1. Externally sync the scope main sweep on '+VFO trigger' at pin A3F4B09 and observe the 'VFO trigger' at pin A3G4B05 (use B probe). Adjust the vertical sensitivity so that the signal covers 4 ± 0.2 cm and adjust the sweep so that the negative swing of the trigger covers 10 ± 0.1 cm as shown in the diagram.



Gate Generator Adjustment No. 1

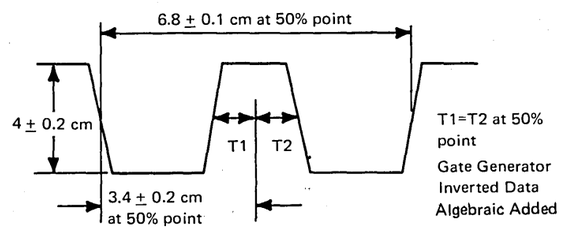
2. Without changing the scope sweep setting, monitor the 'gate generator' at pin A3G4J06. Adjust the vertical sensitivity so that the signal covers 4 ± 0.2 cm. Adjust the bottom potentiometer of the A3G4 card until the leading edge of the signal stops or until the end of the potentiometer is reached. Then adjust the top potentiometer on the same card until the signal covers 6.8 ± 0.1 cm reference as shown on the diagram.



Gate Generator Adjustment No. 2

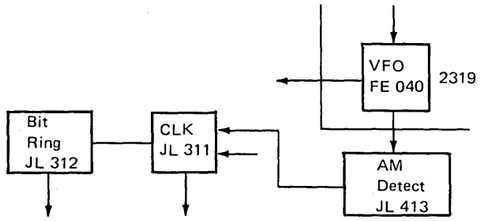
Data Delay Line:

1. Program the delay line at A3L5 for 40 nanoseconds. This is an initial adjustment.
2. Without changing the sweep setting of the scope, observe in algebraic add mode the output of the 'gate generator' at pin A3G4J06 (use probe A) and delayed data at pin A3G4G05 (use probe B). Invert the data input and set the vertical sensitivity so that the signal covers 4 ± 0.2 cm.
3. Select the correct delay taps on the A3L5 card to obtain the relationship shown in the diagram. The scope sweep calibration should be about 20 ns/cm.
4. Remove all jumpers added for the VFO adjustment.



Delay Line Adjustment

ADDRESS-MARK DETECTION



- The VFO/AM area is used to synchronize the hardware controls to the written information field.
- The four parts of the VFO/AM area must be identified before the data field can be read.
- If any part of the VFO/AM area is not recognized, the sequence is normally restarted.
- If the hardware controls have been oriented, reading the wrong sync byte sets the data-check error.

The address mark is used with the sync byte to synchronize the clocking controls with the written field data. The clocking controls are stopped after each field during a read operation. Under the disoriented condition, the search for the eight-byte VFO/AM area is started at random. The entering data from the addressed track of the selected file may be from a data area, a gap area, or from an VFO/AM area. The search for a VFO/AM area, after the file has been oriented, is started just ahead of the VFO/AM area.

The sequence of the eight bytes in the VFO/AM area is:

1. Four bytes of '00'.
2. One byte of 'FF'.
3. Two bytes of address mark.
4. One sync byte.

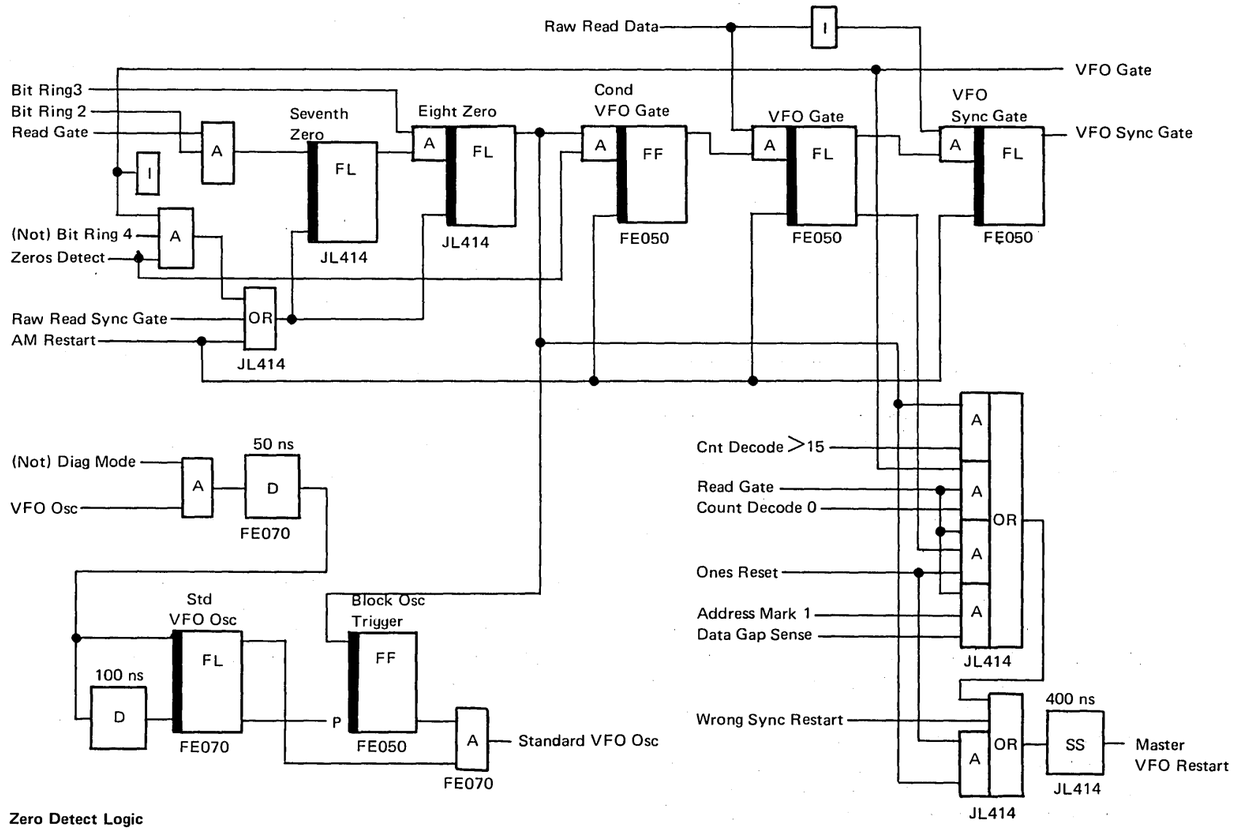
To locate appropriate information, the search must identify each of the four areas. The search involves starting and synchronizing the variable frequency oscillator (VFO) and then synchronizing the bit ring. The sync byte being read is compared with the byte presented by the microprogram.

00 Bytes: After setting the 'read gate', the sequence starts by searching for nine consecutive data-bit zero levels (single-frequency recording). This identifies clock bits to synchronize the start of the VFO. The zeros are counted by advancing the bit ring with the zeros-detect pulse from the VFO controls. The count of nine requires that the bit ring advance beyond its full cycle. A 'seventh zero' latch is set at BR-2 to indicate the end of the first pass. An 'eighth zero' latch is used to indicate that the sequence of eight has been found. The ninth zero is detected within the VFO circuits in the 2319 to ensure accurate setting of the 'VFO gate' latch. With the 'eighth zero' latch set, the next 'zeros detect' pulse sets the 'condition VFO gate' to allow the 'VFO gate' to set with the next 'raw data' pulse.

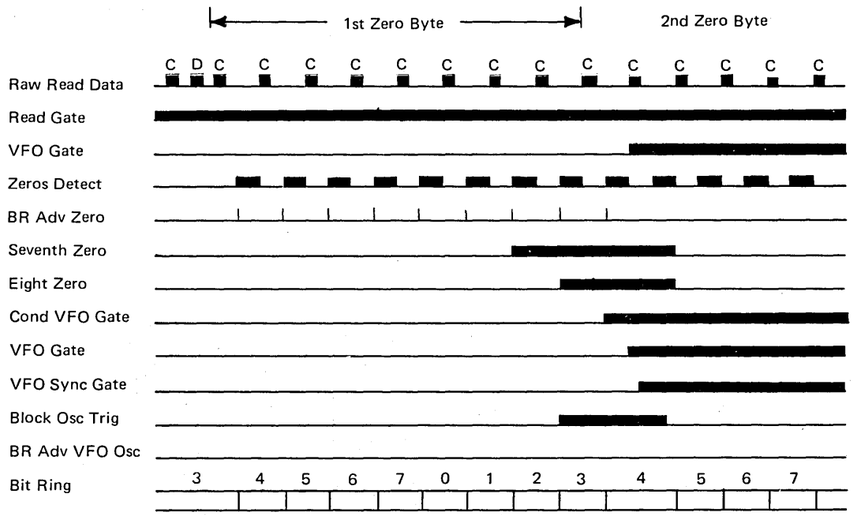
If the sequence of nine zeros is broken, the presence of a data 1-bit resets the bit ring to restart the 0-bit search. When the nine zero bits have counted, the next bit read sets the 'VFO gate' latch. The fall of the bit being read sets the 'VFO sync gate' latch. The output of the 'VFO gate' latch releases the VFO oscillator and the reset clamp on the VFO trigger, starting the VFO with the clock bit. The 'VFO sync gate' latch output gates the 'raw read data' input to the error-detector circuit and to the data-separation gates.

The byte counter is set to a value of eight after the zero bytes (9 zeros) are identified. The output of the VFO drives the bit ring in the normal manner to decrement the count. The remainder of the sequence must be completed within the eight-byte timing, or the sequence is restarted. The output of the 'VFO restart gate' latch gates the counter output to the restart controls.

FF Byte: The byte of 'FF' provides double-frequency recording to establish the clocking frequency before entering the address-mark bytes. The error-detector circuit adjusts the error signal level to obtain synchronism between the input data and the VFO. The gap sensor can now detect a missing bit condition of either clock or data bits. No actual test is made for an FF-byte.



Zero Detect Logic



VFO Zeros Clocking

AM Bytes: The two bytes of address mark each have five missing clock bits. The 'clock gap sense' circuit has a time-out of about three periods. If a clock bit occurs within this period, the time-out is restarted. When the time-out condition occurs for the first time, the 'address mark 1' latch is set if the count decode is 3. The second time-out condition causes the 'address mark 1' latch to reset and the 'address mark 2' latch to set if the count decode is 2. When the 'clock gap sense' signal occurs without the count decode-1 or -2, the AM triggers are not set and the search is restarted. If for any reason a third time-out condition occurs, the 'address mark 1' latch would set again and the next data gap sense causes a restart. The output of the 'address mark 1' latch blocks the recognition of the sync byte if other than two address marks are recognized.

Sync Byte: The sync byte following the second address mark byte has four missing data bits. The data-gap sense has a time-out of about three periods. If a data bit occurs within this period, the time-out is restarted. When the time-out condition occurs, the sync byte has been detected. The bit ring (reset to BR-3) is advanced to BR-4 with the next data pulse. Five sync bytes allow identification of all fields:

1. 0D = Home Address
2. 0B = R0 Count Field
3. 0E = Rn Count Field
4. 0A = All Key Fields
5. 09 = All Data Fields

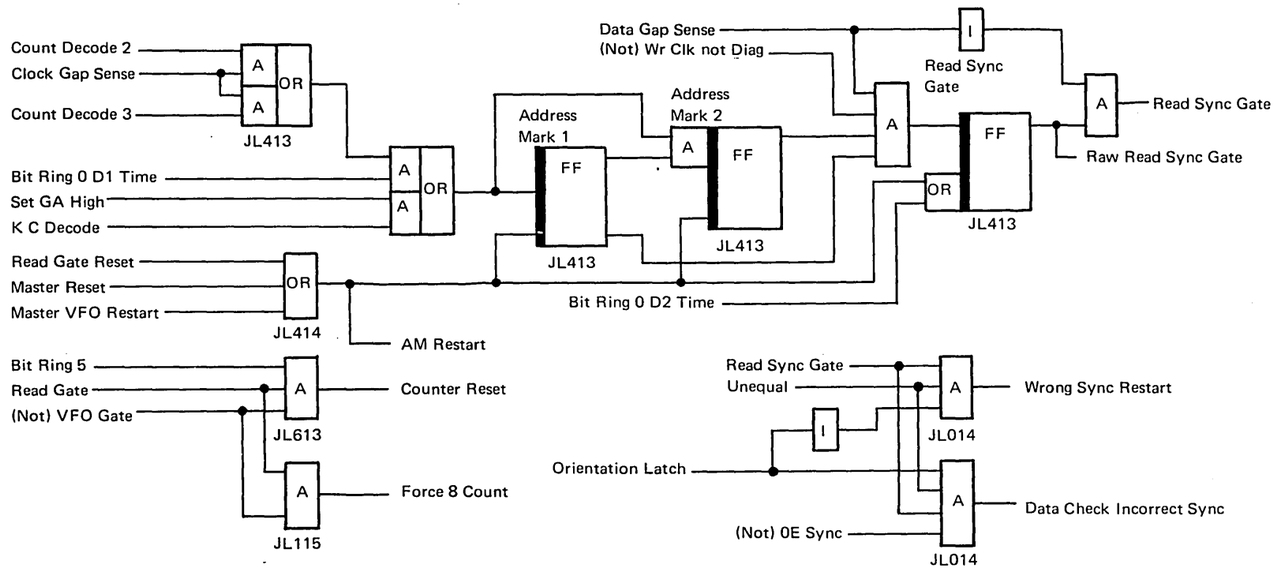
The last three bits of the sync byte are compared with the low-order three bits of the sync byte in the SERDES. If they compare, the field is read for the data operation. When the sync bytes differ, the address-mark search is reinitiated if the file is not oriented. When the file has been oriented, the wrong sync byte indicates that a field has been missed and results in setting the data-check indication.

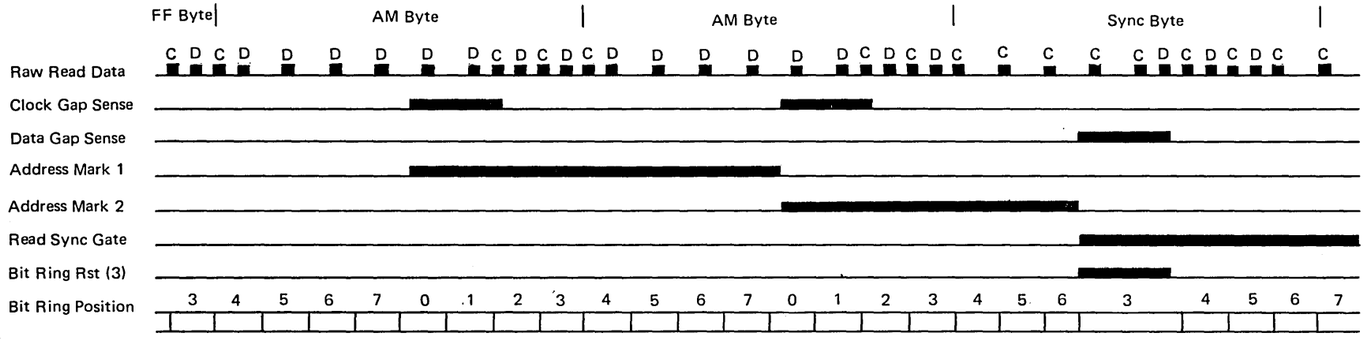
RESTART CONDITIONS

Failure to recognize each part of the VFO/AM area in sequence causes the search sequence to be restarted. The restart conditions include the following:

1. Failure to find nine (9) sequential zero-bits.
2. Recognition of only one address-mark byte.
3. Address-mark bytes occurring during the wrong byte count decode.
4. Wrong sync byte when the hardware controls are not oriented.
5. AM-search time-out (byte count = 0).

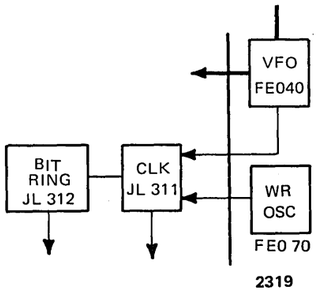
When a restart condition occurs, the master VFO restart signal is developed. This signal resets all of the search sequence latches, the byte counter, and the compare circuits. The search is reinitiated, starting with the zero bits after the reset line falls.





Address Mark Detection

IFA CLOCK AND BIT RING



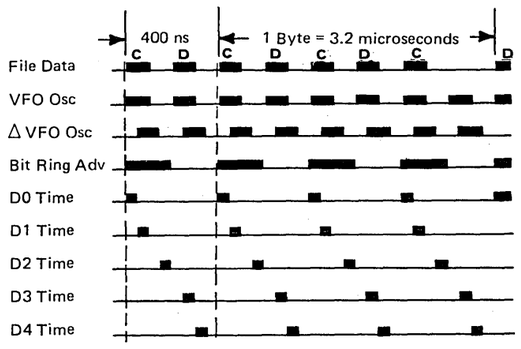
- The clock and bit ring control data gating for both read and write operations.
- The clock is driven from the write oscillator for write operations and from the VFO oscillator for read operations.
- The bit ring is advanced by a pulse developed in the clock circuit.
- When the bit ring is not in use, it is reset to the bit-3 position.

The timing control for file operation contains a clock circuit for bit control and a bit ring circuit for byte control. The same circuits are used for both read and write operations. During write operations, the clock is driven from the fixed-frequency write oscillator. For read operations the 'standard VFO oscillator' signal, synchronized to the incoming data, drive the clock. Either the write oscillator or the 'standard VFO oscillator' is driving the clock during all periods that the control unit is oriented except the VFO synchronization. The clock is also driven during No-Op time-outs with the write oscillator. The bit ring is advanced by a pulse developed from the clock output. Each cycle of the bit ring represents a byte period, and its output advances the byte counter. The clock (driven by the write oscillator), bit ring, and the byte counter are also used for time-outs.

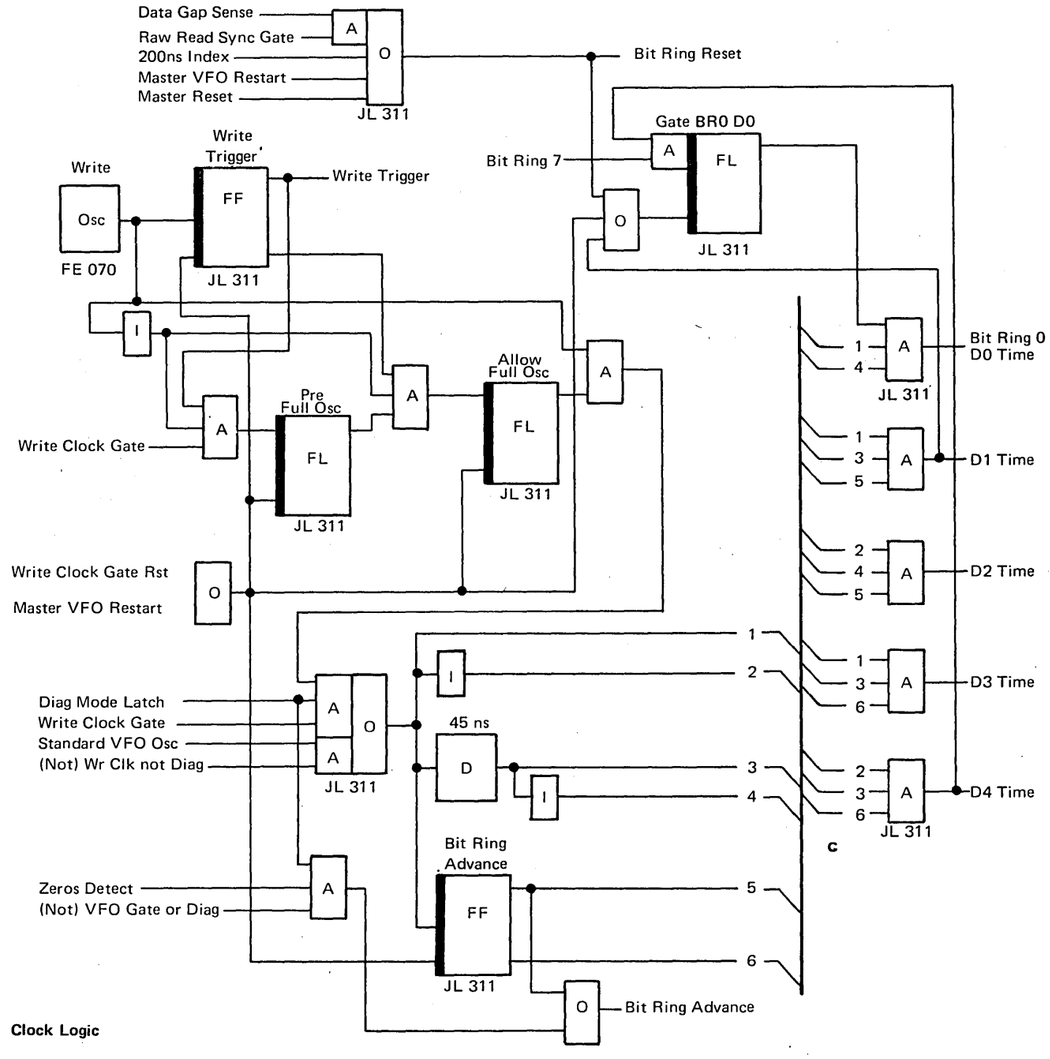
Clock

The write-oscillator and standard-VFO-oscillator lines used to drive the clock are double-frequency drives. The clock consists of a delay circuit to produce a 'delta clock' pulse and the 'bit ring advance' latch to provide single-frequency drive. Five pulses are developed by gating the clock input with these two signals.

The clock outputs are designated D0, D1, D2, D3, and D4. The D0-time pulse is developed only at bit-ring-0 time to gate controls that define the start of a new byte. The remaining four pulses are used in combination with bit-ring outputs to control sequential functions required for bit and byte handling.



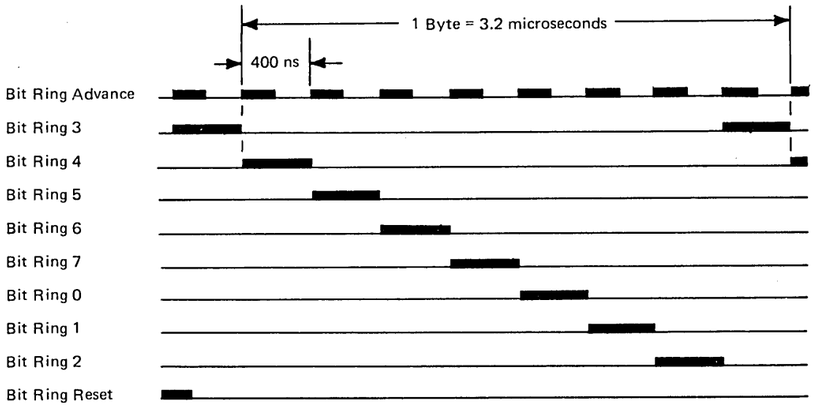
IFA Control Unit Clock



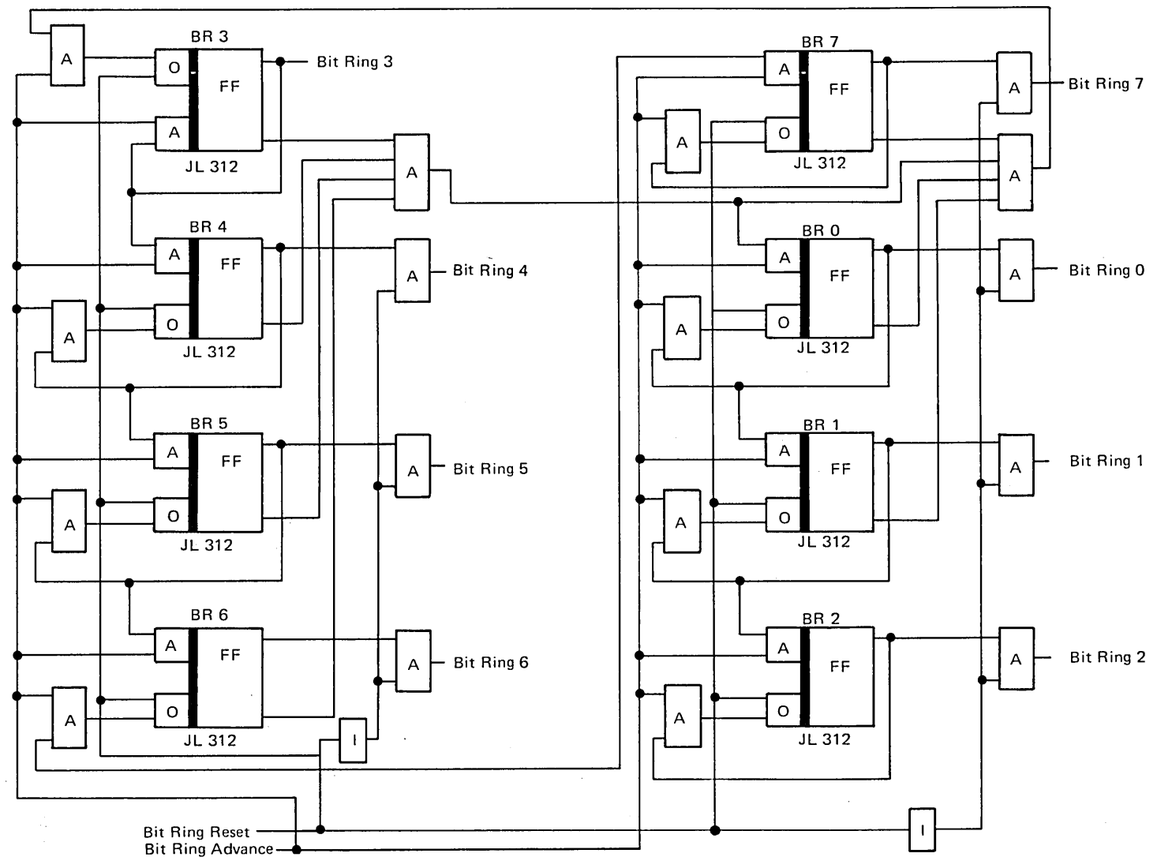
Clock Logic

Bit Ring

The ring consists of eight triggers connected to set one at a time in sequence. The rising output of the 'bit ring advance' latch in the clock circuit drives the ring. The ring is reset to the BR-3 position and started by advancing to the BR-4 position when the data-gap of the sync byte is read. The ring has advanced to the BR-0 position by the time the first data bit is read or written. The ring makes one cycle for each data byte. The bit ring is also used to count the nine zeros at the start of the address-mark search. During the search the ring is advanced with the 'zeros detect' signal.



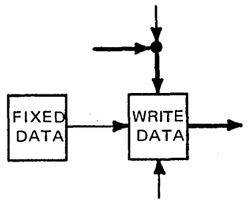
Bit Ring Timing



Bit Ring Logic

Bit Ring resets to position BR 3. Byte time is from BR 0 through BR 7.

WRITE CLOCK/DATA GENERATION



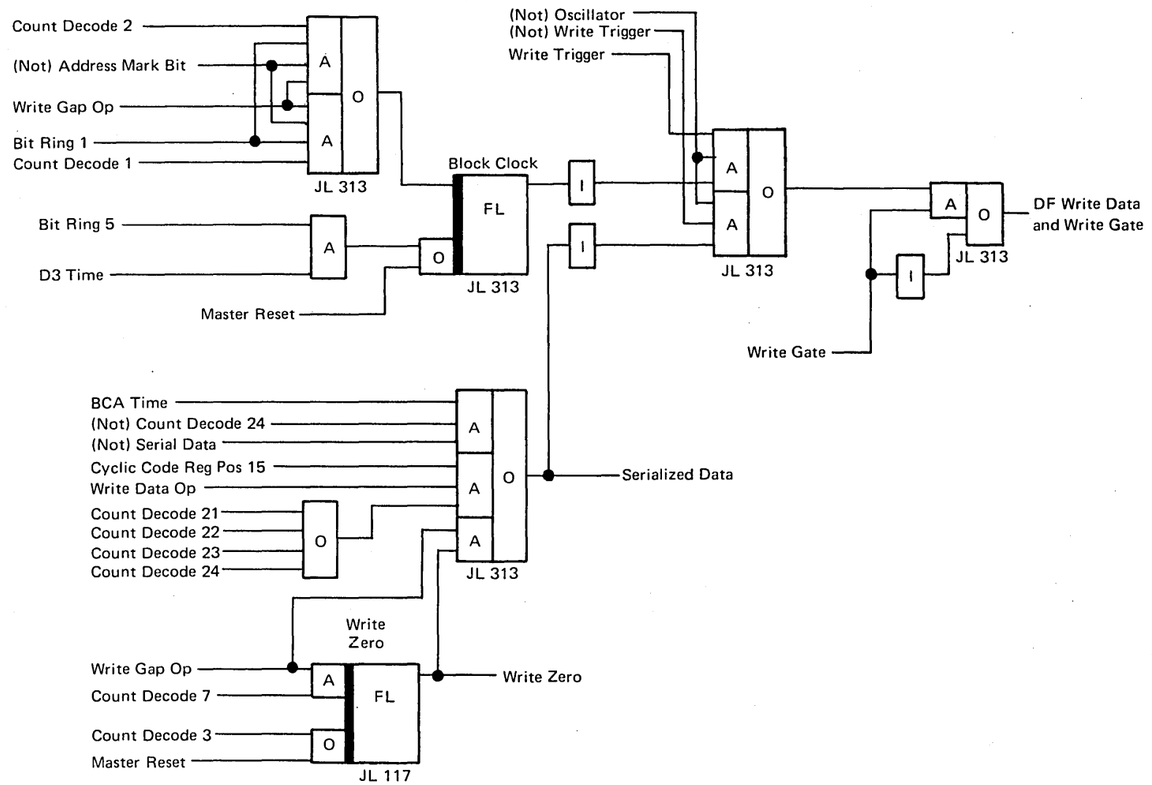
- The double-frequency output of the write oscillator drives the write trigger for single-frequency outputs to define the clock and data bits.
- The write-trigger signals are blocked by the not-data or control lines for the zero-bit level.
- The developed clock and data signals are OR'ed to develop the write signal gated to the selected module.

The output of the write oscillator is used to develop the write-data information fed to the selected file-module. The double-frequency output of the write oscillator drives a binary-connected write trigger with the fall of the pulse to develop two single-frequency output gates. These outputs are used to gate the next oscillator pulse as the clock or data pulse to be written. The zero level of the clock or data pulse is obtained by blocking the respective AND gate.

The clock pulses are blocked only during the five bit-times of each of the two address-mark bytes. For this purpose, the block-clock latch is set at 'bit ring 1' during 'count decode 2' and 'count decode 1'. The latch is reset at BR-5-D3 time in both bytes.

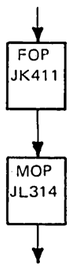
The data pulses are blocked by the zero-bit level of the serial-data line or the output of the cyclic-code register (position 15). The data pulses are also blocked to write the 00-bytes of the VFO area. For the FF-bytes written in the gap areas, no blocking is developed, and thus, all bits are written.

The output of the clock-pulse gate and the output of the data-pulse gate are OR'ed to develop the write-data output. This output is gated by the write-gate signal to feed the selected file-module. When the write-gate signal is down, the output to the file-module is held in the up-level to allow use of the signal coax for read-data input.



Write Data Generation

OPERATION REGISTERS



- The external FOP register is used to buffer the mini-op byte until it is required by the hardware.
- The mini-op transfers to the MOP register when the byte count has been reduced to zero.
- The bit count of the MOP register is tested for out-of-parity conditions that set the bus-out parity indication.

The mini-op codes for IFA operation are set into the FOP external from the external bus-in during the microprogram trap routine. When the execution of the previous operation reaches the 'count 0 gate' condition, the new mini-op is transferred into the MOP register for execution. This sequence is repeated until all of the commands in the chain have been executed.

The bits of the operation registers are defined as follows:

Bit-0 Op Code (Read)
1 Op Code (Write)
2 Address Mark (Omit)
3 Search
4 Scan
5 Index Start
6 Format
7 Skip

MINI-OP CODES

The two high-order bits of the mini-op define the operation code to the file hardware.

(00) No-Op: This mini-op is issued with a count to effect a time-out.

- It is issued without a count to effect a file-control reset.

(01) Write Data: This mini-op is issued along with a count to effect the writing of a field. The sync byte supplied defines the field.

(10) Read Data: This mini-op is issued along with a count to effect the reading of a field. The sync byte supplied defines the field.

(11) Write Gap: This mini-op is issued along with a count to write the gap preceding a write-data operation. The gap ends by writing the address mark and the sync byte. The sync byte ending the gap is supplied with the following write-data mini-op.

MINI-OP MODIFIERS

The low-order six bits of the mini-op are used as modifiers to the basic mini-op code.

Bit-2 Address Mark (Omit): This bit is used with the write-gap mini-op to block the address mark when the erase command is executed. The bit is also set with the read-data mini-op to block the data-check controls when the space-count command is executed.

Bit-3 Search: This bit is used with the read-data mini-op to execute the search-key and search-KD commands. The search bit gates the compare latches for serial-data comparing. The bit is not set for search-HA and search-ID commands because the comparing is performed in the CPU.

Bit-4 Scan: This bit is used with the read-data mini-op to perform the search-KD (scan) commands. The scan bit gates the scan mask-byte (FF) that blocks the compare for that byte. The search bit must also be set to gate the comparing.

Bit-5 Index Start: This bit is used to instruct the hardware to wait for the index pulse before starting this mini-op. For example: when the write-HA command is executed, the index-start bit in the write-gap mini-op starts the writing following the index point. In the case of a read-HA command, the bit is set with the no-op mini-op to initiate the reading at the appropriate point.

Bit-6 Format: This bit is used in conjunction with the write-gap and write-data mini-ops to indicate the formatting sequence. The presence of the bit causes the write controls to remain set at the end of the field because the entire track is being written.

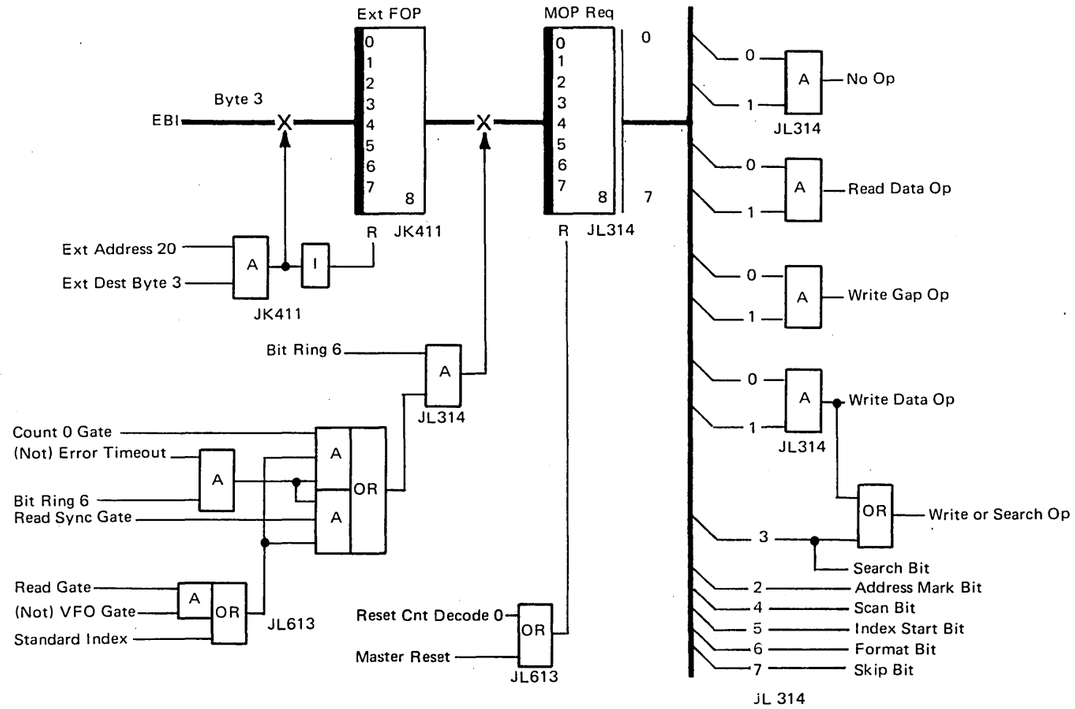
Bit-7 Skip: This bit is used with the read-data mini-op to allow clocking over the field without transferring the data. For example: when a write-data command is performed after a search-ID command, the key field must be accurately clocked-through to find the starting point for the data operation.

FOP REGISTER

The file-operation (FOP) register is byte 3 of the FBAK (external word 20) that is loaded from the external bus-in. It serves as a buffer for the mini-op code in the IFA. The register may be read out to the external bus-out for diagnostic purposes. The register contains nine latches. The parity bit is retained and tested with the parity of the MOP register. The FOP register remains set until the next time it is loaded except for reset by the 'machine reset' line.

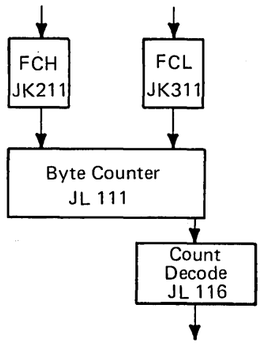
MOP REGISTER

The mini-operation (MOP) register consists of eight latches that control the hardware operation. The FOP register byte is transferred directly into the MOP register during the 'count 0 decode' time. Bit 0 (read) and bit 1 (write) are decoded to develop the no-op, read-data, write-data, and write-gap operation lines. The remaining six modifier bits are used without decoding to control their respective functions. The outputs of the eight latches are used to generate a parity bit that is tested with the parity bit of the FOP register to ensure that the transfer was correct. The register has a forced reset during the index trap when the resulting no-op code causes a time-out in the index gap.



OP Register Entry And Decode

FIELD-COUNT CONTROLS



- Byte counts are used to define the operation through each mini-op.
- The count values are loaded into FCH and FCL externals by the microprogram trap routine.
- The count values are set into the counter in complement and advanced to FFFF=0.
- The counter output is decoded to define functional times within the operation.

The length count (number of bytes) of each field is defined to the hardware by the microprogram when the mini-op is issued. The count value includes the byte count, the four cyclic-check bytes, and the post-record gap as required. A write-gap mini-op has a count equal to the length of the gap. The count value included with the no-op mini-op defines the length of a time-out period.

A 16-bit count value is set into the FCH and FCL registers from the external bus-in during an IFA trap. The new count value is transferred to the byte counter when the previous value in the counter has been decremented to zero. The new mini-op is entered at the same time. The counter outputs are decoded to effect operating functions at their specified times.

COUNT EXTERNALS

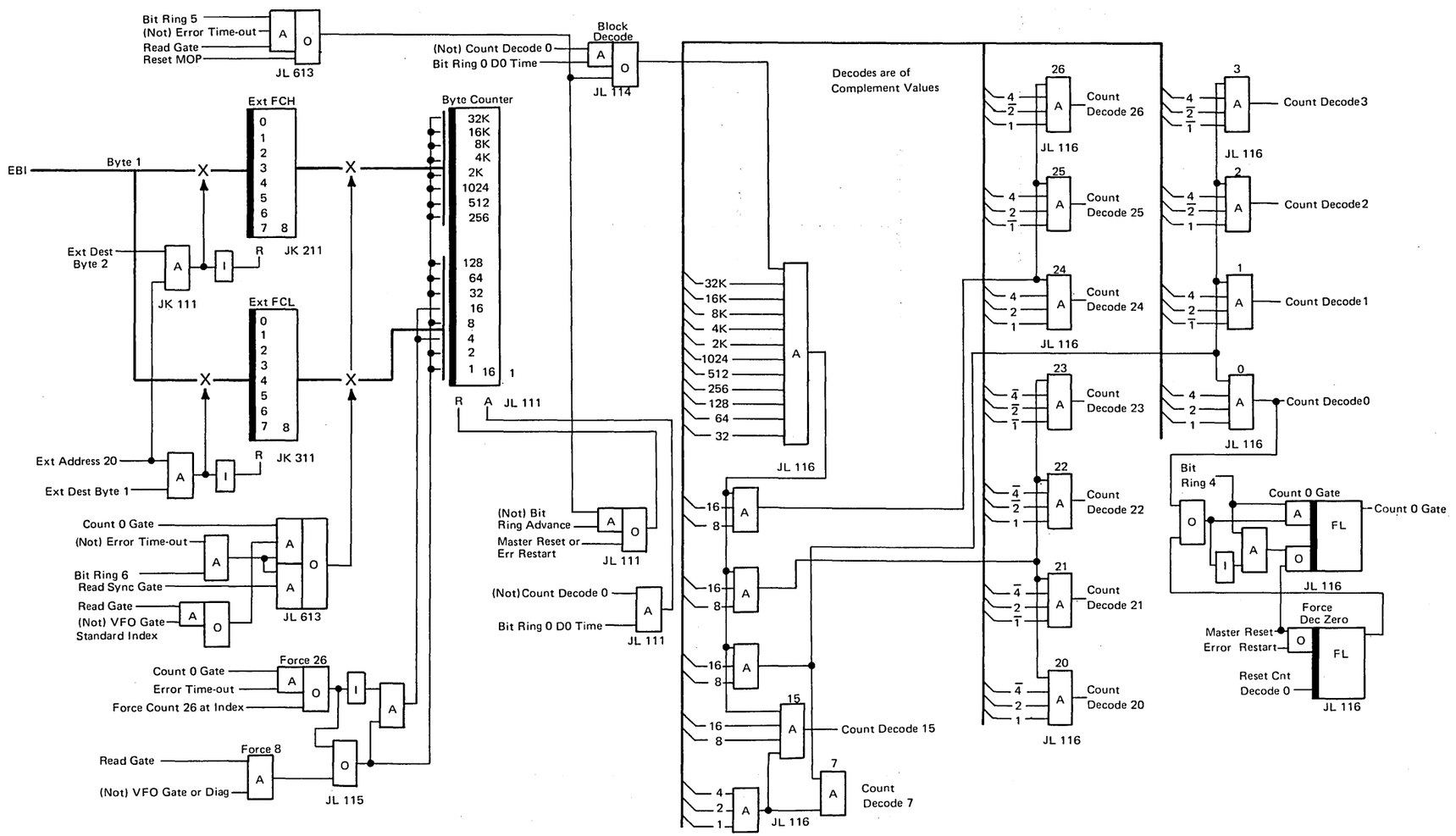
The count value for a mini-op operation is transferred from the CPU in byte 1 (FCH) and byte 2 (FCL) of external word 20 (FBAK). Eighteen latches store the sixteen bits of the binary count value and two parity bits until needed by the byte counter. The count value is available for diagnostic purposes through external addressing. The registers are reset when the next entry is made through the external addressing controls.

COUNTER DECODE

The output of the byte counter is decoded to identify the end of the data area, the cyclic-check area, and bytes of the address-mark area. Because the count value is set in complement, the decode must be made on the complement values. 'Count decode 0' is recognized when the counter positions are all set to the 1-bit level. No decode is made until the counter value reaches 26. At the end of this byte, a 'gate last request' latch is set to stop the share-cycle data requests for write and search operations. The 'decode 25' performs a similar function by setting the 'end data field' latch for read operations. Decodes 24 through 21 gate the cyclic-check operation. 'Decode 20' sets the trap request for a new mini-op. The 'decode 15' turns off the 'erase gate' at the end of a write operation. Decodes 7 and 3 through 0 gate the address-mark bytes. A counter decode of 1 with the trap still in progress causes a command overrun condition. The 'decode 0' also sets a 'count 0 gate. latch to initiate the transfer of the mini-op and the count for the next operation.

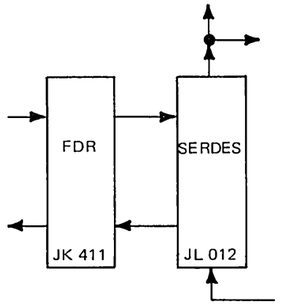
BYTE COUNTER

Any operation that is executing a mini-op has a count value set in the byte counter. The 16-bit count is entered in parallel from the FCH and FCL externals setting the count value in complement at BR-6. The counter is advanced (count decremented) at each BR-0-D0 time. The count value is forced to 26 for an index gap and to eight for timing the address-mark search. The output of all counter triggers are fed to a count decode to gate actions during gap areas and the cyclic-check area. The counter is always reset at 'count decode 0' and at index point when encountered during an operation. It is also reset following a successful address-mark search.



Byte Count Entry and Decode

DATA-TRANSFER CONTROLS



- Write and search data enters the FDR and transfers to the SERDES to be serialized.
- Read data deserializes in the SERDES and transfers to the FDR to enter the CPU.
- The serialized data-bit count is tested against the parity bits being stripped to detect error conditions.

Write or search data enters the IFA through the file data register (FDR) from the external bus-in during an IFA share cycle. At BR-0-D0, the data byte is parallel-transferred to the SERDES register to be serialized for file transfer. The level of the FDR parity bit is tested with the odd/even count of the serialized bits to ensure valid transfer.

Read data (not search) enters the IFA from the file and is deserialized in the SERDES register. At BR-7-D3 when a byte has been assembled, the data is transferred to the file data register (FDR). The data in the FDR is transferred to the CPU on the external bus-out during a share cycle. A parity generator on the SERDES register parallel output develops a parity bit for the byte as it enters the FDR.

FILE DATA REGISTER

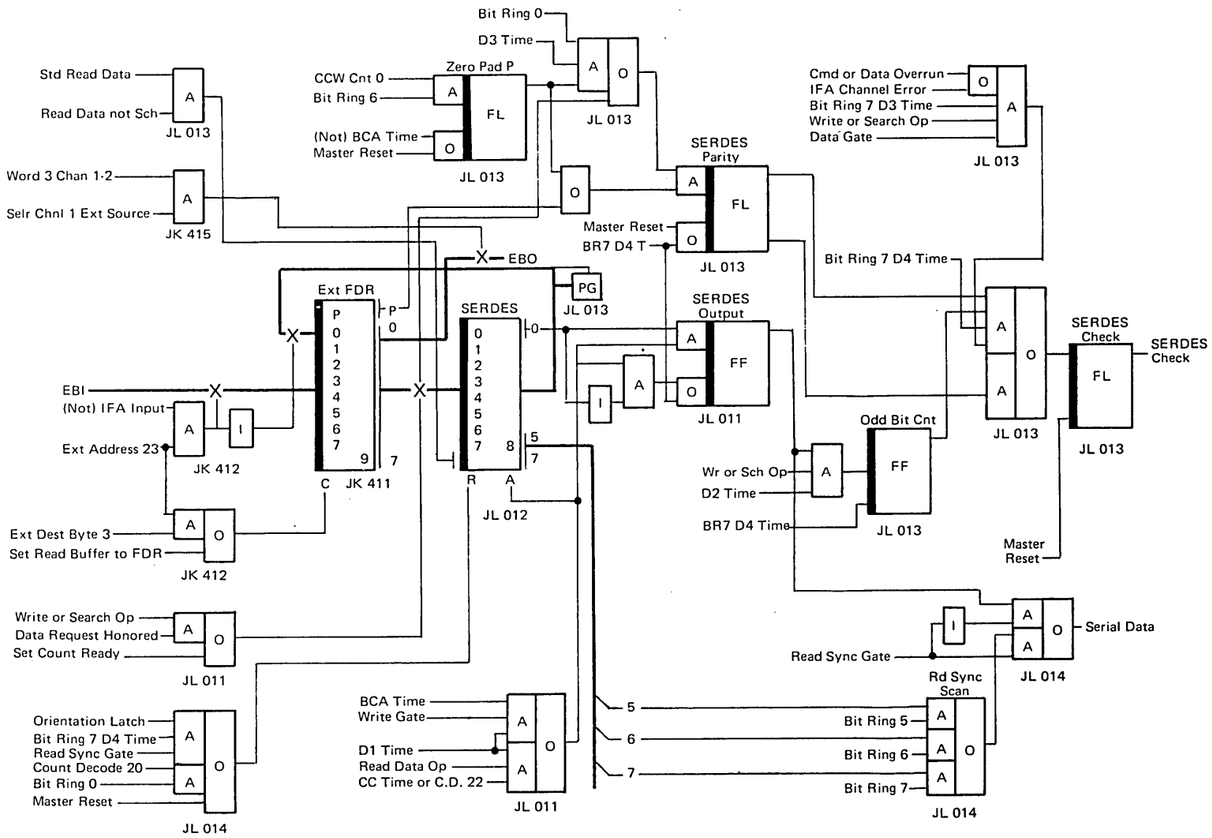
The file data register (FDR) is an external capable of both input and output and is addressed as byte 3 of word 23 (FTAG). The register consists of nine latches for the eight data bits and the parity bit. The latches can be set from either the external bus-in or the SERDES. The data is read from the latches to either the SERDES or the external bus-out. The set, reset, and transfer of FDR data is a function of the share-request controls, and their operation is detailed under "Share Request Controls." The share-request controls contain an 'FDR full' latch to indicate when the data is ready.

SERIAL/DESERIAL (SERDES) REGISTER

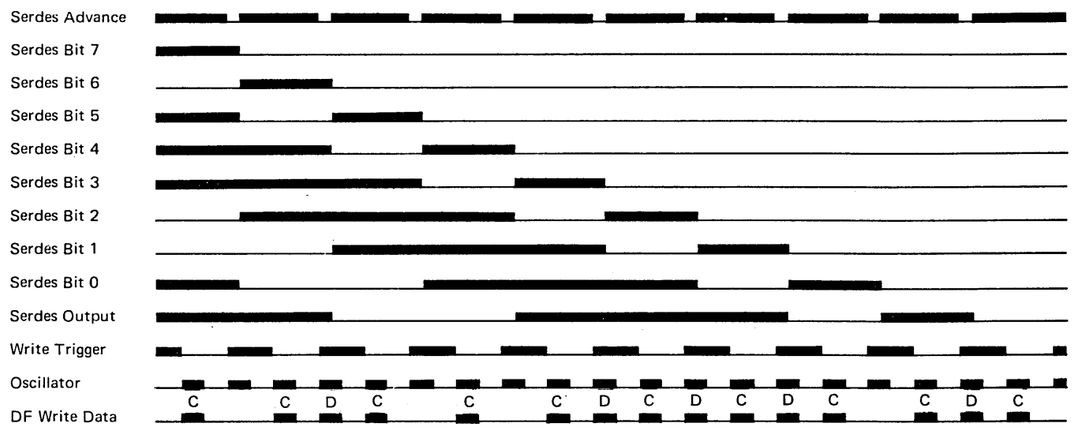
The SERDES consists of eight triggers connected for serial-shift and parallel entry. Only the eight data bits are entered into the register latches. A ninth latch in the parity check circuits is used to retain the parity bit. Serial data enters the register at position 7 gated by the advance pulse. Serial data leaving the register from position 0 is set into an output trigger to provide a one-bit buffer while the next byte is entered. Data bytes are parallel-entered into the register from the FDR at BR-0-D0 on write and search operations. The bits serial to the write generation circuits to write or the compare circuits for search (scan) operations. On read operations, the data bytes are parallel-transferred to the FDR at BR-7-D3 time. The parallel register outputs are used to decode the FF-byte (non-compare mask) on file-scan operations and to develop parity bits on read operations. During read (not search) operations, a decode of a CC-byte on count-decode-22 indicates no BCA. The three low-order bit positions of the SERDES are scanned into the compare circuit with bit ring 5, 6, and 7 for sync-byte compare. The SERDES register is advanced at D1-time for either read or write operations. The register is not reset during data transfer for either read or write operations.

DATA-PARITY CONTROL

The parity bits of the data bytes coming from the CPU are stripped before writing on the file. To replace the parity bits, the IFA develops the cyclic-check information that is written at the end of the field. Two checks are made within the IFA to ensure that the data has been transferred correctly and that the cyclic code is correct. The first check is on the SERDES operation; the odd/even count of the bits leaving the register is compared with the parity bit for each byte. This check sets bit 3 (SERDES Check) in the FSB external. The second check is made by comparing the odd/even count of the bits in the cyclic-code register with the odd/even count of the FDR parity bits of the record field. This check sets bit 0 (CC Hardware Check) in the FSB external. During read operations, the parity bits developed from the SERDES are used for the test. The cyclic-code register test is made following each byte at BR-7-D4.

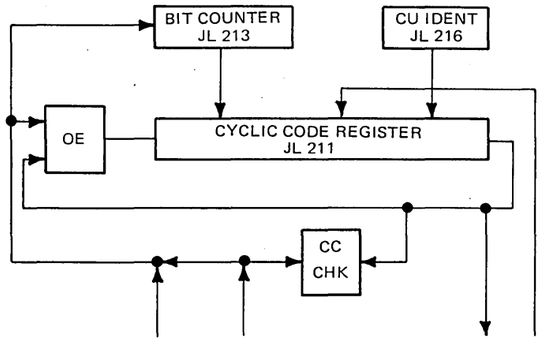


Serial-Deserial (Serdes)



Serdes Output and Write Data

CYCLIC-CHECK CONTROLS



- A cyclic code and a bit count are developed during the writing of each field and then written following the field.
- During read operations, the cyclic-code and bit-count values are again developed and compared with the written values.
- An indicator byte is included in the cyclic-check to identify the writing control unit and file module.
- The operation of the cyclic-code register is checked after each write byte to ensure that the bit combination agrees with the count of entry parity bits.

The cyclic-check controls include the cyclic-code register, the gating for the indicator byte, and the bit-count byte. The four-byte cyclic-check area following each record field is calculated and written during write operations. When a record field is read, the data bytes are used to calculate the cyclic code and the bit count to compare with the written values. The cyclic-code bytes are written by serially shifting the register while feeding the bits to the serial-data line. The indicator byte and the bit-count byte (BCA bytes) are set into the cyclic-code register and serially shifted to write in the same manner. During read operations, the register is again used to serialize the calculated values but the output feeds an OE logic compare with the entry bytes.

CYCLIC-CODE REGISTER

The cyclic-code register consists of sixteen triggers connected as a shift register. The input trigger is position 0, and the output trigger is position 15. The normal data entry for the register is through an exclusive-or circuit into position 0. During data entry, the output of position 15 is fed to the second input of the OR logic circuit to binary-add the bits without carry. The register triggers are reset to the zero state at the start of each field. The output of the register is written in complement to ensure a significant cyclic code for a field written with zeros.

The bit-count byte is transferred true value into cyclic-code register positions 15 through 8. The identifier is entered into the cyclic-code register positions 7 through 0 in complement. Both BCA bytes are written from position 15 in complement. The indicator byte writes true value, and the bit count writes complement value.

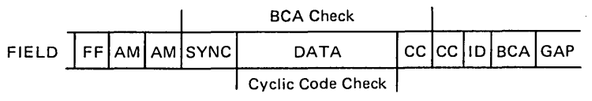
The cyclic-code register is gated to enter the bits starting with the first byte of data in the field and ending with the last byte of the field. The two-byte calculated cyclic code is serially gated from position 15 to write or compare during count-decode 24 and 23. At the start of count-decode 22, the indicator and the bit-count bytes are set into the register. These bytes are serialized by the register to write (or compare bit count) during count decode 22 and 21.

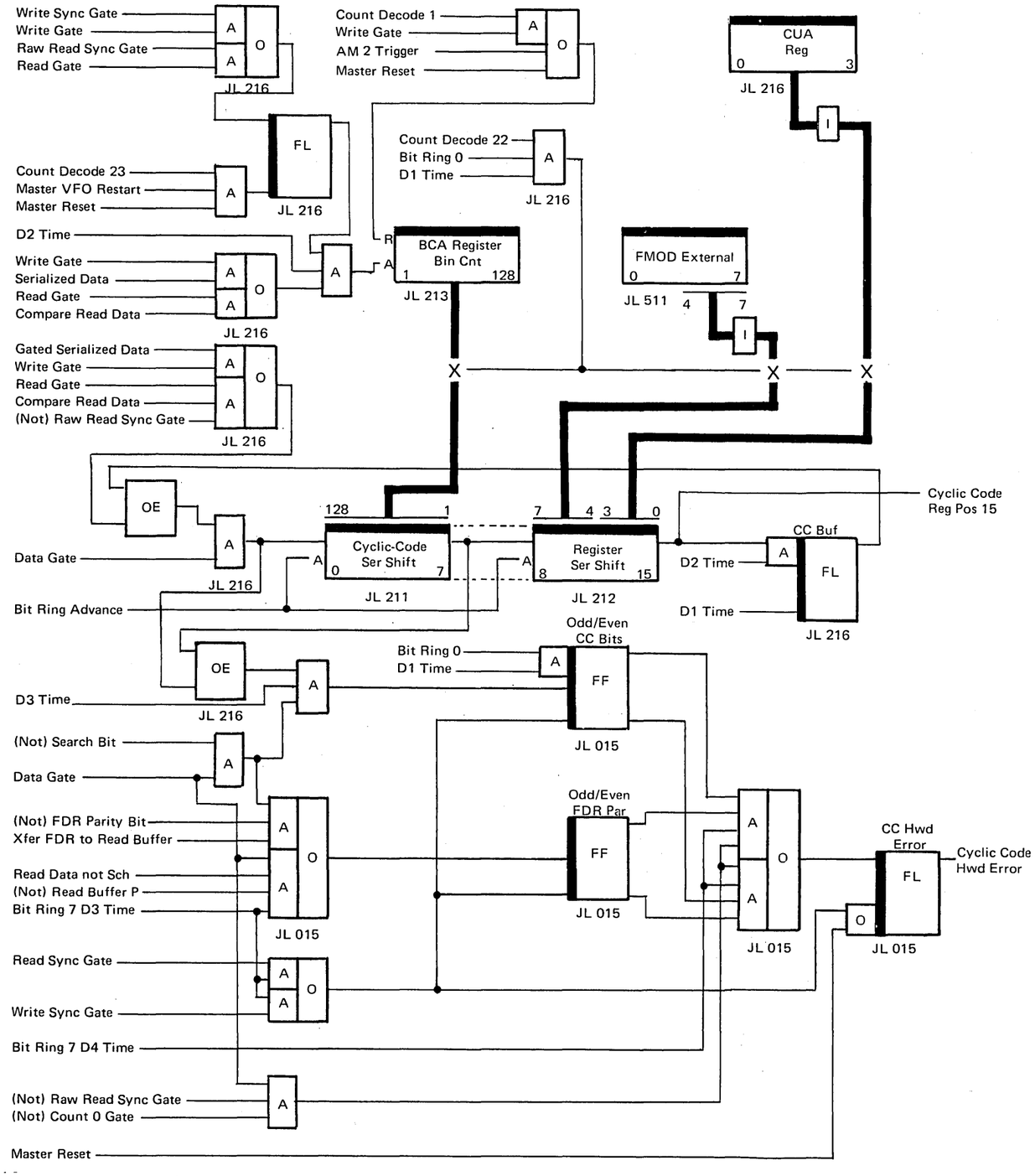
BCA INDICATOR BYTE

The information in the indicator byte comes from two 4-position latch registers. Bits 0 through 3 contain the control-unit address used by the program to effect the write operation. This information is set into four latches by setting the number into the four high-order bits of the FBO and raising the set-CUA tag (bit 4 of FTO). Bits 4 through 7 of the byte contain the number of the physical file module addressed by the write operation. This number is the same as that stored for the low-order four bits on sense byte 4. The value is taken from the module-selected decode in the FMOD-L external that was returned by the selected module over the file interface. The full byte is set into the cyclic-code register (positions 15 through 8) and serialized in complement to write true during 'count decode 22'.

BCA BIT-COUNT BYTE

The bit counter consists of eight binary-connected triggers. Data bits being written or read, starting with the sync byte through the first byte of the cyclic code, are serially gated to the counter. The carry from the high-order position of the counter is lost. Only the low-order eight bits of the binary count are used (255 max.). The true value of the counter is parallel-transferred to positions 7 through 0 of the cyclic-code register at the start of count decode 22. The value serializes in complement to the 'serial data' line during count-decode 21 to either write or compare.





Cyclic-Check Logic

CYCLIC CHECK

During count decode 24, 23, and 21 of read and search operations, the cyclic-code register (pos 15) reads out serially to one side of an OE logic compare in complement. The other side of the OE logic compare is fed serially with the cyclic code and bit count being entered on the 'compare read data' line. The values being read are also in complement because they were written in complement. The output of the OE logic sets the 'data check' latch if any bit position fails to compare. The 'data check' latch also sets for a wrong sync byte when oriented and for a detected index while reading. Sense byte 0 bit 4 indicates the data-check condition when the unit-check status is indicated. If the data check occurred during the reading of a count field, the sense byte 1 bit 0 is also set.

CYCLIC-CODE HARDWARE CHECK

The cyclic-code hardware check operates only on read and write operations. The appropriate parity bits are not developed during the search operation. The hardware makes a check on the operation of the cyclic-code register after each byte is processed. The odd/even count of the bits in the register is compared with the odd/even count of the data-parity bits to that point of the record.

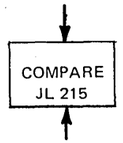
To understand the logic of this checking circuit, it is necessary to understand the resulting bit count in the register. The first byte read into the register binary-adds to zero in the OE logic and enters the 0-7 portion of the register with an odd/even count exactly the same as the first byte. The second byte does the same thing, with the first byte moving to the 8-15 portion of the register.

For the third byte of data, the 8-15 portion of the register is serially fed (pos 15) to the OE logic with the data to binary-add into the 0-7 portion of the register. The odd/even count of the bits at the output of the OE logic is always equal to the odd/even bit count of the two factors entering the logic. The total count of bits in the OE logic output varies with the bit relationship of the bits in the entry factors. The total odd/even relationship for both halves of the cyclic-code register is always equal to the odd-even count of the parity bits of the data bytes entered.

To combine the count of both halves of the register, a second OE logic is used to binary-add these two factors with the output entering a binary trigger for the odd/even count. Because the check must be made following the entry of the new data byte, the odd/even count must include that byte. The output of the OE logic feeding position 0 of the cyclic-code register has the information that serials into the 0-7 portion of the register after entry of the byte. The serial output of position 7 of the cyclic-code register has the information that serials into the 8-15 portion of the register. These two factors are entered into the second OE logic for the total odd/even count.

The parity bit entering the FDR register with the data byte during a write operation and the developed SERDES parity bit for a read operation are counted by the parity-bit binary trigger. The odd/even binary trigger counting the output is tested against the parity-bit binary trigger at the end of each byte. The odd/even trigger is then reset at the end of the byte and set again for the next byte. The parity-bit trigger continues its odd/even count until the end of the field. A difference causes the 'CC hwd error' latch to set as the indicator in the sense information (byte 2 bit 4). An error is also reported as an equipment check (sense byte 0 bit-3).

COMPARE CIRCUITS



- CPU data and data being read from the file record are compared to determine a high, low, or equal condition.
- The microroutine determines the resulting action for the specified command.
- The file-scan operation allows blocking the compare circuits when an FF-byte mask is presented in the CPU data.

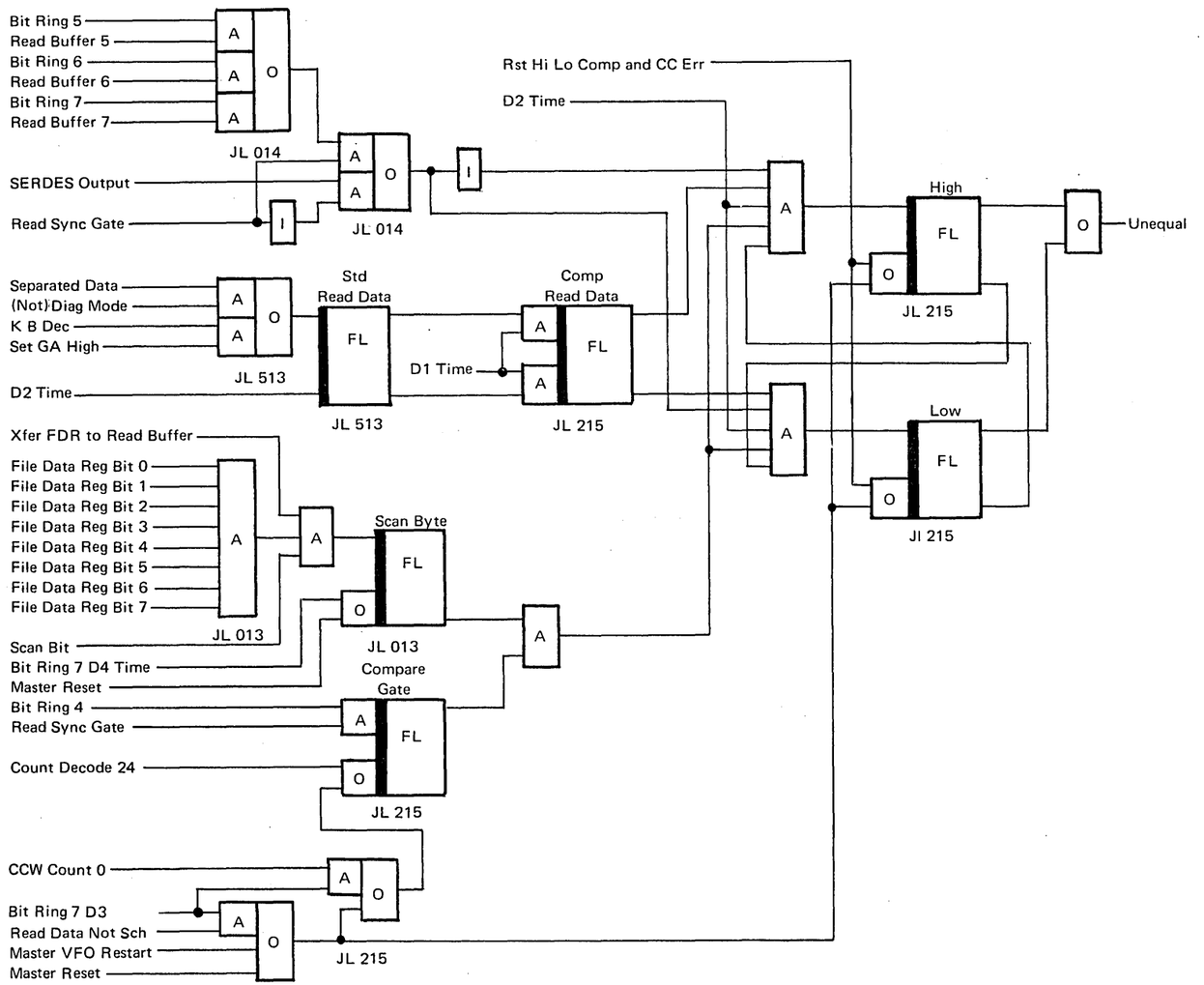
During search and scan operations, data is read from the appropriate field of a file record and compared with data from CPU storage. The CPU data is handled as write information and is serialized by the SERDES to the serial-data line. The 'standard read data' pulses are delayed one bit time by buffering with the 'compare read data' latch to align them with the serial-data pulses.

When the search modifier bit is set in the mini-op, the compare circuits are activated. A 'compare gate' trigger is set with the fall of the 'read sync gate' signal and is reset with the fall of the 'CC calc time' signal to define the field. The output of the 'compare gate' trigger gates the set of the compare latches. The two data sources are fed bit by bit to the set gates of the 'compare high' and 'compare low' latches with one signal inverted. When the bits match, neither compare gate is conditioned. A mismatch of bits causes the appropriate gate to be satisfied, setting the latch.

If either latch is set, indicating a detected mismatch, the input gating for both latches is blocked for the remainder of the compare. The outputs of the 'compare high' and the 'compare low' latches are fed to the CPU for action by the microprogram.

The scan operation differs in that a detected FF-byte in the CPU data entered into the SERDES causes the compare gates to be blocked for the byte.

The compare circuits are also used to compare sync bytes during reading operations. The low-order three bits of the specified sync byte are gated from the SERDES with the bit ring. These three bits are compared with the last three serial-data bits of the file sync byte. The equal-compare output must exist before the field can be read.



Compare Circuits

READ/WRITE CONTROLS

- The control unit raises the read, write, and erase bits on the file-bus-out (FBO) to control the selected file reading and writing.
- Gate latches are set by the operation and timing to control both the file and the control-unit operation.
- The 'write clock gate' selects the write clock and deselects the read VFO during write operations and all control-unit clocking operations.

The IFA control unit has four latches that provide gating to control the read and write functions.

1. Write clock gate latch
2. Write gate latch
3. Erase gate latch
4. Read gate latch

These gate latches are set by the appropriate mini-op code and the timing to start the operation. The outputs of the 'read gate', the 'write gate', and the 'erase gate' latches raise the respective bits in the file-bus-out (FBO) interface to the selected file. These bits along with the control-tag line gate the respective head and amplifier controls to read or write. These bits in the FBO are changed without dropping the control-tag line.

Write Clock Gate Latch

This latch controls the drive selection of the IFA clock. When the latch is set, the clock is driven by the write oscillator and is used for all write functions, clock-through operations and time-outs. When the latch is reset, the IFA clock is driven by the VFO circuits if operating. The normal set for the latch occurs at 'count decode 20' of a read operation because the last twenty bytes (gap) are clocked through. If a write operation follows, the latch remains set for the write operation. When a read function follows, the latch is reset when the count reaches zero. The latch set can be forced by the microroutine and at index time.

Write Gate Latch

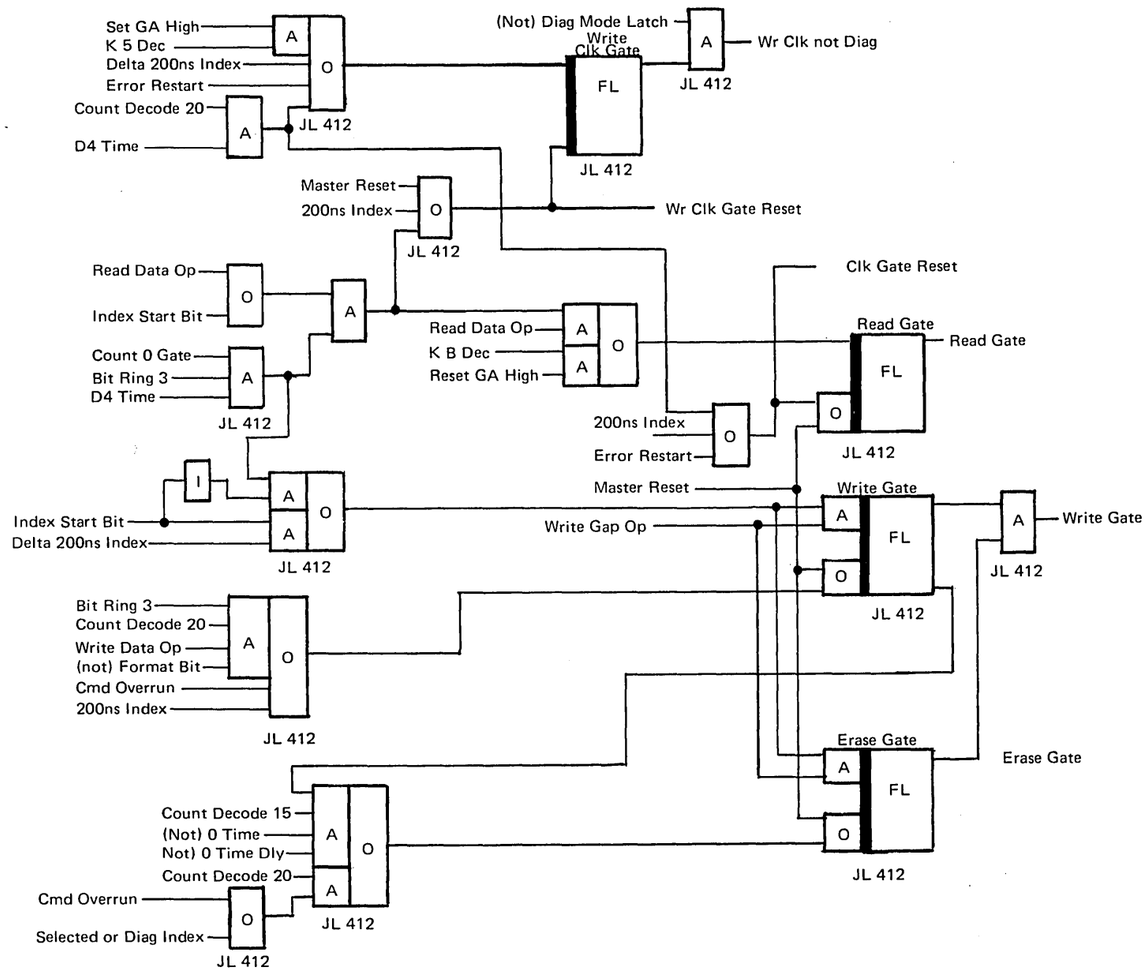
This latch controls the write circuits in the selected file and the control unit. The latch is set when the write-gap mini-op is set with a count of zero. It also sets at index time if the index-start bit is in the mini-op. The latch normally resets when the 'count decode 20' point is reached and it is not a format operation. If it is a format operation, the 'write gate' is not reset until the index point. A command-overrun condition immediately forces the reset of the 'write gate' latch.

Erase Gate Latch

This latch controls the erase head circuits in the selected file. The latch is set under the same conditions as the 'write gate' latch in all cases. The latch normally resets at 'count decode 15' if it is not a format operation. For a format sequence, the 'erase gate' must remain set with the 'write gate' until the index point. The 'selected index' signal resets the 'erase gate' latch. A command-overrun condition forces the reset after the count has reached the decode 20 point. In all cases the 'erase gate' resets about five bytes after the 'write gate' to ensure proper track parameters.

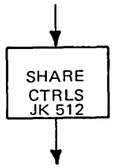
Read Gate Latch

This latch controls the read circuits in both the selected file and the control unit. The latch is normally set with the read-data mini-op at the count zero point to allow the VFO and address-mark search. It also sets at a count of zero when the index point is detected with the index-start bit in the mini-op. The latch can be forced by the microroutine. It is normally reset at the 'count decode 20' time. The reset is forced with the detection of the index point and for an error-restart condition.



Read/Write Gate Controls

SHARE REQUEST CONTROLS



- An IFA share request is developed after the FDR has been loaded during input operations or unloaded during output operations.
- The completion of the serial operation of a byte by the SERDES sets the data-request latch.
- If the count has not reached zero and no error conditions have occurred, the output of the data-request latch initiates the share-request sequence.

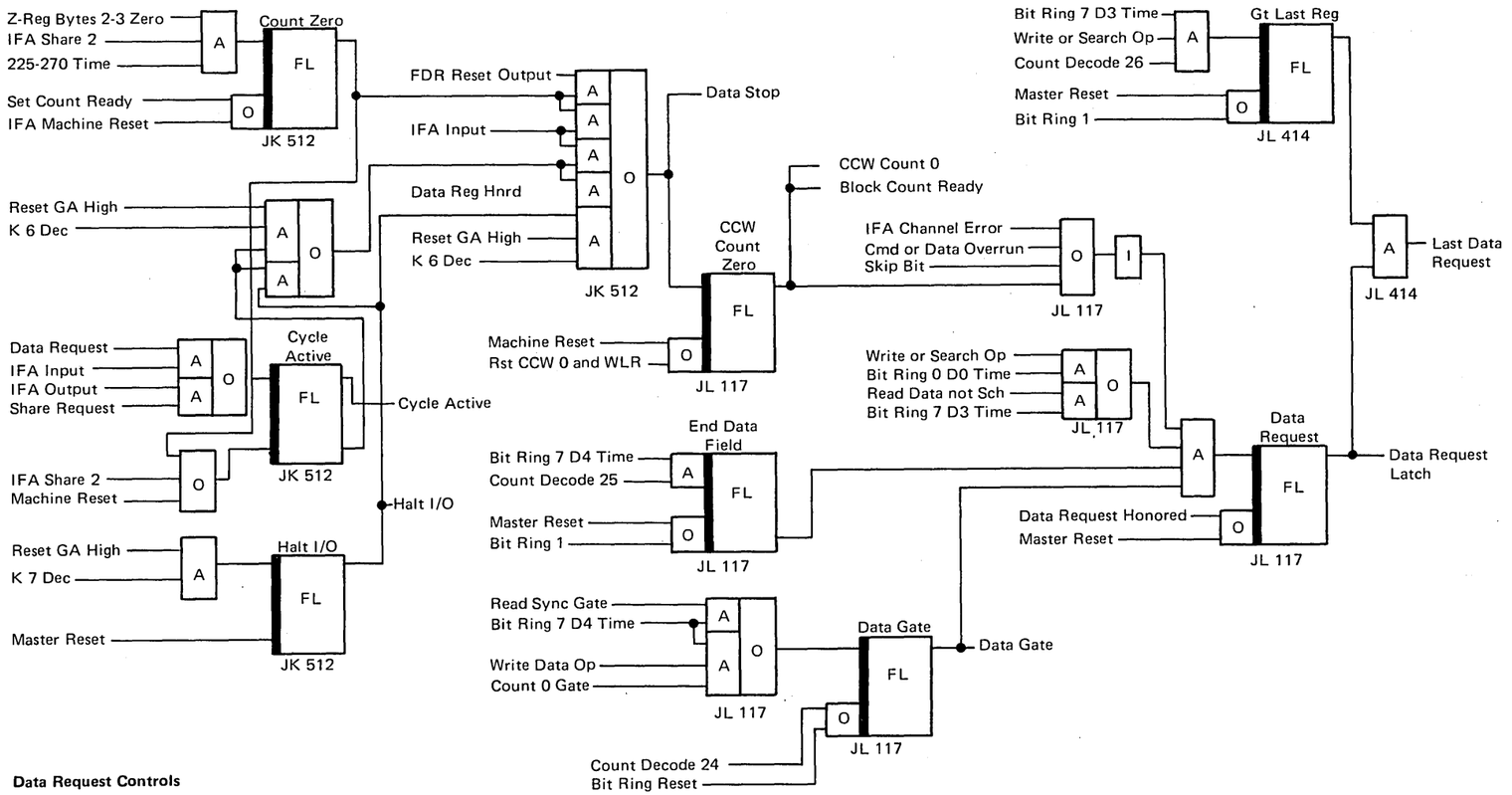
The IFA initiates a share request to the CPU when it has moved a byte of data to the file data register (FDR) on an input (read) operation or removed the byte of data from the FDR on an output (write) operation. The count-ready signal must be present to indicate that there is available CCW count for the transfer. Each time the microroutine loads a data-transfer mini-op and a count value, it also sets the 'count ready' latch for either main storage or control storage along with the input or output latch.

The IFA initiates the share-cycle request by setting the 'data request' latch. The output level of the 'FDR full' latch and the setting of the input/output latches develop the share-request signal. When either the file control indicates the last data request or the CCW count has been reduced to zero, the 'count ready' latch is reset to block further share requests.

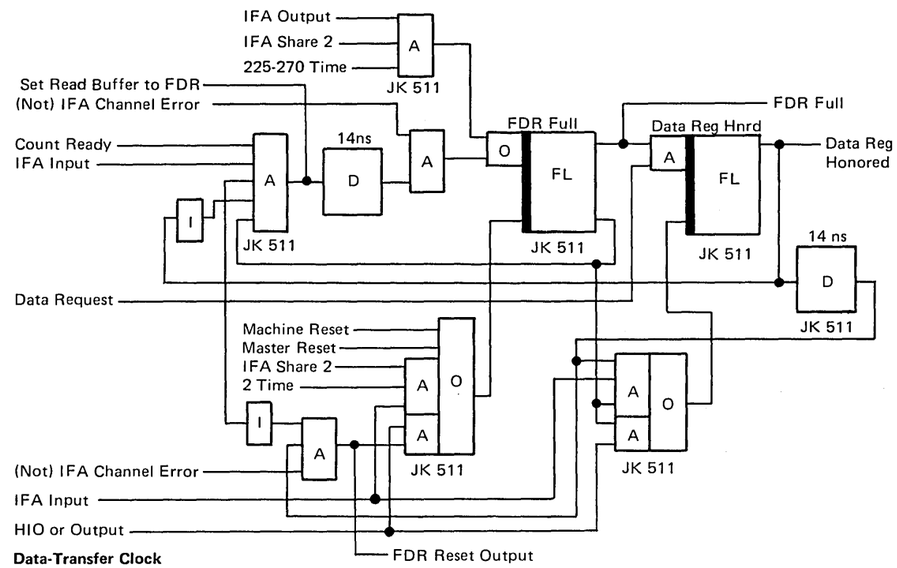
Write or Search Operations

- The data-request latch is set when a byte has been serialized.
- The byte in the FDR is transferred into the SERDES, and the share-request signal is developed.

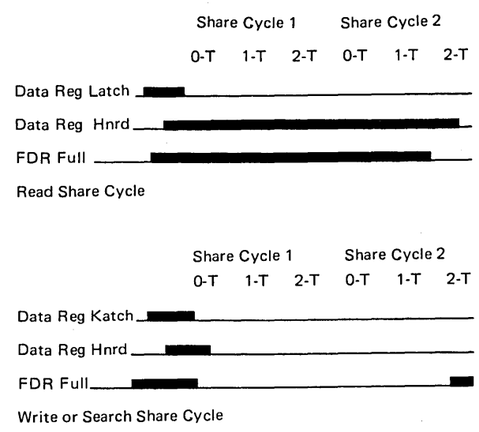
During a write or search operation, the advance to the BR-0-D0 time indicates that the previous data byte has finished serializing and that the new byte in the FDR must be transferred. The 'data request' latch is set if no blocking or error conditions have developed. For a write or search operation, the 'FDR full' latch was set when the new byte was transferred into the FDR. The 'data request honored' latch sets immediately, and the output resets the 'data request' latch. The latch output also gates the transfer of the FDR data to the SERDES. After a delay of 14 ns., the FDR is reset and the 'FDR full' latch is reset for the next transfer. With the 'FDR full' latch reset and the output latch set, the share-request signal is developed if the 'count ready' latch is still set. The reset output of the 'FDR full' latch gates the reset of the 'data request honored' latch, removing the interlock that blocks the next request. The 'FDR full' latch is set during the share-cycle after the new data byte has been loaded into the FDR register.



Data Request Controls



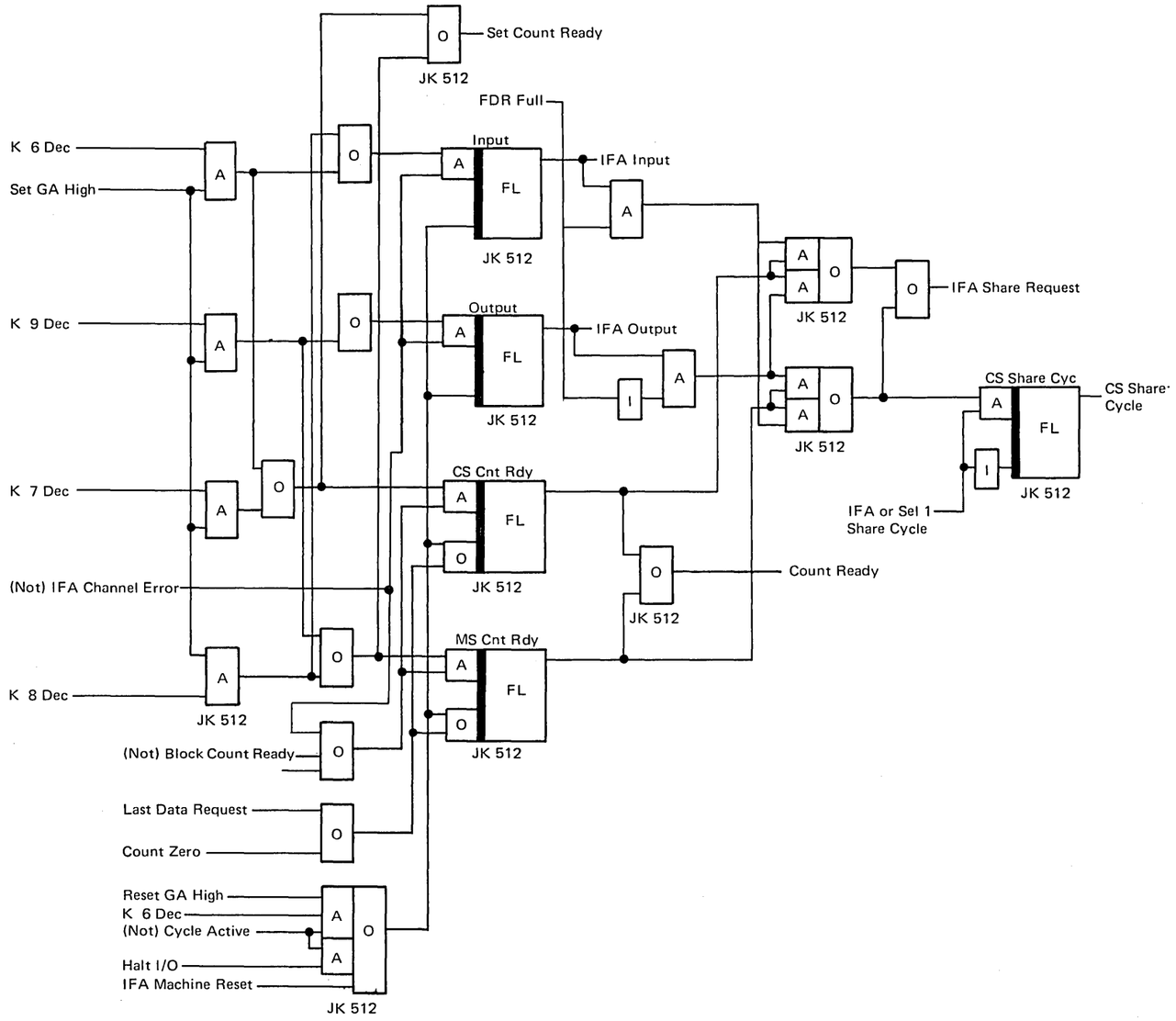
Data-Transfer Clock



Read Operation

- The 'data request' latch is set when a byte has been deserialized.
- The byte is transferred to the FDR, and the share-request signal is developed.

For the read operation, the BR-7-D3 time indicates that a data byte has been deserialized and is ready to be transferred to the FDR. The 'data request' latch is set if none of the blocking or error signals are present. The input operation develops a gating signal to transfer the SERDES to the FDR. After a delay of 14 ns, the 'FDR full' latch is set. The combination of the 'data request' signal and the 'FDR full' signal sets the 'data request honored' latch. The output of the 'data request honored' latch resets the 'data request' latch and blocks a second transfer until the latch is reset. With the 'FDR full' latch set and the input latch set, the share-request signal is developed if the 'count ready' latch is still set. The 'FDR full' latch resets during the share cycle after the new byte is set into the FDR. The reset output of the 'FDR full' latch gates the reset of the 'data request honored' latch removing the interlock that blocks the next request.



Share-Request Data Gates

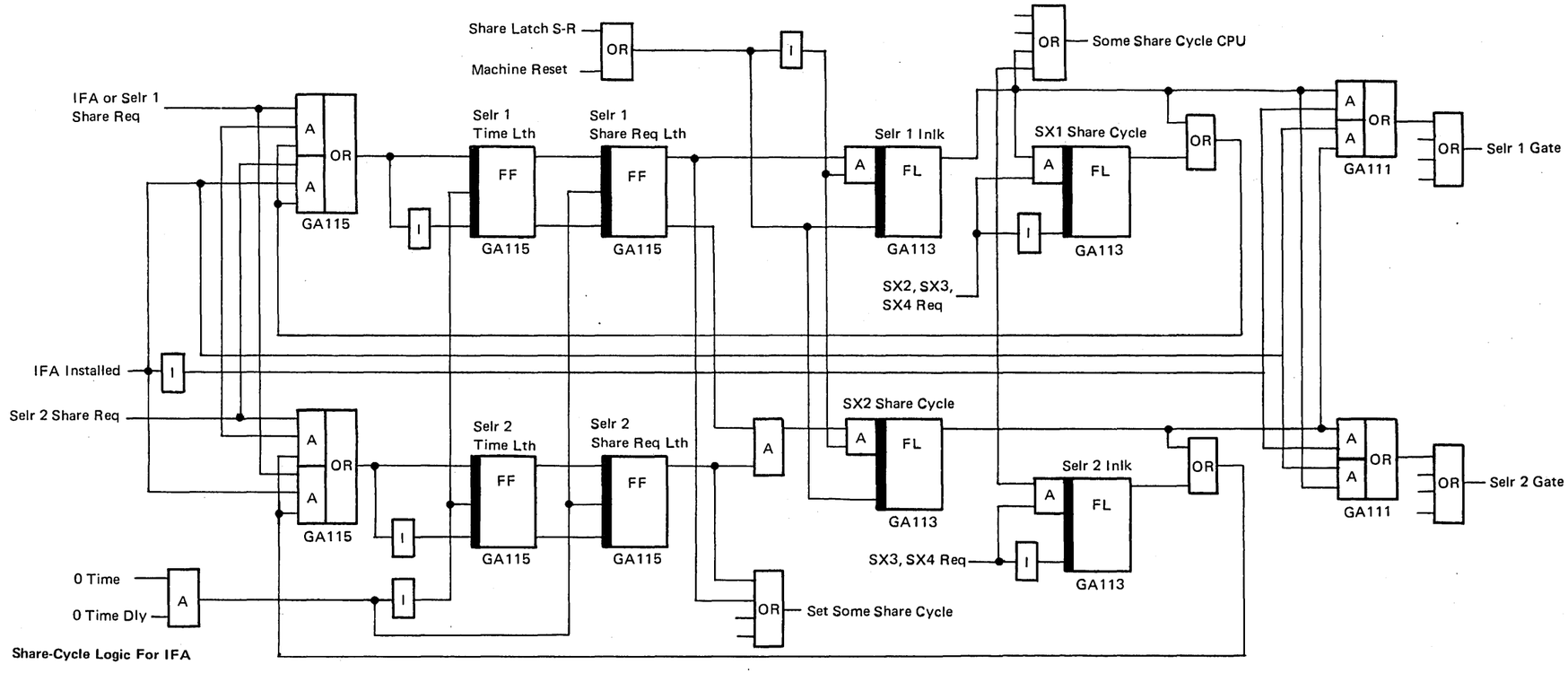
IFA SHARE-CYCLE CONTROLS

- All data is transferred with the selector-channel share-cycle controls.
- The IFA transfers data one byte at a time the same as the non-buffered selector channel.
- In cases of share-cycle contention, channel 2 has first priority followed by the IFA and then channel 3.
- The honored share-cycle request initiates a timing sequence for the transfer.
- A share-cycle storage word is forced into the C-register to effect the transfer.
- The two share cycles follow the pattern of the basic storage word with address and count update.

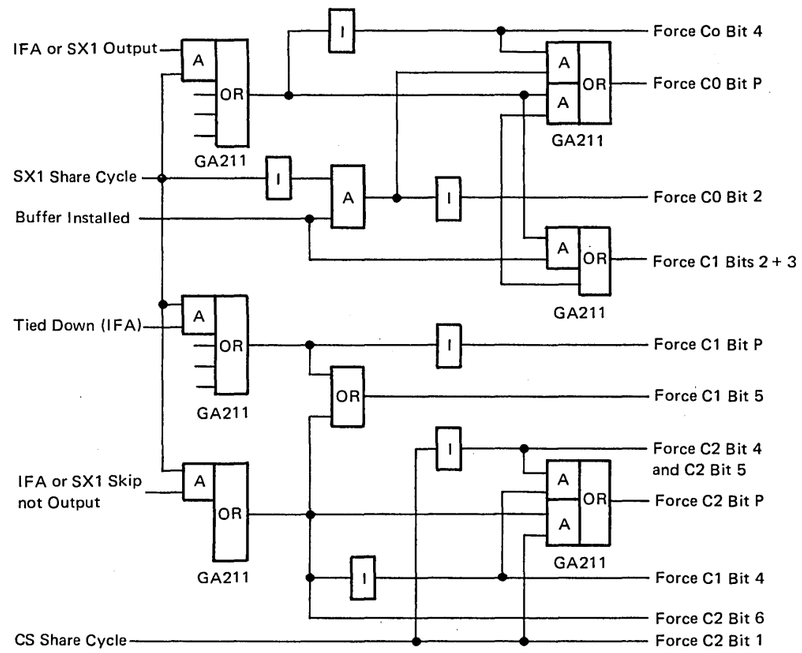
All IFA data transfers are made using the selector-channel share-cycle controls. The current microroutine is stopped for the two control-word cycles (a storage word) for each data transfer. All IFA data transfers are made using the selector-channel share-cycle controls. The current microroutine is stopped for the two control-word cycles (a storage word) for each data transfer. An IFA data transfer is for one byte the same as for the non-buffered channel. The share-cycle transfer makes use of a special storage word that is forced into the C-register and local-storage address gates by the file controls. During the two cycles, the data is moved into or from storage, the data address in load storage is incremented by one, and the count in local storage is decremented by one. The current control-word address is held in the N2 and N3 registers and returned to the M2 and M3 registers during the last half of the second share cycle to continue the operation.

The IFA requests a share cycle when a byte deserialized by the SERDES register moves to the file-data register (FDR) during a read operation. For write and search operations, the request is made when the previous byte in the file-data register moves into the SERDES register to be serialized.

When the selector channels are operating at the same time as the IFA, contention may occur in their share-cycle requests. Channel 2 is assigned first priority in case of simultaneous requests (at sample time). The normal channel priorities for channels 1 and 2 are reversed by gating the IFA to the channel-2 circuits and channel 2 to the channel-1 circuits with an IFA-installed line. At the output of the priority circuits, the two channels are reversed back to obtain the correct addressing controls. The IFA, being second in order, has priority over channel 3. A second request for the IFA or channel already taking a share cycle is blocked for one cycle to ensure reset of the request signal. Once a channel has taken a share cycle, it cannot again request until all of the other channels have had a chance.



Share-Cycle Logic For IFA



C-Register Forced Bits For IFA Share Cycles

Operation	C0 01234567	C1 01234567	C2 01234567	C3 01234567
Output	01100000	10111000	0X00YY00	00001000
Input	01101000	10111000	0X00YY00	00001000
Skip	01101000	10110100	0X00YY10	00001000

X=1 for Control Storage
Y=1 for Main Storage

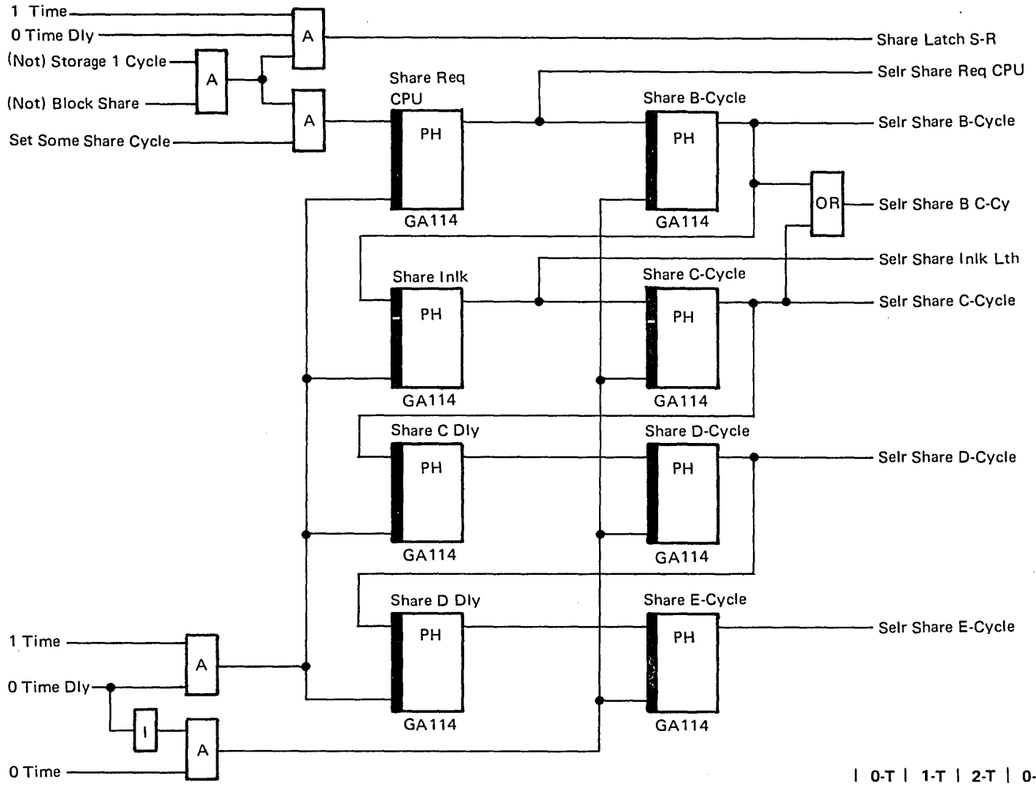
When the IFA or a selector channel has a share request, the timing latch and the share-request latch for the channel are set (IFA sets channel 2) except when the channel is already taking a share cycle or is interlocked by lower-priority requests. Any one of the share-request latches being set initiates the share-cycle sequence. The share-cycle clock is started at 1-time of the following cycle except when that cycle is the first cycle of a storage word. The clock is started by setting the 'selector share request CPU' latch. At the following 0-time, the 'share B cycle' latch is set to define the first of the share-storage cycles. The timing chart shows the relations of the clock outputs and the major control signals.

With the rise of the 'selector share request CPU' signal, the highest-priority channel having its share-request latch set has its share-cycle latch set. The output of the share-cycle latch gates the appropriate addresses and controls for the honored channel during the share cycle. The 'some share cycle CPU' signal developed from the share-latch output blocks the previously addressed control word on the SDBO from entering the C-register and early decodes. At the same time, a storage word for the share cycle is forced into the C-register. The local storage and external addresses for the share cycle are forced through, gating with the channel share-cycle line and are not entered into the C-register. The channel command lines gate the appropriate bits into the C-register for the input, output, or skip control and the direction of the address update.

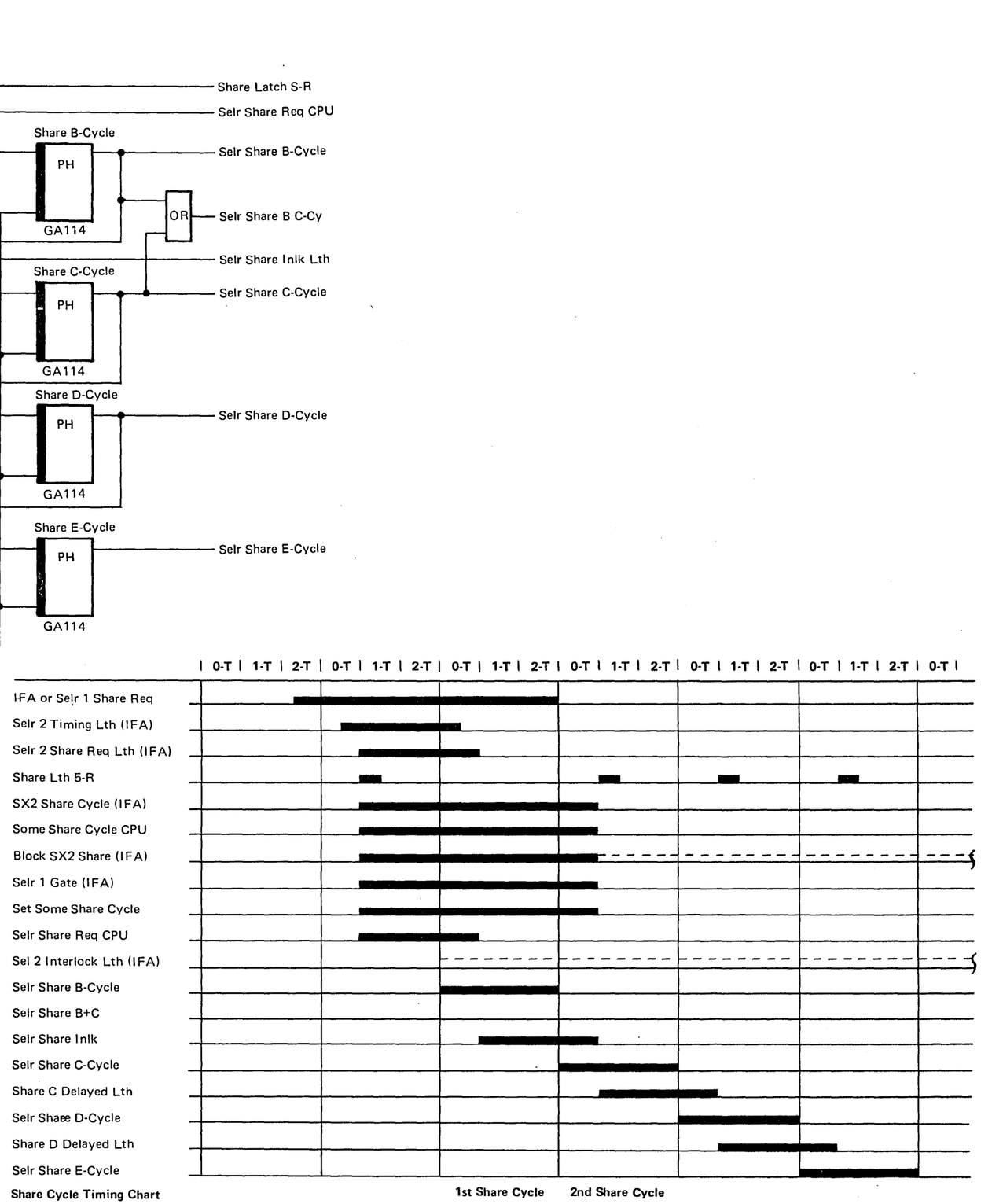
During the first share cycle, the encoded local-storage address containing the data address of the IFA is gated. The data address is set into the M-register for the storage addressing. The address is then incremented by one and destined back to the originating local-storage address. Depending on the direction of the data movement, the appropriate file-data register and storage gates are set. The data follows the normal data paths either from the SDBO to the EBI or from the EBO to the SDBI. In the case of the read function, the data movement occurs at the start of the second share cycle.

During the second share cycle, the encoded local-storage address is incremented by one for the address of the count value of the IFA. The count is decremented by one and destined back to the originating local-storage address.

A share request for a channel can be initiated during the second share cycle to follow immediately. No CPU cycle occurs between the second share cycle of the IFA sequence and the first share cycle of the channel sequence. The interlock that ensures that the channel share request is reset allows one CPU cycle to occur between share cycles for the IFA when no channel request has been made. The third and fourth share-cycle lines from the clock are used for second-cycle destining and retry controls. The IFA share latch is reset at '1 time delay' of the second share cycle after the local-storage access for the count.



Share-Cycle Clock



Share Cycle Timing Chart

INDEX CONTROLS

- All records on the disk file are referenced to a common index point.
- Only the index signal of the selected module is returned to the control unit.
- The index signal causes a sequence of timed pulses to be generated for control.
- By forcing the high-priority (H3) trap, the microprogram is signaled for appropriate action.

The index point indicates that the file disk pack is at the starting position for all tracks. Several operations either must start at the index point; or if reading into the index point, a special sequence must be performed. As a result, a series of index pulses are developed following the sensing of the index point. They occur only when a head has been selected for either reading or writing and not when the file is disoriented.

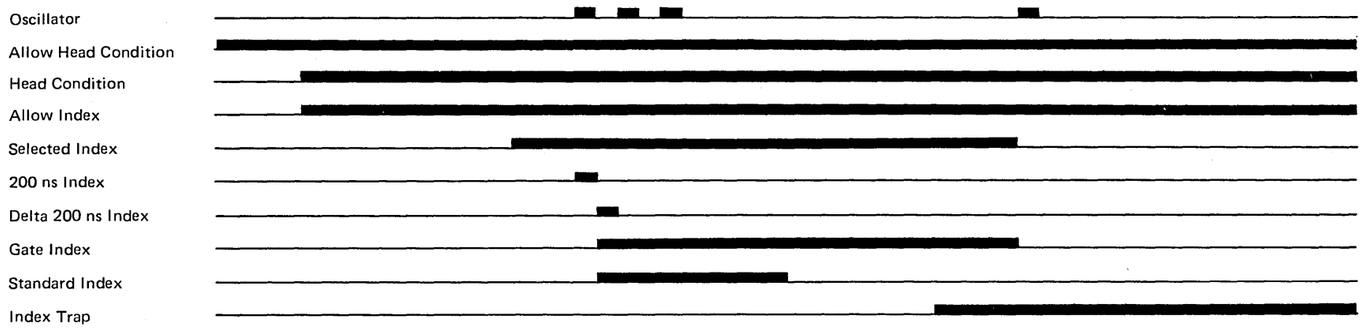
Before the index signal from the selected file can initiate the sequence, the 'allow index' latch must be set. During the initiation sequence for either reading or writing, the select-head bit is set and the control-tag line is raised. At the same time, a value is set into the byte counter for a time-out to allow head selection. When the count is reduced to one, the 'allow head condition' latch is set. At count zero, the 'head condition' latch is set. If the 'selected index' signal is down at this point in time, the 'allow index' latch is set to gate the pulse when it occurs. If the 'selected index' signal is raised at the time of head selection, the 'allow index' latch is not set until the pulse falls. This ensures that a full index pulse is available for the index sequence.

When either the 'selected index' signal occurs or the 'diagnostic index' latch is set, the index sequence is initiated. The sequence is blocked if an error time-out is in progress. The gated index signal provides the gate for the development of the index sequence. With the next rise of the write oscillator, the '200 ns index' trigger is set to provide a reset for the read and write gates. The orientation latch and the bit ring are also reset. The next write-oscillator pulse occurring 200 nanoseconds later resets the '200 ns index' trigger and sets the 'delta 200 ns index' trigger. The output of this trigger provides a gate for the write/erase latches and sets the 'write clock gate' latch. It also provides the gate to set error conditions resulting from the index point. If the track is to be written, the write/erase gates are set immediately. The 'gate index' latch is set to block a repetition of the sequence, and the 'standard index' latch is set. When the third write oscillator pulse occurs, the 'delta 200 ns index' trigger is reset. The 'standard index' latch is reset at the next bit-ring-7 time, ending the basic index sequence.

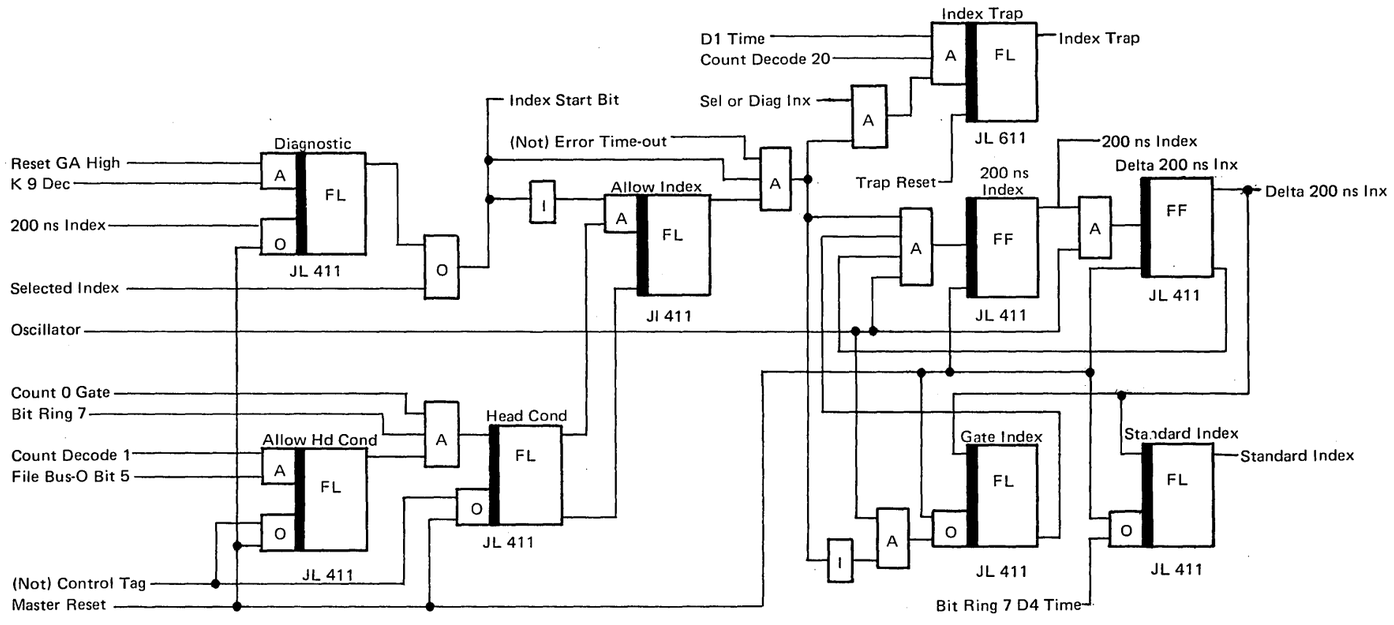
If the index-start bit was not set in the mini-op, the 'standard index' signal resets the MOP and the counter, and forces a count of 26 for an index time-out. When the count is reduced to 20, a request is made for a high-priority trap with the 'index trap' latch set. The microprogram index-trap routine determines if multi-track or overflow-record flags are set to require a head advance. If the head advance is required, the head is advanced and the new head is selected with the normal head-selection time-out. After the head is selected, the home-address field is entered with a forced read-data operation. The original read-data information is stored until the HA entry is completed. The original read-data information is then returned to the FBAK registers, and the operation is continued.

When the index-start bit is included in the mini-op, the operation is started with the 'standard index' signal by transferring the count into the counter. The write gap begins by setting the 'write gate' and the 'erase gate' and writes the FF-bytes of the index gap. The read-data operation begins by searching for the VFO and address mark for the home-address field.

A write operation that encounters the index point while writing results in posting the track-overrun sense (Byte-1 Bit-1). A read operation that encounters the index point while reading a field results in posting the data-check sense (Byte-0 Bit-4).

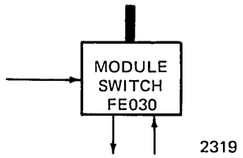


Index-Control Timing Chart



Index Control Logic

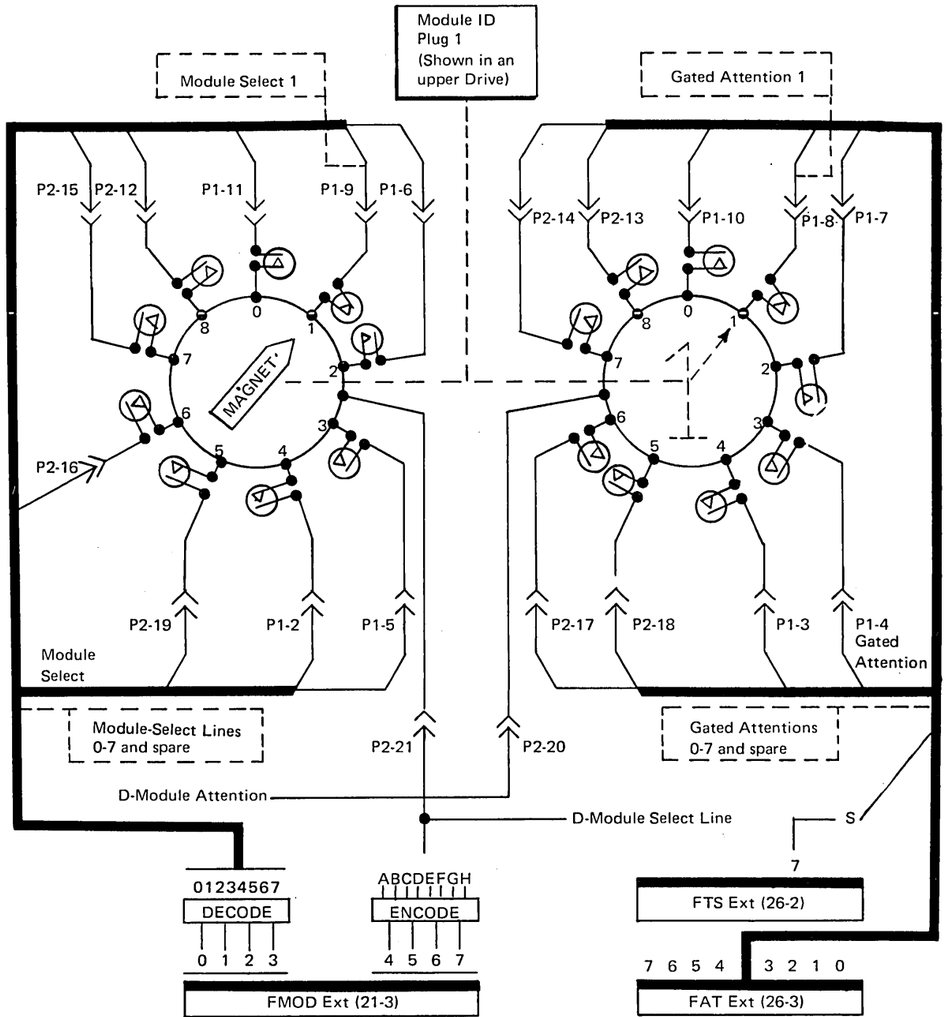
MODULE SELECTION



The IFA can operate with only one disk storage module at a time. The interface lines must be switched from one module to another as required by the program. The programmed address defines the file module to be selected. The selected module gates itself to the interface lines except for the file-data coax. The data switching is performed within the module-select circuits physically located in the 2319-A01 for all eight possible modules.

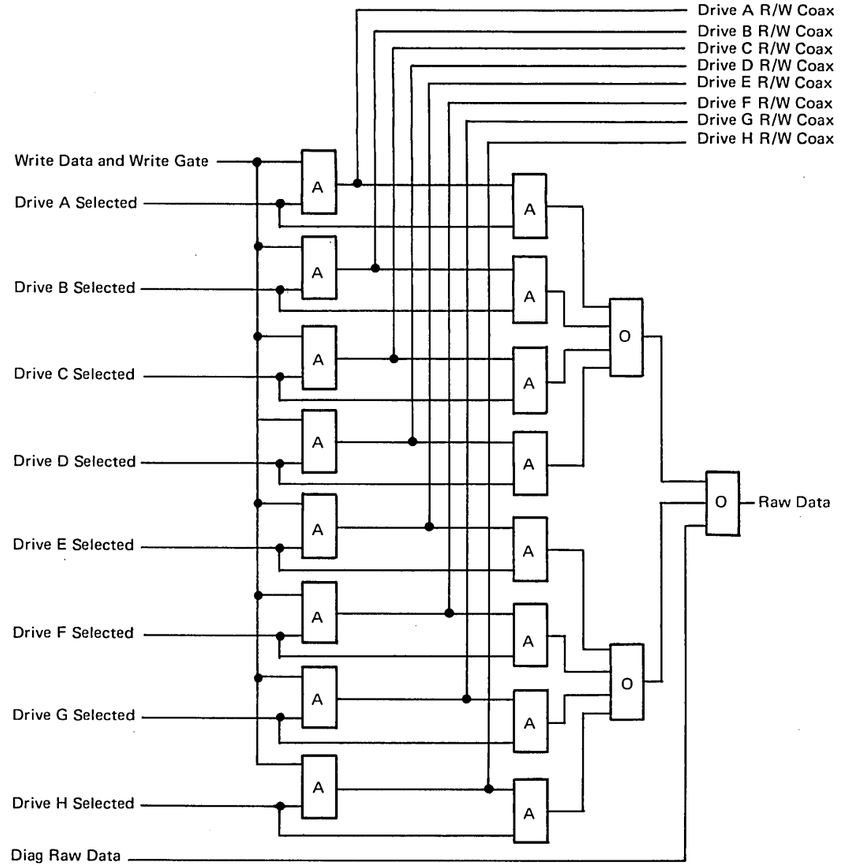
Because the disk drives each have a module selection plug, the selection depends on the location of the module plug for the programmed address defined. The decoded address from the program that is entered into FMOD-U external feeds the module-selected interface lines to the disk storage modules. The module with the indicated plug brings up the module-selected line for its physical address. Within the module, this line gates the controls to the interface. The selected line on the interface gates the data-read and write circuits to the data coax of the selected module. The module-selected line is also encoded and set into the FMOD-L external for use by the IFA.

The file-data coax serves for both read and write operations. The level of the line is normally in the logic-1 level held by the module load resistors when not in use. When writing, the control unit gates the line to the logic-0 level, blocking its use by the module read circuits. The line is taken to the logic-1 level for each bit to be written by the 'write data and write gate' line. For reading, the module gates the file-data coax to the logic-0 level to block its use by the control-unit. The line is taken to the logic-1 level for each bit entering for the raw-data line. The file-data coax lines to unselected lines are held at the logic-1 level by the module load resistors.



Bit	0123	4567	Bit
Module Select 0	1000	0000	A-Module Selected
Module Select 1	1001	0001	B-Module Selected
Module Select 2	1010	0010	C-Module Selected
Module Select 3	1011	0011	D-Module Selected
Module Select 4	1100	0100	E-Module Selected
Module Select 5	1101	0101	F-Module Selected
Module Select 6	1110	0110	G-Module Selected
Module Select 7	1111	0111	H-Module Selected
Spare Module	0111	1111	Mod not Defined

Disk Storage Module Select Circuits



Module Selection For Data

THEORY of OPERATION

DIAGRAM LEGEND

The standard flow-chart convention has been modified slightly for the diagrams in this chapter for ease of explanation. Each chart is accompanied by a text description that covers the operation in greater detail.

The flow-chart diagrams are designed to show the normal execution of the operation with no errors. The basic command operations do not have branch conditions detailed. The start-I/O, halt-I/O, test-I/O, error-trap, ending, and interrupt operations have the major branching conditions detailed. The branch conditions not shown branch out to set appropriate status and result in ending the operation either through the error-trap or the ending routine.

A 'Test for Condition A' indicates that the condition specified must exist for normal operation. A 'Test not Condition B' indicates that the condition is not present. When both conditions of a branch are detailed, a diamond block is shown in normal convention.

BASIC IFA OPERATIONS

- The initial selection sequence provides the status testing and the controls to start a file operation from the program.
- The routine starts by decoding the channel-1 (IFA) address and entering the CAW in the channel controls.
- The IFA routine first determines that the control-unit and module addresses are valid for the system.
- After determining that the addressed file module is not busy, the status indicators are reset before allowing the operation.
- The sense information from the previous operation is reset, and the new information is set into the registers.
- The SIO operation enters a routine to enter the first CCW with the command.
- After determining that the command is valid, the operation stores the information in local storage for execution.
- The command code is decoded, and the operation is routed to the appropriate routine for processing.

All IFA operations are initiated with channel instrucitons. The channel-1 address designates that the instruction is destined for the IFA control unit. After decoding the I/O instruction in the GAIC routine, the operation advances to the GLAA routine to read in the CAW and to decode the channel for execution. The channel-1 (IFA) decode branches the operation to the GPAA routine to perform the initial selection for the IFA operation. The routines up to this point are the same as those for any channel operation.

The GPAA routine first tests for valid control-unit. Tests are made to determine whether the channel is busy or is waiting for an interrupt. A start-I/O instruction cannot be initiated under these conditions, but a test-I/O instruction can be performed. The instruction is decoded to determine the path to be followed. The start-I/O instruction is considered first.

A test is made for the contingent-connection condition that indicates that the control unit has a unit-check status pending. Further tests are made for unselected status and control-unit end before the module is selected. During the module selection, the control-unit address (CUA) is inserted for use on write operations. Tests are now made for unsafe conditions and for possible seek or attention conditions at the file. If conditions indicate that the operation can proceed, the routine is continued.

The command is read from the first CCW and tested for a possible no-op or restore command that should not clear the previous sense information. Otherwise, the sense information is set to zero to start the new operation. The P-register is set for file operation, the CAW is set into FM, and the unit address is set into FW word of local storage. The control-storage address (in FA) for the count storage is initialized for the operation. The retry code is set to 011, and the routine is branched to the BPBA routine to start the operation.

The GPBA and BPBB routines enter the first CCW and decode the command to determine the appropriate routine to continue the operation. The first CCW word is entered into local-storage FD-register, and the command in the 0-byte is tested for the TIC command. If the first command of a command chain is the TIC command, it is an invalid sequence and the program-check status is set. The second CCW word is read in, if the command is other than a TIC, and entered into local-storage FC-register. The command and the storage key are moved to their appropriate operating positions in local storage, and the S-register is set to zero.

The entry being considered valid, the command is decoded by testing two bits at a time until the combination defines either a command group or indicates an invalid command to set the command-reject status. Four command groups and three individual commands are defined to be branched to their appropriate routines:

1. Control	GPBK routine
2. Write	GPBE routine
3. Read	GPBD routine
4. Search	GPBC routine
5. Space Count	GPBF routine
6. Read IPL	GPBK routine and GPBD routine

The command-reject conditions are branched to the file-error (GPCA) routine for handling.

The GPBA routine is also entered during trap routines to obtain the next CCW to continue the operation. If the status-modifier bit is set, the CCW address is advanced by eight to skip the following command (TIC). The next command is tested for a TIC, and if present, its address is entered as the next CCW address and tested for doubleword boundary. At this time, the routine reaches the SIO entry point, and the operation follows the same sequence. If the previous command contained the data-chain flag, the operation branches before decoding the command, with the assumption that it is the same.

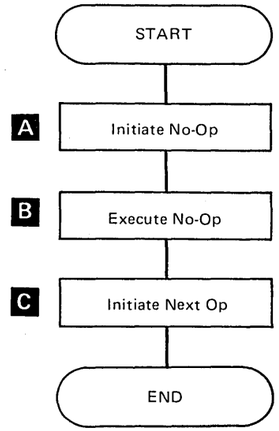
No-Op Mini-Op

0	0	0	0	0	X	0	0
0	1	2	3	4	5	6	7

The no-op mini-op is used to effect time delays during hardware operation by setting an appropriate count value. The delay counts are greater than 20 and cause the usual trap request when the count is decremented to 20. If the mini-op is issued with a count value of zero, a hardware reset is forced. The mini-op is used for head-select delay, index-gap delay, and the recalibrate setup time-out.

For normal time delays and for the hardware reset, no bits are set in the mini-op. When the delay is to start after the index point, the index-start modifier is added to the mini-op.

Entry of the no-op mini-op may occur in a high trap, a low trap, or during an initial selection. The conditions for initial selection are the same as for a low trap in the example. Only the mini-op and the count are loaded during the trap that initiates the operation. No data is transferred to either read or write by the no-op mini-op. The next mini-op to be performed in the sequence is loaded during the trap at count-decode 20. The next mini-op actually starts when the count has decremented to zero.



	Microprogram	IFA Hardware	Disk Drive	Notes
A	Enter from Current Routine or Trap			Clock advancing with write clock if high trap entry.
	Set Condition to Be Delayed			
	Devel No-Op Mini-op	Set FOP		
	Devel Count	Set FCH-FCL		
	Set Write Clock Gate	Gates Write Osc to Clk		
	End High/Low Trap			If count reaches 1 with high trap set, cmd-overrun is set.
B		Count Decode 0		
		Set MOP from FOP		
		Set Counter to FCH-FCL		If count decodes zero, a hardware reset is forced (no time-out).
		Decrement Count		Count continues to decrement for each byte time.
		Decrement Count		
		///		
		Decrement Count		
C	Set High Trap	Count Decode 20		
	Trap Honored			
	Devel Next Op	Set Regs for Next Op		
	End High Trap			If count reaches 1 with high trap set, cmd-overrun is set.

Read Data Mini-Op

1	0	X	X	X	X	0	X
0	1	2	3	4	5	6	7

The read-data mini-op is used to read any defined record field. This includes the home-address, count, key, and data fields as defined in the track and record format. The read-data mini-op is issued to enter a field for search and scan commands. The mini-op is also used by the program for entry of track information and to clock-through fields when oriented.

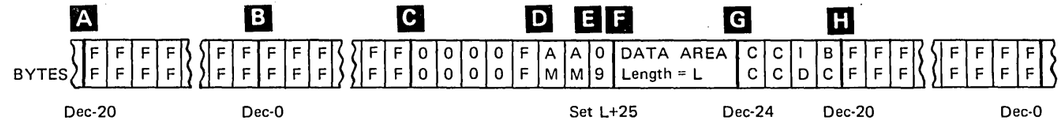
For a straight read operation, only the read bit (bit 0) is set in the mini-op. Modifiers are added to obtain special controls as follows:

Bit 2 Address Mark	Blocks Data Check.
Bit 3 Search	Gates Compare.
Bit 4 Scan	Gates Compare Mask.
Bit 5 Index Start	Starts Read at Index.
Bit 7 Skip	Blocks Data Request.

The serial data on the disk drive is read by the selected head and transferred to the IFA SERDES to deserialize by byte. The assembled byte is transferred to either the control storage or main storage area designated with a selector channel share cycle.

During the post-record gap (last 20 count) the microprogram loads the hardware registers with the mini-op, the required count, the appropriate sync byte, and the data-flow gates. Before reading the field, the VFO clock must be synchronized with the written track and then the bit ring must be synchronized with the byte-defined data. When this is completed, the data is transferred. During the reading of the data, the cyclic code and bit count are computed to compare with the values written with the record field. If the cyclic-check bytes do not compare, the transferred data contains an error that is signaled by the data-check indicator.

	Microprogram	IFA Hardware	Disk Drive	Notes
A	Set High Trap	Count Decode 20		Count advanced with write clock.
	Trap Honored			
	Devel Next Mini-op	Set FOP		
	Devel Count (L+25)	Set FCH-FCL		
	Devel Sync Byte	Set Byte in FDR		
	Devel Data Flow Gates	Set CS or MS Gating		
		Set Data Flow In		
	Set Count Ready	Move FDR to SERDES		
	End High Trap			If count reaches 1 with high trap set; cmd-overrun is set.
B		Count Decode 0		
		Set MOP from FOP		
		Raise Read Gate	Read Amps Gated	
		Drop Write Clock Gate		
		Reset Bit-Ring (3)		
		Advance Bit-Ring	Read Zero Data Bit	Bit-Ring used to count
		Advance Bit-Ring	Read Zero Data Bit	9 consecutive zeros.
		///	///	Zero Detect advs ring.
		Advance Bit-Ring	Read Zero Data Bit	If a one-bit is read,
		Bit-Ring = 9 zeros		the count restarts.
		Set VFO Gate Latch		
		Block Osc Trigger		Gates VFO full pulse.
		Set VFO Sync Gate		
		Force Counter to 8		Defines search time-0.
C		Devel VFO Output	Read a Bit	
		Reset Block Osc Trig		VFO now drives clock.
		Syncs VFO to Read Data	Read FF Byte	
		Release Gap Detectors		
D		Sense Clock Gap	Read AM Byte	
		Set AM-1 Trig		
		Sense Clock Gap	Read AM Byte	If 1 or 3 AM bytes are
		Set AM-2, Rst AM-1		detected, sch restarts.



READ DATA MINI-OP (Read Count, Read Key, Read Data, or Read HA)

	Microprogram	IFA Hardware	Disk Drive	Notes
E		Sense Data Gap	Read Sync Byte High	
		Set Raw Read Sync Gt		
		Rst Bit-Ring to 3		
		Drop Data Gap Sense	Read Sync Data Bit-4	Syncs Bit-Ring to Rec.
		Devel Read Sync Gate		
		Gate Bit-Ring Adv		
		Set Count and MOP		
		Gate SERDES with BR	Read Sync Bits 5,6,7	Bit-ring gates SERDES.
		Compare Bits 5,6,7		If compare is equal, the data is read. If unequal, the search is restarted.
F		Set Data Gate Latch		
		Deserial in SERDES	Read Byte from File	Data from disk drive through the IFA SERDES to either CS or MS as defined by flow gating.
		Devel CC & Bit-count		
		Move Byte to FDR		
	Request Share Cycle	Set Data Request		
	Store Byte in Storage	Decrement Count		
		///	///	
		Deserial in SERDES	Read Byte from File	Transfer continues
		Devel CC & Bit-count		until count reaches 24.
		Count Decode 25		
		Move Byte to FDR		Share request blocked
	Request Share Cycle	Set Data Request		when CCW count = 0.
		Set End Data Field Lch		
	Store Byte in Storage	Decrement Count		
G		Count Decode 24		Check data not stored.
		Block Data Request		
		Serial CC Reg to Comp	Read File CC to Comp	Cyc-Code must compare
		Compare Cyclic-Code		for correct data.
		Count Decode 22		
		Move BCA to CC Reg		
		Serial Ident from CC	Read Ident from File	Ident Byte not comp.
		Count Decode 21		
		Serial Bit-Cnt to Comp	Read Bit-Cnt from File	Bit count must compare
		Compare Bit-Count		for correct data.
H	Set High Trap	Count Decode 20		
		Reset Data Gate	Degate Read Amps	
		Write Clock Gate Set		Count advances with write clock.
	Trap Honored			
	Devel Next Op	Set Regs for Next Op		The operation may be ended without new op.
	End High Trap			If count reaches 1 with high trap set; cmd-overrun is set.

Write Data Mini-Op

0	1	0	0	0	0	X	0
0	1	2	3	4	5	6	7

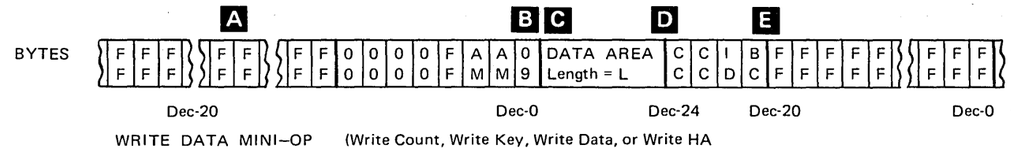
The write-data mini-op is used to write any defined record field. This includes the home-address, count, key, and data fields as defined in the track and record format. The write-data mini-op writes only the actual data field defined except during formatting, when the 20-byte post-record gap is also written.

For an update write operation, only the write bit (bit 1) is set in the mini-op. For a format write operation, the format bit (bit 6) is added to hold the write circuits active through the post-record gap.

The data from either the control-storage or the main-storage area designated is transferred byte by byte with a selector-channel share cycle to the IFA FDR. The data is serialized by bit in the SERDES to write on the disk drive.

During the last 20 bytes of the previous write-gap operation, the microprogram loads the hardware registers with the write-data mini-op, the required count, the appropriate sync byte, and the data-flow gates. Before performing the write-data operation, the write-gap operation including the writing of the sync byte must be completed.

When the gap is completed, the data is transferred. During the writing of the data, the cyclic code and the bit count are computed to be written at the end of the data record. Checks are made on the odd/even bit count serialized and the bit count of the computed cyclic code to set the data check and CC-hardware-check indicators. For an update write operation the write-gate is dropped after writing the BCA bytes (count-decode20). The erase gate is dropped at count-decode 15. These gates are not dropped for a format-write operation, and the writing continues with forced FF-bytes.



	Microprogram	IFA Hardware	Disk Drive	Notes
A	Set High Trap	Count Decode 20		Clock advanced with write clock.
	Trap Honored			
	Devel Next Mini-op	Set FOP		
	Devel Count (L+25)	Set FCH-FCL		
	Devel Sync Byte	Set Sync Byte in FDR		
	Develop Data Flow Gates	Set CS or MS Gating		
		Set Data Flow Output		
	Set Count Ready	Move FDR to SERDES		
	End High Trap			If count reaches 1 with high trap set, cmd-overrun is set.
B		Count Decode 0		Gates set by previous write-gap op.
		Set MOP from FOP		
		Set Counter to FCH-FCL		
		Decrement Count		
		Serialize from SERDES	Write Sync Byte on File	Sync byte is last byte of write-gap op (Also enters BCA).
		Develop Bit-count		
C		Set Data Gate Latch		Data from either CS or MS, as defined by the flow gating, feeds through SERDES to the disk drive.
	Request Share Cycle	Set Data Request		
	Read Byte from Storage	Set Byte into FDR		
		Move FDR to SERDES		
	Request Share Cycle	Set Data Request		
		Decrement Count		
	Read Byte from Storage	Set Byte into FDR		
		Serialize from SERDES	Write Data Byte on File	
		Devel CC & Bit-count		
		///	///	Sequence repeats for each byte.
		Move FDR to SERDES		
	Request Share Cycle	Set Data Request		
		Decrement Count		
	Read Byte from Storage	Set Byte into FDR		
		Serialize from SERDES	Write Data Byte on File	
		Devel CC & Bit-count		
		Count Decode 25		
		Set End Data Field Lch		Transfer continues until count reaches 24. Share request blocked when CCW count =0; remainder of field is written with zeros.
		Block Data Request		
		Move FDR to SERDES		
		Decrement Count		
		Serialize from Serdes	Write Data Byte on File	
		Devel CC & Bit-count		
D		Count Decode 24		
		Block SERDES Output		
		Gate CC-Reg-15 to Write		
		Serialize from CC-Reg	Write First CC Byte	First byte enters BCA. CC-reg output is complemented to write.
			Write Second CC Byte	
		Count Decode 22		
		Set Bit-count in CC-Reg		
		Set CUA into CC-Reg		CUA and FMOD-L are set in complement.
		Set FMOD-L into CC-Reg		
		Serialize from CC-Reg	Write First BCA Byte	Ident byte writes true.
			Write Second BCA Byte	Count byte writes comp.
E	Set High Trap	Count Decode 20		
		Reset Write Gate	Degate Write Amps	If the write op had format bit set, the write and erase gates remain set.
	Trap Honored			
	Devel Next Op	Set Regs for Next Op		
	End High Trap			
		Clock Decode 15		
		Reset Erase Gate	Degate Erase Amps	The operation may be ended without new op. If count reaches 1 with high trap set, cmd-overrun is set.

Write Gap Mini-Op

1	1	X	0	0	X	X	0
0	1	2	3	4	5	6	7

The write-gap mini-op is used to write the gap area just prior to a defined data area. This includes the home-address, count, key, and data fields as defined by the track and record format. The length of the gap written varies with the associated data field to which it is coupled. The gap preceding key and data fields has a length of 21 bytes. The home-address field follows a 73-byte index gap. The gap that precedes a count field is 23 bytes plus a variable length that allows for data skew in the previous record. The last eight bytes of all gaps contain the VFO and sync areas.

For the normal update write, the write-gap mini-op contains only the read bit (bit 0) and the write bit (bit 1). Modifiers are added to obtain controls as follows:

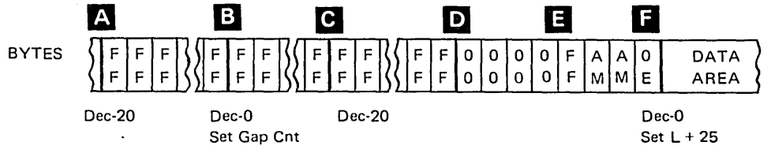
Bit 2 Address Mark	Blocks writing AM.
Bit 5 Index Start	Starts gap at Index.
Bit 6 Format	Indicates format write sequence.

For the write-gap operation, no data is transferred from storage. The gap contains FF-bytes, 00-bytes, and AM-bytes that are forced in the IFA hardware. The sync-byte written as the last byte is entered by the microprogram as part of the following write-data operation.

During the post-record gap (last 20 count), the microprogram loads the hardware registers with the mini-op and the count for the operation. The write-gap operation is started when the previous count reduces to zero. The write and erase gates are set at this point except when they are already set from a previous format-write operation.

The write gate allows both the clock and data bits to write an FF-byte if no blocking circuit is activated. The 00-bytes are developed by blocking the data bits. The address-mark bytes are developed by blocking the appropriate clock bits. The erase command sets the address-mark modifier in the mini-op to cause the AM-bytes to write as FF-bytes. The count decode defines the sequence of the bytes.

At count-decode 20, a trap is taken to enter the associated write-data operation. At this time, the sync byte for the field is entered to write during the count-decode-0 time of the write-gap operation. The write and erase gates remain set through the following write-data operation.



WRITE GAP MINI–OP

	Microprogram	IFA Hardware	Disk Drive	Notes
A	Set High Trap	Count Decode 20		Clock advanced with write clock.
	Trap Honored			
	Devel Next Mini-op	Set FOP		
	Devel Count	Set FCH-FCL		
	End High Trap			If count reaches 1 with high trap set, cmd-overrun is set.
B		Count Decode 0		
		Set Write Gate Lch	Gates Write Amps	
		Set Erase Gate Lch	Gates Erase Circuits	
		Set MOP from FOP		
		Set Counter to FCH-FCL		
		Decrement Count		
		Allow Clk and Data Wr	Write FF-Bytes on File	Count is decremented after each byte.
C	Set High Trap	Count Decode 20	Write FF-Byte on File	FF-Bytes Continue to write on file.
	Trap Honored			
	Devel Next Op & Count	Set FOP, FCH, FCL Regs		
	Devel Sync Byte	Set Byte in FDR		Sync byte writes as last write-gap byte.
	Devel Data Flow Gates	Set Data Flow Gates		
	Set Count Ready	Move FDR to SERDES		
	End High Trap			If count reaches 1 with high trap set, cmd-overrun is set.
D		Count Decode 8	Write FF-Byte on File	
		Decrement Count		
		Count Decode 7		
		Set Wr Zero (Blk Data)	Write 00-Byte on File	With the data bits blocked only the clock are written.
		Decrement Count		
			Write 00-Byte on File	
		Decrement Count		
			Write 00-Byte on File	
		Decrement Count		
			Write 00-Byte on File	
		Decrement Count		
E		Decode Count 3		
		Reset Write Zero	Write FF-Byte on File	
		Decrement Count		
		Count Decode 2		
		Block 5 Count Bits	Write AM-Byte on File	If AM modifier is set FF is written for AM.
		Decrement Count		
		Clock 5 Count Bits	Write AM-Byte on File	
		Decrement Count		
F		Count Decode 0		
		Set Write Sync Gate		
		Set Bit Count Time		
		Gate SERDES to Write		Sync byte entered into FDR by write-data op.
		Serialize from SERDES	Write Sync Byte	

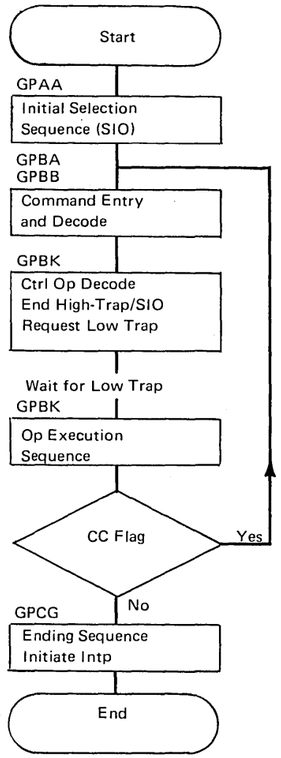
Basic Control Command Execution

- Control commands are executed with the file disoriented and with the low-priority trap.
- The no-op command is processed immediately if it occurs during the initial selection.
- Each of the control commands is handled with an assigned portion of the GPBK routine.
- If another command is not chained, the operation branches to the ending routine to store status.

Control commands are processed with the GPBK routine after disorienting the file. The control commands are decoded in the GPBB routine, and with the exception of the space-count command, are branched to the GPBK routine where the final decode is performed. When the control command is decoded as a no-op command and the entry is through the initial sequence, the ending status is presented and the operation returns to the GPBA routine for the next CCW.

If a control command enters the GPBK routine through either the high trap or the initial sequence, the initializing controls are reset and a request for low trap is initiated. This allows other high-priority devices to obtain service while the IFA is disoriented. If the previous operation had the format bit set, the controls cannot be disoriented until the index point is passed.

When the low-trap request can be honored, the operation is resumed with the final decoding of the command. Each of the control commands has a routine for handling the operation within the GPBK routine. At the completion of the selected routine, the operation is tested for possible error conditions. If no error exists, a test is made for the CC flag to determine whether the command is chained. When the command is chained, the operation returns to the GPBA routine for the next CCW. The low trap is not ended until the new CCW has entered its first mini-op into the control unit. If the command-chain flag was not present or an error existed, the operation is branched to the GPCG routine to store status and to end the operation.



Basic Data Command Execution

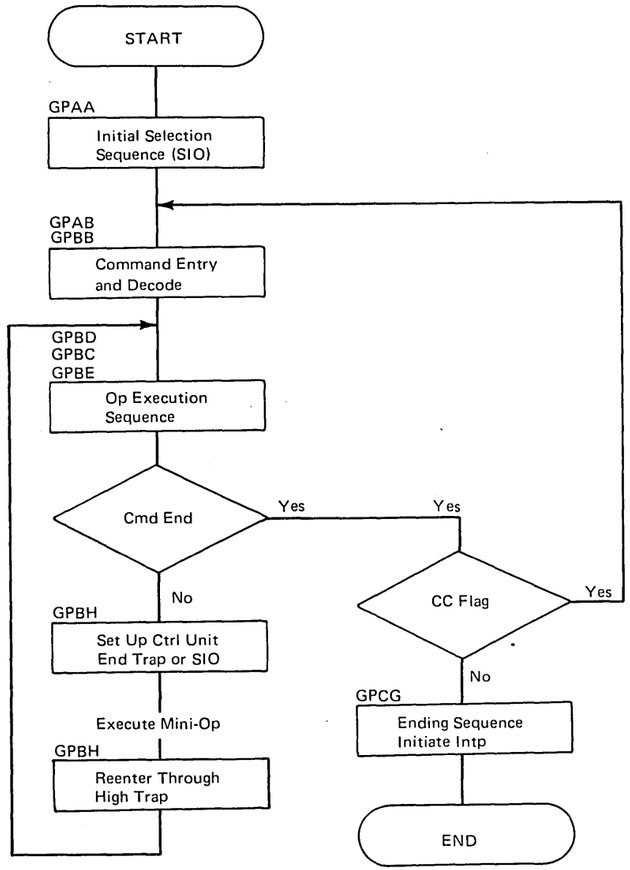
- Data commands are executed by sequentially setting the appropriate mini-ops.
- The mini-op information is entered into the control unit during a high trap.
- When the control unit finishes a mini-op, a request is made for another high trap to enter a new mini-op.
- At the end of a command execution, the command-chain flag returns the operation for another CCW command.
- If another command is not chained, the operation branches to the ending routine to store status.

Data commands are processed with a sequence of mini-ops to control the operation. The command is decoded in the GPBB routine and branched to the appropriate routine for execution. The execution routine provides additional decoding to define fields or special modifiers for the operation. The routine provides tests for sequence requirements and file-mask conditions to determine whether the operation can be executed. For some operations, the physical position or orientation of the record requires variations in the operation sequence. After the starting point is determined for the operation, the first step of the execution is selected.

A link address is set and the operation is branched to the GPBH routine to set information into the control unit. This information includes the mini-op, the count, and the sync byte; it also includes setting the appropriate gates for the operation. When the controls are set, the operation is suspended until the control unit has executed the mini-op. If the command is the first of the chain, the initial sequence routine is ended with a return to I-cycles. When the entry is through the trap routine, the trap ends and the CPU returns to the interrupted routine.

When the control unit detects that the count has been reduced to 20, the high-trap request is set for the next mini-op information. High-trap entry is made through the GPBH routine to update the link address and return the operation to the data-sequence routine. After appropriate tests, a new link address is stored and the operation branched to the GPBH routine to set the mini-op information into the control unit. After entering the information, the trap is ended. This sequence is repeated until the command has been executed.

With the final entry through the high trap into the data-sequence routine, a further test is made for the command-chain bit. When it is present, the operation returns to the GPBA routine to enter another command. After decoding for the appropriate data-sequence routine, the operation branches to the indicated routine. The high-trap routine is not ended until the information has been sent to the control unit for the first step of the new command. If the command-chain bit was not present, the operation branches to the GPCG routine to store status and to end the operation.

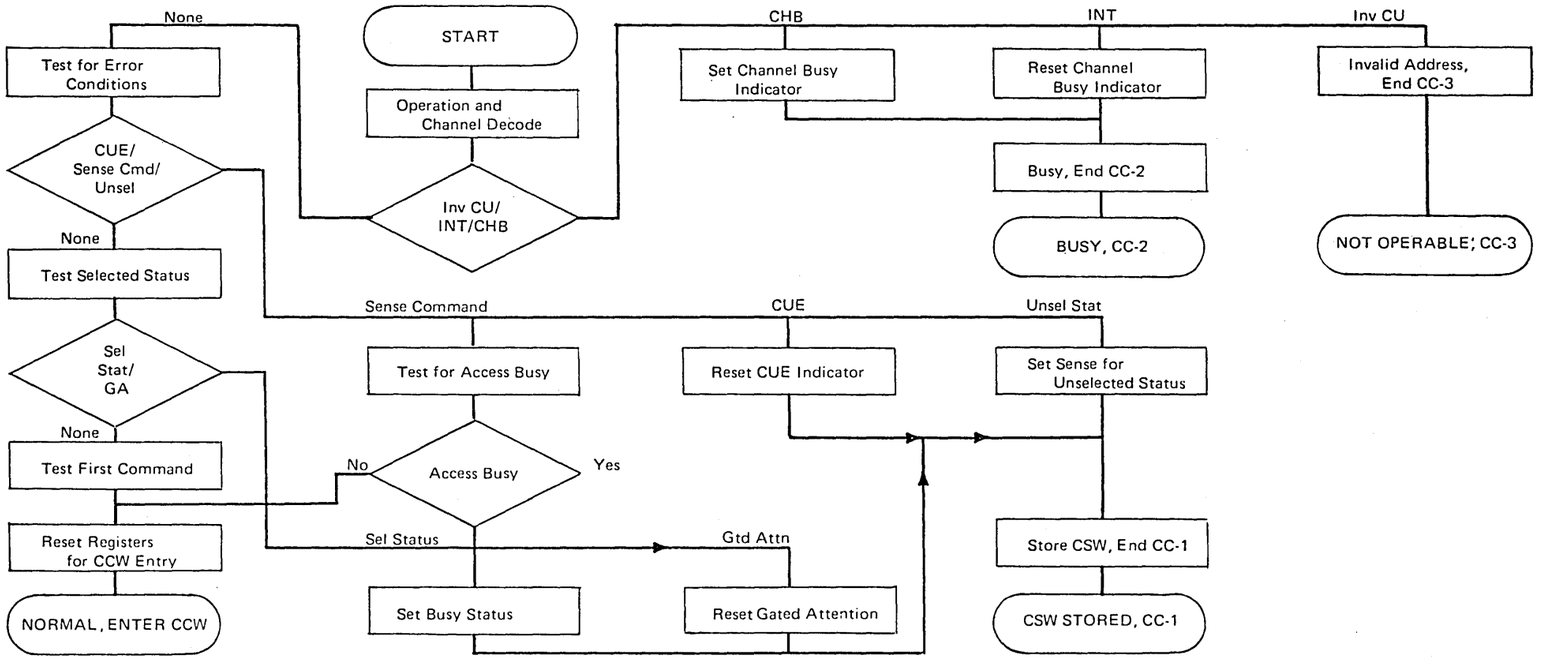


INITIAL SELECTION,START I/O

The channel Start I/O instruction is used to initiate all disk-file operations through the IFA. The IFA has been assigned the channel-1 address to the program. When the channel routines detect the channel-1 address, they branch the operation to the IFA. The CAW carries the normal CCW address to effect the operation. The IFA first determines the status of the control unit and the addressed drive for operation. If they are ready for operation, the initial selection is continued with the entry of the first command (CCW). If instead an interrupt is pending, the control unit is busy, or the control-unit address is invalid, the operation is ended with the condition code set to 1, 2, or 3 respectively.

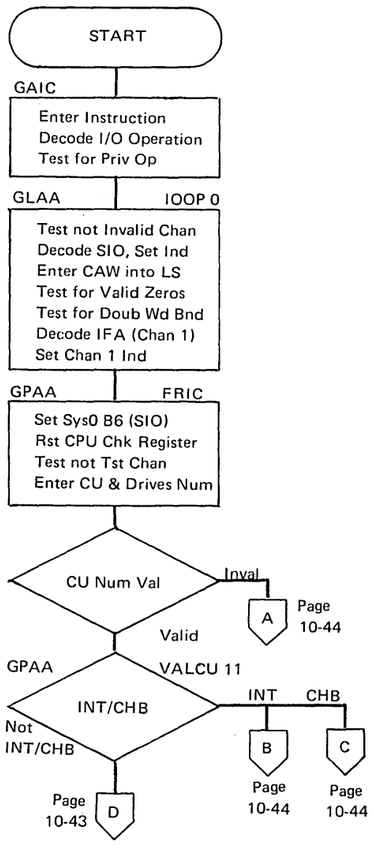
The initial selection routine continues by testing for CUE status, first command sense, and unselected status on the file interface lines. The addressed module is then selected to test for status conditions. If none of these conditions exist, the initial selection routine branches to the CCW entry and decode routine.

The status or error conditions result in ending the initial selection routine after posting the CC-1 status and storing the CSW.



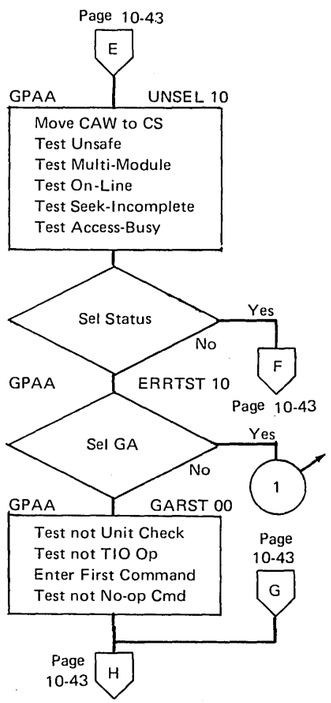
A Operation and Channel Decode

- The start I/O instruction is entered during the normal I-cycles.
- The instruction is decoded as a channel operation and branched to the channel routine.
- The CAW is entered and tested for validity.
- A decode of channel-1 branches the operation to the IFA routine.
- Tests are made for valid control-unit address, the interrupt latch, and the channel-busy latch to determine the routine to follow.
- A valid CU address and no INT or CHB condition is required to continue the initial selection.



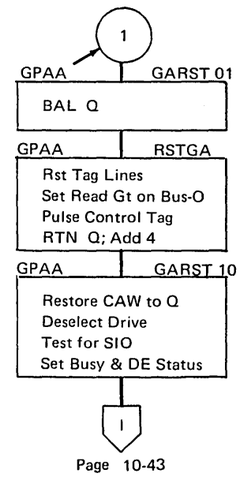
D Test First Command

- The first command is tested for a possible no-op or restore command that retains the old sense information.
- Zero sense bytes when the command is an active operation.



I Reset Gated Attention

- Set read gate on bus-out and pulse the control tag line to reset the module attention.
- Set busy and DE status.

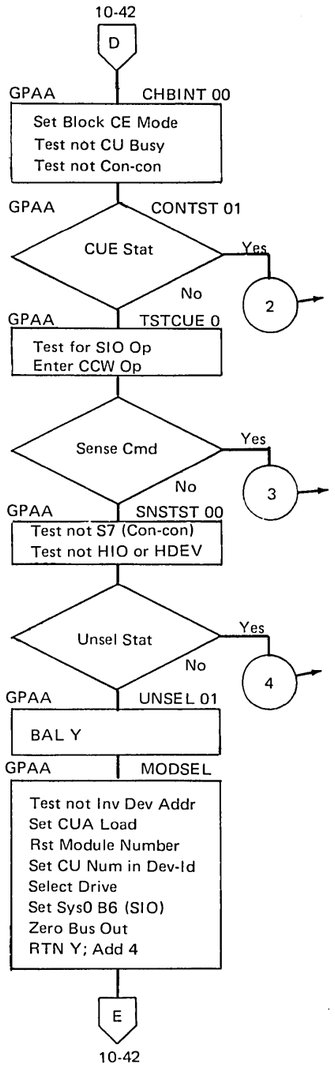


B Test for Error Conditions.

- A test is made for the erase-to-index from the previous operation.
- If the CUE status is pending, the routine is branched to the CC-1 ending.
- A test is made of the first command specified to determine whether it is the sense command.
- A test is made for any unselected status conditions indicating an equipment failure.
- If none of these conditions exist, the operations is allowed to continue with the initial selection.

C Test Selected Status

- The addressed module is selected to test for its operation conditions.
- The status lines are tested to branch any unusual conditions to the CC-1 ending.
- The 'gated attention' line is tested to determine whether the attention latch is set in the module.
- If none of the status conditions or the gated-attention are present the initial sequence routine is continued.



E Reset Registers for CCW Entry

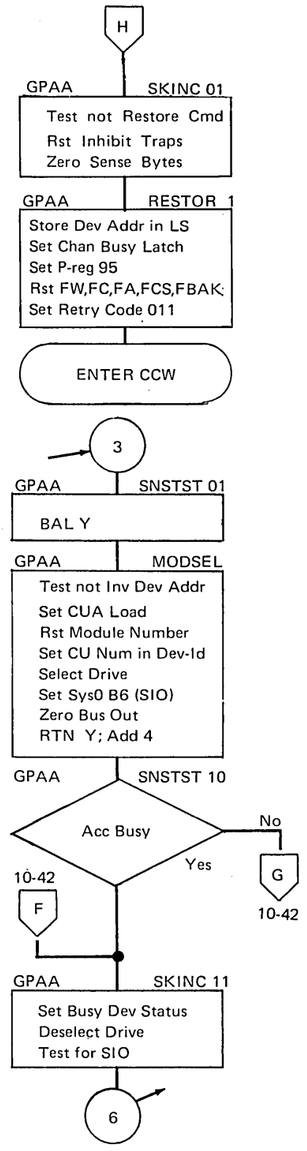
- The CCW address from the CAW is entered into local storage as the current CCW address.
- The operating registers are set to zero to start the operation with the first command.
- The retry code is set to code 001, indicating a successful initial selection.

F Test for Access Busy

- If the device address is valid, the CU address is set into the device-identifier register.
- After selecting the module, a test is made to determine whether the access is busy.
- When the access is not busy, the routine returns to the initial selection ending.

G Set Busy Status

- A busy access causes the routine to branch to the CC-1 ending to store CSW.



Reset CUE Indicator

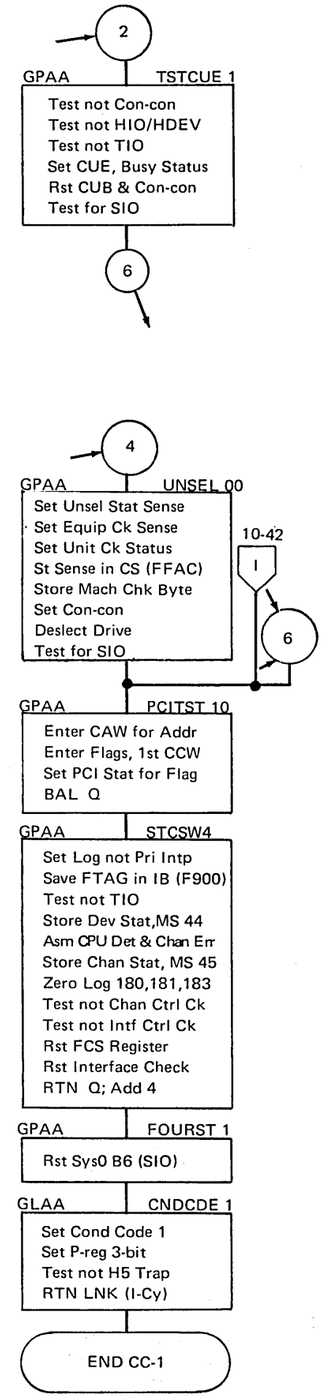
- Set CUE and busy status to indicate the condition to the program.
- Reset the control-unit busy and contingent-connection latches.

J Set Sense for Unselected Status

- Set unselected-status and equipment-check sense bits to indicate the condition.
- Set the unit-check status bit.
- Store sense information in control storage.
- Set CU and device address in control storage for reference.

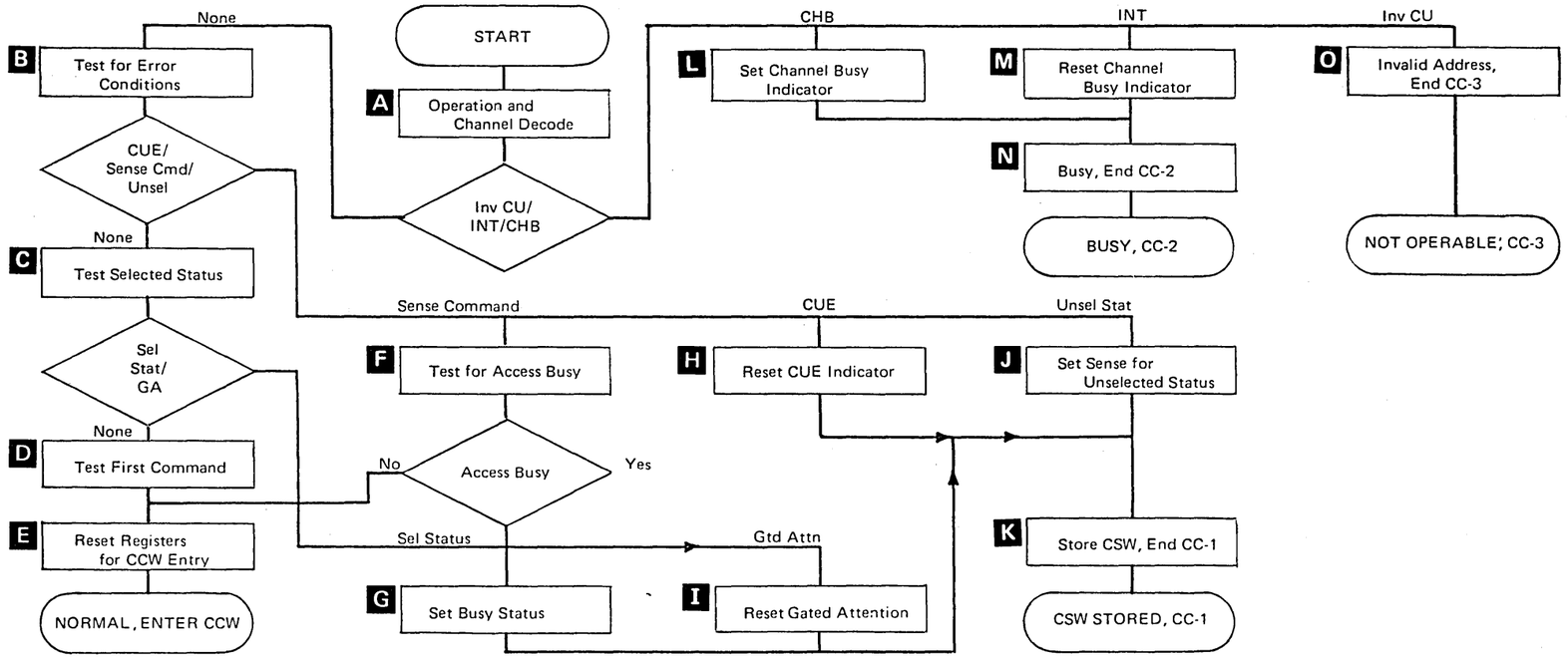
K Store CSW, End CC-1

- Set PCI status if the first CCW contained the PCI flag.
- Set log indication for not primary interrupt.
- Store channel and device status in CSW.
- If either channel-control check or the interface-control check status bits are set, the extended log is posted. (See "IFA Interrupt Routine.")
- The channel-busy latch is reset, and the module is deselected.
- The CC-1 indication is set, and the operation is returned to I-cycles.



The channel-busy and interrupt conditions that were tested early in the initial selection routine are branched to set the CC-2 indication and return the operation to I-cycles. The only operating difference for the two conditions is that the channel-busy indicator is set for the CHB condition.

When it is determined that the instruction control-unit address does not agree with that assigned to the IFA, the operation is ended after posting CC-3. If a device address is presented that is invalid, the operation ends with the CC-1 posted and the unit-check and intervention-required conditions posted.



L Set Channel Busy Indicator

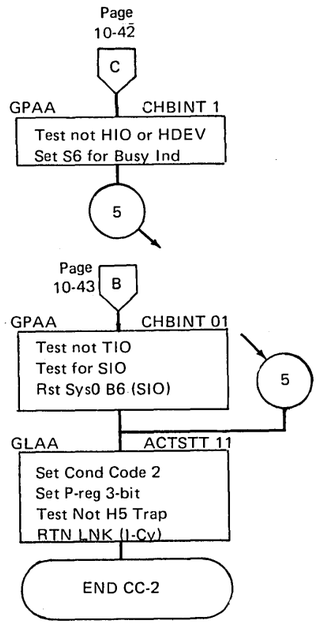
- The channel-busy indicator is set to show that the operation found the CHB latch set.

M Reset Channel Busy Indicator

- The channel-busy indicator is reset to show that the operation found the CHB latch reset.

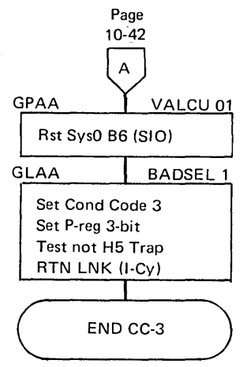
N Busy, End CC-2

- Reset the 'inhibit traps' latch before setting the ending condition.
- The CC-2 indication is set, and the operation is returned to I-cycles.



O Invalid Address, End CC-3

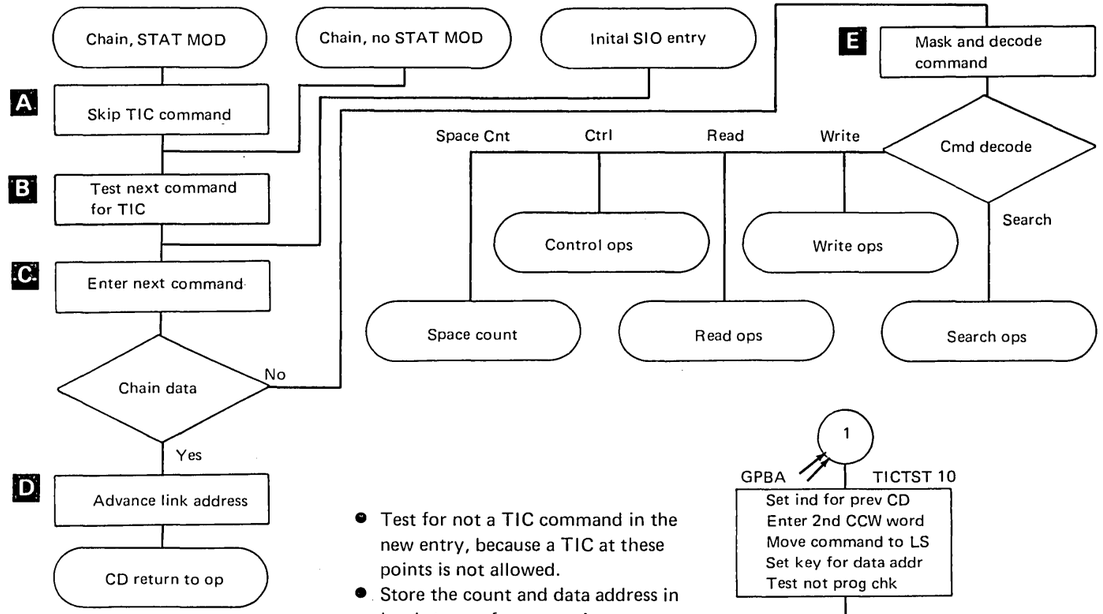
- Reset the inhibit traps latch before ending the routine.
- The CC-3 indication is set, and the operation is returned to I-cycles.

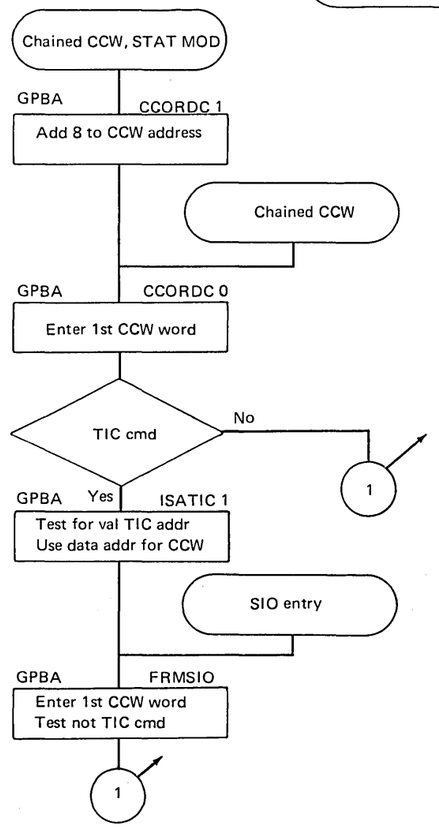


CCW Entry and Decode

The CCW entry and decode routine is used to enter new CCW as it is required for processing the operation. The CCW entry routine can be entered under four conditions. These include the initial selection, a CC entry with status modifier, a CC entry without status modifier, and a CD entry. The last two use a common entry but with separate exit conditions. In all cases the next CCW is entered and first tested for the TIC command. If the TIC is allowed, another command is entered from the TIC'ed to address (a second TIC is not allowed). The DC entry is exited after storing the new count and data address.

Initial selection and CC entries decode the new command to branch the operation to the appropriate routine for processing. The read-IPL and sense commands are routed to the control routine for secondary decoding.





A Skip TIC Command

- When entering with the status-modifier bit set, the CCW address is advanced by eight (one CCW).

B Test Next Command for TIC

- Enter the next command from the current CCW address
- A test is made for the TIC command that calls for entering the command from the data address of the CCW.
- If the first command entered is not a TIC, the routine is branched to retain the information.

C Enter Next Command

- Entry at this point is either from initial selection or a TIC'ed-to condition.

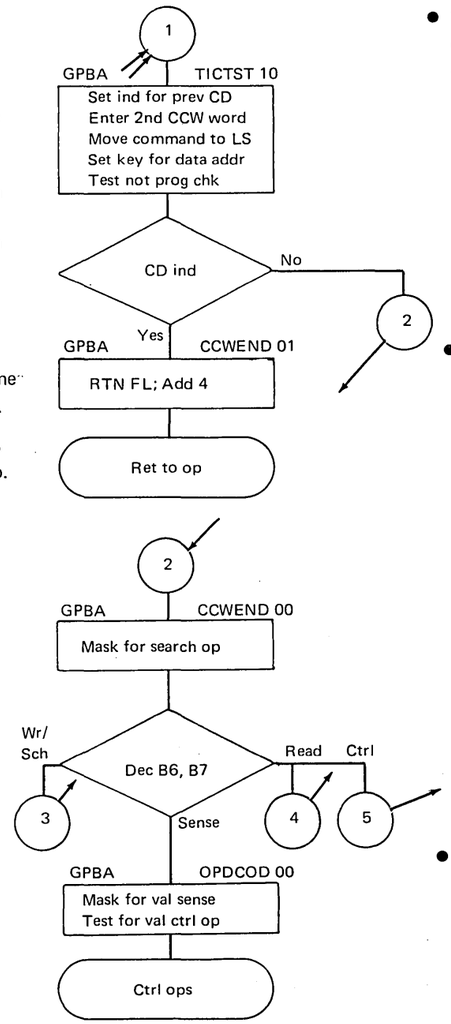
- Test for not a TIC command in the new entry, because a TIC at these points is not allowed.
- Store the count and data address in local storage for processing.
- Test for previous CD flag to branch the CD entry.

D Advance Link Address

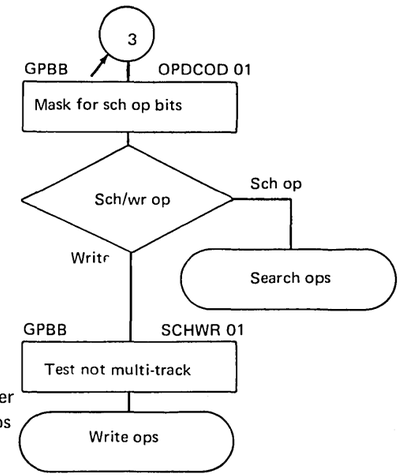
- A CD entry left its processing routine through a branch and link function.
- Four is added to the link address to re-enter the routine at the next step.

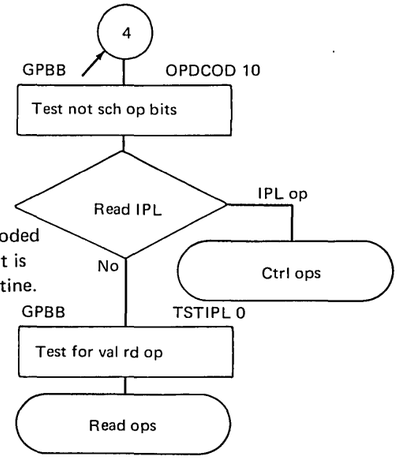
E Mask and Decode Command

- The four conditions of bits 6 and 7 of the command code define the initial decode.
- The sense branch after being tested for validity is branched to the control-op routine for processing.

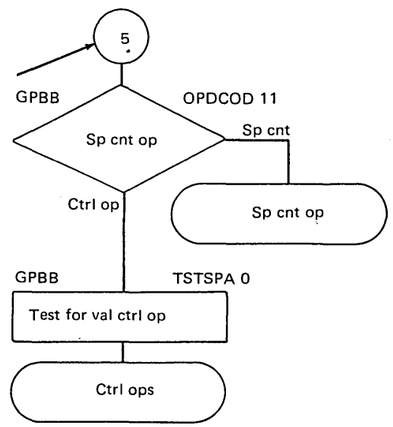


- The write/search branch is further decoded to separate the write-ops and the search-ops.





- The read branch is further decoded for the read-IPL command that is handled by the control-op routine.



- The control branch is further decoded for the space-count command that is handled by its own routine.

SENSE COMMAND

- The sense command reads up to six bytes of sense information stored in the file control into the specified address in main storage.
- The command can be executed only after the operating command sequence is completed or ended because the sense information is stored in the ending routine.

The sense command provides the means of reading the operation-developed sense information into CPU main storage. The data address of the command specifies the starting main-storage address for entry of the sense information. The normal count for the sense command is six bytes. The SLI flag must be set on if any other count value is used. The sense command is normally issued by the supervisor program in its analysis of the status conditions. The contents of the sense bytes are defined under "Status and Sense Indications."

The sense information is not stored until the operating sequence is completed or ended through some error condition. This means that the information is not available until the ending sequence has been completed. The contingent-connection indicator is set in the FGT external when the sense information contains an indicator that causes the unit-check status. If the contingent-connection indicator is set, the microroutine does not allow operation with a different file module. The sense command can be processed on the previously selected module only.

The sense command operation is more fully detailed under "Sense Command" in the following pages.

	Byte 0	Byte 1	Byte 2	Byte 3
Bit 0	Command Reject	Data Check in Count Field	Unsafe	Ready
Bit 1	Intervention Required	Track Overrun		On Line
Bit 2	Bus-Out Parity	End-of Cylinder	Serdes Check	Unsafe
Bit 3	Equipment Check	Invalid Sequence	Selected Status	Write Current Sense
Bit 4	Data Check	No Record Found	Cyclic-Code Check	Pack Change
Bit 5	Overrun	File Protected	Unselected File Status	End-of Cylinder
Bit 6	Track Cond Check	Missing Address Marker		Multi-Module Select
Bit 7	Seek Check	Overflow Incomplete		Seek Incomplete

Byte 4: Physical File Address (See Text)
Byte 5: Record Overflow Codes (See Text)

Sense Bytes

- Six sense bytes are provided to report the file conditions to the program.
- Sense information defines conditions occurring in the control unit and the disk drive reported in the status.

The file control has provisions to transfer six bytes of information during the execution of the sense command. These bytes contain the details of conditions indicated in the status bytes along with other information that may be required for recovery from an error condition. Those conditions that are a direct expansion of the status set the unit-check status bit when they occur. The chart of sense bytes shows the information in the first four bytes. The information for all six bytes is defined in the text.

SENSE BYTE-0

Bit-0 Command Reject: This bit indicates that the command presented cannot be performed. The command may be invalid or its information invalid. The sequence of commands or protected conditions may have been violated. An additional sense bit may define the reason for the rejection.

Bit-1 Intervention Required: This bit indicates that the addressed disk file is not connected or for some reason is not ready.

Bit-2 Bus-Out Parity: This bit indicates that a data-parity error was detected during the transfer of information from the CPU to the IFA. The check is made in the IFA.

Bit-3 Equipment Check: This bit indicates that an unusual condition was detected in the control unit or the disk file. The condition is defined by bits in sense byte 2.

Bit-4 Data Check: This bit indicates that an error was detected in the data coming from the file by the cyclic-code or BCA information. Reading into index and a wrong sync byte when oriented also set the data-check bit. An error in a count field also sets byte 1 bit 0. The bit is not set for clock-through fields.

Bit-5 Overrun: This bit indicates that the CPU did not respond to either a share-cycle or a trap request in time to execute the command properly.

Bit-6 Track Condition: This bit indicates that an operation was attempted on a track that was flagged as defective. In multi-track operation, switching to a defective track or switching from an alternate track also sets the bit.

Bit-7 Seek Check: This bit indicates that the file was unable to complete the seek sequence. The address may be incorrect or it may be a hardware failure. It may also occur as the result of multi-track switching if a head-compare check is detected.

SENSE BYTE 1

Bit-0 Data Check in Count Field: This bit indicates that the indicated data-check (byte 0, bit 4) was detected while reading the count field. The error was detected by the cyclic-code or BCA checks.

Bit-1 Track Overrun: This bit indicates that a write operation was not completed by the time the index point was detected.

Bit-2 End of Cylinder: This bit indicates that the file has advanced beyond the last address of the cylinder on a multi-track operation before the end of the command-chain sequence.

Bit-3 Invalid Sequence: This bit indicates that the sequence of commands (CCWs) has violated the accepted practice. The 'command reject' (byte 0, bit 0) is also set.

Bit-4 No Record Found: This bit indicates that the index point has been passed twice without finding the record on a single-track operation. It is also set with the 'missing address mark' (byte 1, bit 6) when the HA or R0 field cannot be found.

Bit-5 File Protected: This bit indicates that the file-mask provisions are violated by the command. This includes basically the seek and write commands, but it also includes multi-track operations if the head seek is inhibited.

Bit-6 Missing Address Mark: This bit indicates that two successive count fields read had the same sequence flag (bit 0). The intervening address mark was missed. The bit is set with 'no record found' (byte 1, bit 4) if the HA or R0 fields cannot be found.

Bit-7 Overflow Incomplete: This bit indicates that a record overflow operation has been stopped. The track condition (byte 0, bit 6) is also set when the overflow was to a defective track or from an alternate track. The 'file protected' (byte 1, bit 5), 'seek check' (byte 0, bit 7), or 'end of cylinder' (byte 1, bit 2) indicators are set when the respective conditions cause the stop.

SENSE BYTE 2

Bit-0 Unsafe: This bit indicates that an unsafe operating condition was detected in the file. These conditions include simultaneous read and write controls, multiple-head selection, and write/erase driver control failure.

Bit-1: Reserved

Bit-2 SERDES Check: This bit indicates that a bit has either been lost or gained when the parallel-transferred data from the CPU was serialized during a write operation.

Bit-3 Selected Status: This bit indicates that the microprogram decode of the selected file status has yielded contradictory results.

Bit-4 Cyclic-Code Check: This bit indicates that a malfunction in the operation of the cyclic-code checking hardware has been detected.

Bit-5 Unselected Status: This bit indicates that one of the file status lines is active without any of the files being selected. This indicates a malfunction of either the status or selection controls.

Bit-6: Reserved

Bit-7: Reserved

SENSE BYTE 3

These bits present the level of the selected file interface lines.

Bit-o Ready

Bit-1 On Line

Bit-2 Unsafe

Bit-3 Write Current Sense

Bit-4 Pack Change

Bit-5 End of Cylinder

Bit-6 Multi-Module Select

Bit-7 Seek Incomplete

SENSE BYTE 4

The bits of byte 4 identify the physical disk drive assigned to the given address. Only the four low-order bits are assigned. The same four bits are written as the low order of the BCA indicator byte at the end of each field.

Bits 0123 4567	Physical Drive
0000 0000	A
0000 0001	B
0000 0010	C
0000 0011	D
0000 0100	E
0000 0101	F
0000 0110	G
0000 0111	H
0000 1111	Module not defined.

Note: Module not defined occurs when the identifier plug for the given address has not been inserted.

SENSE BYTE 5

This byte is zero at all times except when an 'overflow incomplete' (byte 1, bit 7) occurs.

0123 4567	Hex	Interrupted Condition
0000 0110	06	A read command was in progress.
0000 0101	05	A write command was in progress.
0010 0101	25	A search-KD-equal was in progress, and the record is equal to this point.
0100 0101	45	A search-KD-high was in progress, and the record is equal to this point.
0110 0101	65	A search-KD-high or -equal was in progress, and the record is equal to this point.
0101 0101	55	A search-KD (any type) was in progress, and the record at this point is low. A search-KD-equal was in progress, and the record is unequal. The status-modifier must not be set for the ending.
0111 0101	75	A search-KD-high or search-KD-high/equal was in progress, and the record to this point is high. The status modifier must be set for the ending.

SENSE BIT ERROR CONDITIONS

The sense bit error condition chart shows the common sense display of bytes 0 and 1. The bits in the remaining bytes carry their individual definition and may occur with these displays. The chart shows the bit combinations in their left to right display sequence with byte 0, bit 0 shown first.

Sense Indication	Explanation	Error
Byte 0, B0 Command Reject	The IFA has received an invalid command code.	Program
Byte 0, B7 Seek Check	An invalid seek address was received by the IFA.	Program
Byte 1, B7 Overflow Inc.	Overflow has been attempted, either to a defective track or from an alternate track.	Program
Byte 1, B3 Invalid Seq	The IFA has received an invalid sequence of commands.	Program
Byte 1, B5 File Protected	A command that violates the file mask has been issued.	Program
Byte 0, B1 Intervention Req	The specified file is not "on line" and ready.	Equipment
	The specified file is not available.	Equipment
Byte 0, B3 Equipment Check	An unusual condition has been detected. (The condition is indicated in sense byte 2.)	Equipment
Byte 2, B0 Unsafe	A File malfunction has been detected.	Equipment
Byte 0, B4 Data Check	A data error has been detected by the cyclic-check during the reading of the file record.	Equipment
	Wrong sync byte detected with the control oriented.	Equipment
	Read operation read into index.	Equipment
Byte 1, B0 Data Chk Cnt	The data error occurred while reading a count field.	Equipment
Byte 0, B5 Overrun	The CPU did not respond to either a share request or a trap request in time to execute the command.	Equipment
Byte 0, B6 Track Condition	An operation other than HA or R0 was attempted on a defective flagged (B6=1) track.	Program
	A multiple track operation has attempted to switch to the next track from an alternate track.	Program
Byte 0, B7 Seek Check	A seek operation is incomplete because of hardware failure.	Equipment
	Home address did not compare on multiple track operation head switch.	
Byte 1, B7 Overflow Inc	Home address did not compare on an overflow operation head switch.	Equipment
Byte 1, B1 Track Overrun	Writing was not completed by the time the index point was detected.	Program
Byte 1, B2 End of Cylinder	An end of cylinder condition was reached before satisfying the CCW chain in a multiple track operation.	Program
Byte 1, B4 No Record Found	Two index points were detected on a single track without finding a match condition.	Program
Byte 1, B6 Miss Addr Mark	The HA or R0 field could not be found.	Equipment
Byte 1, B5 File Protect	A multiple track operation violates the seek mask.	Program
Byte 1, B7 Overflow Inc	An overflow operation violates the seek mask.	Program
Byte 1, B6 Miss Addr Mark	Two successive count fields had equal flag bit 0 settings indicating a skipped record.	Equipment

Sense Bit Error Conditions

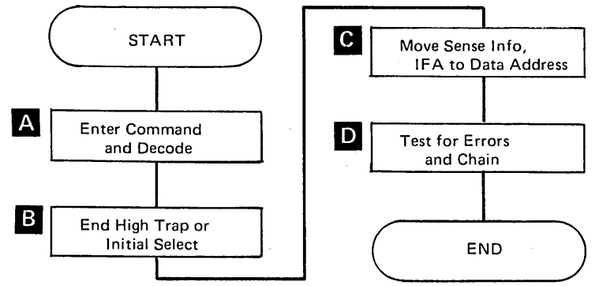
Sense Command

SIO–CC

The sense command provides the means of transferring the six bytes of developed sense information to main storage. The sense information contains the error conditions that resulted from the last operation. The sense command is normally the first command of a chain issued by an error-correction routine. It cannot be the last command of a data sequence because the sense information is not posted until the ending routine. It is assumed for this diagram that the operation originates with the SIO instruction and ends chained to the next command.

The sense command is decoded as a control command. The initial selection or high-trap sequence entering the command is ended, and a low-priority trap requested to process. If the control unit had been previously oriented with the disk file, the orientation controls are reset.

The sense command stores the sense information at the designated data address of the command. This information has been retained in control storage and external registers by the control unit.



	Byte 0	Byte 1	Byte 2	Byte 3
Bit 0	Command Reject	Data Check in Count Field	Unsafe	Ready
Bit 1	Intervention Required	Track Overrun		On Line
Bit 2	Bus-Out Parity	End-of Cylinder	Serdes Check	Unsafe
Bit 3	Equipment Check	Invalid Sequence	Selected Status	Write Current Sense
Bit 4	Data Check	No Record Found	Cyclic-Code Check	Pack Change
Bit 5	Overrun	File Protected	Unselected File Status	End-of Cylinder
Bit 6	Track Cond Check	Missing Address Marker		Multi-Module Select
Bit 7	Seek Check	Overflow Incomplete		Seek Incomplete

Byte 4: Physical File Address (See Text)
Byte 5: Record Overflow Codes (See Text)

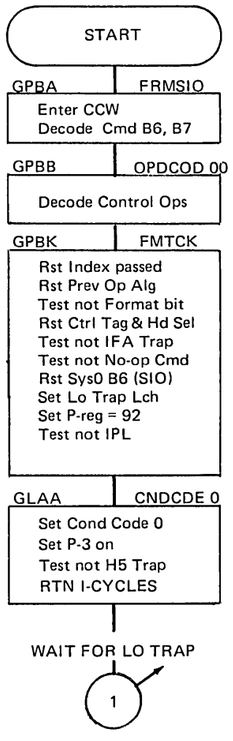
Sense Bytes

A Enter and Decode Command

- The command is entered by the GPBA routine after entry from the SIO initiation.
- The sense command is decoded as a control command, causing a branch to the GPBK routine.
- The control-unit orientation is ended if the operation enters from a command-chain sequence (except when the format bit was set in the previous operation).

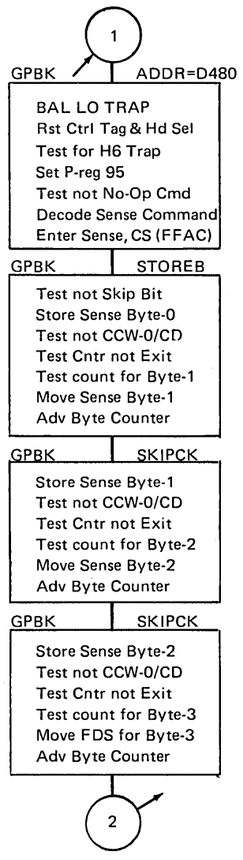
B End High Trap or Initial Selection

- Entry from the SIO initiation or the high-priority trap results in requesting a low-priority trap for processing.
- The initial selection entry ends the SIO instruction and returns to I-cycles after posting CC-0.
- An entry from the high trap to enter the command results in ending the trap and returning to the link address.



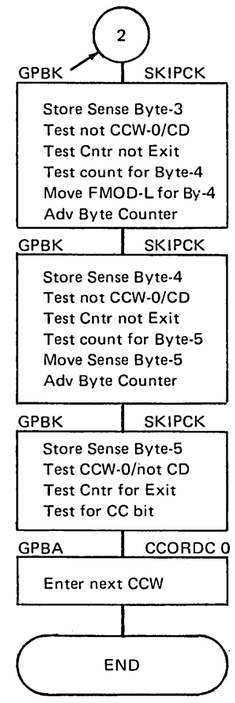
C Move Sense Information, IFA to Data Address.

- Sense bytes 0, 1, 2, and 5 are entered from control storage.
- Sense byte 3 is the disk-file status as indicated by the FDS external.
- Sense byte 4 contains the addressed module number as stored in the lower portion of FMOD.
- The six sense bytes can be stored in different locations by a sequence of commands with the chain-data flag.
- A partial transfer of the sense information can be made if the SLI flag is set.



D Test for Errors and Chain

- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for ending.
- The CC flag is tested to determine whether another command should be entered or the operation should be ended.



REMEMBER

There is a Reader's Comment Form at the back of this publication.

CONTROL COMMANDS

- Control commands are used to initiate a function sequence and to set the file mask; no data transfer is made to/from the file.
- The seek and recalibrate commands position the selected file access and select the head.
- The file mask inhibits all or a portion of the seek and write commands to protect the written file.
- The space-count command allows recovery of data fields of the record when the count field is defective.

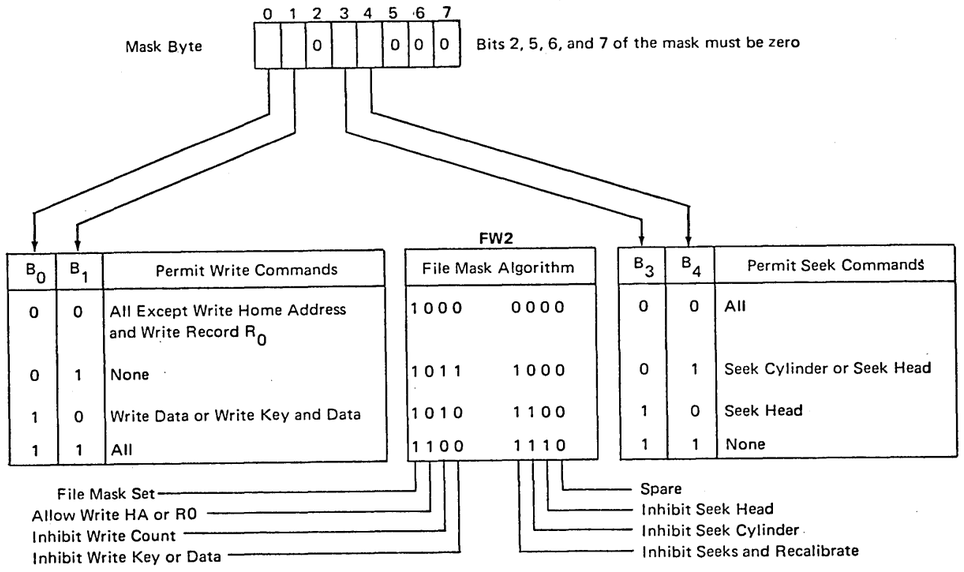
Command Codes	Single Track Hex	Single Track Binary	Multi-Track Hex	Multi-Track Binary
Control,				
No Operation	03	0000 0011	--	---- ----
*Set File Mask	1F	0001 1111	--	---- ----
Restore (2321 Cmd)	17	0001 0111	--	---- ----
*Recalibrate	13	0001 0011	--	---- ----
*Seek	07	0000 0111	--	---- ----
Seek Cylinder	0B	0000 1011	--	---- ----
Seek Head	1B	0001 1011	--	---- ----
*Space Count	0F	0000 1111	--	---- ----

The control commands perform a number of non-data functions that with the exception of the seek commands have little in common with each other. All except the space-count command disorient the file-control timing relation with the disk file. Although these commands do not involve the transfer of file data, some of them do require one or more bytes of information from main storage to augment the command. This information is transferred during the initiating microroutine except for the case of the space-count command.

No Operation (No-Op) command performs no actual function. Because it disorients the file-control timing, the command can be used to skip a record when it is placed between a search and a read command. This same condition causes a malfunction when the operation is not intended. Attempting a write command following the no-op command results in an invalid-sequence indication. The no-op command transfers no information to or from main storage. Except when the erase-to-index function is in progress, the no-op command is performed as a command immediate operation.

**Set File Mask* command is used to insert the one-byte file mask that defines the permissive write and seek functions that may be performed. Without inserting the file mask, all seek commands and write commands except the write-HA and write-R0 commands may be executed. The mask may be inserted at the start of the command sequence or at some later point. It should not be set between related data commands, because the file timing is disoriented. The set-file-mask command must not be set twice within a command sequence or the invalid sequence indicator is set. The data address of the command is for the one-byte mask. The file-mask chart shows the bit assignments for the mask byte and the algorithm set in local storage.

*These commands are more completely detailed on the following pages.



File Mask Decode and Algorithm

Restore (2321) command is not used for the IFA operation, but the command is not rejected. The reaction to the command by the control unit is the same as for the no-op command. The file-control timing is disoriented. The restore command is not performed as a command immediate.

**Recalibrate* command causes the selected file access arm in the 2319 to seek to cylinder 000. The head and cylinder addresses for the file are set to zero for subsequent operations. The recalibrate command does not follow the sequence used for the normal seek operations. The command defines the new address without additional information bytes from main storage. If seeks are inhibited by the file mask, the recalibrate command cannot be executed.

**Seek* command causes the selected file access arm in the 2319 to move to a new cylinder location and to set a new head selection. The data address of the command locates the six-byte file address located in main storage to be used for the operation. The seek address diagram defines the content of the address bytes. Bytes 0, 1, 2, and 4 are not required to address the 2319 files, but they must be set to zero to conform with other file systems. The file control compares the new cylinder address to the address presented by the selected file as its current address. From these figures, the direction and the amount of movement is determined and supplied to the selected file. The new address is then set into the selected file cylinder address register (CAR). The head address specified by the file address is also sent to the selected file without testing the old address. If full seeks are inhibited by the file mask, the seek command cannot be executed.

	BIN		CYL		HD	
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Binary	0000 0000	0000 0000	0000 0000	0000 0000 to 1100 1010	0000 0000	0000 0000 to 0001 0011
Hexadecimal	00	00	00	00 to CA	00	00 to 13
Decimal	00	00	00	00 to 202	00	00 to 19

Six Byte Seek Address

Seek-Cylinder command for the IFA is exactly the same as the full seek command. This command does not use the two high-order bytes (Bin) of the file address and is required for selection in some file systems. The high-order bytes are still tested for a valid address (zeros). For the IFA, the file mask is interpreted to allow full-seek or cylinder-seek for either condition.

Seek-Head command causes the selected file in the 2319 to change its head selection without moving the access arm. The four high-order bytes of the file address are not used, but they must be a valid IFA file address. The new head address is set into the selected file head address register without changing the cylinder-address register. If head seeks are inhibited by the file mask, the seek-head command cannot be executed.

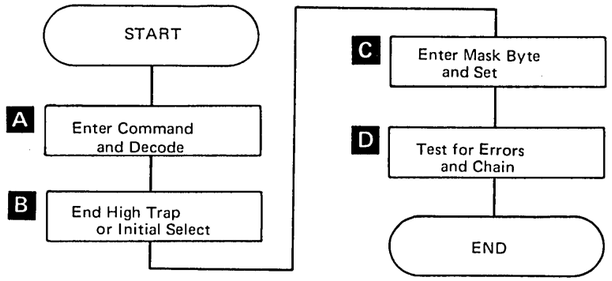
**Space-Count* command is used as an aid in recovering information from a defective track. If a poor read area is detected in the count field of a record, it is possible to retrieve the key and data fields by using the space-count command. The command attempts to identify the count-field location but does not enter the information or post an error. The data address of the command command defines three bytes in main storage to be used as the key length and the data length in reading the record. The defective record is usually located through a search on the previous record.

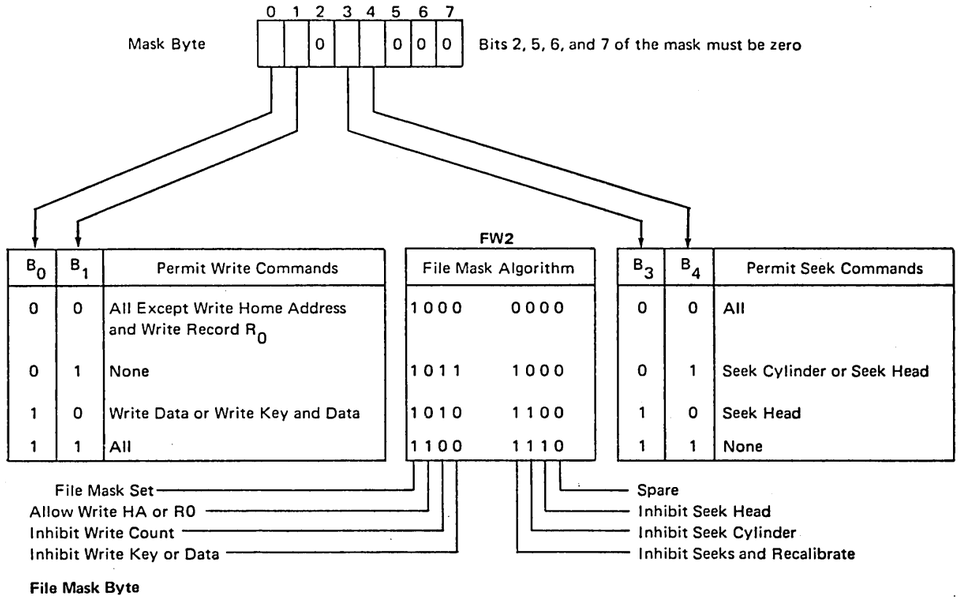
Set File Mask, High Trap - CC

The set-file-mask command provides the means of entering a mask byte that defines the allowable write and seek commands to be processed. The entered mask applies for the remainder of the chaining sequence. The mask is reset at the end of the chaining sequence. The mask can be set at any point in the sequence, but it can not be changed with another mask command. In the reset status, all seeks are allowed and all write operations except write-HA and write-R0. It is assumed for this diagram that the set-file-mask command is the first in the sequence and that it is chained to another command.

The set-fuel-mask command is decoded as a control command. The initial selection or high-priority trap sequence entering the command is ended and a low-priority trap requested to process. If the control unit had previously been oriented with the disk file, the orientation controls are reset.

The mask is entered from the specified data address and decoded. The assigned bits are translated into an algorithm as shown in the chart. The algorithm is stored in local storage register FW2 for use.





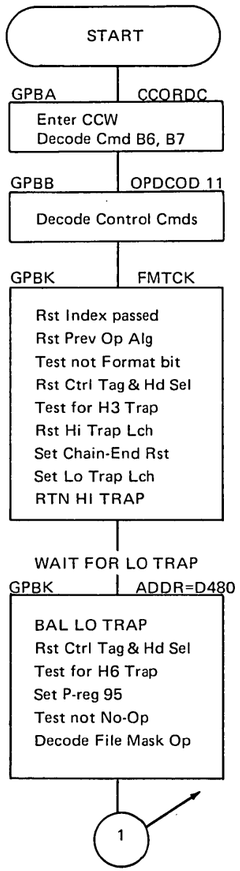
File Mask Byte

A Enter Command and Decode

- The command is entered by the GPBA routine with entry from either the SIO initiation or chaining.
- The set-file-mask command is decoded as a control command, causing a branch to the GPBK routine.
- The control-unit orientation is ended if the operation enters from a command-chain sequence (except when the format bit was set in the previous operation).

B End High Trap or Initial Selection

- Entry from the SIO initiation or the high-priority trap results in requesting a low-priority trap for processing.
- The initial selection entry ends the SIO instruction and returns to I-cycles after posting CC-0.
- An entry from the high trap to enter the command results in ending the trap and returning to the link address.

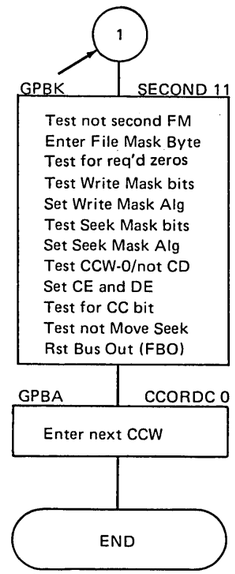


C Enter Mask Byte and Set

- The mask byte is entered from the data address and tested for zeros in bits 2, 5, 6, and 7.
- Bits 0 and 1 are decoded to set the write algorithm.
- Bits 3 and 4 are decoded to set the seek algorithm.

D Test for Errors and Chain

- The command flags are tested for the appropriate bits to allow continuing the operation.
- If the CC flag is present, the next command is entered.

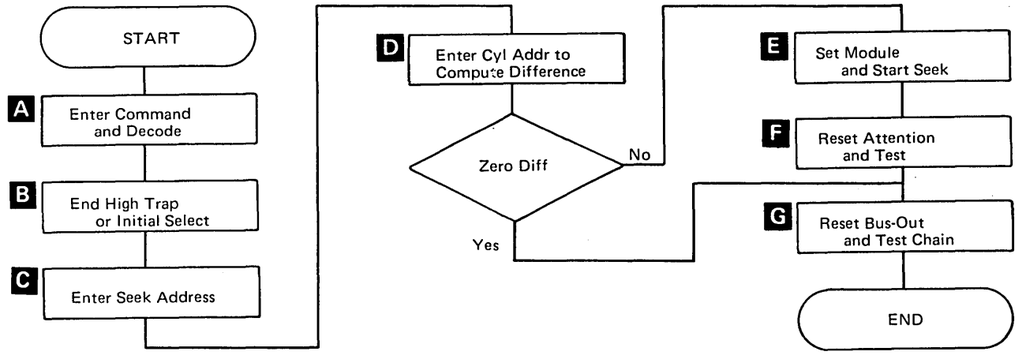


Seek Commands, SIO - CC

The seek commands provide the means of moving the access mechanism of the selected module from its existing position to the address designated. At the same time, a new head may be selected. The three seek commands are handled by the same microroutine. The resulting operation follows one of two patterns defined by whether the access arm is moved. A seek command may be issued at any point in the command sequence, but the type of seek specified must be allowed by the file mask. It is assumed for this diagram that the seek command is originated by the SIO instruction and ends chained to the next command.

The seek commands are decoded as control commands. The initial selection or the high-trap sequence entering the command is ended, and a low-priority trap is requested to process. If the control unit had been previously oriented with the disk file, the orientation controls are reset.

The seek command enters a six-byte address from the address designated by the data address of the command. Byte 3 contains the cylinder address to which the access arm is to be moved. Byte 5 contains the head address to be selected. The remainder of the bytes must be zeros.



	BIN		CYL		HD	
	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
Binary	0000 0000	0000 0000	0000 0000	0000 0000 to 1100 1010	0000 0000	0000 0000 to 0001 0011
Hexadecimal	00	00	00	00 to CA	00	00 to 13
Decimal	00	00	00	00 to 202	00	00 to 19

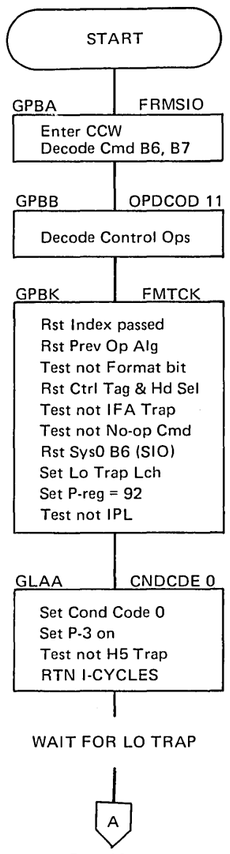
Six-Byte Seek Address

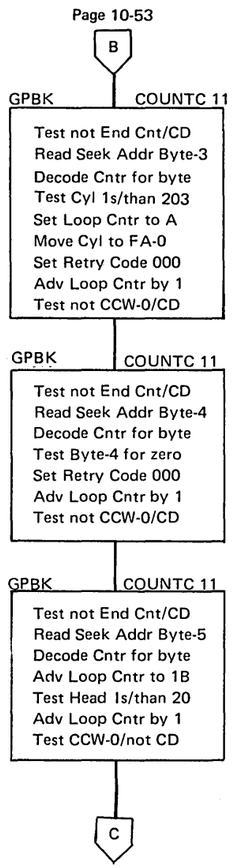
A Enter Command and Decode

- The command is entered by the GPBA routine after the entry from the SIO initiation.
- The seek commands are decoded as control commands, causing a branch to the GPBK routine.
- The control-unit orientation is ended if the operation enters from a command-chain sequence (except when the format bit was set in the previous operation).

B End High Trap or Initial Selection

- Entry from the SIO initiation or the high-priority trap results in requesting a low-priority trap for processing.
- The initial selection entry ends the SIO instruction and returns to I-cycles after posting CC-0.
- An entry from the high trap to enter the command results in ending the trap and returning to the link address.



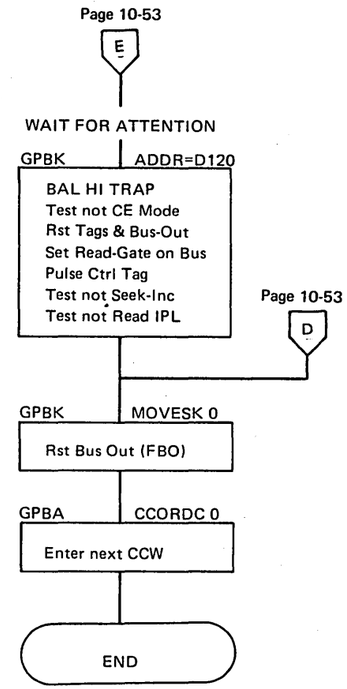


F Reset Attention and Test

- The module attention is reset by setting the read-gate bit on the bus and pulsing the control-tag line.
- The seek incomplete indicator is tested to determine that the operation was completed by the module.

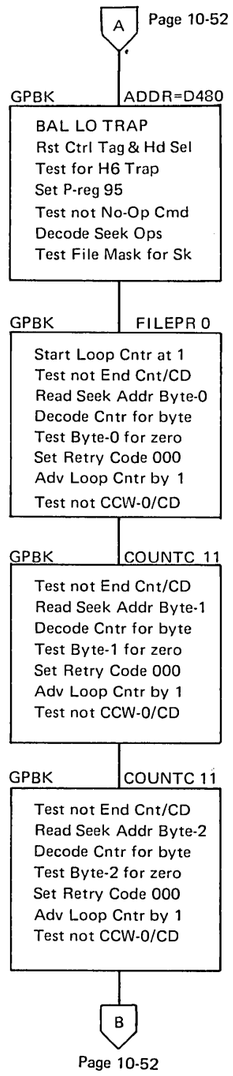
G Reset Bus-Out and Test Chain

- The file bus-out register is cleared before the operation is ended.
- If the CC flag is present, the operation enters the next command: otherwise, the operation is ended.



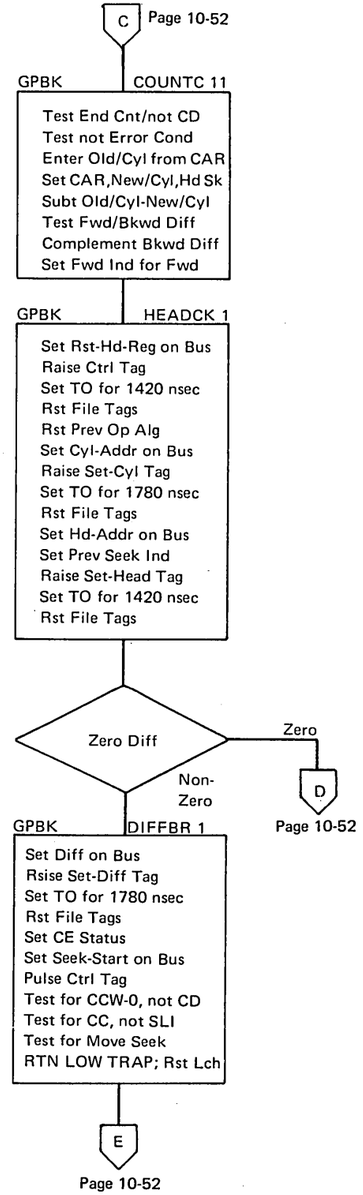
C Enter Seek Address

- The six bytes of the seek address are entered one byte at a time and tested.
- Bytes 0, 1, 2, and 4 are tested for zeros.
- Byte 3 containing the cylinder address is tested for validity (less than 203).
- A further test of an invalid address is made to determine whether it is 255 used to force a seek incomplete.
- Byte 5 containing the head address is tested for validity (less than 20).



D Enter Cylinder Addresses to Compute Difference

- The selected module CAR register is entered for the old cylinder address.
- A difference of zero or a head seek command is branched after setting the head register to the specified address.
- Set module cylinder address register and time out for set.
- No access movement takes place when the cylinder address is not changed.
- A backward seek difference is complemented. A forward seek difference sends a direction indicator to the module with the head address.



E Set Module Registers and Seek Start

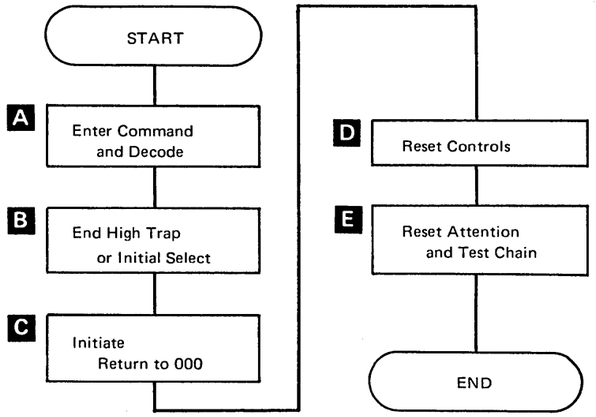
- Set difference register and time out for set.
- Set seek-start on bus and pulse control tag.
- Wait for module to raise the gated attention signal when the seek is completed.

Recalibrate, SIO - CC

The recalibrate command is used to return the access mechanism to cylinder 000 when it has been determined that the file is not reading the cylinder specified by the previous seek. In normal operation the recalibrate command is the first of a chaining sequence, but this is not a requirement. If a set-file-mask command is issued before the recalibrate command, the mask must permit cylinder seeks. It is assumed for this diagram that the operation originates with the SIO instruction and ends chained to the next command.

The recalibrate command is decoded as a control command. The initial selection or high-trap sequence entering the command is ended, and a low-priority trap is requested to process. If the control had been previously oriented with the disk file, the orientation controls are reset.

No seek address is required with the recalibrate command because the return to zero implies the address. The head and cylinder address registers are reset to indicate the zero addresses. The attention line is raised by the selected module in the same manner as for the normal seek operation.

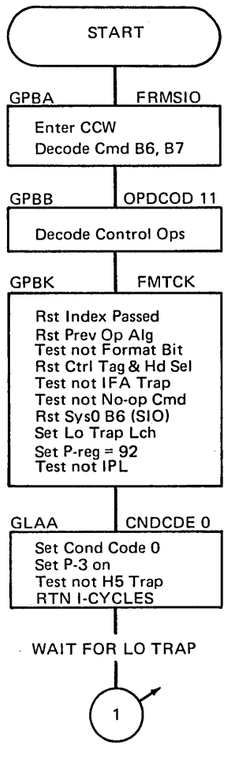


A Enter and Decode Command

- The command is entered by the GPBA routine after entry from the SIO initiation.
- The recalibrate command is decoded as a control command, causing a branch to the GPBK routine.
- The control-unit orientation is ended if the operation enters from a chain-command sequence (except when the format bit was set in the previous operation).

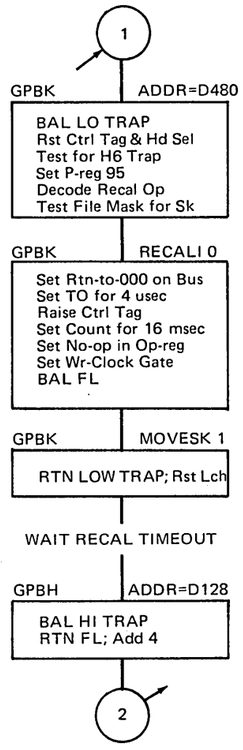
B End High Trap or Initial Selection

- Entry from the SIO initiation or the high-priority trap results in requesting a low-priority trap for processing.
- The initial selection entry ends the SIO instruction and returns to I-cycles after posting CC-0.
- An entry from the high trap for a new command results in ending the trap and returning to the link address.



C Initiate Return-to-000

- The decoded recalibrate command tests the file mask to determine whether full seeks are permitted.
- The return-to-000 bit is set on the bus, and the control tag line is pulsed.
- A no-op mini-op with a count for 16 milliseconds is used to time out the control delay.
- The write-clock gate is set to advance the counter.

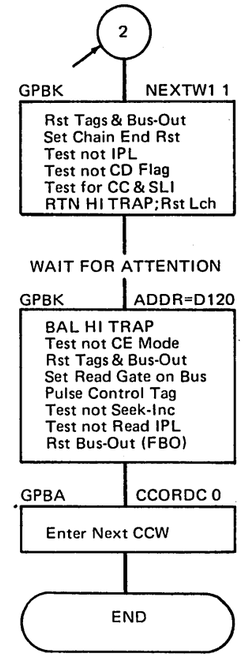


D Reset Controls

- Following the recalibrat time-out, the bus-out and tag lines are cleared.
- The 'chain end reset' line is raised to reset the control-unit latches before the operation is continued.
- Tests are made for the appropriate command flags to continue the operation.
- The control operation waits for the gated attention response from the addressed module before continuing.

E Reset Attention and Test Chain

- The module attention is reset by setting the read-gate bit on the bus, and the control tag line is pulsed.
- The seek-incomplete indicator is tested to determine that the operation was completed by the module.
- If the CC flag is present, the operation enters the next command; otherwise, the operation is ended.

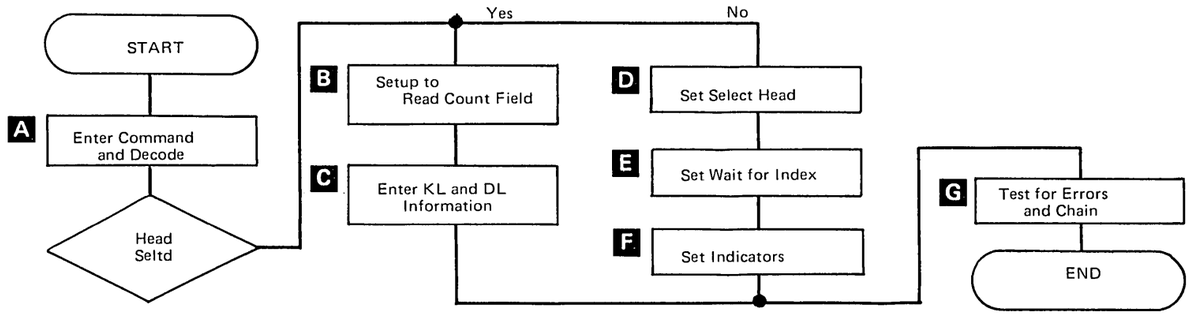


Space Count Command, CC - CC

The space-count command provides the means of salvaging data from records that have developed a defect in the count field. While processing the command, the KL and DL information is entered from the data address specified in main storage. The key and data-field information can then be read in the usual manner with the appropriate command chained.

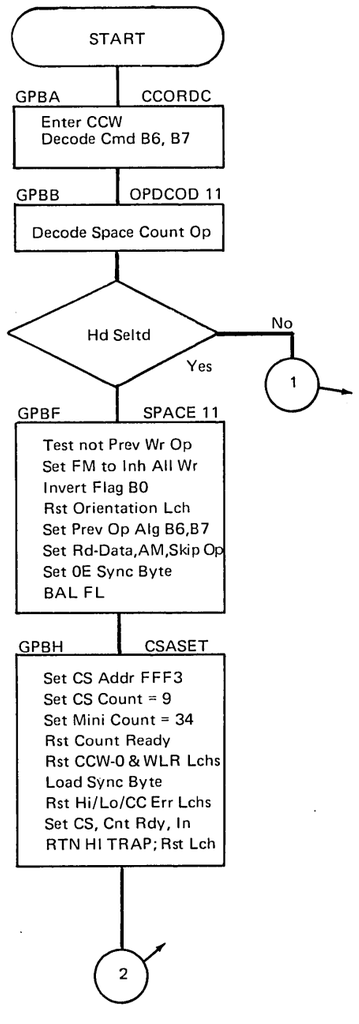
The space-count command is normally issued chained from a previous command to orient the control unit. When chained to a search-ID, the space-count command applies to the following record. If the space-count command is issued when the head is not selected, the command selects the head and orients the control unit to the index point. When followed by a second space-count command, the operation applies to R0.

The routine for the space-count command is similar to that of the read-count command except that the AM and skip modifiers are set in the mini-op. Following the dummy read operation, the KL and DL information is entered from main storage.



A Enter Command and Decode

- The space-count command normally enters chained from a previous command.
- The space-count command is decoded first as a control command and then is branched to a special routine.

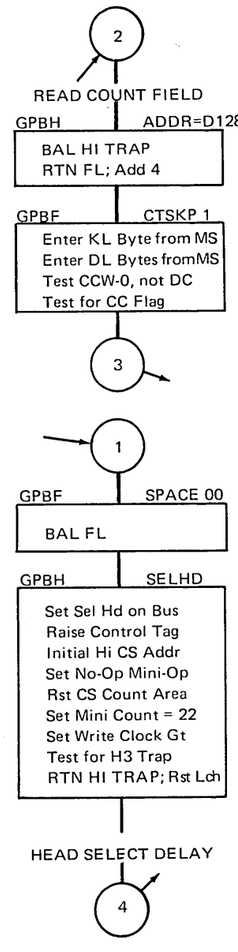


B Setup to Read-Count Field

- Test not previous write operation and set the file-mask to inhibit all write commands.
- The read-data mini-op with the AM and skip modifiers is set to clock-through the count field.
- The control-storage address and count information for entry of the count field.
- The 0E sync byte is entered to identify the count field.
- The read data into control-storage gating is set for the operation.
- The flag B0 is inverted to show the correct odd/even indication.

C Enter KL and DL Information

- The KL and DL information is entered into control storage from the data address specified by the CCW.
- The command flags are tested for the appropriate bits to continue the operation.

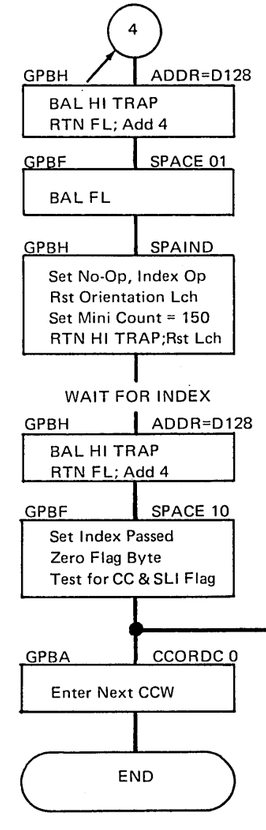


D Set Select Head

- If the head is not selected when the command is entered, the head is selected and the control unit is oriented to the index point.
- Set select-head bit on the bus-out and raise the control tag line.
- Set count of 22 for selection time-out.
- The write-clock gate is raised to control the countdown.

E Set Wait for Index

- Set the no-op mini-op with the index modifier to locate the index point.
- The orientation latch is reset, and a count of 27 is entered for the delay.



F Set Indicators

- The index-passed latch is set to indicate starting at the beginning of the track.
- The flag byte is zeroed because the first record is R0.
- The command flags are tested for appropriate bits to allow continuing the operation.

G Test for Errors and Chain

- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for ending.
- When the CC flag is present, the next command is entered.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

WRITE COMMANDS

- Write commands divide into track-formatting commands and data-updating commands.
- Format commands are used in a sequence to create the initial track record format.
- The data-updating commands write new field information within the formatted record.
- The erase command in effect destroys the data of a written record because the area is rewritten without address marks.

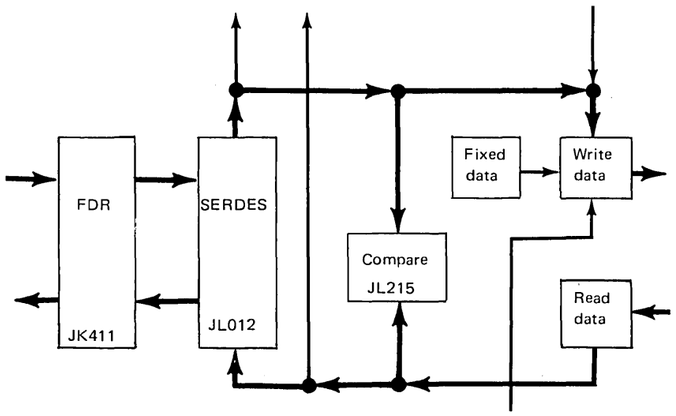
Command Codes	Single Track Hex	Single Track Binary	Multi-Track Hex	Multi-Track Binary
Write			----	----- -----
*Home Address	19	0001 1001	----	----- -----
Record Zero	15	0001 0101	----	----- -----
*Count/Key/Data	1D	0001 1101	----	----- -----
Special CKD	01	0000 0001	----	----- -----
Erase	11	0001 0001	----	----- -----
*Data	05	0000 0101	----	----- -----
Key and Data	0D	0000 1101	----	----- -----

All of the write commands, including the erase command, move information from main or control storage to write on the specified area of the track record. The write information for home-address and count fields is transferred as a block to control storage and then written from there. Because the write operations destroy any previously written information, the file control requires a chaining sequence that defines the track and record before writing. The write information for home-address and count fields is transferred as a block to control storage and then written from there. Because the write operations destroy any previously written information, the file control requires a chaining sequence that defines the track and record before writing. The write commands divide into two groups. The first five commands in the list represent one group used to format track records. The last two commands in the list represent the second group and are used for normal record updating. A write operation starts in the gap following the 20-byte post-record area of the previous field and continues through the post-record gap of the current field. When two consecutive fields are written, the write controls remain set.

The format write commands are used to define the content and capacity of records on a track. Writing the count field defines the number of bytes that can be stored in the key field, if used, and the data field. Until the track format is changed, each field can accommodate only data records of the length defined by the count field. If the suppress-length-indication flag is set, a shorter length field can be written but the file control fills the remainder of the record with zeros and the cyclic-check bytes are written at the end of the defined field. Following a format write command, the remainder of the track is erased unless another format write command is chained.

*These commands are more completely detailed on the following pages.

During all write operations, information from main storage enters the IFA FDR register over the EBI using share cycles. As required, the data byte is transferred from the FDR to SERDES and a share-cycle request refills the FDR. The bytes are serialized by the SERDES to feed the write-data generation circuits, where it is gated by the clock to develop double-frequency clock/data information. This information is fed to the 2319, where the module switching gates the selected file co-ax to carry the information to the file write circuits. The special bytes used in the associated gap areas are developed in the write-data generation circuits.



***Write Home Address (HA)** command writes a new home-address field on the track after locating the index point. This is a track-formatting command. This is the only write command that can be written without a previous search command or a continuing write sequence. It does require a previous setting of the file mask to allow writing the home-address field.

Write Record Zero (R0) command writes the count, key, and data fields of the track descriptor record. This is a track-formatting command. This command requires either a previous write-HA command or a successful search-HA command to execute. A file mask that allows the write-R0 record must have been previously set.

***Write Count, Key, and Data (CKD)** command writes the full record area for records R1 through Rn. This is a track-formatting command. The key and data fields can be written with this command during formatting, or the count field can be written alone if the SLI flag is set. In the latter case, the key and data fields are written with zeros supplied by the control unit. The key field can be omitted if the key length is set to zero in the written count field. When a data length is set to zero in the count field, that record defines the end-of-file to the system. This command must be chained from either a successful search-ID-equal, a write-R0, or another write-CKD command. In the case of the successful search, the read-data or the read-KD command can occur between the search and the write commands. The write-CKD command must be allowed by the file mask in order to execute.

Write Special CKD command is identical to the normal write-CKD command except that the overflow-record flag (bit 1) is set in the flag byte. The special write command is used to format-write all segments of an overflow record except the last. In subsequent read and search operations, the overflow-record flag bit indicates that another segment of the record follows on the next track. The command sequence required to execute is the same as for the write-CKD command. The file mask must allow the write-CKD command.

Erase command is used to remove record information after a track-overflow condition. The operation is the same as with the write-CKD command except that no address marks are written. The chaining and file-mask requirements are the same as for the write-CKD command. The file control remains busy during the erase command.

***Write-Data** command writes the data field of the record with new information from main storage. The remainder of the track is not changed. The data length specified in the written count field defines the number of bytes that can be written. If DL=0, the EOF indicator is set and no data is written. This command requires that a successful search equal command of either the identifier or the key field precede the write operation. The file mask must allow writing the field.

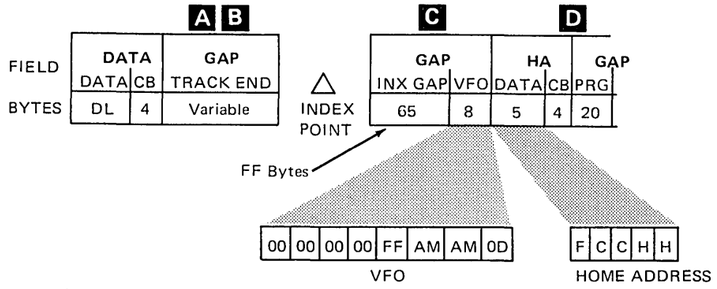
Write Key and Data (KD) command writes both the key and the data fields of the record with new information from main storage. The count field of the record is not changed. The key and data field lengths written in the count field indicate the number of bytes that can be written in each field. If the key length is zero, only the data field is written. If DL=0, the EOF indicator is set and no data is written. This command requires a successful search-ID-equal precede the write operation. The file mask must allow writing the key and data fields.

Write HA, CC - Write R0

The write-HA command is a format command used to write the identifying address at the start of the track. The command writes the 65-byte (753 bytes if long gap is written) index gap (FF bytes), the pre-record and VFO area, the five-byte home address, the check bytes, and the 20-byte post-record gap (FF bytes). The write-HA command must be preceded by the set file mask to allow writing the area. The command must be followed by the write-R0 command, or the remainder of the track is erased (written with zeros). It is assumed for this diagram that the write-HA command is entered through command chaining after the mask is set to allow. The command ends with a chain to the write-R0 command. The Write-CKD Command diagram shows the erase ending used if the command sequence is ended.

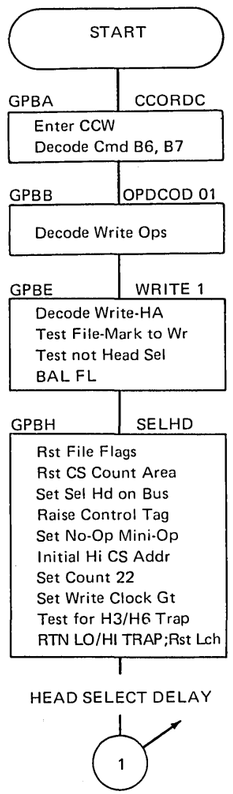
The write—HA command is decoded as write command and tested against the file mask. If allowed, the head must first be selected to route the write signals. The routine then sets the write-gap sequence with the index bit set so that the control unit is oriented and writing starts following index detection.

The home-address data is transferred from the data address specified to the home-address area in control storage. The address information is needed for subsequent control. The actual writing of the address is from control storage.



A Enter Write-HA Command After a Set-File-Mask Command.

- The write-HA operation must be allowed by the file mask or an invalid sequence is reported.
- The control unit is not oriented when the operation starts because the head is not selected.
- The head-select bit is set on the bus, and the control tag line is raised.

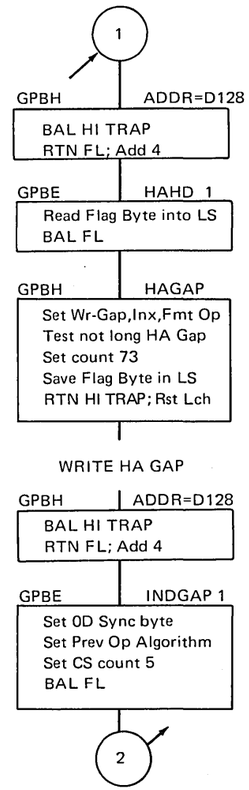


B Set Write-Gap Controls to Start After Index

- A count of 73 is set for writing the index gap and the VFO area.
- The write-gap mini-op is used with the index and format bits set.
- The start of the operation waits until the index is detected, and then writes the gap with FF bytes followed by the VFO.
- The sync-byte is not entered until the write-data controls are set.

C Set Write-Data Controls for the Home Address Field.

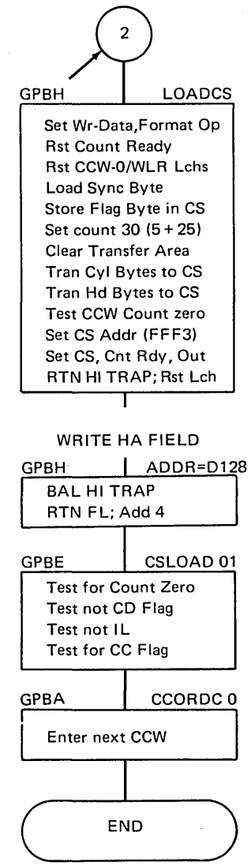
- The home-address field information is transferred from main storage to control storage before writing.
- The 0D sync byte is entered before the completion of the gap operation.



- A count of 30 (5 + 25) is entered with the write-data-format mini-op.
- The read data from control-storage gate is used during the writing of the home-address field.
- The twenty bytes of the post-record-gap area are written with FF-bytes at the end.

D Test for Command Chaining

- Test command flags for the appropriate bits to allow continuing the operation.
- If the CC flag is present, the next command is entered.
- When the CC flag is not present, the operation is ended with the erase-to-index sequence as shown in the write-CKD operation.

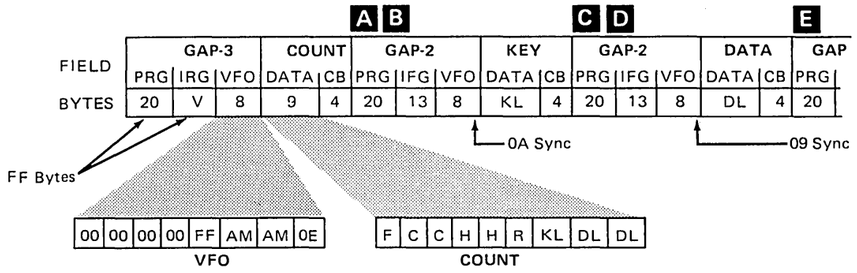


Write Data, Search ID - CC

The write-data command provides the means to update the information stored in the data field of a record. The command writes the interfield gap, the VFO area, the data information, the check bytes, and the 20-byte post-record gap. The write-data command must be preceded by a successful search-ID-equal or search-key-equal to identify the record. Writing is also allowed in pre-identified sequences (a write-data command following a write-data command). The write-data command must be allowed by the file mask, or the operation ends in an invalid sequence. There are no requirements for additional commands to be chained to the write-data command. It is assumed for this diagram that a successful search-ID-equal was performed and that the write command is command-chained to the next command.

The write-data command is decoded as a write command and tested to determine whether it is allowed. The operation starts to write gap immediately following the post-record gap of the previous key field, or the count field if the key field is not used.

The data-field information is transferred from the specified main-storage address.

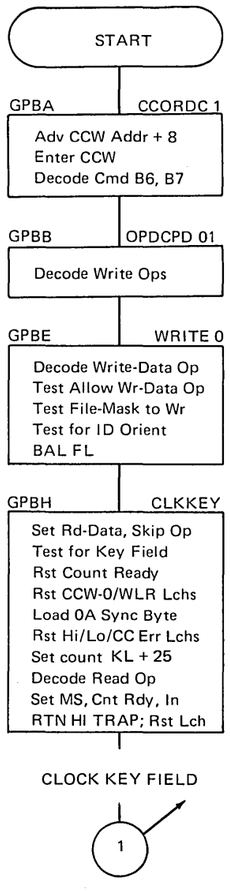


A Enter Write-Data Command After a Successful Search-ID

- When orientation is from the search-ID, the key field, if present, must be clocked-through.
- The write-data operation must be allowed by both the file mask and the previous-op algorithm.

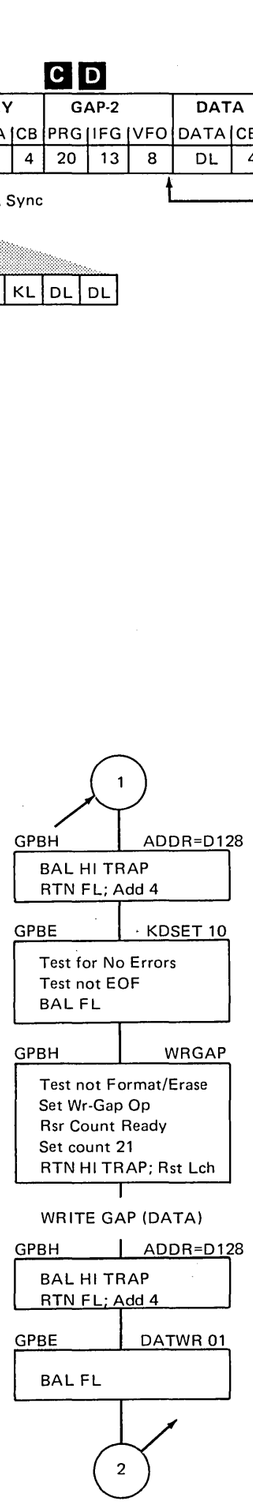
B Set Clock-Through Key Field Controls

- The key field is clocked with the read-data-skip mini-op.
- A count of KL plus 25 is entered, and a sync byte of 0A is set.
- The store data in main-storage gating is set, but no data is transferred.



C Set Controls for Writing the Interfield Gap (Data)

- A count of 21 is set to write the fixed interfield gap.
- The first thirteen bytes are written with FF-bytes followed by the eight-byte VFO area.
- The write-gap mini-op is set to write the gap.
- The sync byte is not entered until the write-data controls are set.

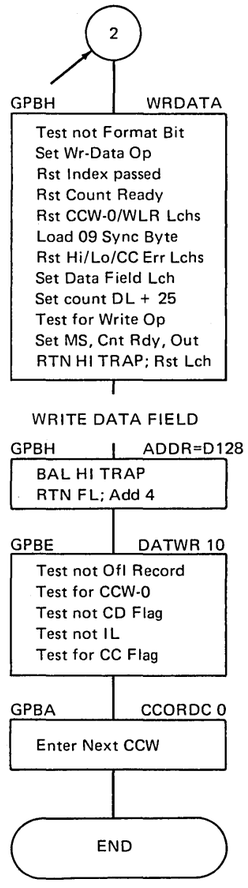


D Set Controls for Writing the Data Field

- The 09 sync byte is entered before the completion of the write-gap operation.
- A count of DL plus 25 is entered along with the with the write-data mini-op.
- The read data from main-storage gating is used during the writing of the data field.
- The twenty bytes of post-record gap area are written with FF-bytes at the end.

E Test for Errors and Chain

- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for ending.
- The CC command flag is tested to determine whether another command should be entered or the operation should be ended.

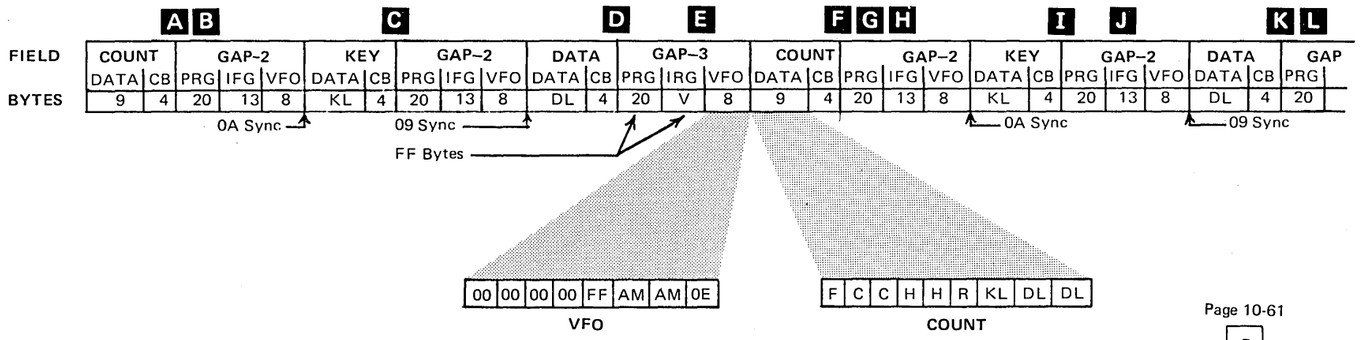


Write CKD, Search ID - CD - Format End

The write-CKD command is a format command used to initially write a defined record on the track. The command writes the gaps, the VFO areas, the record information, and the check bytes for the count, key, and data fields. The write-CKD command must be preceded by the write-R0, another write-CKD, or a successful search-ID-Equal. The write-CKD command must be allowed by the file mask. The command must be followed by another write-CKD, or or the remainder of the track is erased (written with zeros). It is assumed for this diagram that a successful search-ID-equal preceded and that the write-CKD command was the last in the sequence. The operation contains a chain-data sequence and the format ending.

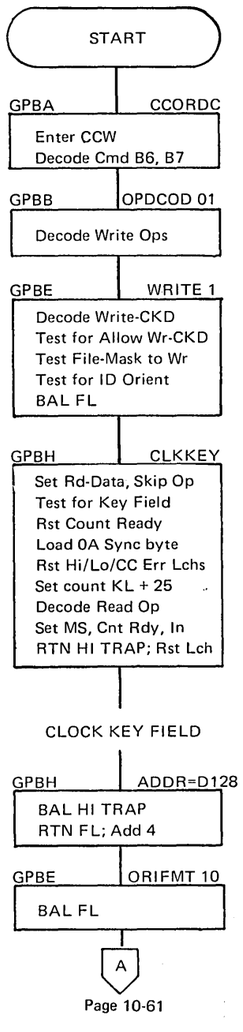
The write-CKD command is decoded a s a write command, but it must be allowed by the file mask and the previous-op algorithm. The operation starts to write gap immediately following the post-record gap.

The count-field data is transferred from main storage to control storage and is written from control storage. The key- and data-field information is written from the specified main-storage addresses.



A Enter Write-CKD Command After a Successful Search-ID-Equal

- When orientation is from the search-ID, the key and data fields for that record must be clocked-through.
- The write-CKD operation must be allowed by both the previous-op algorithm and the file mask.

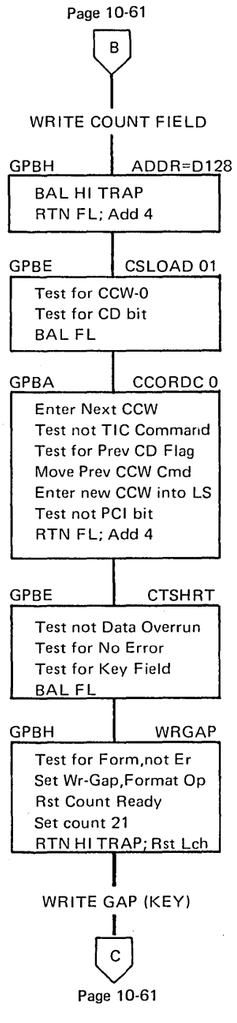


B Set Clock-Through Key-Field Controls

- The key field is clocked with the read-data-skip mini-op.
- A count of KL plus 25 is entered, and a sync byte of 0A is set.
- The store data in main-storage gate is set, but no data is transferred.

F Detect Chain Data

- When CCW = 0, test for chain-data flag for additional CCW.
- Enter next CCW, retaining the previous command, and continue operation.

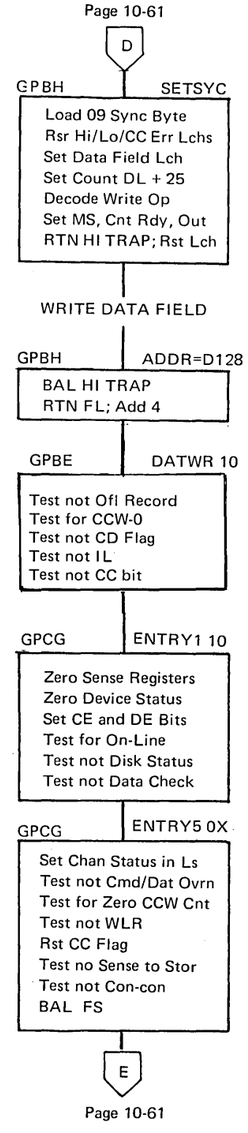


G Set Controls for Writing the Interfield Gap (Key)

- A count of 21 is set to write the fixed interfield gap.
- The first thirteen bytes are written with FF-bytes followed by the eight-byte VFO area.
- The write-gap-format mini-op is set to write the gap.
- The sync byte is not entered until the write-data controls are set.

K Set the Ending Controls If the Command Is Not Chained

- The operation branches to the ending routine if the command is not chained.
- The error conditions are tested to set the status and sense registers.
- The interrupt latch is set; the previous-op algorithm and the file mask are reset.



C Set Clock-Through Data-Field Controls

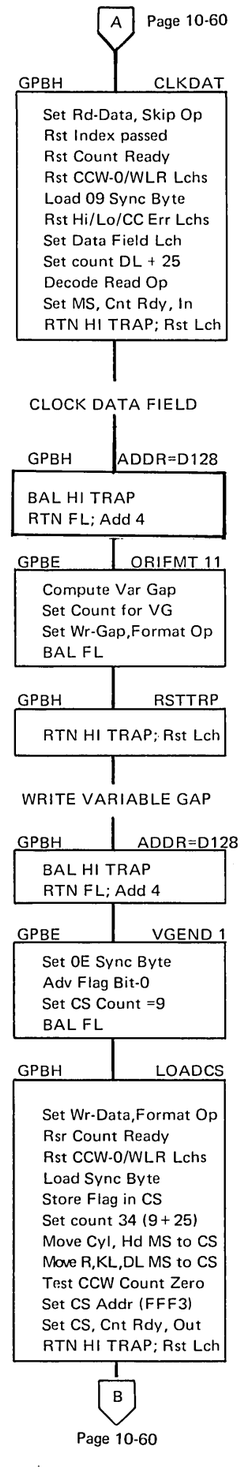
- The data field is clocked with the read-data-skip mini-op.
- A count of DL plus 25 is entered and a sync byte of 09 is set.
- The store data in main-storage gate is set, but no data is transferred.

D Set Controls for Writing the Inter-record Gap (Gap-3)

- An amount equal to 4.3% of the KL and DL counts in the previous record is added to the fixed 21-byte count for entry.
  Var Gap Computation:
  Add KL to DL
  Expand to 11 times
  Add 24 to Byte-2
  Drop Byte-3 (Div by 256)
- The write-gap-format mini-op is set to write the gap.
- The gap is written with FF-bytes except for the last eight bytes containing the VFO.
- The sync byte is not entered until the write-data controls are set.

E Set Controls for Writing the Count Field

- The count-field information is transferred from main storage to control storage before writing.
- The 0E sync byte is entered before the completion of the gap operation.
- A count of 34 (9 + 25) is entered with the write-data-format mini-op.
- The read data from control-storage gate is used during the writing of the count field.
- The twenty bytes of the post-record gap area are written with FF-bytes at the end.



H Set Controls for Writing the Key Field

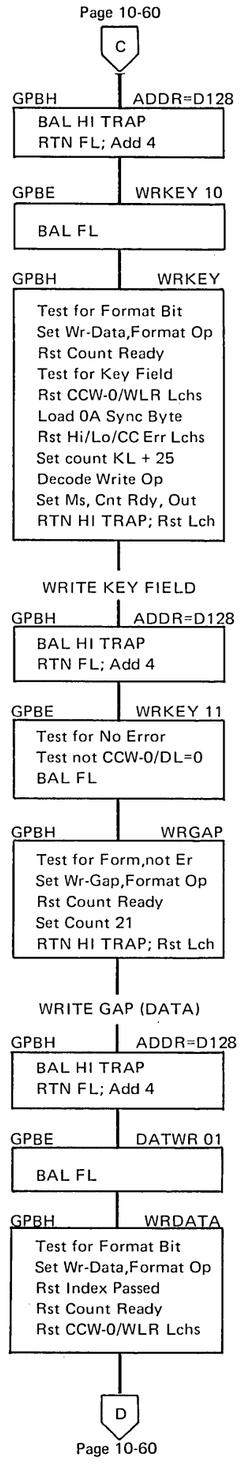
- The 0A sync byte is entered before the completion of the write-gap operation.
- A count of KL plus 25 is entered along with the write-data-format mini-op.
- The read data from main-storage gate is used during the writing of the key field.
- The twenty bytes of the post-record gap area are written with FF-bytes at the end.

I Set Controls for Writing the Inter-field Gap (Data)

- A count of 21 is set to write the fixed interfield gap.
- The first thirteen bytes are written with FF-bytes followed by the eight-byte VFO area.
- The write-gap-format mini-op is set to write the gap.
- The sync byte is not entered until the write-data controls are set.

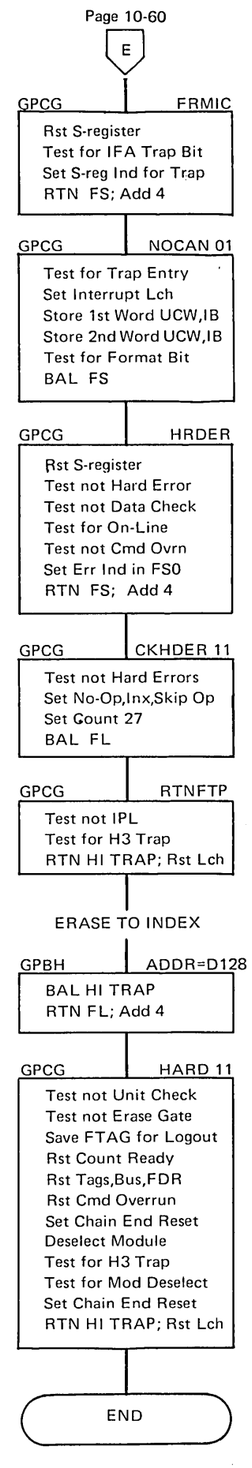
J Set Controls for Writing the Data Field

- The 09 sync byte is entered before the completion of the write-gap operation.
- A count of DL plus 25 is entered along with the write-data-format mini-op.
- The read data from main-storage gate is used during the writing of the data field.
- The twenty bytes of the post-record gap area are written with FF-bytes at the end.



L Set the Erase-to-Index Controls to End the Format Sequence

- The erase-to-index controls are set if the previous mini-op contained the format bit.
- A count of 27 is entered along with the no-op-index-skip mini-op to cause a trap when the index point is reached.
- A final test is made for error conditions after the erase-to-index operation.
- The file is deselected, and the chain-end reset is raised to end the operation.



REMEMBER

There is a Reader's Comment Form at the back of this publication.

READ COMMANDS

- Read commands transfer information read from the selected file into main storage.
- Read commands can be performed on either a single-track basis or on a multi-track basis.
- The read-IPL command forces a recalibrate operation before starting to read data from R1 of the track.

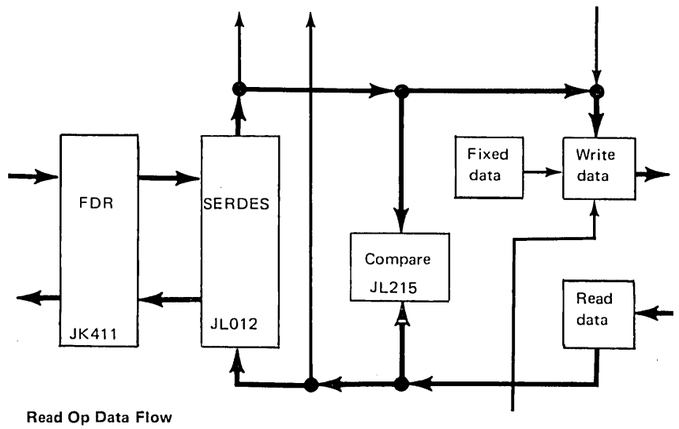
Command Codes	Single Track Hex	Single Track Binary	Multi-Track Hex	Multi-Track Binary
Read,				
Home Address	1A	0001 1010	9A	1001 1010
Count	12	0001 0010	92	1001 0010
*Data	06	0000 0110	86	1000 0110
Key and Data	0E	0000 1110	8E	1000 1110
Count/Key/Data	1E	0001 0110	9E	1001 1110
Record Zero	16	0001 0110	96	1001 0110
IPL	02	0000 0010	--	--- ---

All of the read commands, with the exception of the read-IPL, are similar. They cause information read from a specified area of the record to enter either control storage or main storage, replacing the previous data. Read commands do not require a previous successful search to execute, but a search is desirable to ensure reading the correct record. Read operations start with the eight-byte VFO/AM area to sync the clock and read through the cyclic-check bytes. When two consecutive fields are read, each field is treated separately with the controls reset after each field.

Read commands can be performed on either single-track or multi-track basis. In the latter case, when successive records are read through command chaining, the control unit advances to the next sequential head each time the index point is passed. This allows reading all of the records from the start to the end of a cylinder without interruption. Bit-0 of the command byte controls the multi-track function. A seek operation must be performed before the multi-track operation to ensure that the correct starting head number is in local storage. The read commands are not inhibited by the file mask.

During all read operations the information read is sensed by the read amplifiers and fed over the selected file co-ax to the module switch. The read information is gated to the raw-data line and fed to the VFO circuits, where the data bits are removed from the double-frequency information. The VFO also develops the read clocking to drive the control unit. The data bits are deserialized by the SERDES into bytes. When a byte is completed, it is transferred from the SERDES to the FDR and a share-cycle request is made. During the share cycle, the byte is transferred from the FDR over the EBO to the CPU storage. The home-address and count fields enter the control storage, and the key and data fields go into main storage.

*These commands are more completely detailed on the following pages.



Read Op Data Flow

**Read-Home-Address (HA)* command reads the home-address field from the track after waiting for the index point. The information goes to control storage to be available to the control-unit and then is transferred to main storage as a block. No search operation is required to execute.

Read Count command reads the count field of the next record into control storage to make it available to the control-unit. The information is then transferred to main storage as a block. No previous search is required, but the operation should be defined by the previous record to ensure reading the correct record. The R0 count field cannot be read with this command.

**Read-Data* command reads the data field following a successful search or from the next record, depending on the sequence. A previous search is not required but should be used to ensure reading the correct data field. The data field for R0 can be obtained with this command if the search-ID-equal command is used to locate the record. If DL = 0, the EOF indicator is set.

Read Key and Data (KD) command reads the key and data fields of the same record or the next record into main storage. To read the same record, the file must be oriented following the count field (search—ID). No previous search is required but should be used when a specific record is required. The key and data fields for R0 can be obtained with this command if the search-ID-equal command is used to locate the record. If the key length is zero, only the data field is read by the command. If DL = 0, the EOF indicator is set.

Read Count, Key, and Data (CKD) command reads the next full record from the track and stores the information in main storage. A previous search is not required, but using the search-ID-equal command for the previous record ensures the correct record. This command cannot be used to obtain the R0 record. If the key length is zero, only the count and data fields are transferred. If DL = 0, the EOF indicator is set.

Read Record Zero (R0) command reads the full R0 record from the track after locating the index point. The information is transferred to main storage. No previous search is required. If DL = 0, the EOF indicator is set.

**Read IPL* command is used to read the first record from the selected file (cylinder 000, head 00) into main storage to start the program load sequence. This command can be used in a start-I/O sequence when the program requires. The command is forced from the console load key when the file address is specified. The file controls initiate a recalibrate command to move the access arm to cylinder 000 and head 00. When the access arm is ready, the file control forces the read-data command after waiting for the index point. If DL = 0, the EOF indicator is set.

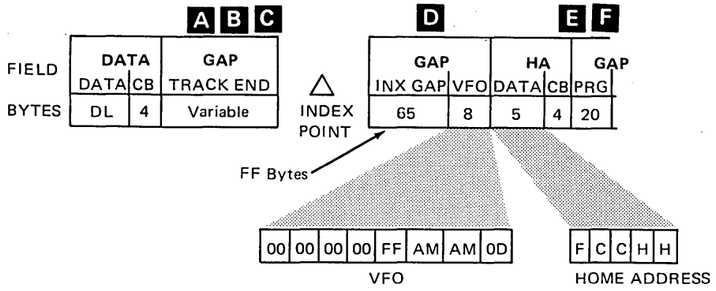
Twenty-four bytes of data are transferred to the first twenty-four positions of main storage. In this information are the PSW and two CCWs to allow continuing the reading of data from the file. Subsequent commands of read data enter the remaining part of the program. The entry continues until the chaining flag is removed to allow ending the sequence.

Read HA, SIO–CC

The read-HA command provides a means of entering the present track address into main storage. The command requires no special command sequence. The control-unit is reoriented at the index point before reading. A common usage for the command is to orient the control unit at index before a search sequence. It is assumed for this diagram that the read-HA command is the first command of a command-chaining sequence.

The read-HA command is decoded as read command. If the head is not selected, the selection becomes the first step of the operation. After waiting for the index point, the read controls are set for the operation. Following the index point, the VFO is synchronized and the address-mark and sync byte are detected before the field is read.

The home-address field information is read into control storage during the read operation. At the end of the field, the information is transferred from control storage to the specified data address in main storage.

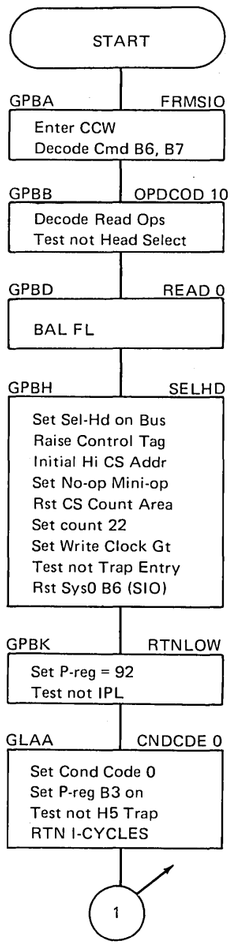


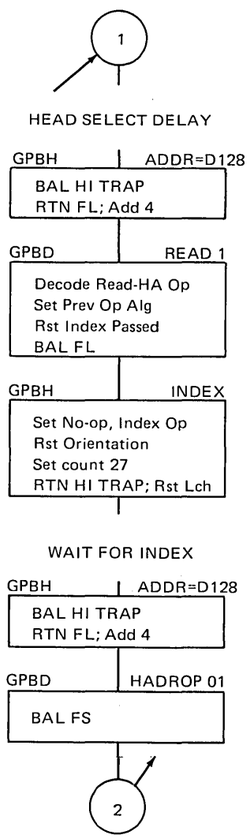
A Enter the Read-HA Command as the First Command of a Chaining Sequence

- Entering as the first command, the control unit is not oriented and the head is not selected.
- The read-HA command does not require consideration of either the file mask or the previous-op algorithm.

B Set Head-Select Controls

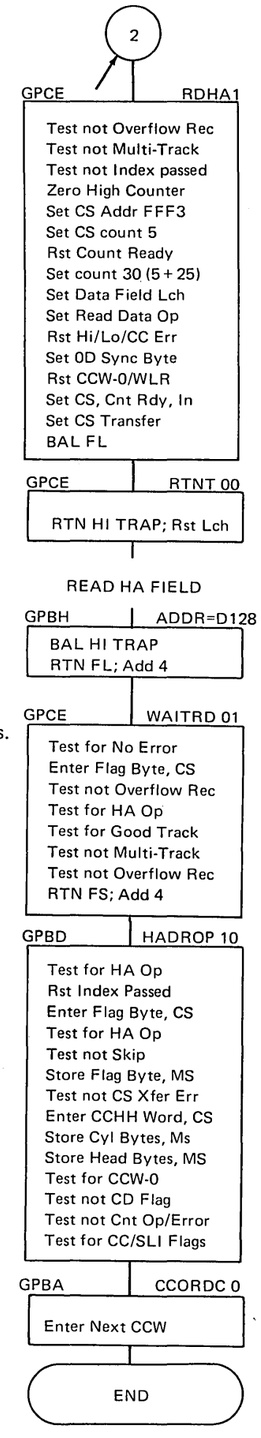
- Set select-head bit on the bus and raise the control tag line.
- Set a count of 22 for a selection time-out.
- The write-clock gate is raised to control the countdown.





C Set Wait for Index

- A no-op mini-op is set with the index-hold modifier to wait for the index point.
- A count of 27 is set to time a delay into the index gap.
- The index-passed latch and the orientation latch are reset.



D Set Controls for Reading the Home-Address Field

- Set up the control-storage address and count for entry of the home address.
- Set the read-data mini-op without modifiers and a count of 30 for the operation.
- The 0D sync byte is entered to identify the field.
- The read data into control-storage gating is set for the operation.
- Counter is forced to 8 for AM search.

E Test for No Errors and Enter Home Address into Main Storage

- Test for good track and no transfer errors.
- Transfer the home-address information into main storage if the command skip flag is not set.

F Test for Errors and Chain

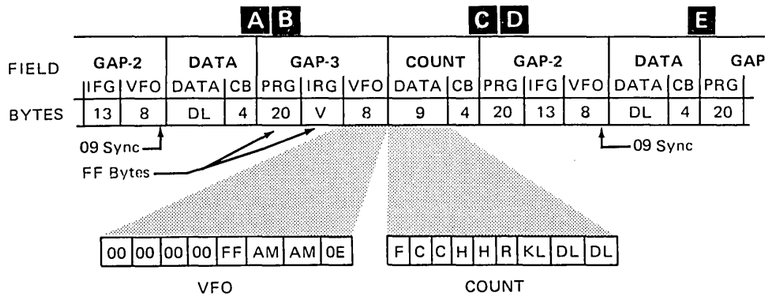
- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for ending.
- The command flags are tested for appropriate bits to allow continuing the operation.
- If the CC flag is set, the next command is entered.

Read Data, Read Data – CC

The read-data command provides the means of entering the information recorded on the data field of the record. No specific command sequence is required for this command, but it is desirable to have identification of the record position before reading. It is assumed for this diagram that a sequence of read-data commands is being performed. The orientation is from the previous data field.

The read-data command is decoded as a read command. With the track oriented at the end of the previous data field, the count field must be read for the field-length information and then the key field, if present, must be clocked-through. (This diagram assumes no key field.) Following the count field, the controls are set to read the data field of the next record.

The data-field information is entered into the specified data address except when the command skip flag is set.

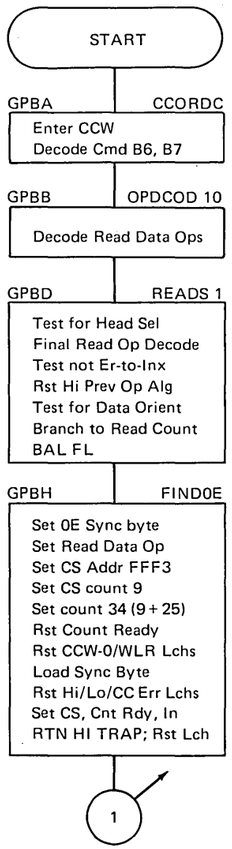


A Enter the Read-Data Command Following a Previous Read-Data Command

- When the command follows another read-data command, the control unit is oriented at the end of the data field.
- The read-data command does not require consideration of either the file mask or the previous-op algorithm.

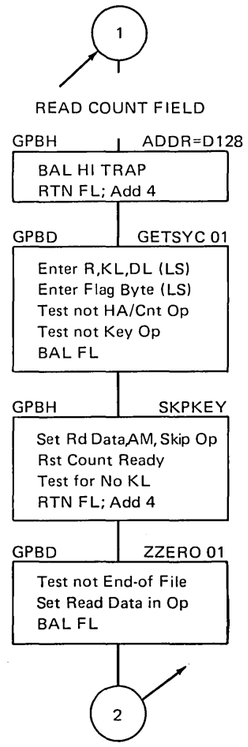
B Set Read Count Field into Control Storage

- Set up control-storage address and count for entry of the count field.
- Set read-data mini-op without modifiers and a count of 34 for the operation.
- The 0E sync byte is entered to identify the field.
- The read data into control-storage gating is set for the operation.
- Counter forced to 8 for AM search.



C Skip Key Field Controls

- The flag and length information read from the count field is entered into local storage.
- Testing the key length for zero determines that the record has no key field.
- A partial setup is made for a skip-key sequence before the length is tested.

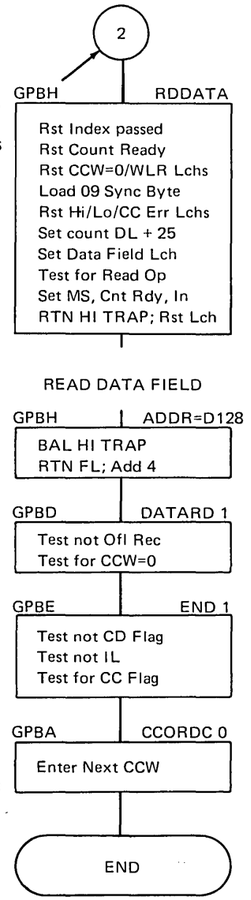


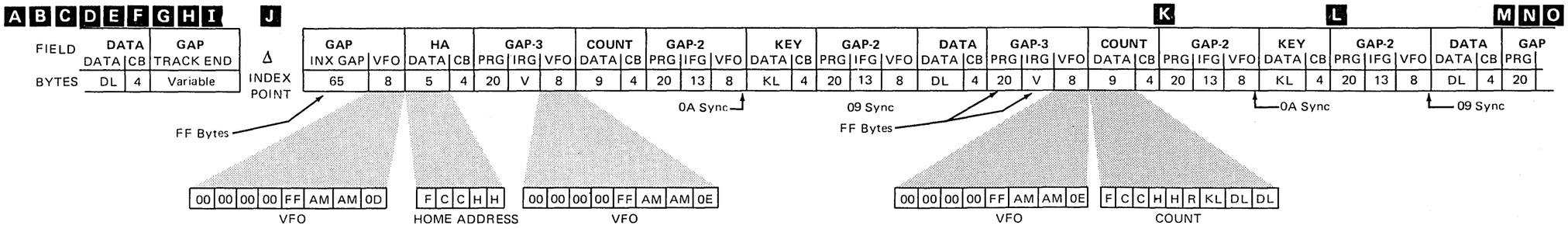
D Set Controls for Reading the Data Field

- Set read-data mini-op without modifiers and a count of DL plus 25 for the operation.
- The 09 sync byte is entered to identify the field.
- The read data into main-storage gating is set for the operation.

E Test for Errors and Chain

- The CPU machine check-circuits detect transfer errors and branch the routine to the IFA controls for ending.
- The command flags are tested for appropriate bits to allow continuing the operation.
- If the CC flag is set, the next command is entered.





Load IPL Operation (Read IPL)

The read-IPL command provides the means of entering program material from a disk file. Normal usage is from the operation of the console load key with the address switches set for a disk file. The read-IPL command can also be initiated from the start-I/O instruction.

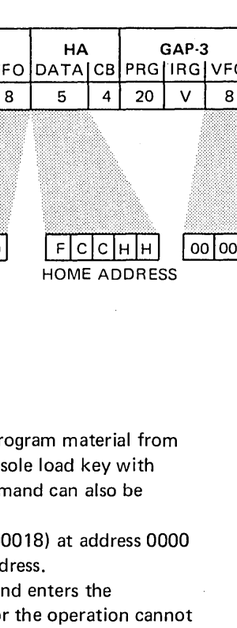
. The initial load controls enter a CCW (02000000 60000018) at address 0000 of main storage. The CAW is entered with a zero CCW address.

The command first decodes as a recalibrate command and enters the control-op routine. (The file-mask must allow full seeks or the operation cannot be performed.) After the cylinder 000 head 0 has been set up, the command is transferred to the read-op routine as a read-data command.

Record-1 data field of 24 bytes is read into the first 24 positions of main storage. The first doubleword contains the starting PSW for the entering program. The next two doublewords contain two CCWs that are command-chained. The CC flag was forced for the read-IPL. The operation continues with these and other chained CCWs to enter the full program.

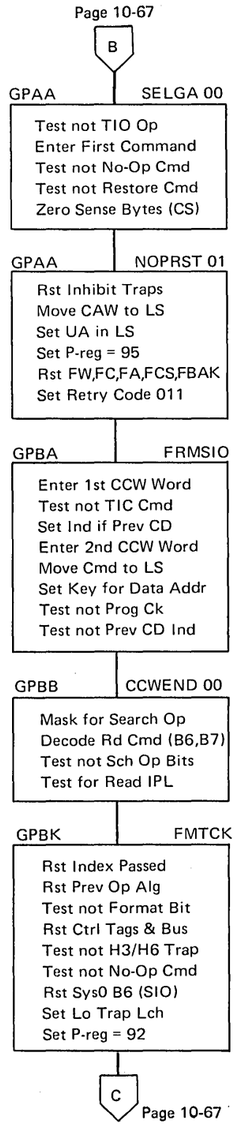
A Set Up Load IPL

- The address of module loaded with the disk pack for the IPL is entered in the console address switches.
- Operating the console load key forces a power-on reset before starting the load operation.
- The Q-register is zeroed for the CAW entry.
- A first CCW of 02000000-60000018 is entered at MS address 0000.
- The address switches are entered and tested for validity.
- An address decode of channel 1 branches the operation to the IFA routines.



C Enter First CCW

- Enter first CCW word and test that it is not the TIC command.
- Enter second CCW word and store both words in local storage.
- Decode read-IPL command and branch the operation to the control-op routine for the recalibrate sequence.



F Reset Recalibrate Controls

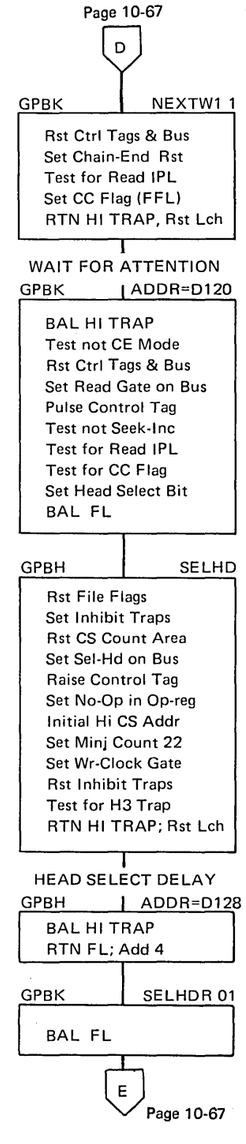
- Drop control tag line and reset the control unit with chain-end-reset.
- Wait for module to respond with its attention signal.

G Reset Gated Attention

- Set read-gate on bus-out and pulse the control tag line to reset module attention latch.
- Test the seek-incomplete indicator to determine that the recal was completed.
- Force the CC flag into the register to continue the operation.

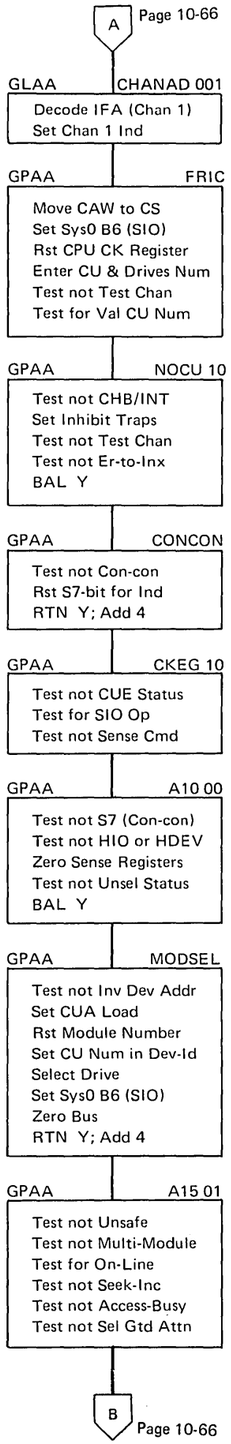
H Set Head-Select Controls

- Set head-select bit on the bus and raise the control tag line.
- Set count of 22 for selection time-out using the write clock.



B Perform Initial Selection

- Tests are made for valid control-unit address, the interrupt latch, and the channel-busy latch to determine whether the operation can be performed.
- A test is made to determine that the erase-to-index from a previous operation is not in progress.
- A test is made of the first command to determine that it is not the sense command and the sense registers are reset.
- A test is made for any unselected status conditions indicating equipment failure.
- The addressed module is selected to determine whether there is any selected status to prevent operation.
- Set the P-register for IFA operation and zero the operating registers.

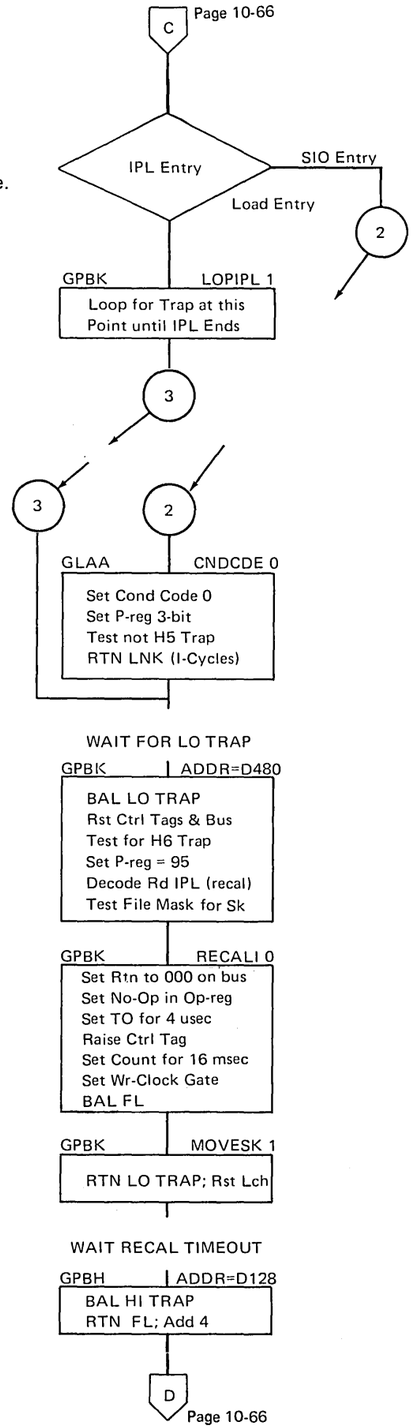


D End Initial Selection

- Request low trap and loop at this point if the entry is from the load-IPL sequence.
- If the read-IPL is initiated by a start-I/O instruction, the operation is returned to I-cycles.

E Set Controls for Recallbrate Operation

- Test file mask to determine that the recalibrate is allowed.
- Set return-to-000 on bus-out and time-out for 4 usec.
- Raise control tag line and set recalibrate time-out for 16 msecs.



I Set Wait for Index Controls

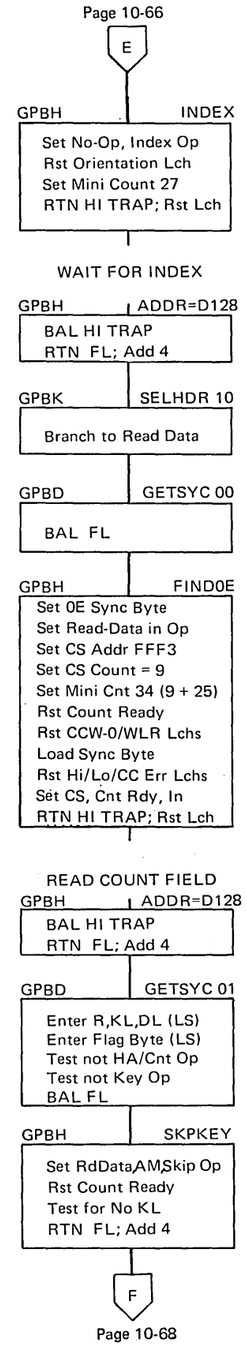
- The no-op mini-op is set with the index-hold modifier to wait for the index point.
- A count of 27 is set for a delay after index detection.

J Set Controls for Reading the Count Field of Record 1

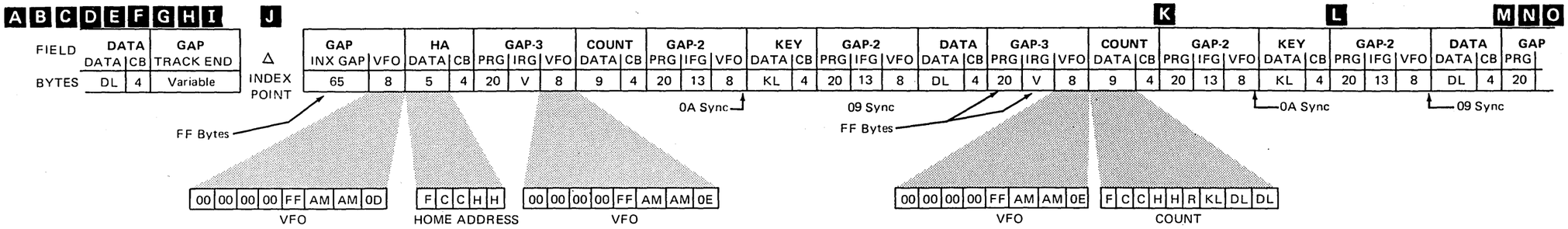
- The operation is transferred to the read-op routine as a read-data command.
- The count field of record 1 must be located and read for the length information.
- The control-storage address and count are set to enter the count-field information.
- The 0E sync byte is set to identify the count field.
- The read-data mini-op is set without modifiers along with a count of 34.
- The read data into control-storage gating is used to enter the information.
- Counter is forced to 8 for AM search.

K Set Controls to Skip-Through the Key Field

- Enter count-field information into local storage for processing.
- The key field may not be present, but it is not stored because this is not a key operation.
- The read-data mini-op is set with the AM and skip modifiers to skip-through the key field.
- If no key field is present, the mini-op is not used.



Load (IPL Operation (Read IPL) Contd.

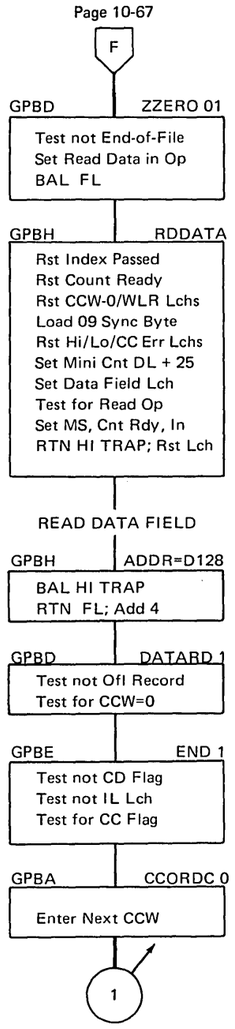


L Set Controls for Reading the Data Field

- Set the read-data mini-op without modifiers for the operation.
- The 09 sync byte is set to identify the data field.
- The DL (24) plus 25 is set into the counter for the transfer.
- The read data into main-storage gating is used to enter the 24 bytes starting at address 0000.
- Counter is forced to 8 for AM search.

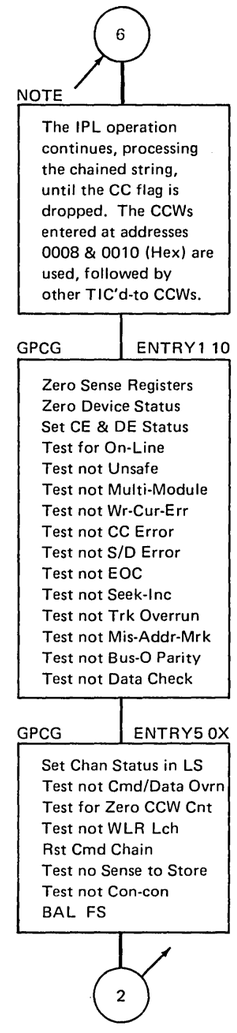
M Test For Errors and Chain

- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for ending.
- The command flags are tested for the appropriate bits to allow continuing the operation.
- Because the CC flag was forced during the read-IPL routine, the next CCW is entered from address 0008.
- The load-IPL operation continues by processing chained CCWs until the chain is broken before ending.



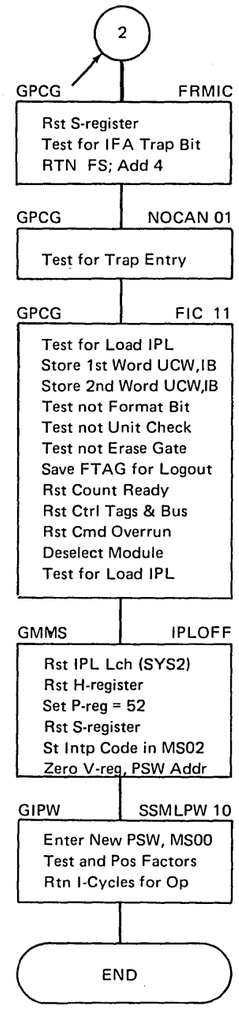
N Test Error Conditions for Ending

- The previous information in the status and sense registers is cleared.
- The channel-end and device-end status bits are set.
- The status conditions of the disk file are tested, and appropriate sense bits are set.
- The control-unit error conditions are tested, and appropriate sense bits are set.
- The sense information is stored, and the required status bits are set for the ending.



O Set Interrupt Buffer and End Operation

- The UCW information and the status are stored in the interrupt buffer in control storage.
- If no unit-check conditions were detected, the operation ends without posting a logout.
- The load-IPL operation branches to the GMMS routine to reset the load indicator.
- The P-register is adjusted for CPU operation.
- After setting the PSW entry address to 0000, the new PSW is entered.
- The PSW factors are tested and positioned as the current PSW before returning the operation to the I-cycles routine.



SEARCH COMMANDS

- Search commands compare main-storage data with data read from a track to locate a specified matching condition.
- The search commands can be performed on a single-track or a multi-track basis.
- Only a search equal command that is not truncated allows subsequent writing of information.

Command Codes	Single Track Hex	Single Track Binary	Multi-Track Hex	Multi-Track Binary
Search,				
Home Addr Equal	39	0011 1001	B9	1011 1001
*Identifier Equal	31	0011 0001	B1	1011 0001
Identifier High	51	0101 0001	D1	1101 0001
Ident Equal/High	71	0111 0001	F1	1111 0001
*Key Equal	29	0010 1001	A9	1010 1001
Key High	49	0100 1001	C9	1100 1001
Key Equal/High	69	0110 1001	E9	1110 1001

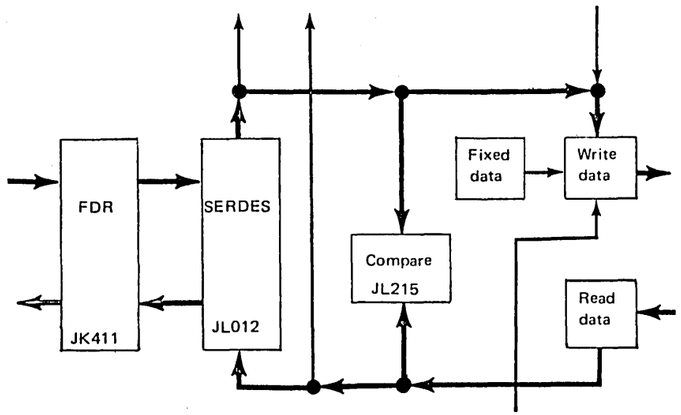
Search commands fall into two groups. The search equal group is used for positive identification of an area and is required for all writing sequences and some reading sequences. The search high and search high/equal groups are normally used for table lookup operations. This group cannot be used to locate for a writing sequence. Both groups of commands perform the same function. The data from the specified track is compared with data from a specified address in main storage. The output condition of the compare unit sets the status-modifier bit as an indicator when the result matches the command type. The search commands are not inhibited by the file mask.

The normal command-chain sequence for a search operation is the appropriate search command followed by a TIC-8 command and then the read or write command to be executed. The search command is performed; and if the match condition does not occur, the TIC-8 command is read to re-execute the search command. This continues until the status-modifier bit is set, indicating that the match has been made. The status-modifier bit causes the TIC-8 command to be skipped, so that the next command to be executed is the read or write.

Search commands can be performed on a single-track basis or on a multi-track basis. For a single-track search, all records on the track can be searched for a match. If no match is found, the operation is ended after passing the index point twice.

In a multi-track search, when the index point is reached, the control unit advances the head selection to the next sequential head and continues the search. If no match is found, the search continues through the last track of the cylinder before ending the operation. Bit 0 of the command byte defines the multi-track function. A seek operation must be performed before the multi-track operation to ensure that the correct head number is in local storage. If the full first track is to be searched, either a read-HA or read-R0 command should precede the search.

*These commands are more completely detailed on the following pages.

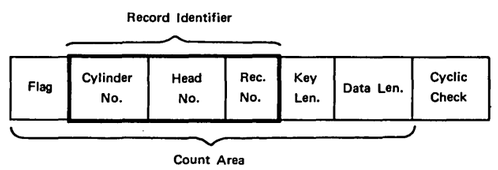


A normal search is for the full area of the field specified. A short or truncated search may be performed by using a smaller CCW count and the SLI flag set on. If that portion of the record meets the requirements, the status-modifier bit is set. No write commands can be performed after a short search.

During search operations with key and data fields, two data paths are present to provide the two sets of information to the compare circuits. The CPU and the main data paths of the IFA are set up for a write operation. Information from CPU main storage transfers with share cycles one byte at a time to enter the FDR register. As a byte is required, it transfers to the SERDES from the FDR and a request is made for another share cycle. The SERDES serializes the byte bit-by-bit to feed one side of the compare circuit (it does not feed the cyclic-check circuits). For the second path, information is being read serially from the selected file to feed the other side of the compare circuits and the cyclic-check circuits (it does not feed the SERDES). The comparison is made bit by bit as the information is entering.

Search operations on the home-address field and the count-field identifier are compared in the CPU ALU and do not use the IFA compare circuits. The information from the selected file is entered into control storage using a normal read operation. The comparison is performed following the full field entry.

Search Home Address (HA) Equal command compares the home address as read from the track with data from the specified address area of main storage. In this case the track information enters control storage and is compared to main storage by the CPU ALU. An equal compare sets the status-modifier bit.



**Search Identifier (ID) Equal* command compares the five bytes of the record count field that represent the identifier with the specified data address area in main storage. In this case the track information enters control storage and is compared to main storage by the CPU ALU. An equal compare sets the status-modifier bit.

Search ID High command compares the five-byte track record identifier with the data from the specified area of main storage. The information is read into control storage and compared by the CPU. If the value of the track record identifier is greater than the value in storage, the status-modifier bit is set.

Search ID Equal or High command compares the five-byte track record identifier with the data from the specified area of main storage. The information is read into control storage and compared by the CPU. If the value of the track record identifier is equal to or greater than the value in storage, the status-modifier bit is set.

**Search Key Equal* command compares the key field being read from the track record with the data from the specified address of main storage. The information is compared by the IFA compare circuits. An equal compare sets the status-modifier bit.

Search Key High command compares the key field being read from the record track with the data from the specified address area in main storage. The information is compared by the IFA compare circuits. If the value of the track key field is greater than the value in storage, the status-modifier bit is set.

Search Key Equal or High command compares the key field from the track with the data from the specified address area in main storage. The information is compared by the IFA compare circuits. If the value of the track key field is either equal to or greater than the value in storage, the status-modifier bit is set.

Scan Commands

- The scan commands are selective search commands.
- An FF-byte stored in any position of the search argument in main storage suspends the compare for that byte of the record.
- The scan commands can be chained in either single-track or multi-track mode for operation.

Command Codes	Single Track			Multi-Track	
	Hex	Binary		Hex	Binary
Search (Scan),					
* Key/Data Equal	2D	0010	1101	AD	1010 1101
Key/Data High	4D	0100	1101	CD	1100 1101
Key/Data Eq/High	6D	0110	1101	ED	1110 1101

The scan commands provide a means of selectively searching file records. Instead of requiring that all bytes of a record match the stored argument, the scan commands allow selective bytes to be matched. When any byte of the search argument is set to FF, the IFA suspends the compare for that byte. The scan commands extend the search from the key field through the data field. The search is of the data field only, when no key field is written. If DL = 0, the EOF indicator is set and the data is not compared.

The scan function provides three commands that can be used for either single-track or multi-track operation. These commands are normally chained in the same manner as the normal search commands, the TIC command being used to repeat the scan-search until the desired record is found. The status-modifier bit is presented when the selected bytes of the record match the masked argument as defined by the command. A short or truncated search is also possible with the scan commands by setting the CCW count to the lower number of bytes.

For the single-track scan commands, all records on the addressed track can be read for the match. If no match is found, the operation is ended after passing the index point twice.

For multi-track scan operations, the records on the addressed track are read until the index point is encountered. If no match has been detected, the control-unit advances the head selection to the next sequential head and continues the search. The operation continues and the heads are switched with each passing of the index point until the match is found or the end-of-cylinder condition is reached. A seek operation must be performed before the multi-track sequence so that the correct head number is in local storage. Either a read-HA or a read-R0 command should precede the scan-search to ensure a search of all the records on the first track.

Search Key and Data Equal command compares the scan argument in main storage with the information read from the key and data fields of the current record. An equal compare indication by the IFA compare circuits sets the status-modifier bit.

Search Key and Data High command compares the scan argument in main storage with the information read from the key and data fields of the current record. If the value of the track record is greater than the argument in main storage, the status-modifier bit is set.

Search Key and Data Equal or High command compares the scan argument in main storage with the information read from the key and data fields of the current record. If the value of the track record is either equal to or greater than the argument in main storage, the status-modifier bit is set.

Continuing Scan Commands

- Continuing scan commands allow restarting interrupted record-overflow operations.
- These commands are originated from sense byte 5 when the operation is interrupted.
- The continuing command defines the original command or the status-modifier setting if the final condition has been determined.

Command Codes	Single Track			Multi-Track	
	Hex	Binary		Hex	Binary
Continue Scan,					
Search Equal	25	0010	0101	A5	1010 0101
Search High	45	0100	0101	C5	1100 0101
Search Equal/High	65	0110	0101	E5	1110 0101
Not Status Mod	55	0101	0101	D5	1101 0101
Set Status Mod	35	0011	0101	B5	1011 0101
or	75	0111	0101	F5	1111 0101

When the scan commands are issued for long records written in the record-overflow mode, the CCW count is for a record on two or more tracks. If the record execution is interrupted for any reason, the operation is halted to allow the program to correct the condition. The normal reason for this interruption is the detection of a defective or alternate track following head switching. The control unit stores the indication of the initiating command and the match condition reached in sense byte 5. From this information, the CSW, and the original CCW sequence, the program can restart the operation after seeking to the appropriate track.

The conditions that must be defined when the scan-search is interrupted are:

1. The original command condition of high or equal if the match has not been decided.
2. The final setting for the status-modifier bit if the match has been decided.

For example, if during an equal search the first segment was high, the continuing command says do not set the status-modifier bit. If the first segment had ended with an equal condition, the search-equal continuing command is given.

Continue, Search Equal command calls for a continuance of the scan-search-equal command from the point that the operation was interrupted. The match condition has not yet been decided. The remainder of the record is compared in the normal manner. The status-modifier bit is set if the remainder of the record compares with the argument.

Continue, Search High command calls for continuance of the scan-search-high command from the point that the operation was interrupted. The match condition has not been decided. The remainder of the record is compared in the normal manner. The status-modifier bit is set if the remainder of the record is greater than the argument.

Continue, Search High/Equal command calls for continuance of the scan-search-high/equal command from the point that the operation was interrupted. The match condition has not been decided. The remainder of the record is compared in the normal manner. The status-modifier bit is set if the remainder of the record is either equal to or greater than the argument.

Continue, Do Not Set Status Modifier command calls for continuance of any scan-search command from the point that the operation was interrupted. The match condition has been determined as a mismatch and the status-modifier bit is not to be set for the ending status. The operation is completed to determine that no error conditions develop.

Continue, Set Status Modifier command calls for continuance of any scan-search command from the point that the operation was interrupted. The match condition has been determined as a match, and the status-modifier bit is to be set for the ending status. The operation is completed to determine that no error conditions develop.

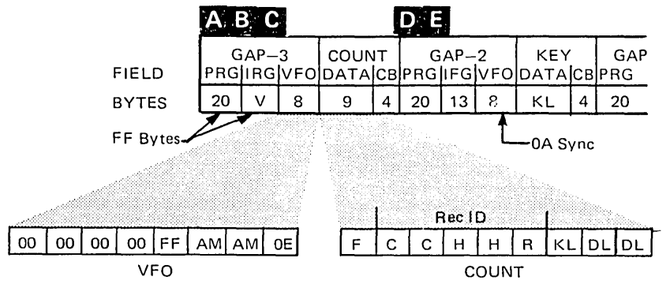
*These commands are more completely detailed on the following pages.

Search ID Equal, SIO - Equal Match

The search-ID-equal command provides the means of locating a specified record on the track or cylinder. The command does not require a specific command sequence to perform, but it is normally followed by the TIC command. When the search ends in a match, the TIC command that follows is skipped and the second following command is performed. If the match does not occur, the TIC command causes the search command to repeat on the next record. If it is desired to start with the first record of the track, the search-ID command should be chained from the read-HA command. The search-ID-high and the search-ID-high/equal commands are performed in the same manner but with different match conditions. (A write operation can follow only the search-equal command.) It is assumed for this diagram that the search-ID-equal command is the first in a command sequence and that the operation ends in a match.

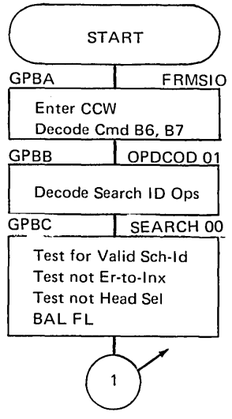
The search-ID command is decoded as a search command. Because this is the first command, the head must be selected. (The control unit is not oriented.) Orientation occurs when the first count field is identified. In the case of a second search, the key- and data fields are not clocked-through, and the count field is located in the same manner as for the first search. If the search results match the command requirements, the status-modifier bit is set to cause the next command to be skipped (TIC).

The count-field information is entered into control storage in the same manner as for a read-count field operation. Following the entry, the IFA enters the information from the specified data address in main storage for comparison. The compare operation is performed in the CPU ALU. The main-storage information is not retained for subsequent searches.



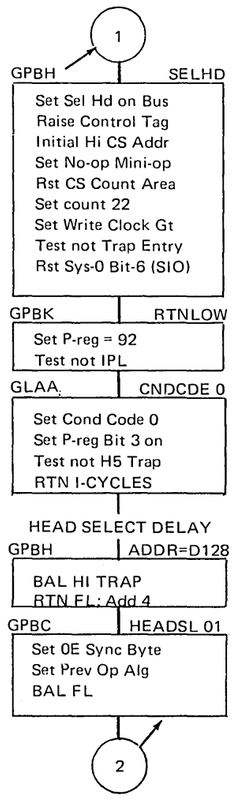
A Enter the Search-ID Equal Command as the First Command

- When the search command is the first command of the sequence, the head must be selected.
- The search-ID command does not require consideration of either the file mask or the previous-op algorithm.



B Set Select-Head Controls

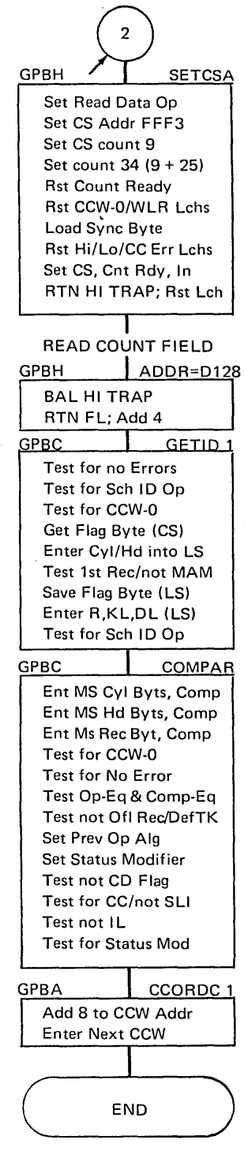
- Set select-head bit on bus and raise the control tag line.
- Set count of 22 for selection time-out.
- The write-clock gate is raised to control the countdown.



C Set Controls for Reading a Count Field

- Set up the control-storage address and count for entry of the count field.

- Set the read-data mini-op without modifiers and a count of 34 for the operation.
- The 0E sync byte is entered to identify the field.
- The read data into control storage gating is set for the operation.
- Counter is forced to 8 for AM search.



D Test for No Errors and Compare with Identifier From Main Storage

- Test for good track and no transfer errors.
- Enter identifier information from main storage one byte at a time.
- Compare with identifier read from file using the CPU ALU.
- Set status-modifier bit if ALU compare matches conditions specified by the command.

E Test for Errors and Chain

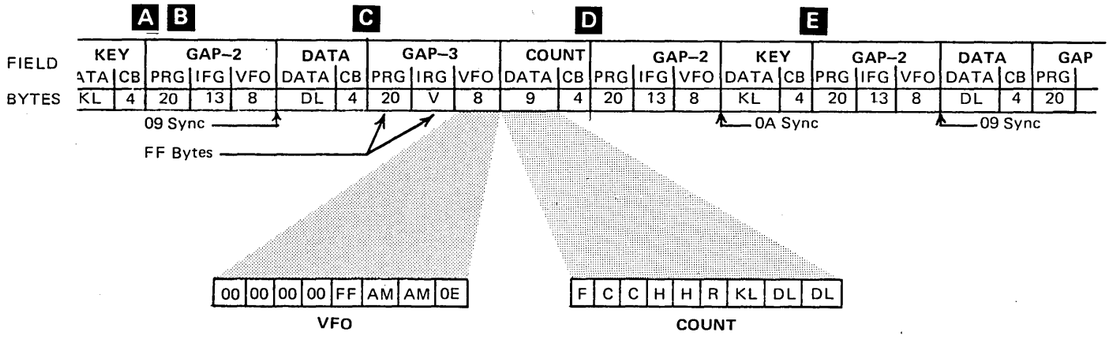
- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for ending.
- The command flags are tested for appropriate bits to allow continuing the operation.
- If the CC flag is set, the next command is entered.

Search Key, Search Key - No Match

The search-key commands provide a means of identifying a record from the associated key information. The commands may be used to locate the record in the manner of the search-ID commands, or they may be used for positive identification in random storage systems. The search-key commands are normally followed by the TIC command. The search-key command may specify equal, high, or high/equal matches for the compare. When the search ends in a match, the TIC command that follows is skipped and the second following command is entered. If the match does not occur, the TIC command causes the search command to repeat on the next record. If it is desired to start the search with the first record of the track, the search-key command should be chained from read-HA command. It is assumed for this diagram that search-key command followed an unseccessful search-key command and that the present search did not result in a match.

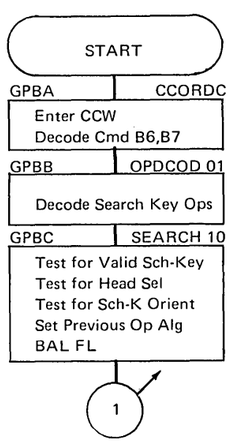
The search-key commands are decoded as search commands. Because a previous search command was performed, the head is selected. The count field for the next record must be read in to obtain the length information. This is followed by the search-key operation. If the search results in a match, the status-modifier bit is set to cause the next command to be skipped (TIC).

For the search-key operation, the key-field information being read enters one side of the compare unit as a read operation. The information from the specified data address in main storage enters the SERDES and is serialized to feed the other side of the compare unit. The key-field information read also enters the cyclic-code and bit-count registers to perform the data checks. The information from main storage has the normal write checks.



A Enter the Search Key Command After a Previous Search

- Because the search is not the first data command, the head is already selected.
- The search-key command does not require consideration of either the file mask or the previous-op algorithm.

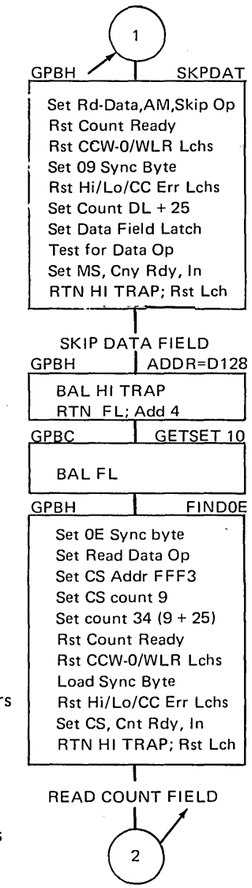


B Set Skip-Through Data Field Controls

- The data field is clocked with the read-data-AM-skip mini-op.
- The operation is the same as the clock-through operation except that a data check condition cannot occur.
- A count of DL plus 25 is entered and a sync byte of 09 is set.
- The store data in main storage gating is set, but no data is transferred.

C Set Controls for Reading a Count Field

- Set up control-storage address and count for entry of the count field.
- Set the read-data mini-op without modifiers and a count of 34 for the operation.
- The 0E sync byte is entered to identify the field.
- The read data into control-storage gating is set for the operation.

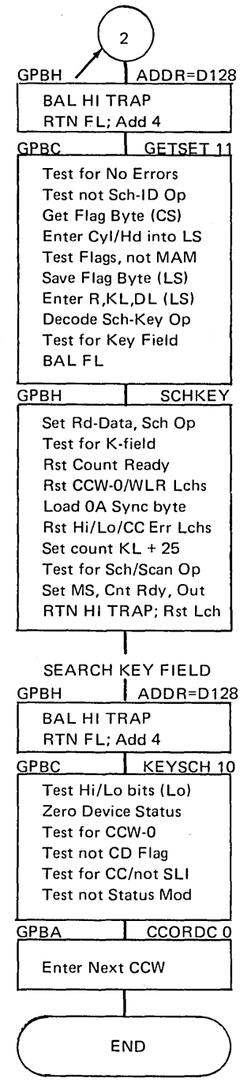


D Set Controls for Searching the Key Field

- Transfer the count-field flags and lengths to local storage for the operation.
- Set the read-data mini-op with the search modifier along with a count of KL plus 25.
- The 0D sync byte is entered to dientify the field.
- The read data from main-storage gating is set for the operation.
- The search argument is transferred from main storage to SERDES and serialized.
- The serialized argument is compared with the key-field information being read.

E Test for Compare, Errors, and Chain

- Set the status-modifier bit if the compare output matches the conditions specified by the command.
- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for the ending.
- The command flags are tested for appropriate bits to allow continuing the operation.
- If the CC flag is set, the next command is entered.

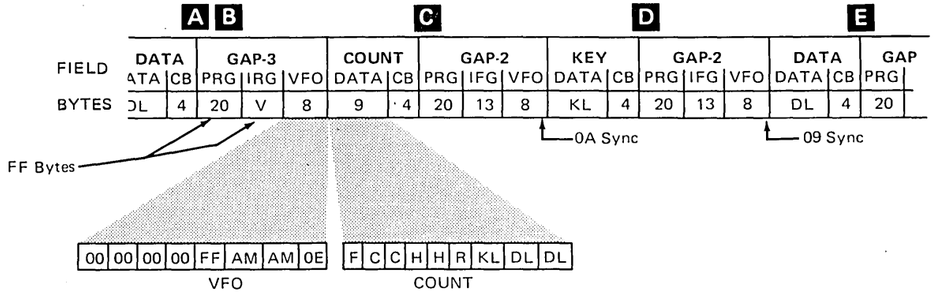


Search KD (Scan), Sch KD - Eq Match

The search-KD (scan) commands provide the means of selective searching specific portions of records for defined information. The commands differ from the conventional search key in that when the stored argument contains an FF-byte that position is not compared. The normal command sequence for locating and identifying the record is read-count, search KD, and TIC-16. The identifier for the record to be scanned is stored for use when the match occurs. The search-KD command may specify equal, high, or high/equal matches for the compare. When the search (scan) ends in a match, the TIC command that follows is skipped and the second following command is entered. If the match does not occur, the TIC command causes the read-count and the search-KD commands to be repeated for the next record. It is assumed for this diagram that the search-KD was command-chained directly to a previous search-KD command. This causes the read-count operation to occur within the search-KD command. It is also assumed that a match occurred with the current command.

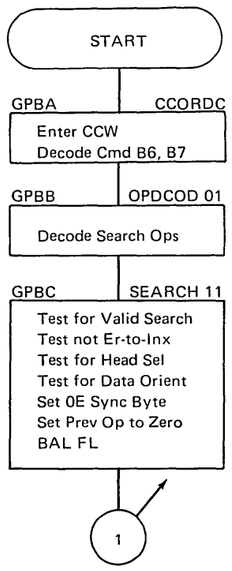
The search KD (scan) commands are decoded as search commands. Because a previous command was executed, the head is selected. The count field must be read to obtain the length information. This is followed by the search-KD operation that includes first the key field; then the data field. The mini-op scan modifier is set along with the search modifier to recognize the FF-bytes. If the search results in a match, the status-modifier bit is set to cause the next command to be skipped (TIC).

For the search-KD operation, the key-and data-field information being read enters one side of the compare unit except during the bytes indicated by the FF-bytes. The information from the specified data address in main storage enters the SERDES and is serialized to feed the other side of the compare unit. The entire key- and data-field information read also enters the cyclic-code and the bit-count registers to perform the data checks. The information from main storage has the normal write checks.



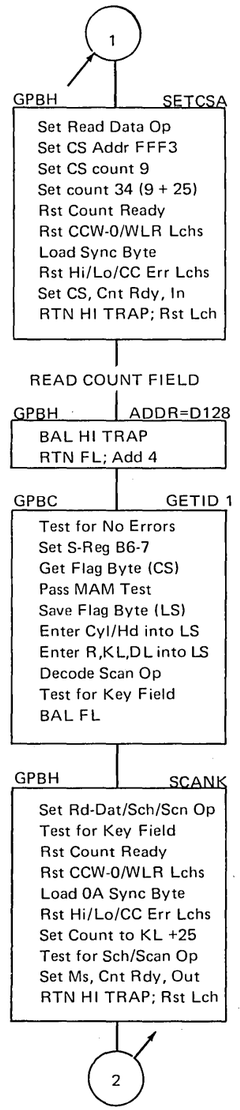
A Enter the Search-KD Command After a Previous Search

- Because the search is not the first data command, the head has been selected.
- The search-KD command does not require consideration of either the file mask or the previous-op algorithm.



B Set Controls for Reading the Count Field

- Set up control-storage address and count for entry of the count field.
- Set read-data mini-op without modifiers and a count of 34 for the operation.
- The 0E sync byte is entered to identify the field.
- The read data into control-storage gating is set for the operation.



C Set Controls for Searching the Key Field

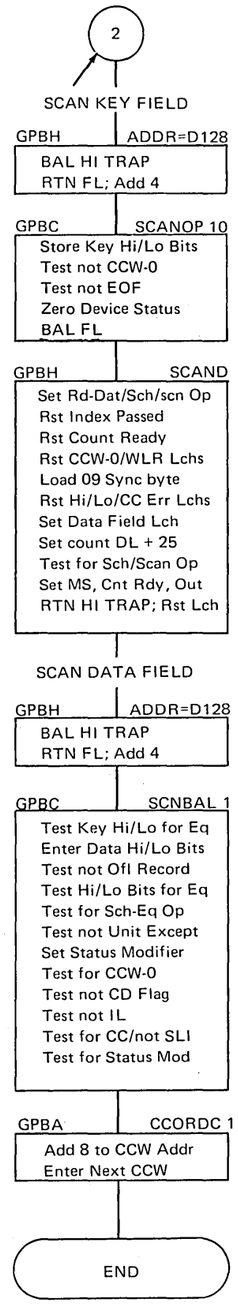
- Transfer the count-field flags and lengths to local storage for the operation.
- Set the read-data mini-op with the search and scan modifiers along with a count of KL plus 25.
- The 0C sync byte is entered to identify the field.
- The read data from main-storage gating is set for the operation.
- Counter is forced to 8 for AM search.
- The search argument is transferred from main storage to the SERDES and serialized.
- The serialized argument is compared with the key-field information being read (FF bytes are not compared).

D Set Controls for Searching the Data Field

- Set the read-data mini-op with the search and scan modifiers along with a count of DL plus 25.
- The 09 sync byte is intered to identify the field.
- The read data from main-storage gating is set for the operation.
- Counter is forced to 8 for AM search.
- The search argument is transferred from main storage to the SERDES and serialized.
- The serialized argument is compared with the data-field information being read (FF bytes are not compared).

E Test Compare, Errors, and Chain

- Set the status-modifier bit if the compare output matches the conditions specified by the command.
- The CPU machine-check circuits detect transfer errors and branch the routine to the IFA controls for the ending.
- The command flags are tested for appropriate bits to allow continuing the operation.
- If the CC flag is set, the next command is entered.



TRAP AND ENDING OPERATIONS

IFA operating sequences use a group of trap routines to allow trapping into CPU operations and to end the operations. These are discussed under a common heading because of the direct relation between the error-trap routine and the ending.

Trap Routines

- Trap routines allow the hardware to request CPU (IFA) time to process a condition encountered.
- An honored trap request interrupts the current operation to process the trap routine.
- When the routine is completed, the CPU returns to the interrupted operation.

The trap routines, when requested through hardware conditions, are allowed to interrupt the existing operation as soon as the operation priority permits. The address of the next step of the existing microroutine is stored as a link for return. The trap controls insert the address of the sequence for the requested trap. When the trap sequence is completed, the link address is entered to resume the interrupted sequence.

The IFA has six trap routines that can be requested by hardware conditions. Four of the traps are of high priority, using the H3 indicator for trap control. This priority is below the machine-check trap but above all other channel traps. Two of the traps are of lower priority, using the H6 indicator that allows interruption by other channel traps. The following chart shows the traps, their priority, and the starting address of the trap microroutine:

Gated Attention Trap	H3	D120	GPBK
*Index Trap	H3	D124	GPCE
Mini-Op (High) Trap	H3	D128	GPBH
*Error Trap	H3	D12C	GPCG
Control (Low) Trap	H6	D480	GPBK
Diagnostic Trap	H6	D48C	GPD0

Gated Attention Trap is requested when the selected file has completed a seek operation and has responded with its attention signal. The trap routine responds by forcing the attention to reset and then proceeds with the next chained command.

**Index Trap* is requested when the index point is encountered during an operation without the index-start op-bit being set. If data is transferring, this represents an overrun and must be handled by the trap routine. In cases where data is not transferring, the indicators are tested to determine whether the head should be advanced and the operation continued on the next track. When the head is not advanced, a test is made for a previous index trap, indicating that the record is not on the track. The indicators are set, and the operation is branched to the ending routine.

Mini-Op (High) Trap is set at the count-decode-20 time for each mini-op performed unless an error developed. This routine immediately enters the link stored in the FL-register to resume the command operation in progress. The trap is ended when the operation has loaded the next mini-op information and stored a new return link.

**Error Trap* is set at count-decode-20 time if an error condition has developed in the mini-op processing. When the error-trap sets, the mini-op trap is not set. The error indicators are set before allowing the operation to branch either back to the command operation or to the ending routine.

Control (Low) Trap is requested by the microroutine when a control command is processed. When the previous operation is completed, the file controls are disoriented; thus the high priority is not needed. The trap ends after the control command is initiated.

Diagnostic Trap is set when the CE panel switches (2319) are set to perform inline diagnostics. The customer's program must not be using the IFA at the time. The trap ends after the diagnostic operation specified has been initiated.

* These traps are more completely detailed on the following pages. The remainder of the traps appear within other flow charts in this section.

Ending Sequence

- The ending sequence tests for error conditions to develop the appropriate sense and status bytes.
- Ending during initial selection causes the status to store in the CSW.
- After initial selection, the status is stored in the interrupt buffer until interrupt.

The ending sequence provides the housekeeping to record the result of the operation and to reset the necessary controls. The entry to the routine can be as the result of detected error conditions or it can be the result of completion of the programmed sequence. In either case the routine tests for all error conditions and stores the appropriate sense and status bytes. During initial selection, the status is stored in the CSW. When the operation is no longer in initial selection, the status is stored in the CSW. When the operation is no longer in initial selection, the status is stored in the interrupt buffer, starting at control-storage address F904 for access by the interrupt routine. The logout information is stored as required in this routine, if the operation is still in initial selection and the logout mask permits. The following sections show the information stored for each of these classifications.

Sense Bytes

- The sense information is stored in control storage for access by the sense command.

The first four bytes of sense information are stored at control-storage address FFAC, from which the information is transferred by the sense command. This section contains only a chart of the four sense bytes. The detail of these bytes is discussed under "Sense Command."

	Byte 0	Byte 1	Byte 2	Byte 3
Bit 0	Command Reject	Data Check In Count Field	Unsafe	Ready
Bit 1	Intervention Required	Track Overrun		On Line
Bit 2	Bus Out Parity	End of Cylinder	Serdes Check	Unsafe
Bit 3	Equipment Check	Invalid Sequence	Selected Status	Write Current Sense
Bit 4	Data Check	No Record Found	Cyclic-Code Check	Pack Change
Bit 5	Overrun	File Protected	Unselected File Status	End of Cylinder
Bit 6	Track Cond Check	Missing Address Marker		Multi-Module Select
Bit 7	Seek Check	Overflow Incomplete		Seek Incomplete

Status Bytes

- Bits 32 through 39 of the CSW define the unit status.
- Bits 40 through 47 of the CSW define the channel status.

The status bits carry the standard System/360 channel designations, but in some cases are redefined for the IFA operation. The CSW bit numbers are shown after the bit name.

Unit-Status Byte

Bit 0 Attention (CSW-32) — This bit is not used for IFA/2319A operations.

Bit 1 Status Modifier (CSW-33) — This bit is set as the result of a successful search-command operation to indicate that the condition has been found. The status-modifier bit is also set with the busy status (bit-3) to indicate control-unit-busy.

Bit 2 Control Unit End (CSW-34) — This bit is set if the control-unit-busy has been presented and the condition has ended. The control-unit-end bit is also set with the unit-check status (bit 6) when a check condition occurs after the device end.

Bit 3 Busy (CSW-35) — This bit presented alone in an initial status indicates that the device is busy. The busy bit is also presented with the status modifier (bit 1) to indicate that the control unit is busy with an erase sequence.

Bit 4 Channel End (CSW-36) — This bit is set at the end of the channel operation of each command.

Bit 5 Device End (CSW-37) — This bit is set with the channel-end bit for most commands. The device-end bit is set later when the access mechanism is ready after a seek or recalibrate command. Device-end is also presented when an attached drive goes from not-ready to ready condition.

Bit 6 Unit Check (CSW-38) — This bit indicates that an error condition has been detected in the program or in the file hardware. These conditions are defined by the sense bytes stored by the control unit. The sense information is available to the program through the sense command. If the unit-check bit is not set, it may be assumed that no significant sense information is stored.

Bit 7 Unit Exception (CSW-39) — This bit indicates that an end-of-file condition has been detected for any data operation except read-count, write-CKD, search-key, or search-ID commands. The unit-exception bit is also set for a data length of zero.

Channel Status Byte

Bit 0 Program Controlled Interrupt (CSW-40) — This bit indicates that the interrupt posting in the CSW was caused by the PCI flag in the CCW.

Bit 1 Incorrect Length (CSW-41) — This bit indicates that the CCW count and the file record length did not agree and that the SLI flag was not set in the CCW.

Bit 2 Program Check (CSW-42) — This bit indicates that some address, coding, or sequence in the program is not correct for the defined operation. The operation and chaining are suppressed.

Bit 3 Protection Check (CSW-43) — This bit indicates that the storage protect key was violated for storage access during a file share-cycle. The condition is detected by the CPU circuits but is referred to the IFA for posting. The operation and chaining are suppressed.

Bit 4 Channel Data Check (CSW-44) — This bit is set to indicate a bus parity error during IFA share-cycles.

Bit 5 Channel Control Check (CSW-45) — This bit indicates that the CPU detected an error other than storage protect or storage

wrap during a file share-cycle. The condition is detected by the CPU circuits but is referred to the IFA for posting. The operation and chaining are suppressed.

Bit 6 Interface Control Check (CSW-46) — This bit indicates that a mini-op command condition existed during the operation.

Bit 7 Chaining Check (CSW-47) — This bit is not set by the IFA.

Extended Logout

- The IFA stores its logout in the same area as the selector channels.
- The conditions for extended logout storage by the IFA are the same as for a selector channel.

The IFA stores an extended logout either as part of the CC-1 ending routine in initial selection or as part of the interrupt-handling routine when conditions require. No extended logout is stored unless either the channel-control or the interface-control check status bits are set. The logout must also be allowed by the logout mask (CR14 bit 2), and there must be space to store the full logout.

The IFA logout is stored in the same area as the selector-channel logout starting at the address designated by the IOEL pointer (storage location 172). If the pointer address does not permit the full required logout to be stored within main storage, the logout is not stored. A short logout (10 words) is stored for IFA check conditions only. A long logout (19 words) is stored if the CPU machine-check conditions also exist. Byte 0 of word 1 defines the valid log as a byte count.

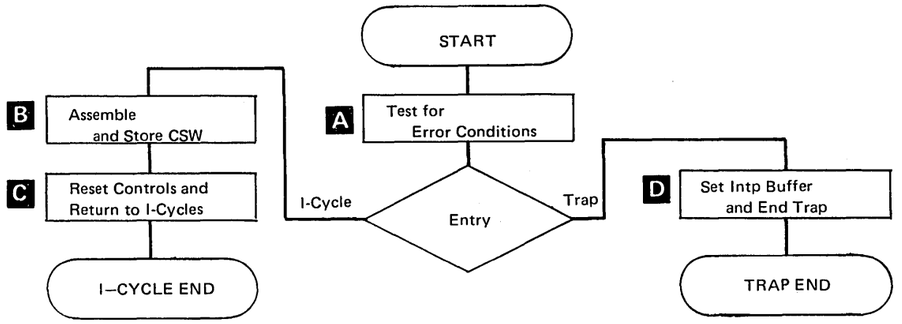
	Byte 0	Byte 1	Byte 2	Byte 3
Word 1	Log Length	FCH	FCL	FOP
Word 2	FFL	FCS	FST	FGL
Word 3	FTO	FTI	FBO	FDR
Word 4	MCKA (Ext 07)			
Word 5	MCKB (Ext 06)			
Word 6	FDRL			
Word 7	FD (LS 28)			
Word 8	FC (LS 29)			
Word 9	FM (LS 2A)			
Word 10	FW (LS 2B)			
Word 11	Retry Counts (CS F800)			
Word 12	Retry Register 1 (CS F804)			
Word 13	Retry Register 2 (CS F808)			
Word 14	Retry Register 3 (CS F80C)			
Word 15	Retry Register 4 (CS F810)			
Word 16	Control Word (CS F814)			
Word 17	System Register (CS F818)			
Word 18	I Register (CS F81C)			
Word 19	U Register (CS F820)			

IFA Extended Logout Assignments

Ending Sequence Flow Chart

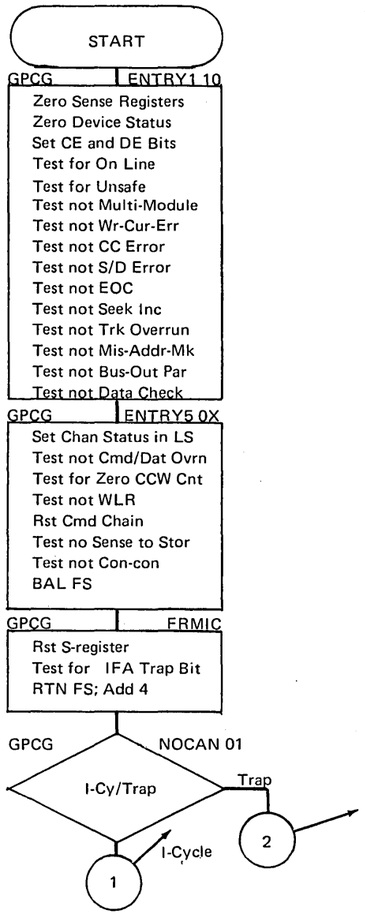
Entry to the ending sequence is from two normal points in operation. When no errors occur in the operation or when the errors are of a programming nature, the normal routine branches the operation to the ending routine. If there is a hardware-detected error resulting in an error trap, the operation goes to the ending routine if no functions are required from the operating routine. These conditions are discussed under the "Error Trap Sequence." Two ending conditions exist depending on the entry condition. An initial selection entry returns to I-cycles, and a trap entry ends the trap.

The ending sequence tests the file status and error conditions to set the appropriate sense and status indicators. For the initial selection entry, the status is stored in the CSW and the operation is ended with the CC-1 posted. A trap entry sets the interrupt latch and stores information in the interrupt buffer. In both cases the module is deselected and the controls are reset.



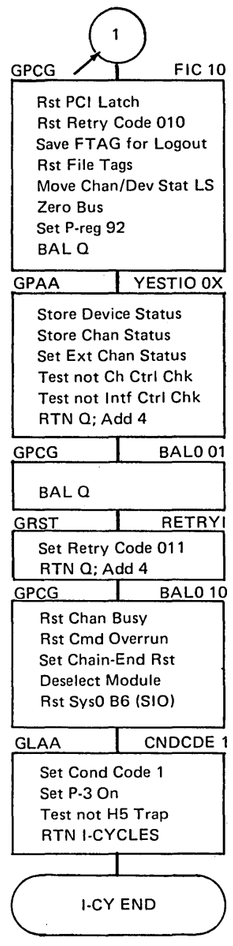
A Test for Error Conditions

- The previous information in the sense and status registers is cleared.
- The channel-end and device-end status bits are set.
- The status conditions of the disk file are tested, and the appropriate sense bits are set.
- The control-unit error indicators are tested, and appropriate sense bits are set.
- Store sense information and set required status for ending conditions.
- The routine branches between a trap entry and an initial selection entry.



B Assemble and Store CSW

- The FTAG register is stored for later use in logout.
- Store channel and device status in the CSW.

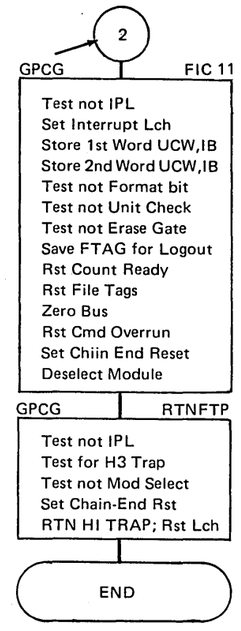


C Reset Controls and Return to I-cycles

- The module is deselected and the control-unit is reset.
- The operation is ended after posting CC-1 and returning to I-cycles.

D Set Interrupt Buffer and End Trap

- The interrupt latch is set to cause an interrupt to store and handle the CSW conditions.
- The UCW information in local storage is stored in the interrupt buffer in control storage.
- The module is deselected, and the control unit is reset.



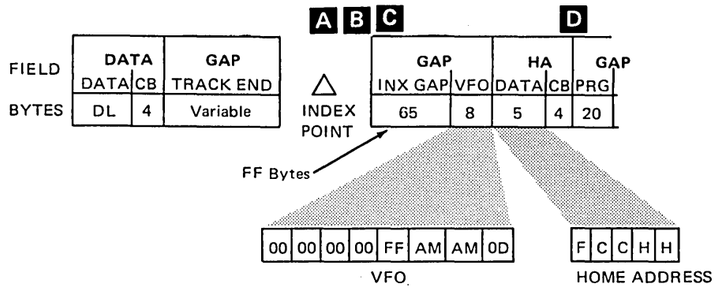
Index Trap to Advance Head

When the index point is encountered without the index-hold modifier being set in the mini-op, the hardware forces a no-op mini-op with a count of 27. When that count reaches 20 the index trap is set instead of the normal high trap. The index trap can result in one of three routines:

1. A normal single-track operation without head advance.
2. A multi-track operation to advance the head.
3. A record-overflow operation to advance the head.

The latter two routines are similar in operation and differ from the first in the head advance. In all cases, except when the command is to read either the HA or R0 field, the home-address field is read into control storage. If the head has been advanced, the new head address is compared to ensure that the next track is being read. The previous mini-op and count are re-entered to continue the operation.

For this diagram, the multi-track advance head routine is shown. The operation starts with the honoring of the index trap and ends with the return from the high trap following the HA entry.

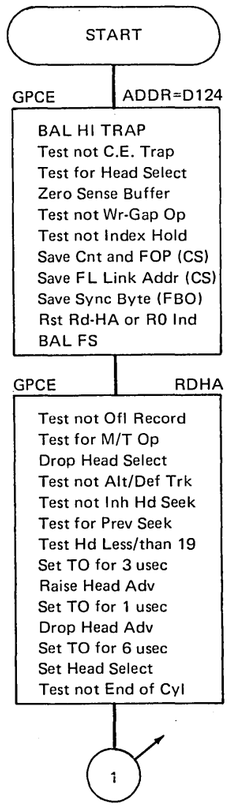


A Index-Trap Entry

- The entry conditions are tested to determine that the trap is required.
- The FOP, the counter, and the sync byte of the current operation are saved in control storage.

B Advance-Head Controls

- Test to determine that the multi-track (or overflow) operation was in process.
- Drop the head selection and determine that the present head address is less than 19.
- Test the file mask for previous seek and that the head seek is allowed.
- Set the head-advance bit on the bus and time out for the advance.
- Raise the head-select line and test for possible end-of-cylinder indication.

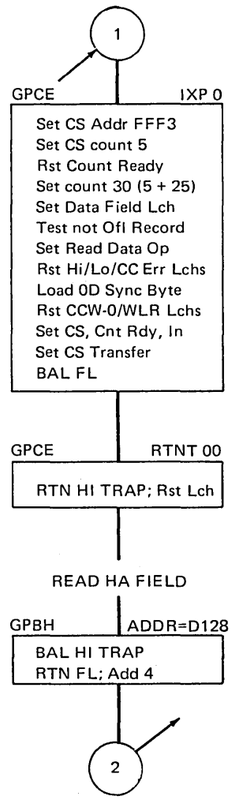


C Set the Controls to Enter the HA Field

- Set control-storage address and count for entry of the HA field.
- The 0D sync byte is entered to identify the field.
- The read data into control-storage gating is set for the operation.

D Test for Correct Head Switching

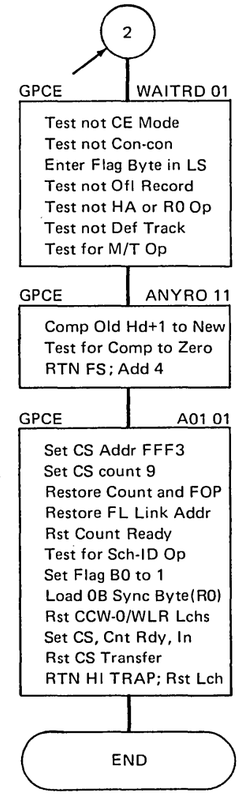
- The overflow-record and single-track routines are branched at this point.



- The flag byte is tested for a possible defective track.
- The old head address plus 1 is compared with that read from the HA field.

E Set the Controls for the Previous Operation

- Set control-storage address and count for entry of the count field.
- Restore FOP and count control storage.
- The 0B sync byte is entered to identify the R0 record-count field.
- The read data into control-storage gating is set for the operation.
- The return from the high trap ends the index-trap sequence to advance the head.

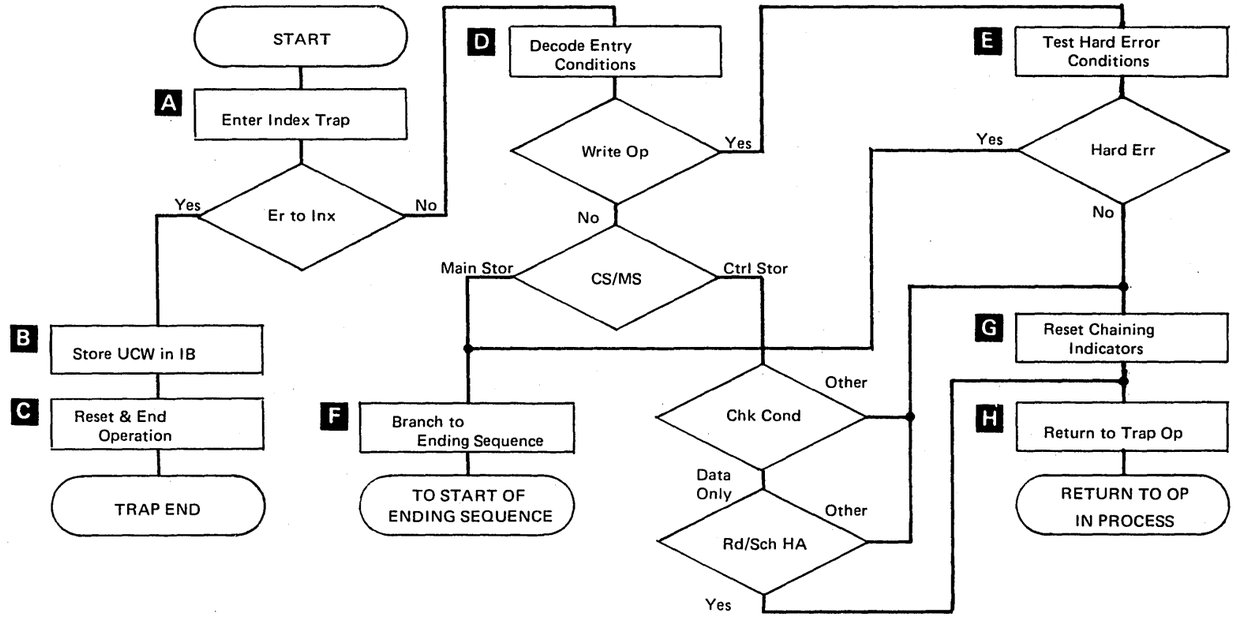


Error Trap Sequence

An error trap is initiated by the hardware when one of the following errors is detected during hardware operation:

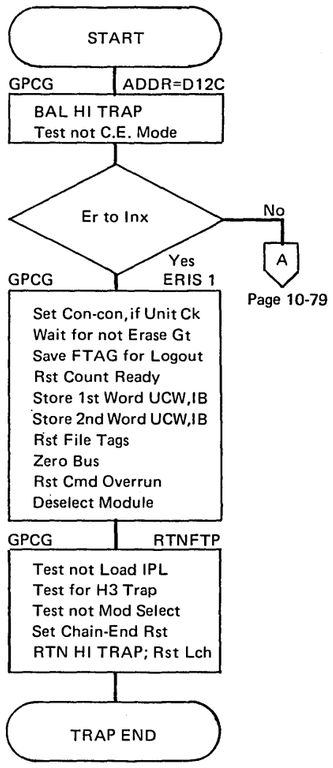
Multi-module selected	Unsafe
Wrong sync	Missing address mark
MOP parity check	Loss of on-line
Command overrun	Write track overrun
Data overrun	Write current error
IFA channel error	SERDES check
Data check	CC hardware error

The error trap occurs in place of the normal high trap when the count has reduced to 20. The error-trap routine can end in one of three routines depending on conditions. If an erase-to-index sequence is in progress, the routine stores the UCW in the interrupt buffer and ends the operation and the trap. When no erase-to-index is in progress, the operation and the check conditions are tested to determine the routine. Most conditions branch to the start of the normal ending sequence to post the sense and the status before ending the operation. Control-storage operations and write commands without errors are returned to the operation in progress to end that sequence before ending the operation.



A Enter Index Trap

- The error trap is a high-trap level with an address of D12C.
- A test is made to determine that the operation is not in CE mode.
- The routine is branched into two paths depending on the presence of the erase-to-index indicator.



B Store UCW in IB

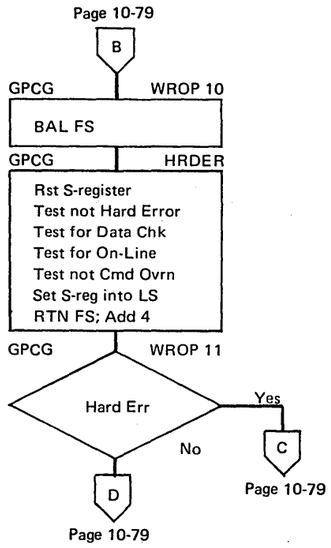
- The contingent-connection latch is set if there is a unit-check condition.
- After the erase gate falls, the FTAG register is saved for logout.
- The current UCW information is stored in the interrupt buffer in control storage.

C Reset and End Operation

- The bus-out lines are cleared, and the file tag lines are reset.
- The module is deselected, and the control unit is reset.
- The H3 bit is reset, and the operation is returned to the trapped-into operation.

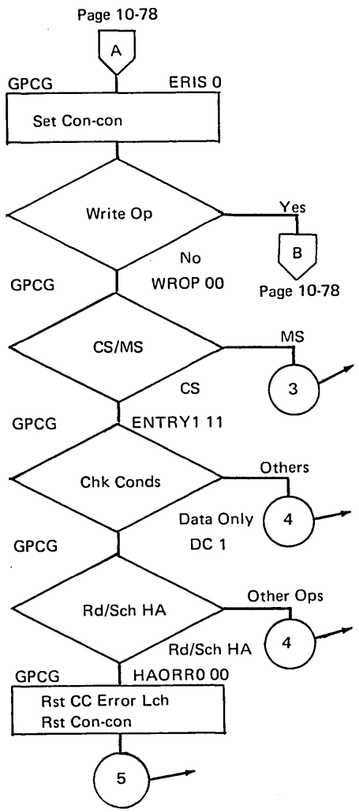
E Test Hard Error Conditions

- The hard errors unsafe, not on-line, and multi-module status are tested in the FDS register.
- The data check is tested in the FSD register and the command overrun is tested in the FGT register.
- The hard error test branches the write operations a direct ending and a return to the processing routine.



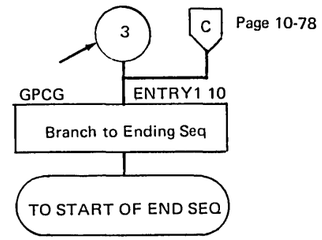
D Decode and Entry Conditions

- The contingent-connection latch is set to indicate the condition.
- The command is tested for a write operation that requires a test for hard errors.
- The storage entry is tested to determine whether main or control storage was being used.
- Control-storage read operations are further tested for check conditions and for HA operations.
- All conditions except a data-only error on an HA operation reset the chaining flags before continuing.



F Branch to Ending Sequence

- Read operations, search operations, and write operations with hard errors are branched directly to the ending routine.

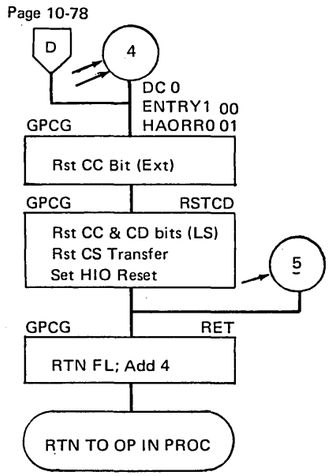


G Reset Chaining Indicators

- CC flag indicator in the FFL external is reset.
- The CC and CD flags in local storage FC0 are reset to end the chaining.

H Return to Trap Operation

- A count of 4 is added to the link address stored before ending the last trap.
- The operation returns to the point in the processing routine that would have been entered by the high trap.

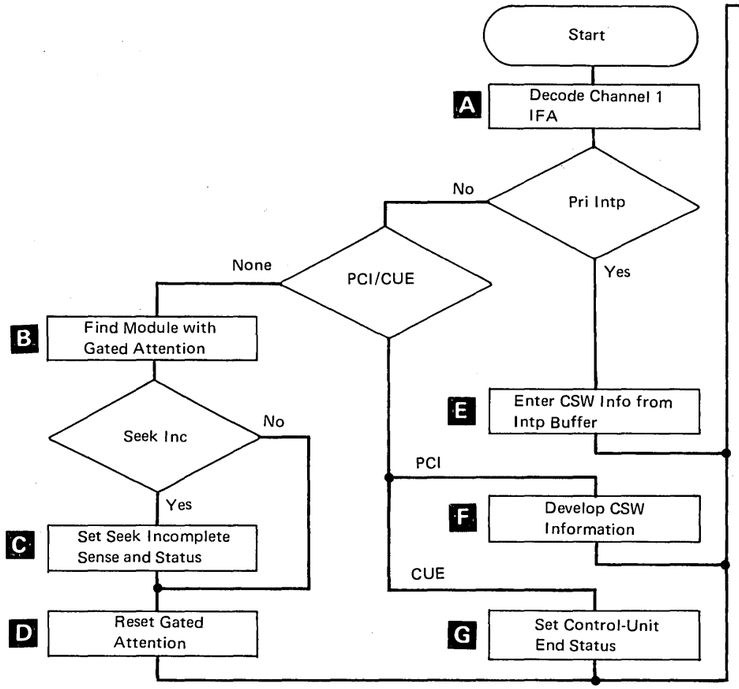


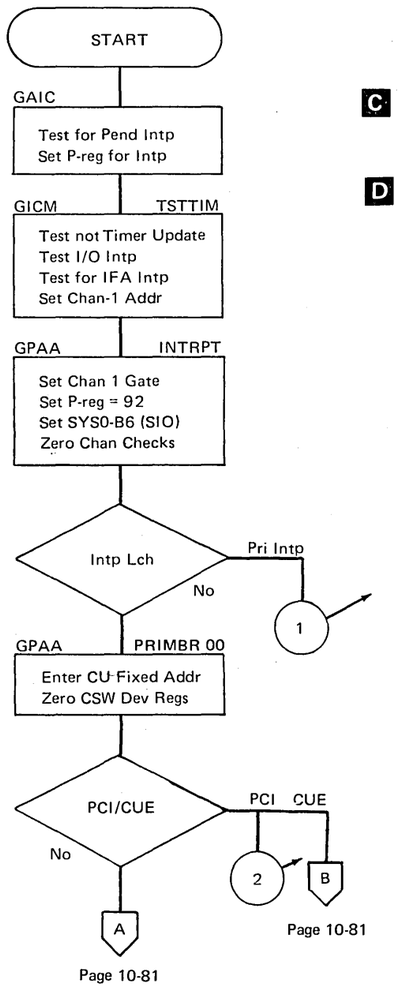
IFA INTERRUPT ROUTINE

The IFA interrupt routine handles the interrupt requests intiated by the IFA and disk files. The operation is similar to the interrupt-handling routine for selector channels. Entry to the routine occurs when the CPU honors a request of the IFA through a test of channel requests.

The interrupt routine tests for the type of interrupt (primary, CUE, PCI, and GA) to determine the path through the routine. The previous ending routine has posted the sense and status conditions in control-storage buffers. For the PCI interrupt, the information is still active in local storage. The status information is updated as required and stored in the CSW in main storage. The machine independent log is stored, and then the routine returns to the channel routines to interchange the PSWs.

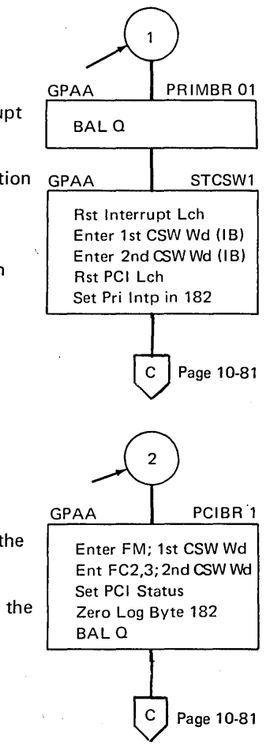
If either the channel-control or interface-control check conditions are posted, the IFA routine posts the machine independent log area along with the CPU log if differences exist.





A Decode Channel 1 IFA

- When the I-cycles routine detects an interrupt request, it is decoded for the appropriate device.
- The IFA request decodes as channel 1 and is branched to the IFA GPAA routine for processing.
- The appropriate P-register and SYS0 information is posted for the operation.
- Tests are made of the primary, PCI, and CUE indicators to determine the path.



E Enter CSW Information from Interrupt Buffer

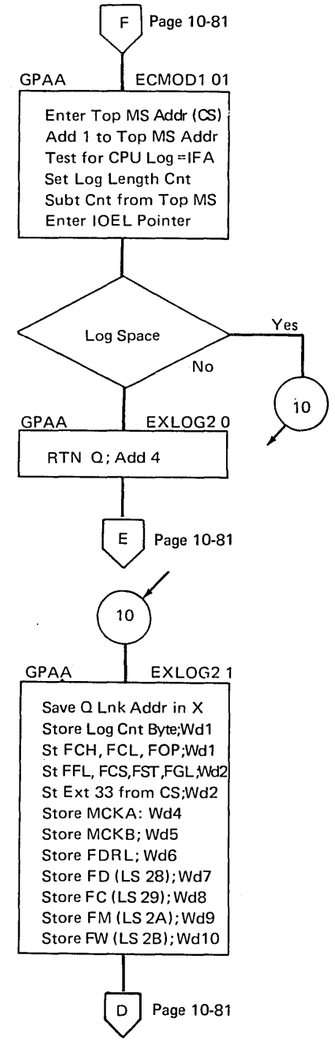
- The previously stored UCW information in the interrupt buffer is entered for update.
- The interrupt latch and the PCI latch are reset.

F Develop CSW Information

- The CSW words are assembled from the FM and FC registers.
- The PCI status bit is entered to show the type of interrupt.

J Extended Log Setup

- Compare CPU log with IFA log to determine whether CPU log needs to be posted.
- Test log pointer to determine whether space is allowed in main storage for required words.



K Store Extended Log

- The link address in the Q-register is saved in the X-register to make the Q-register available.
- The IFA externals and local-storage registers are stored first.
- If the byte count has not reduced to zero, the CPU stored log is written to the end of the byte count.

B Find Module with Gated Attention

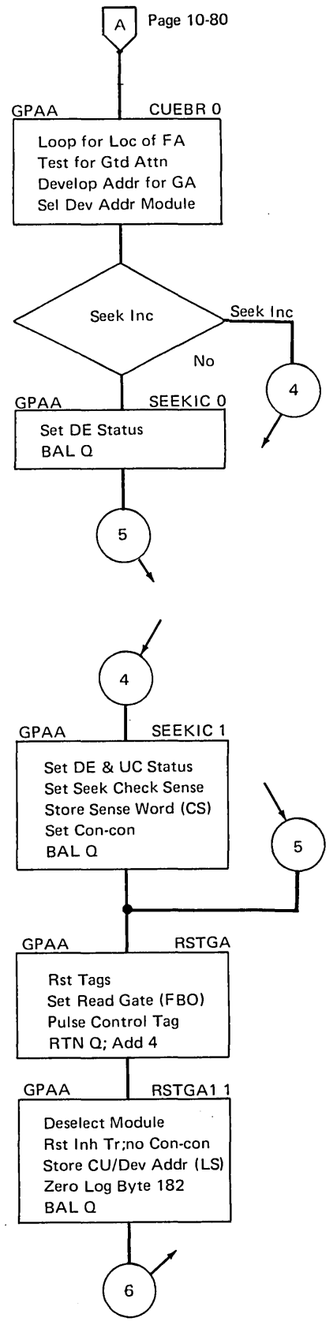
- If none of the interrupt conditions tested are present, the gated attention condition exists.
- The FAT register is tested to find the module that the gated attention applies.
- After selecting the indicated module the seek-incomplete indicator is tested.
- If no seek incomplete exists, the DE status is posted.

C Set Seek-Incomplete Sense and Status

- The DE and unit-check status are posted along with the seek-check sense.
- The sense word is stored in control storage, and the contingent-connection latch is set to indicate.

D Reset Gated Attention

- The read gate is set on bus-out, and the control tag line is pulsed to reset the GA.
- The module is deselected, and the CU and device addresses are stored in local storage.



G Set Control-Unit End Status

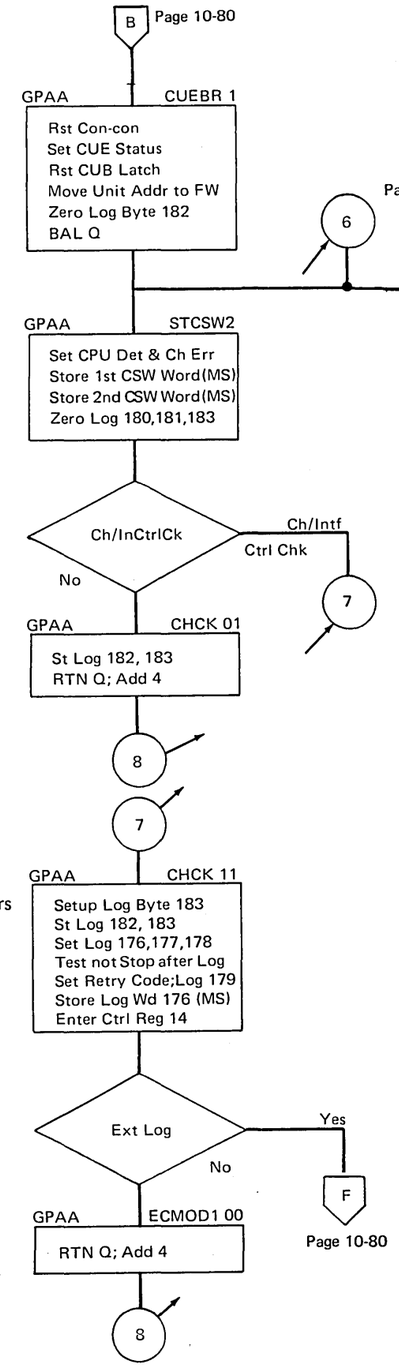
- The control-unit end status is entered to show the type of interrupt.
- Move unit address into FW register.

H Test for Logout Requirements

- Post machine independent log information.
- Test log mask bit to determine whether logout is allowed.
- When set to stop after log, the log is posted starting at MS 512 without reference to the logout mask or the IOEL pointer.

I Store CSW Words

- Enter CPU-detected errors and channel errors into the CSW status.
- Store the resulting CSW words into the CSW in main storage.
- Test for possible channel/interface control status that requires posting the log.



L Store Log Identifier

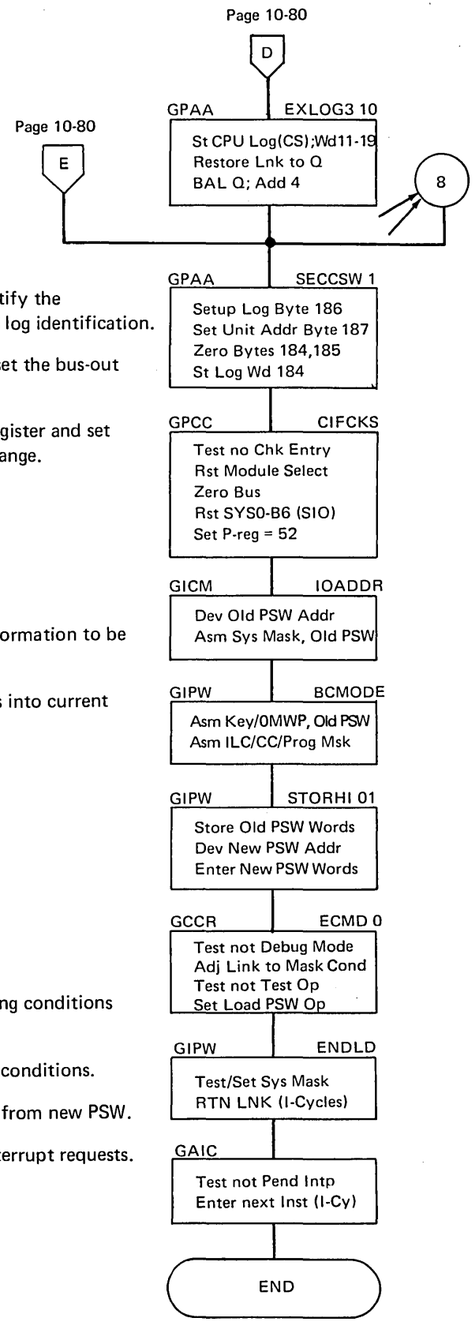
- Store information to identify the origination of the channel log identification.
- Deselect the module and set the bus-out to zero.
- Remove B6 from SYS0-register and set P-register for PSW interchange.

M Interchange PSW Words

- Assemble current PSW information to be stored as old I/O PSW.
- Enter new I/O PSW words into current PSW locations.

N Test for Other Interrupts

- Adjust new PSW to masking conditions before continuing.
- Test and set system mask conditions.
- Enter I-cycle link address from new PSW.
- Test for possible other interrupt requests.



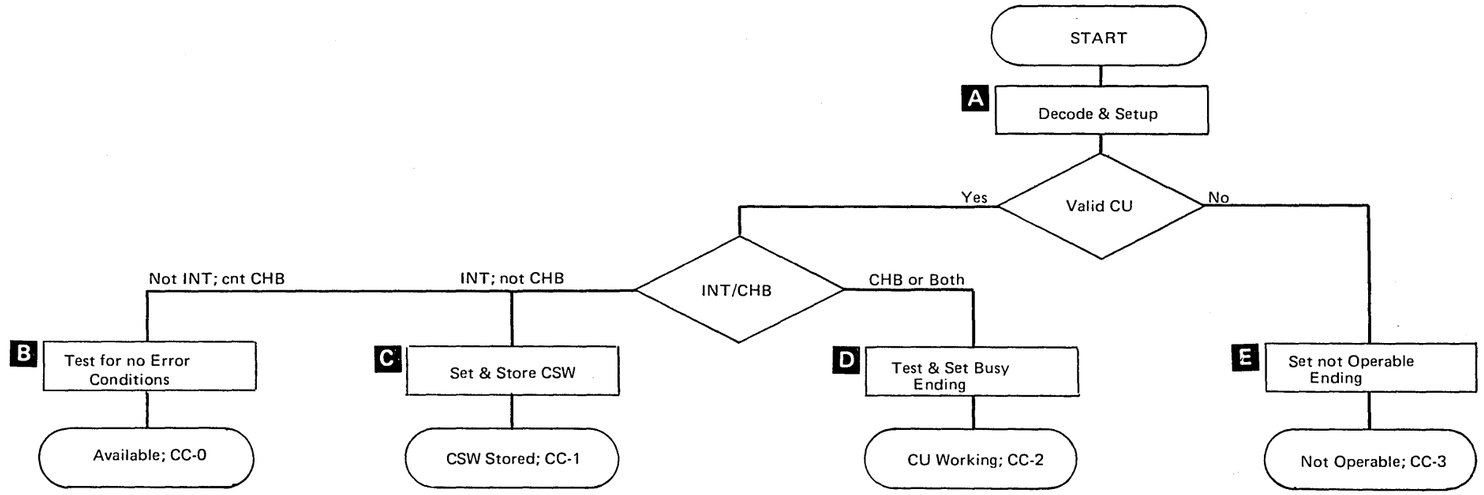
IFA TEST I/O OPERATION

The Test I/O (TIO) instruction is used to determine the operating conditions of the file and controls. At the end of the operation, the routine sets one of the four condition codes as its indication.

CC-0	Available for use.
CC-1	CSW has been stored.
CC-2	Control unit working.
CC-3	Control unit not operable.

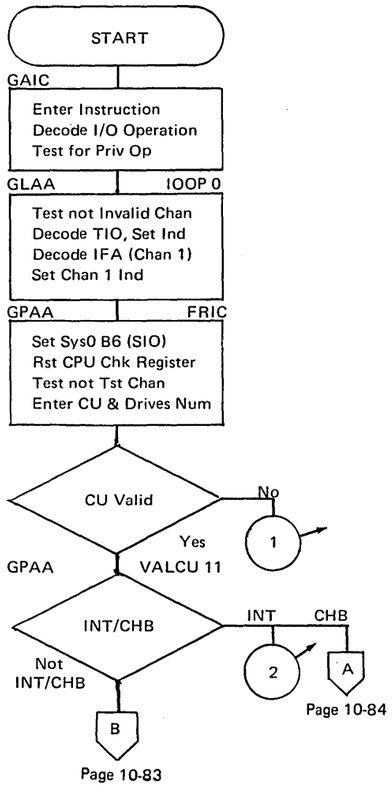
The TIO instruction is entered in the I-cycles routine and decoded as an I/O operation. The operation branches to the channel routine for channel decode. A decode of channel 1 is branched to the IFA GPAA routine. The TIO instruction does not use the CAW information.

The flow chart shows the major branching for the four ending conditions. The detection of error conditions causes further branching as indicated in the comments.



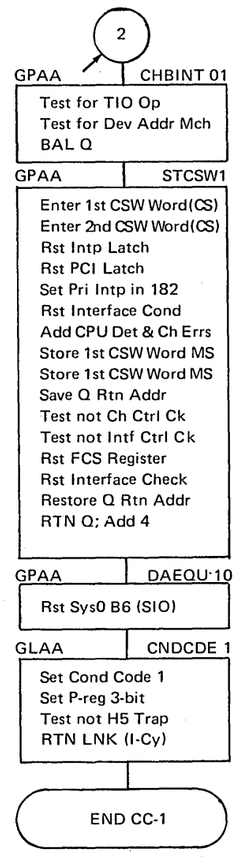
A Decode and Setup

- The TIO instruction is read during I-cycles and decoded as an I/O operation.
- The channel controls decode the channel-1 address as IFA and branch the operation to the GPAA routine.
- The CAW is entered by the routine, but the information is not used.
- The control-unit and module addresses are entered and tested for validity.
- The control-unit validity, the interrupt latch, and the channel-busy latch determine the initial branching.



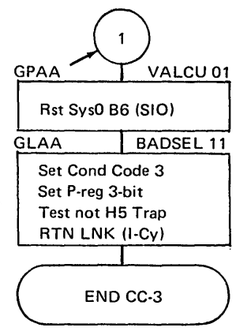
C Set and Store CSW

- CC-1 is set when an interrupt is pending for the addressed module; the CSW is stored by the routine.
- The previously assembled CSW information is entered from the interrupt buffer in control storage.
- The CPU-detected errors and the channel-status information are added to the CSW before storing in main storage.
- The presence of either the channel or interface control-check status or the contingent-connection indicator causes the logout ot be stored.
- After the sequence is completed, the operation is ended with the setting of CC-1 and returning to I-cycles.



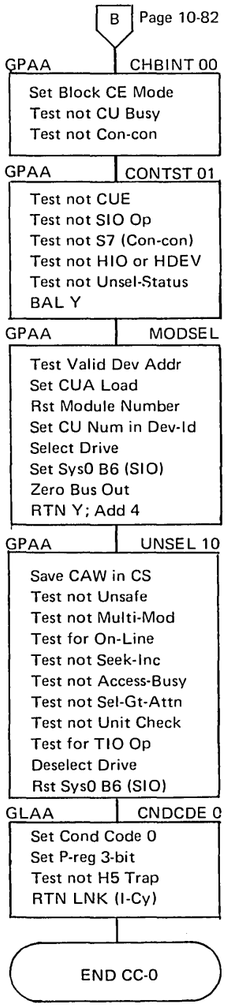
E Set Not Operable Ending

- CC-3 is set when an invalid control-unit address is used.
- An invalid or inoperable module address results in posting CC-1.



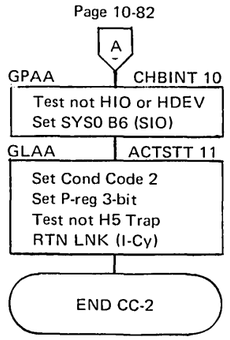
B Test for No Error Condition

- CC-0 is set when the routine finds no pending conditions and the control unit and the addressed module are operable.
- The module address is tested, and the appropriate UCW information is developed.
- The module is selected, and the status byte is tested to determine whether the file module is ready.
- If any status condition is present, the operation is branched to the CC-2 ending.
- Tests are made for contingent-connection (con-con) and the control-unit-end conditions that indicate that the control-unit is still busy.
- The module is deselected and the CC-0 indication is posted if no status conditions exist.



D Test and Set Busy Ending

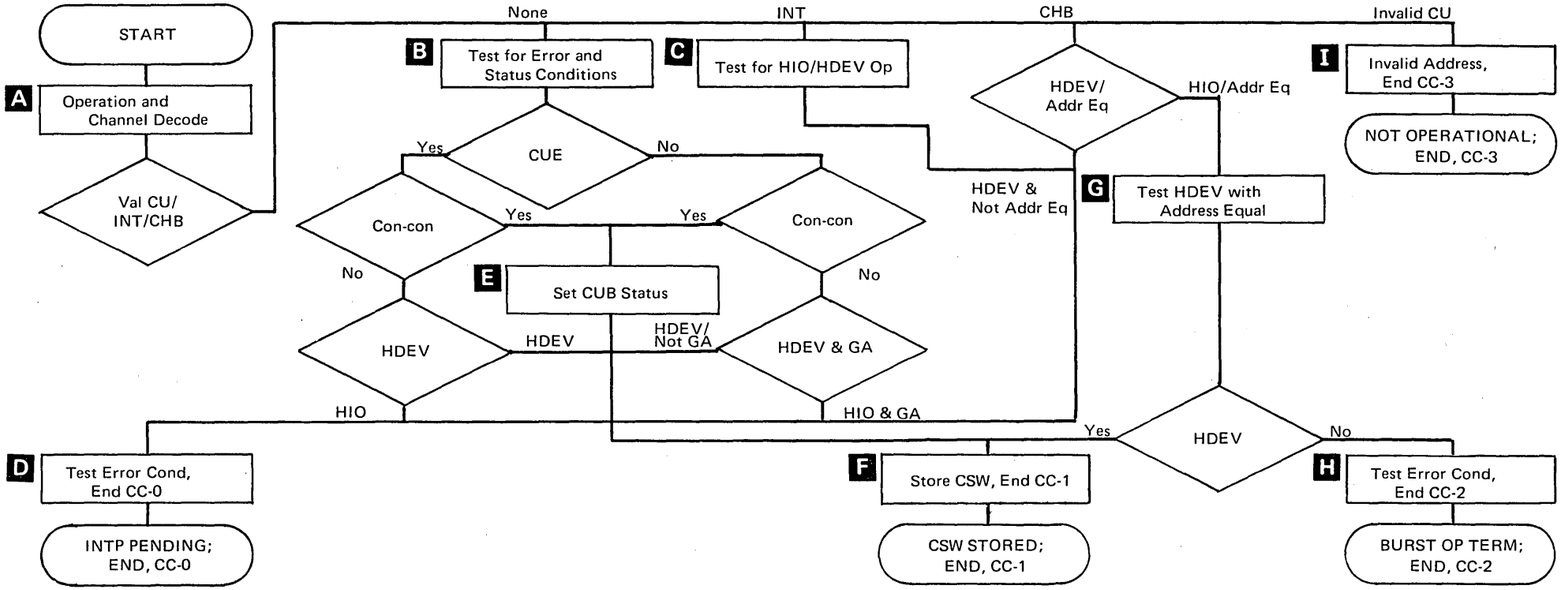
- CC-2 is set when the control unit is busy as the result of the previous IFA operation.
- The CC-2 indication is posted if no error conditions are detected.



IFA HALT I/O OR DEVICE OPERATION

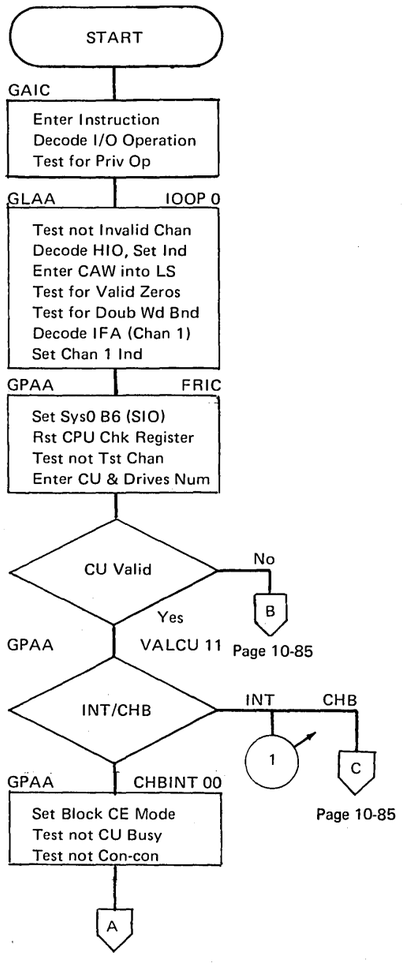
The channel Halt I/O instruction is used to program-stop a started operation on the IFA. This instruction has two forms to permit stopping either all IFA operation (HIO) or stopping an addressed device (HDEV). When either form of the instruction is given with a channel-1 address, the operation is routed to the IFA for handling. The routine takes four major branches, depending on address validity, channel busy, and interrupt-pending conditions. The pending conditions must apply to the addressed device when that form of the instruction is used. The operation ends by setting one of the four condition codes with the following meaning:

CC-0	Interrupt Pending
CC-1	CSW Stored for Addressed Device
CC-2	A Burst Operation was Terminated
CC-3	Not Operational (Invalid address)



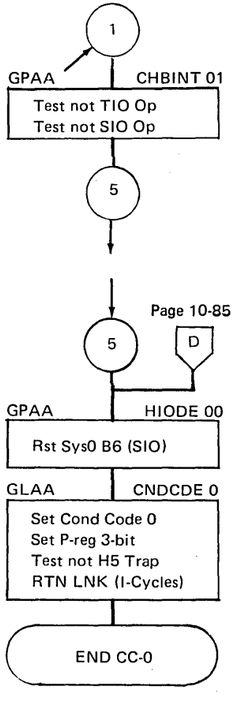
A Operation and Channel Decode

- The halt I/O instruction is entered during the normal I-cycles.
- The instruction is decoded as a channel operation and is branched to the channel routine.
- The CAW is entered and tested as a normal function, but the information is not used.
- The decode of channel 1 branches the operation to the IFA routine.
- Tests are made for invalid CU address, the interrupt latch, and the channel-busy latch to determine the routine to follow.
- A valid CU address without either the interrupt or channel-busy latches forms a fourth path.



C Test for HIO/HDEV Operations

- Tests are made for other possible operations because this is a common initial-selection routine.



D Test Error Condition, End CC-0

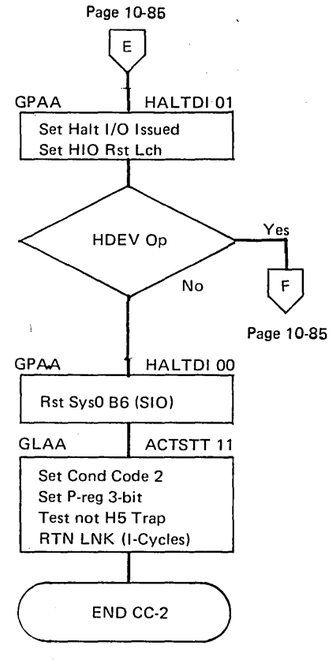
- Before the operation is ended, SIO indicator is reset and the inhibit traps latch is reset.
- The operation returns to the selector-channel routine to set the condition code 0.
- After setting the P-register, the routine returns to I-cycles using the stored link address.

G Test HDEV with Addr Equal or HIO

- A channel-busy condition is tested for an address-equal condition.
- The HDEV operation with the address unequal is ended as an interrupt pending.
- An HIO operation results in ending a burst operating condition before ending.
- The HDEV operation with an equal address results in ending after posting the CSW.

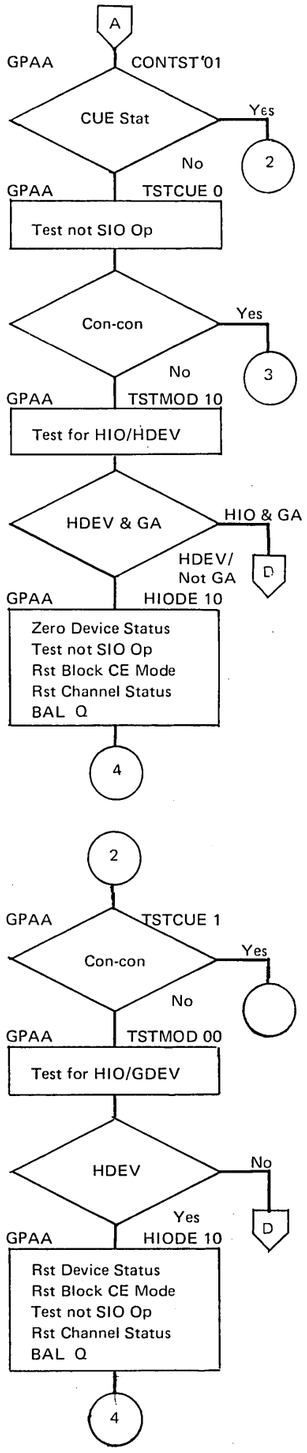
H Test Error Conditions, End CC-2

- A HDEV operation branches to store the CSW.
- If it is not an HDEV, the operation branches to end the routine as a burst op termination.
- The SIO indicator is reset, and the inhibit-trap latch is reset before branching to the selector-channel routine to set condition code 2.
- After setting the P-register, the routine returns to I-cycles using the stored link address.



B Test for Error and Status Conditions

- Tests are made for operating conditions to set appropriate indicators and to determine the ending condition code.
- The control-unit-end status condition determines whether the gated-attention condition should be tested.
- The contingent-connection condition branches the operation to store the CSW after setting the CUB latch.
- The HDEV operation without gated-attention condition branches to store the CSW.
- The HIO operation and gated-attention conditions branch to the interrupt pending ending.

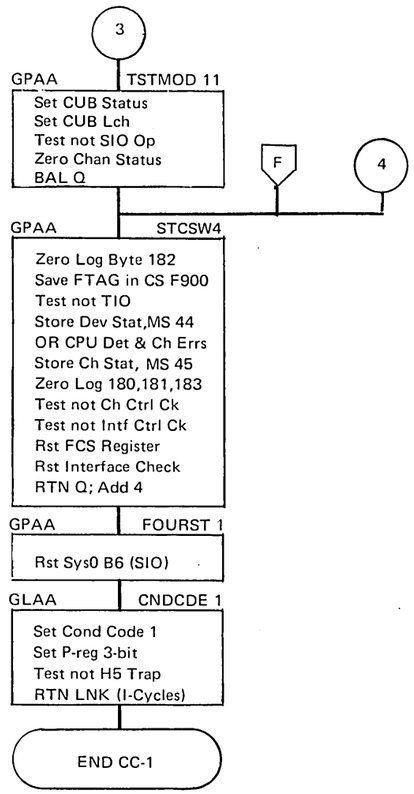


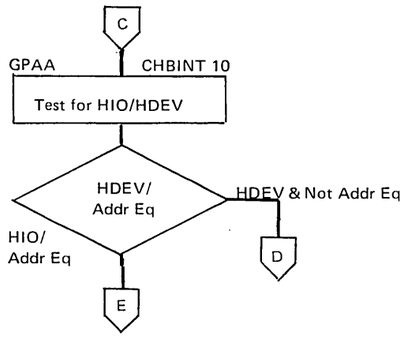
E Set CUB Status

- With the contingent-connection condition set, both the control-unit busy status bit and the latch are set.
- After ensuring that the operation is the HIO/HDEV, the routine branches to set the CSW.

F Store CSW, End CC-1

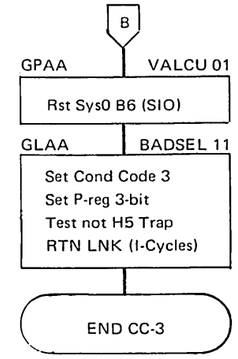
- The previous channel status is reset before entering the new status.
- The FTAG information is stored in control storage for possible logout.
- The device status developed is stored in main-storage address 44.
- The channel status developed is stored in main-storage address 45.
- The independent log area is zeroed.
- The extended logout is executed if either the channel-control or interface-control check status was posted.
- The channel-busy latch is reset, and the module is deselected.
- The SIO indicator is reset, and the inhibit-trap latch is reset before branching to the selector-channel routine to set condition code 1.
- After setting the P-register, the routine returns to I-cycles, using the stored link address.





I Invalid Address, End CC-3

- An invalid control-unit address causes the operation to branch to a non-operational ending.
- The SIO indicator is reset, and the inhibit-traps latch is reset before branching to the selector-channel routine to set condition code 3.
- After setting the P-register, the routine returns to I-cycles using the stored link address.



MAINTENANCE FEATURES

CE PANEL

A CE panel for IFA operation is provided under the rear cover of the 2319-A01 box. The controls and indicators function with the inline diagnostic tests included in the IFA microroutines. The tests are used to operate the logic and hardware of a disk-storage module identified with the "spare" module plug. The tests are not intended to test the IFA control-unit operation.

The normal use for the CE panel is when a fault has been indicated in a module during customer operation. After switching his work to another file module, the problem module can be serviced online while the customer continues his operation. The "spare" module identifier plug and either the CE disk pack or an unused formatted disk pack are installed. A test is selected with the test-select switch and the desired head or cylinder selection made on the switches. When the allow-CE-mode switch is set to the ON position, the selected test will be performed when the IFA control unit is not busy. A complete test is performed, and then the control unit is returned to the customer's program. Details of the inline diagnostic tests are covered in this manual under "Inline Diagnostics" and in microroutine GPD0.

If no error occurs during the test, the test is repeated after a delay of 200 microseconds or when the control unit is again not busy. When an error is detected during the test, the repeat is taken unless the error-disable switch is set to ON. Error conditions are displayed in the error-display indicators on the CE panel for analysis. The meaning of the display is detailed in this manual under "Inline Diagnostics" and in microroutine GPD0. When the error-disable switch is ON, error-display indicators are reset and the test is repeated to allow observation.

The allow-CE-mode and error-disable switches should be set to the OFF position when the CE panel is not in use.

Lights and Switches

Error Display: This group of eight lamps are the indicators for the FED external register. The microroutine sets the FED latches for the error indication when an error stops the test. The lamps remain lit until the operation is taken out of CE mode or the test is repeated.

CE Mode: This lamp is lighted when the IFA has honored the test request and remains lit until the test is complete.

CE Error: This lamp lights when the test stops with an error indication set in the error display. The lamp remains lit until the FED latches are reset.

Probe: This lamp is lighted when the probe latch is set from the wired conditions existing in the disk-file logic. The lamp remains lit until the lamp test switch is operated.

Allow CE Mode: This switch is used to place the IFA in CE mode to perform the inline diagnostic tests. The switch should be in the OFF position when tests are not being performed.

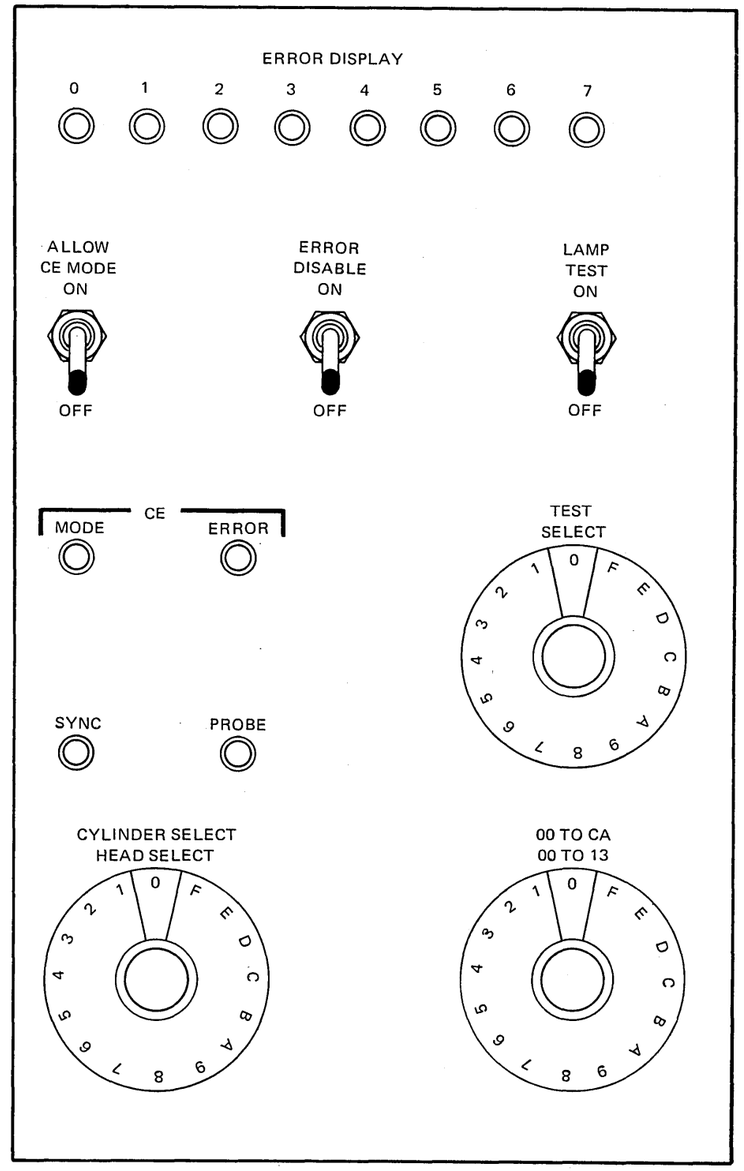
Error Disable: This switch disables the error-stop control when an error is detected during a test. When the switch is in the ON position, the test is repeated following a delay time-out or when the IFA is not busy. The switch should be in the OFF position when tests are not being performed.

Lamp Test: This is a spring-return switch used to test the indicator lamps on the CE panel. Operating the switch also resets the probe latch and turns off the probe lamp when released.

Test Select: This is a sixteen-position rotary switch used to enter the number of the inline diagnostic test to be performed. The switch should not normally be turned during a test, but the test number is read in and stored by the microroutine.

Head Select/Cylinder Select: These two rotary switches are used to set values into the FHC external register. The value set is treated as a head selection or cylinder selection as required by the specified test. The value set should be valid for its use in the test.

Sync: This is an output jack that supplies a sync pulse for an oscilloscope during test operations. The sync pulse originates in the IFA area of the CPU and provides a timing related to the test start.



2319-A01 CE Panel

INLINE DIAGNOSTICS

- The IFA has microroutines that operate with the CE panel on the 2319-A01.
- Tests can be initiated from the CE panel to be performed on a specified drive during normal customer operation.
- These tests check the hardware and control of the disk-drive unit and not the IFA control.
- The test selected repeats until an error occurs and then stops with the error indicated on the CE panel.
- The error stop can be bypassed to allow observation of the error under repeat conditions.

The inline diagnostics provided for the IFA allow troubleshooting a defective disk drive while the customer continues to operate the system. The defective drive must have the "spare" module identifier plug inserted in order to perform the inline diagnostics. A CE disk pack or a formatted scratch disk pack (with home addresses written) should be installed on the disk drive to prevent damage to the customer's records. The diagnostics are performed by setting the switches on the IFA CE control panel at the rear of the 2319-A01 frame. When the customer's program is not using the IFA and there are no pending conditions, the diagnostic controls can take over the control unit.

The test-select switch on the CE panel determines the inline diagnostic test to be performed. All sixteen positions of the switch are active because two numbers have been assigned to some tests to make decoding simpler.

Select	Test
0	Read Single Track
1	Restore Seek
2	Write Single Track
3	Disk Speed
4	Read Full Cylinder
5	Restore Seek
6	Write Full Cylinder
7	Cylinder Address Register
8	Chaining Test
9	Forward-Backward Seek
A	Write Single Track
B	Sequential Seek
C	Chaining Test
D	Backward-Forward Seek
E	Write Full Cylinder
F	Unsafe Condition

The specified test is performed once by the diagnostic controls, and then a 200-millisecond time-out is taken to allow the customer's program to regain control. The test is repeated at the end of the time-out if the IFA control unit is not busy. The diagnostic operation normally stops with errors indicated on the CE panel. The error indications can be disabled to allow scoping or other observation of the operating conditions.

GENERAL STATUS INDICATIONS		UNSAFE TEST INDICATIONS		SPECIAL CONDITION INDICATIONS	
HEX	INDICATION	HEX	INDICATION	HEX	INDICATION
21 to 3F	Drive Read/Write Failure. Bit-0 = 0 Bit-1 = 0 Bit-2 = 1 Bit-3 Write Current Failure. Bit-4 Data Check Occurred. Bit-5 Missing Address Mark. Bit-6 High Compare Detected. Bit-7 Low Compare Detected.	06	Unsafe indicator set.	01	No Record Found.
		07	EOC indicator set.	02	Return to Head 00 Failed.
		0A	Both unsafe and EOC set.	02	Failed to detect index at maximum interval. (Test 3)
		A1	Failed to set EOC with advance to head-20.	04	Return to Cylinder 000 Failed.
		B-	Force multiple head select (Y4 + Y8).	04	Index pulses occurring to close together. (Test 3)
		C-	Force read and erase gates.	08	Seek 2nd Pass Failed (Dif not = Sw)
		D-	Force write gate without erase gate.	08	Index pulses occurring too far apart. (Test 3)
		E-	Force erase gate and seek start.	0D	CAR failed to restore to original address. (Test 7)
41 to 7F	Control Unit failure. Bit-0 = 0 Bit-1 = 1 Bit-2 Channel Data or Control Check. Bit-3 Data or Command Overrun. Bit-4 Cyclic Code Hardware Error. Bit-5 Write Track Overrun. Bit-6 Bus-Out Parity Error. Bit-7 Serializer/Deserializer Error.	-2	Failed to reset Unsafe at index (with head-tag and bus-out 1).	09	CAR failed to reset to 00.
				0A	CAR failed to load to FF.
		-4	Failed to set Unsafe indicator.	0B	CAR failed to load to value in CE head/cylinder switches.
		-8	Unsafe set at entry of force test.	0C	Seek Fwd/Bkwd Diff of Sw Failed.
				0C	Index trap occurred before head conditioning. (Test 3)
				0E	Write Test not Allowed (Def/Alt)
				0F	Head Selected not Equal to Switches.
81 to FF	Disk status displayed. Bit-0 = 1 Bit-1 Off Line Bit-2 Unsafe. Bit-3 Busy. Bit-4 Pack change. Bit-5 EOC detected. Bit-6 Multi-module selected. Bit-7 Seek incomplete.			10	No module selected.
				11	Machine check during inline.
				13	An R0 key field was detached.
				FF	Gated attention failed to reset.
				03	SEEK INITIATED BUT NO GTD. ATT. IN 210 MS
				05	FAILED TO DETECT BUSY ON RESTORE SEEK
				06	
				07	

Error Condition Indications

It is desirable to follow a sequence of tests as defined by the diagnostic test chart to ensure that the error indications are not taken out of context. The error indicators are keyed to the test being performed, and thus the lights indicate only the error bits stored. These indications are defined in the Error-Condition Indications chart.

Error Condition Indications

When the test sequence is allowed to stop with an error, the eight error-indicator lights on the 2319-A01 CE panel can be read as two hex digits that define the error. If the test sequence is not stopped for the error, the error indicators are reset with the restart.

The indications shown as special conditions in the chart have special meaning for the test indicated. The unsafe test indications apply to the unsafe condition test only but do not include all of the indications that could exist. All of the tests may report with the general indications for file-status conditions. The general indications are divided into three groups that can be identified by the two high-order bit indicators.

No.	Test Operation
7	Cylinder Address Register Test
3	Disk Speed Test
0	Read Single Track Test
4	Read Full Cylinder Test
F	Unsafe Condition Test
2, A	Write Single Track Test
6, E	Write Full Cylinder Test
1, 5	Restore Seek Test
8, C	CHAINED TEST

Recommended Inline Test Sequence

GPDO INLINE ROUTINES

Diagnostic Test Decode

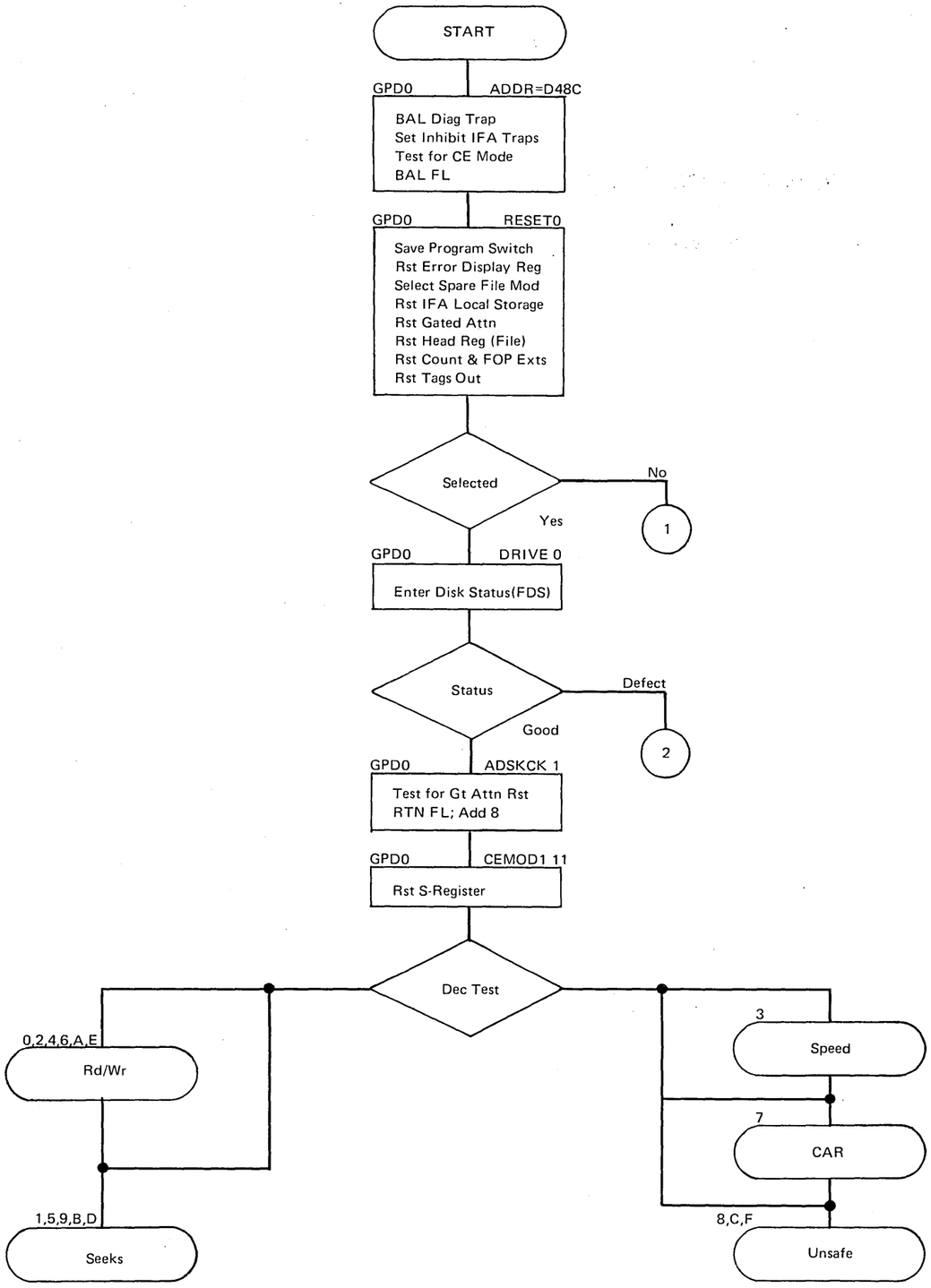
The inline diagnostic test is initiated by inserting the "spare" plug in the file module to be tested, setting the CE panel select switches, and then setting the allow-CE-mode switch to the ON position. The test can start as soon as the IFA control unit is not busy.

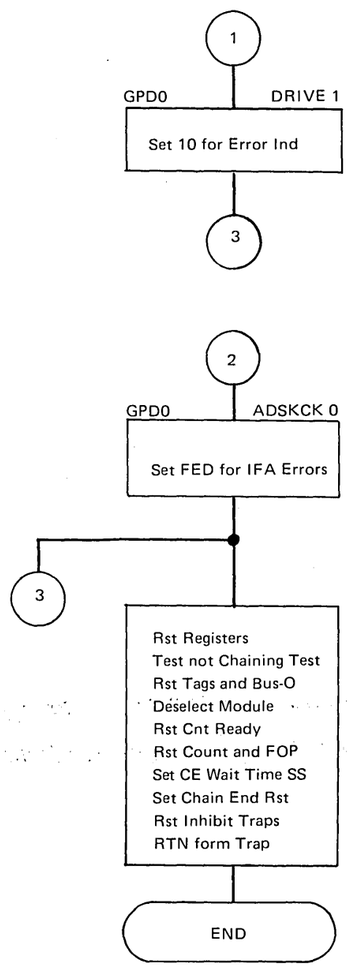
The initiating conditions request a diagnostic low-priority trap to perform the operation. The routine selects the spare module and resets the registers for the test. If no module is selected, the routine posts an error indication of 10 and goes to the ending. With a module selected, its file status is entered and tested. Any status results in posting the status as the error indicator and ending the operation. When no error or status is detected, the test selection is entered and decoded to branch the operation to the appropriate routine.

The error-ending condition resets the registers and operating latches. The module is deselected, and the operation returns from trap. The 200 ms delay is set to block the next request.

Diagnostic Chaining Test (8 or C)

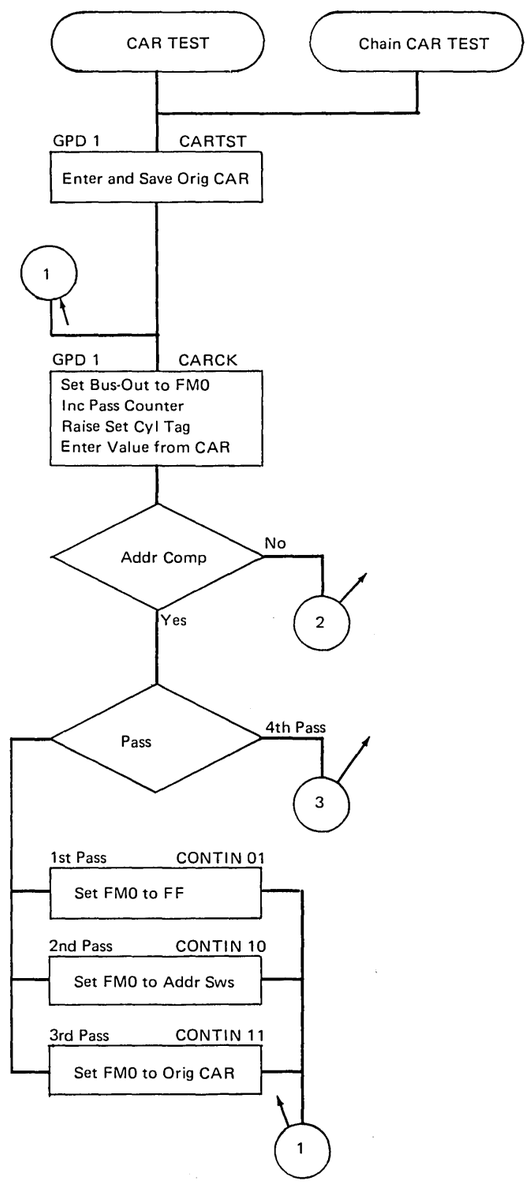
The chaining test allows performing four of the diagnostic inline tests in sequence to obtain an overall performance indication. The routine first performs the unsafe test followed by the CAR test, the write full cylinder test, and the seek restore test. The address switches on the CE panel are set for the desired test cylinder. The test error stops following any portion that contains a detected error. The error indication defines the test and the nature of the error.

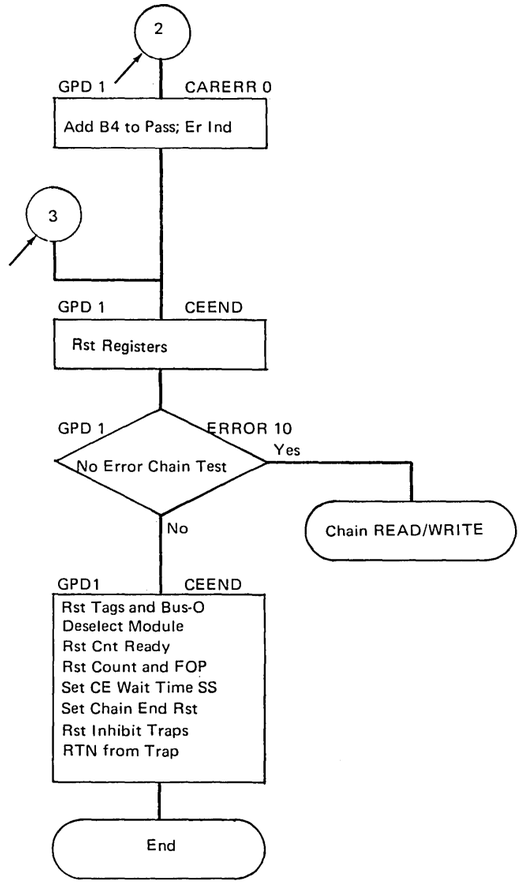




Cylinder Address Register (CAR) Test (7)

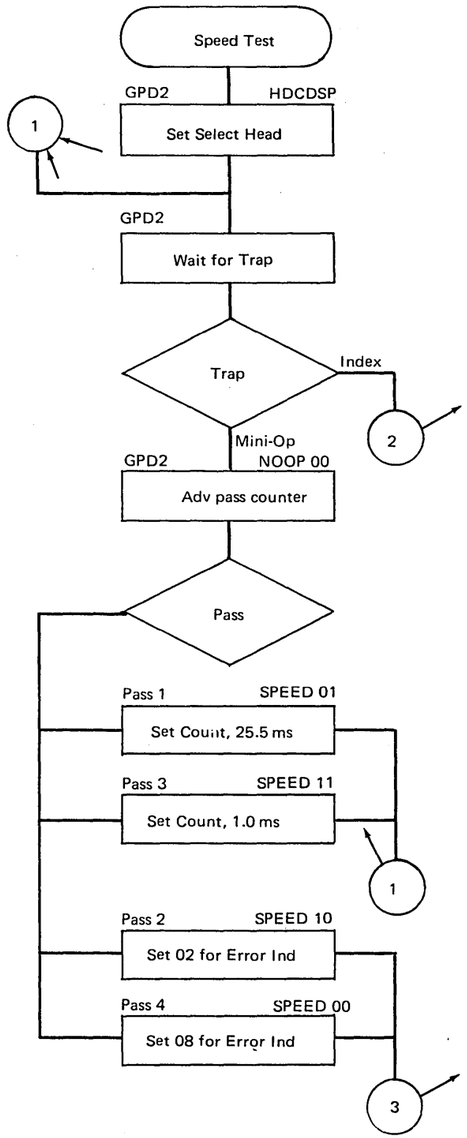
The CAR test checks the ability of the cylinder address register in the disk file to set and reset. The existing address in the register is read in and stored. The first test sets an address of 00 into the CAR and then reads the entry back for test. The second test sets an address of FF into the CAR and then reads back to test. The third test enters the cylinder select value set in the address switches into the CAR and then reads the entry back to compare with the switches. The final test returns the original value to the CAR and then reads the entry back to compare with the stored value.

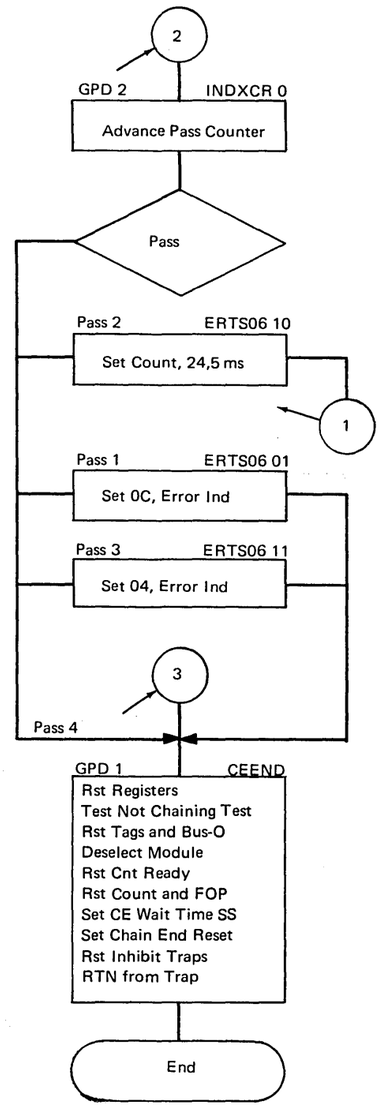




Disk Speed Diagnostic Test (3)

The disk speed test times the period between successive index pulses to determine if the speed is within tolerance. The specified tolerance for disk speed places the index pulses 24.5 to 25.5 milliseconds apart. After conditioning the head, a count equal to 25.5 milliseconds is set with the no-op mini-op. Unless the speed is very slow, the index trap should occur before the mini-op trap. The next test is made with a count for 24.5 milliseconds set. Unless the speed is high, the mini-op trap occurs before the index point is reached. A third test is made with a count for 1 millisecond. If the index trap occurs before the mini-op trap, the speed is within tolerance.





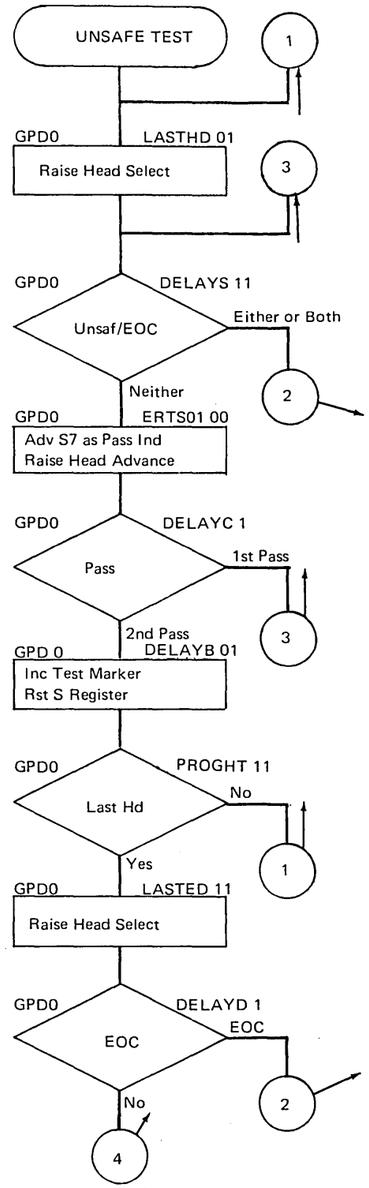
Diagnostic Unsafe-Condition Test (F)

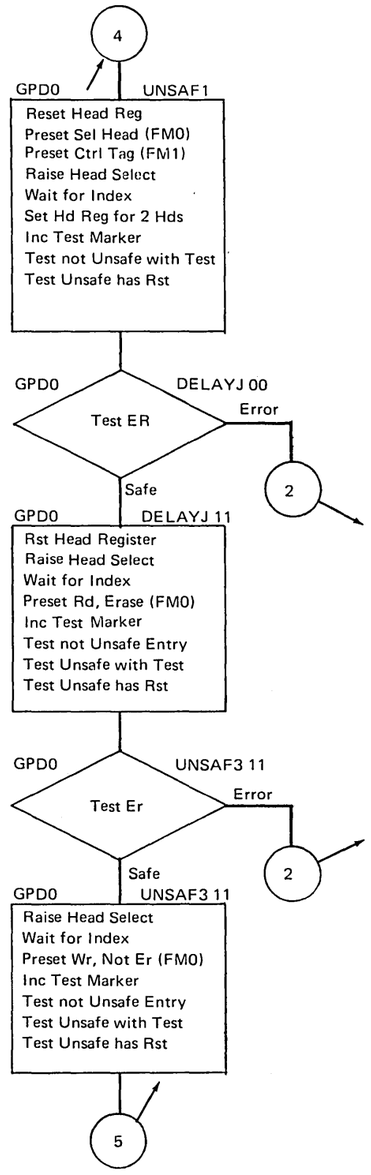
The unsafe-condition test provides a sequence of forced tests that cause the disk file unsafe indicators to set and reset. When an error is detected, the test operation is ended with the sequence number and the error condition in the indicators.

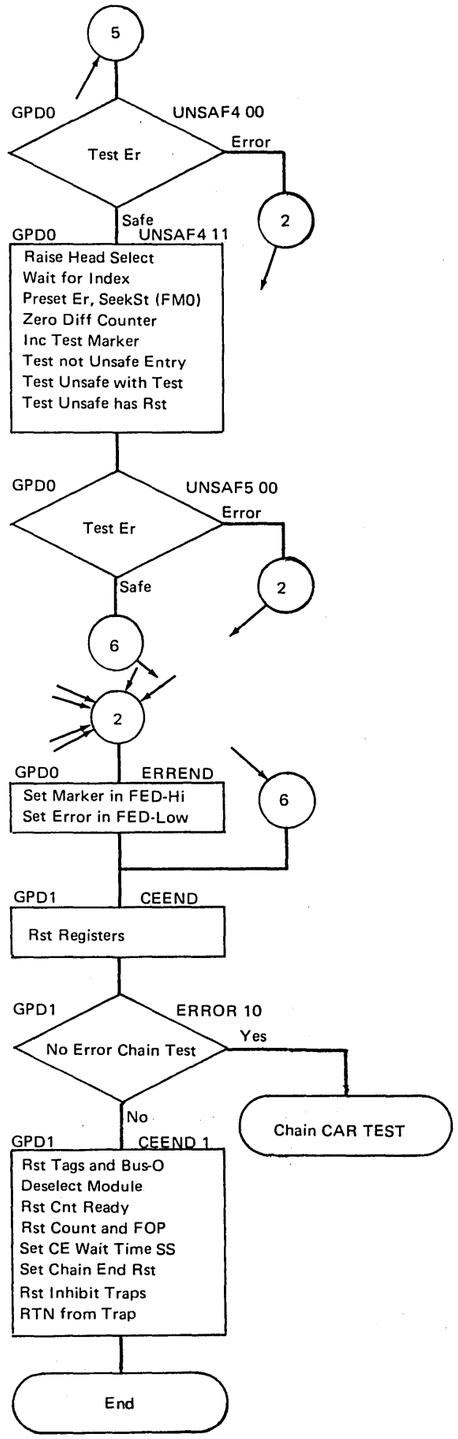
Alternate heads are conditioned in sequence with a test for unsafe and end-of-cylinder indications. Error conditions are reported if any drive lines are affected. This test posts error indications of 06 for unsafe, 07 for EOC, and 0A for both conditions.

Following head selection sequence, the head advance is forced again to determine that the end-of-cylinder indication operates. A failure to set the EOC indicator gives an error indication of hex-A1.

Subsequent tests force the unsafe indication with combinations of two heads selected, read and erase gates, write and not erase gates, and erase gate with seek start. A head is conditioned and the wait for index trap operation initiated before each test. Tests are made for the unsafe condition before the test, after the forced test, and after a forced reset. Hex-A through hex-E in the high-order four bits define the force unsafe test. Bits 4 through 6 of the error indication define the unsafe test found in error.







Read/Write Diagnostic Test

READ SINGLE TRACK TEST (0)

When performing the read test single track, the operation repeats using the head selection set in the address switches for the track. The head is selected and, after waiting for index, the home address field is read into control storage. A test is made to ensure that the track is not flagged defective before proceeding with the reading of record-0.

The count field is read into control storage to obtain the field length of the data field. The data field is read with the skip bit set because there is no storage area assigned. The test provides a check on the reading controls and the cyclic-check circuits. An error trap is initiated for any errors detected during the transfer. The following error indications are set in the trap sequence.

READ FULL CYLINDER TEST (4)

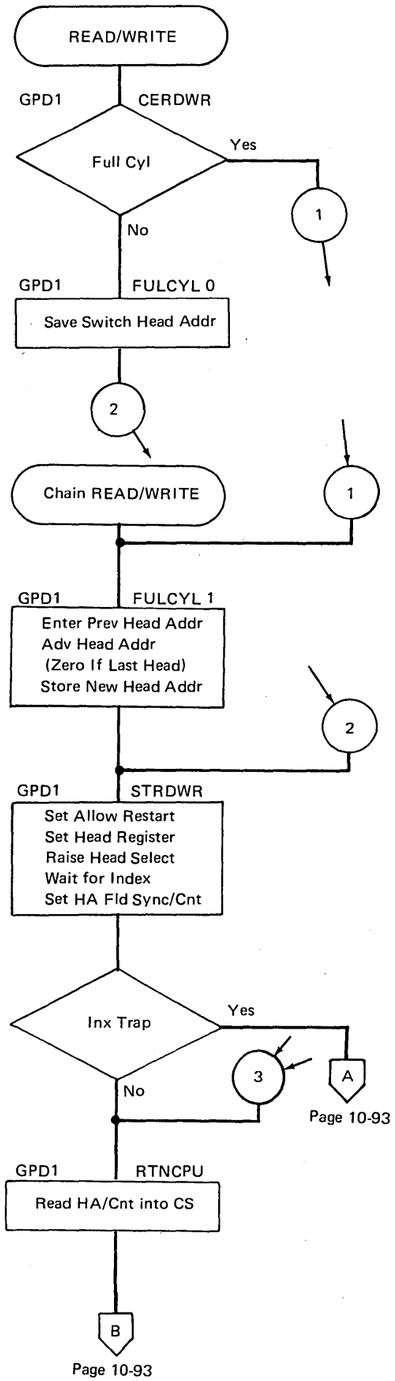
When performing the read test full track, the head address that was stored in the previous operation is used. The operation is the same as for the single track test except that before ending the stored address is advanced by plus-1. With each repeat of the test, a new head is used for each test. The head advance continues with each test until it reaches 20, then the stored value is set to 00. Thus, if the test is allowed to continue, it reads all of the tracks in the cylinder and then repeats.

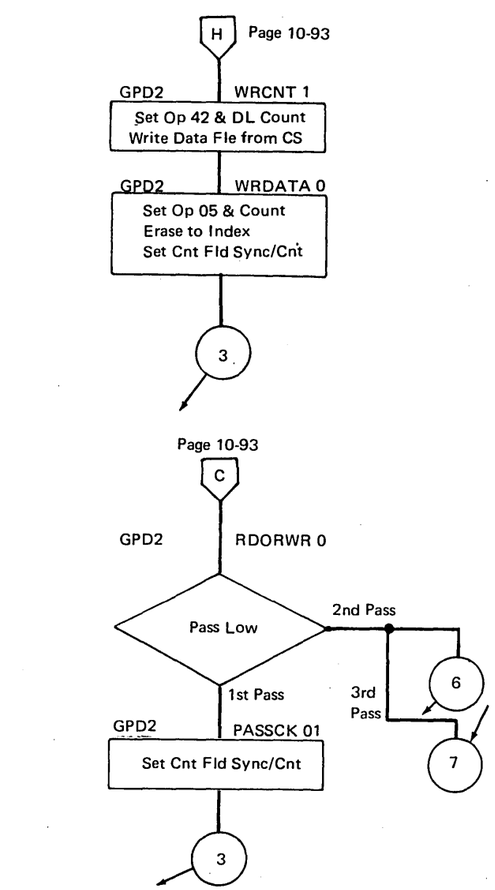
If trouble is detected in the full cylinder test with a specific head, the condition may be repeated using the single track test. An error trap is initiated for any errors detected during the transfer. The following error indications are set in the trap sequence.

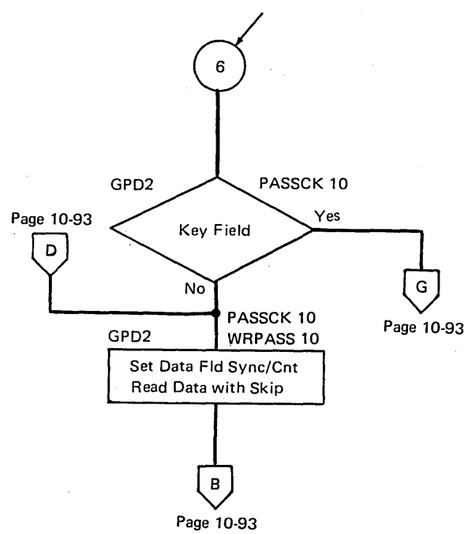
WRITE SINGLE TRACK TEST (2 OR A)

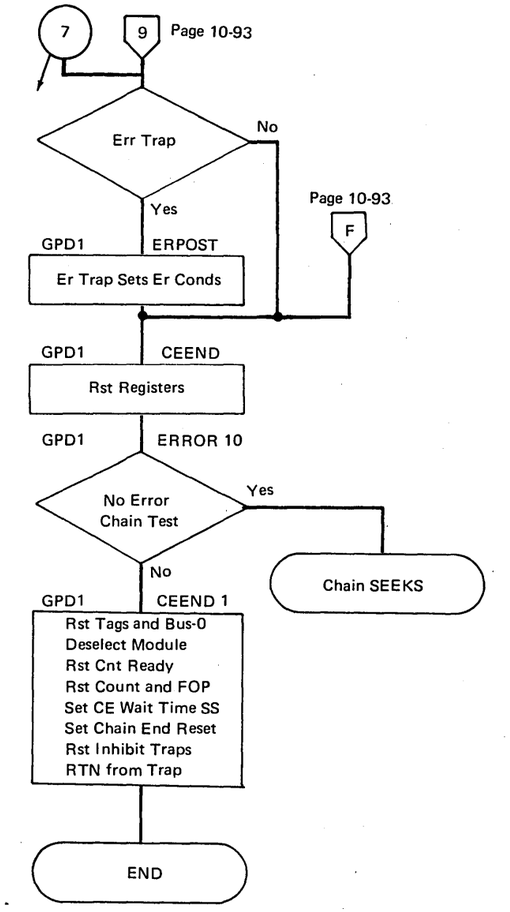
When performing the write test single track, the operation repeats using the head selection set in the address switches for the track. The head is selected and, after waiting for index, the home address field for the track is read into control storage. A test is made to ensure that the track is not flagged defective or alternate before proceeding with the writing.

The record-0 count field gap is written followed by the count field information set into control storage. This information contains the cylinder and head addresses from the home address followed with record-0 and the length information. A data length count of 6144 was entered to write nearly a full track. After writing a data field gap, the operation continues by writing the data field from control storage starting at the low order and continuing for 6144 bytes. The remainder of the track is erased to index as a format write sequence.







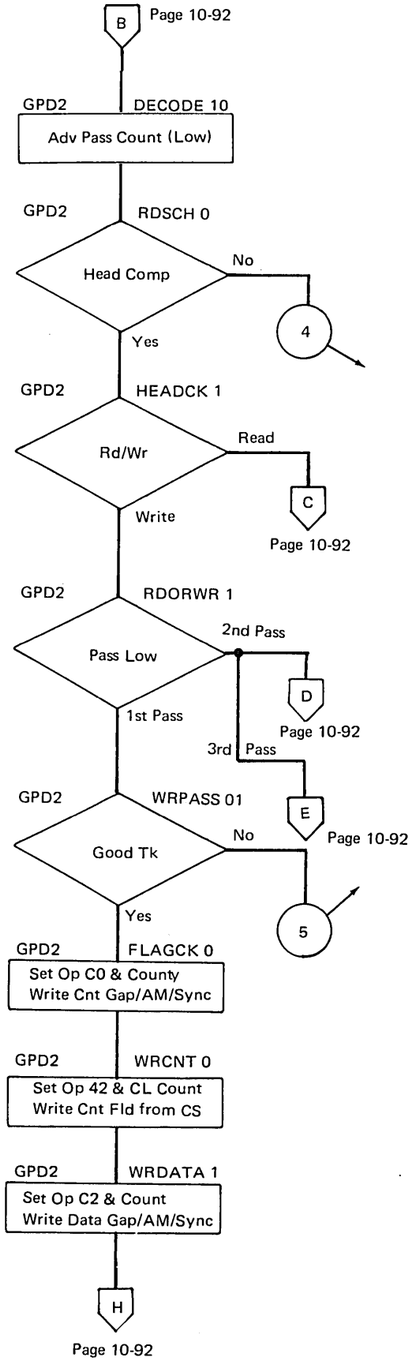


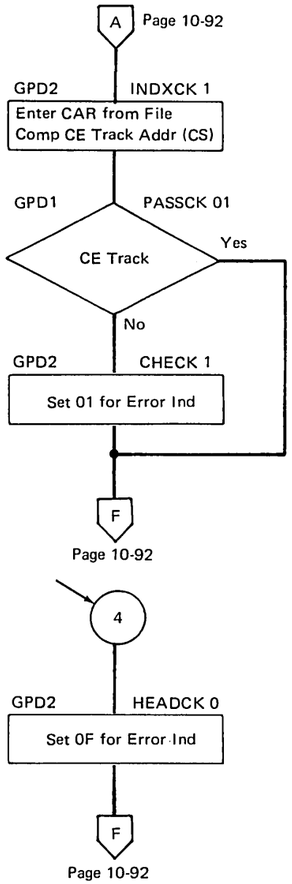
To prove the writing operation, a read operation follows covering the home address, count field, and the data field. The information of the data field is only tested by the cyclic-check in the same manner as for the normal read test. An error trap is initiated for any errors detected during the transfer. The following error indications are set in the trap sequence.

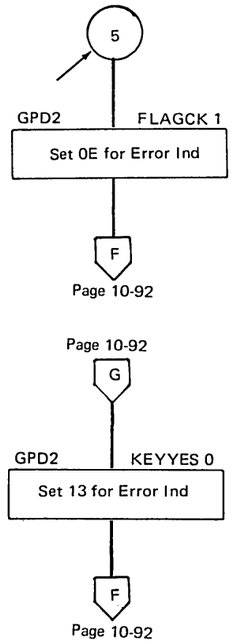
WRITE FULL CYLINDER TEST (6 OR E)

When performing the write test full cylinder, the head address that was stored in the previous operation is used. The operation is the same as for single track except that before ending, the stored address is advanced by plus-1. With each repeat of the test, a new head is used for each test. The head advance continues with each test until it reaches 20, then the stored value is set to 00. Thus, if the test is allowed to continue, it reads all of the tracks in the cylinder and then repeats.

If trouble is detected in the full cylinder test with a specific head, the condition may be repeated using the single track test. An error trap is initiated for any errors detected during the transfer. The following error indications are set in the trap sequence.







Seek Diagnostic Tests

RESTORE SEEK TEST (1 OR 5)

The restore seek test provides a means of returning the access mechanism to cylinder 000 followed by a seek to the cylinder address set in the address switches. The test provides the means of moving the access arm into position for subsequent read or write tests. A test is made after each movement by reading the home address of the aligned track and comparing with the expected address.

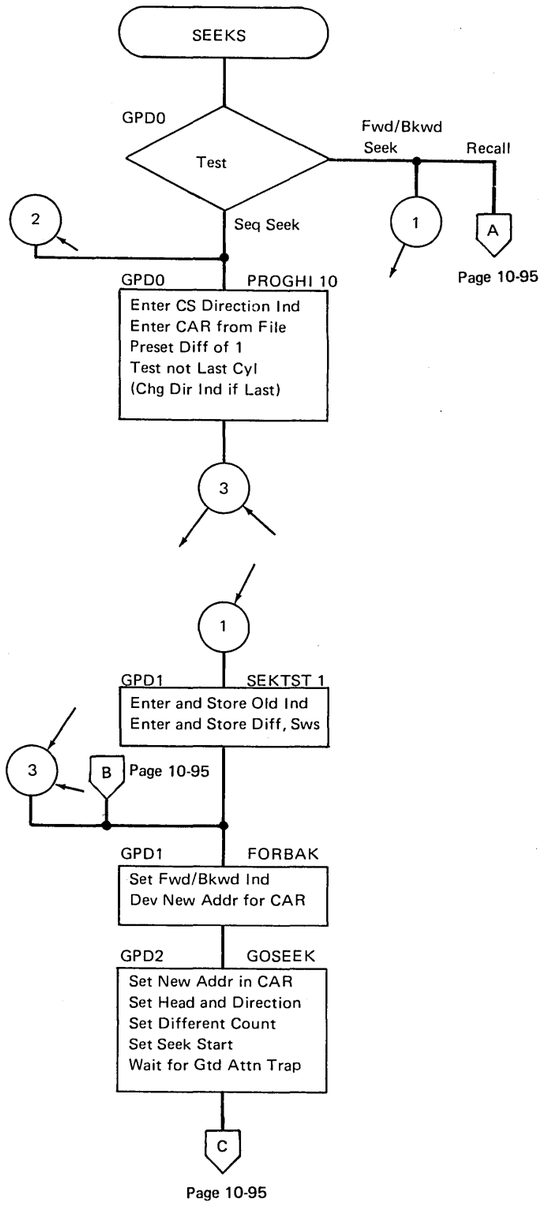
FORWARD SEEK TEST (9)

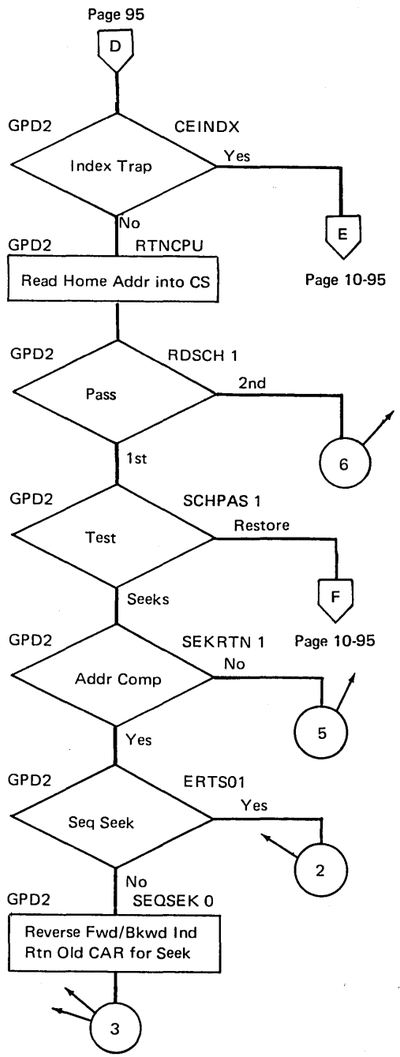
BACKWARD SEEK TEST (D)

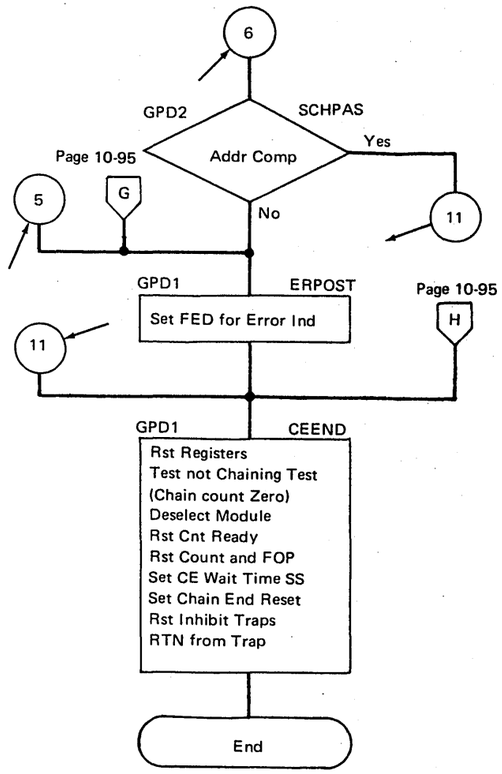
Two seek tests are provided to allow performing the operations at the extremes of addressing. The first test (9) executes a plus difference seek and follows it with an equal minus difference seek. The second test (D) executes the minus seek first and follows with the plus seek. After each seek, the home address for the aligned tracked is read and compared expected address. For these tests the address switches are used for cylinder difference. The value entered is used for both plus and minus movements. Care must be taken never to set a difference greater than the number of cylinders remaining in that direction. If the value is greater, the access will go to a crash stop.

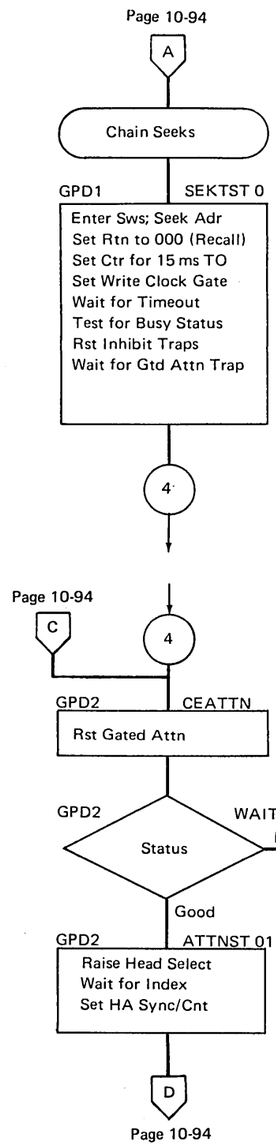
SEQUENTIAL SEEK TEST (B)

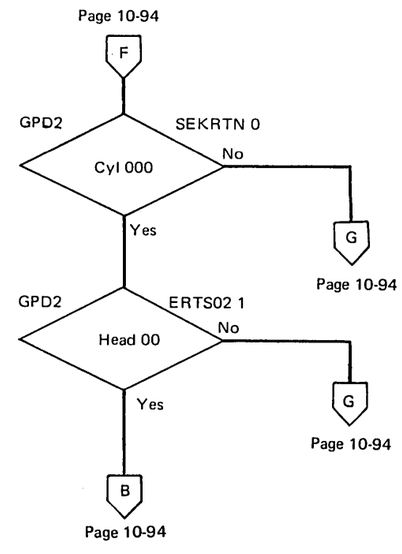
The sequential seek test causes the access to advance one track at a time. One test cycle advances the access twice if no error occurs on the first advance. The direction of advance is controlled by a direction indicator stored at FFF0 bit-0. The sequential seeks continue in the direction of the indicator until cyl-000 or cyl-255 is reached and then the indicator is reversed to continue the test. After each seek the home address is read to determine if the seek reached the computed cylinder number. The address switches are not used for the sequential seek test. The one difference is set by the microroutine and the CAR register is updated for each operation.

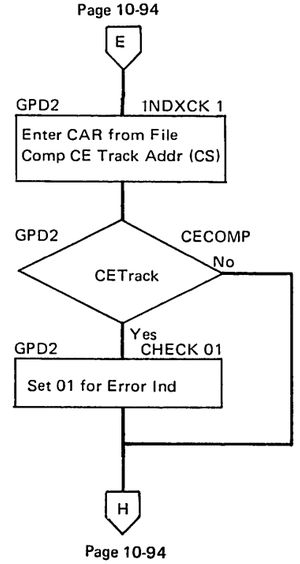







Page 10-94
A
Chain Seeks
GPD1
SEKTST 0
Enter Sws; Seek Adr
Set Rtn to 000 (Recall)
Set Ctr for 15 ms TO
Set Write Clock Gate
Wait for Timeout
Test for Busy Status
Rst Inhibit Traps
Wait for Gtd Attn Trap
4
Page 10-94
C
4
GPD2
CEATTN
Rst Gated Attn
GPD2
WAIT2 1
Status
Defect
Good
G
Page 10-94
GPD2
ATTNST 01
Raise Head Select
Wait for Index
Set HA Sync/Cnt
D
Page 10-94


Page 10-94
F
GPD2
SEKRTN 0
Cyl 000
No
Yes
G
Page 10-94
GPD2
ERTS02 1
Head 00
No
Yes
G
Page 10-94
B
Page 10-94


Page 10-94
E
GPD2
INDXCK 1
Enter CAR from File
Comp CE Track Addr (CS)
GPD2
CECOMP
CETrack
No
Yes
GPD2
CHECK 01
Set 01 for Error Ind
H
Page 10-94

Error Trap Sequence

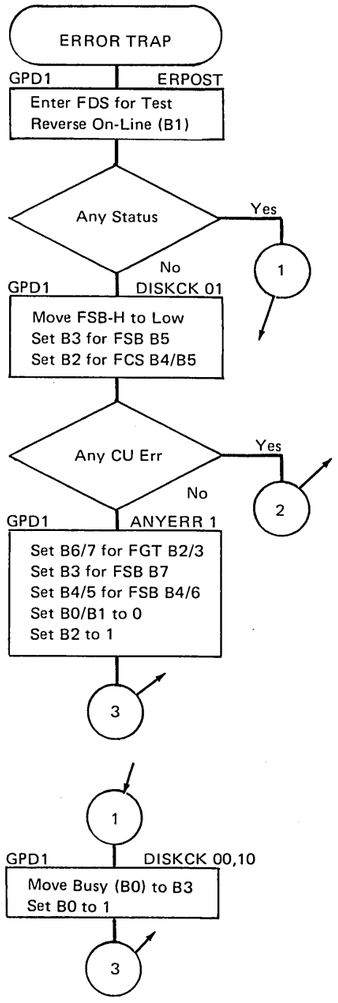
When either a read or write test is performed, the normal error conditions are set and result in setting the error trap at count-20. This condition can occur following any field processed. The normal error trap entry is branched to the inline diagnostic routine GPD1.

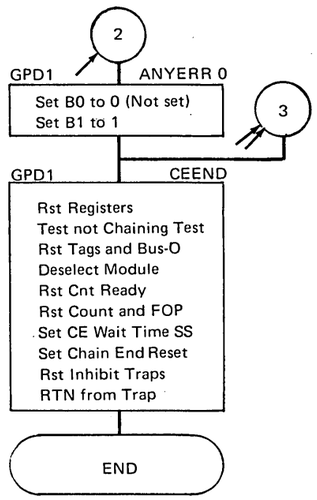
The routine first tests the FDS external for any disk file status conditions. If any disk status condition exists, the FDS information is set into the FED register to display the error. The busy-bit is moved to B3 and Bit-0 is set to 1 to indicate disk status.

If no disk status is posted, the control unit status conditions in external FSB and FCS are tested. If any control unit status exists, the assembled information is set into the FED register to display error. Bit-0 is set to 0 and bit-1 is set to 1 to indicate control unit status.

When neither disk status nor control unit status conditions exist, the routine tests read/write status conditions exist, the routine tests read/write status conditions in externals FGT and FSB. It is assumed that if no other status was set, a read/write condition exists. The assembled information is set into the FED register to display the error. Bits-0 and -1 are set to 0 and bit-2 is set to 1 to indicate the read/write status.

After posting the status, the routine ends the test with the normal CE resets and the timeout. The setting of the error switch determines if the test sequence is halted.





IFA POWER SUPPLIES and INTERFACE

POWER SUPPLIES

All power for the IFA and the associated disk drives is supplied by the 3145. No additional line connections are used.

The IFA logic on the A-gate is supplied by the CPU power supplies for that gate. Power originates from the motor-generator set at 208 V, 400 Hz, 3 phase AC. CB-2 and CP-101 control the ac power to TR-101 that reduces the voltage and rectifies it to dc. This dc output is then regulated by Reg-101 to produce the +1.25 V and -3.0 V lines to the IFA area (A-B1) of the A-gate. These voltages are set for their CPU requirements.

The dc voltages for the 2319-A01 and associated disk drives are supplied by five supplies in the 3145 power frame and one supply in the CPU frame. These supplies are designated as follows:

Volts	Supply
-3.0	TR-1 Reg-1 from 400 Hz.
+3.0	TR-1 Reg-2 from 400 Hz.
-36	TR-1 Reg-3 from 400 Hz.
+6.0	TR-1 Reg-5 from 400 Hz.
+36	TR-2 from 400 Hz.
+12	TR-108 from 50/60 Hz. (CPU)

The four regulators for the first four voltages are located under a cover at the upper rear of the 3145 power frame. The input power is controlled by CB-5 and CP-1 from the 400 Hz supply. They are transformed and rectified by TR-1 located to the left of the regulator cabinet. These are standard Mid-Pack regulators that have transistor voltage regulation and a circuit breaker over current protection. Their outputs feed the mixer board to the right of the supplies to feed the DB-jacks for the disk drives. These supplies are used only for the IFA facility.

A +36 volt supply located to the left of the regulator cabinet also goes to the mixer board to feed the IFA connectors. The +36 V supply is not adjustable from its nominal design setting.

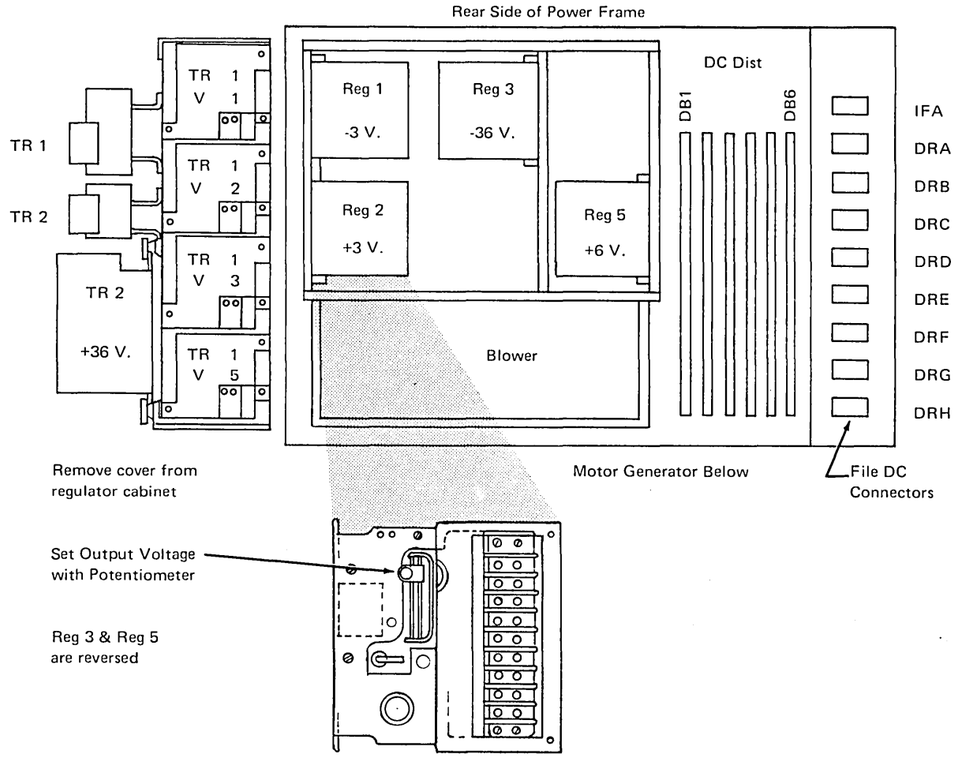
The +12 volt supply that feeds the IFA jack to the 2319-A01 mixer board comes from TR-108. This supply is located on the left side of the CPU frame behind the console and feeds the CPU mixer board. Initial regulation of this supply is provided by a ferroresonant transformer. Final regulation is by an amplifier card on a 16 V output. The TR-108 supply also feeds the 7.25 ac voltage to the 2319-A01 CE panel through the IFA jack. The voltage levels of the TR-108 are set for CPU requirements. The 50/60 Hz input power is controlled by CB-4, CB-3, K27, and F102.

During the power-ON sequence, all of the IFA dc power supplies are turned ON during sequence 1. The 3 phase ac power is also applied to the disk drives through the J63 jack on the end of the power cable. During the I/O start sequence the 'sequence pick' line is raised to allow the disk drives start in sequence through their own controls.

During a power-OFF sequence, the 'controlled ground' line opening causes the disk files to unload their heads and remove the access mechanism from between the disks. Until the access arm is in its home position the 'heads extended' line remains UP. The CPU power-OFF sequence can not remove power from the disk drives until all units have dropped the 'heads extended' line remains UP.

Adjustments

The four dc Mid-Pack supplies for the disk files and the 2319-A01 are individually adjustable to the requirements of the IFA facility. Each regulator has a small knurled knob that controls the voltage set potentiometer on the SMS card. These voltages should be set by monitoring the voltage levels at the disk drive on TS-4 or TS-5. The proper operating level is specified in the 2319-A01 Installation Manual.



IFA DC Power Supplies

IBM 2319-A01 INTERFACE

The 2319-A01 must be physically butted to the left side of the 3145 power frame. Any additional disk drive boxes are placed side by side butted against the left side of the 2319-A01. All cabling between the power frame and the 2319-A01 and its associated disk drives is routed within the box area. Each of the cables is pre-connected at one end in the appropriate box and shipped with that unit. The other ends of these cables are plugged into an alotted areas in the other box.

These cables can be traced by use of CPU ALD pages WF102, WF111, WF121, and WF122. The 2319-A01 connections are shown on ALD WF002.

AC Cable

A 208 V., 50/60 Hz, 3 phase cable originates in the ac distribution of the 3145 power frame and terminates in the J63 jack. The J63 jack connects to the ac connector cable in the 2319-A01 where it feeds the TS-1 blocks at both sides of the box. A jack is provided at the left side of the box to plug the cable for the next box, if used. When multiple disk drive boxes are used, each box plugs into the one to its right and provides connection for the box to the left.

Simplex Cables

Each of the disk drives has a simplex cable that originates in either a TS-4 or a TS-5 block in the associated box. These cables plug into the assigned jack located at the upper left rear of the power frame and labeled DBA through BDH. Each cable is plugged into the jack designated by its position in the disk file subsystem. Each cable carries the dc power to the logic board of the respective disk drive. These cables also carry the R/W coax and the selected module line between the DB-jacks and the drive logic.

IFA Mixer Board DC Cable

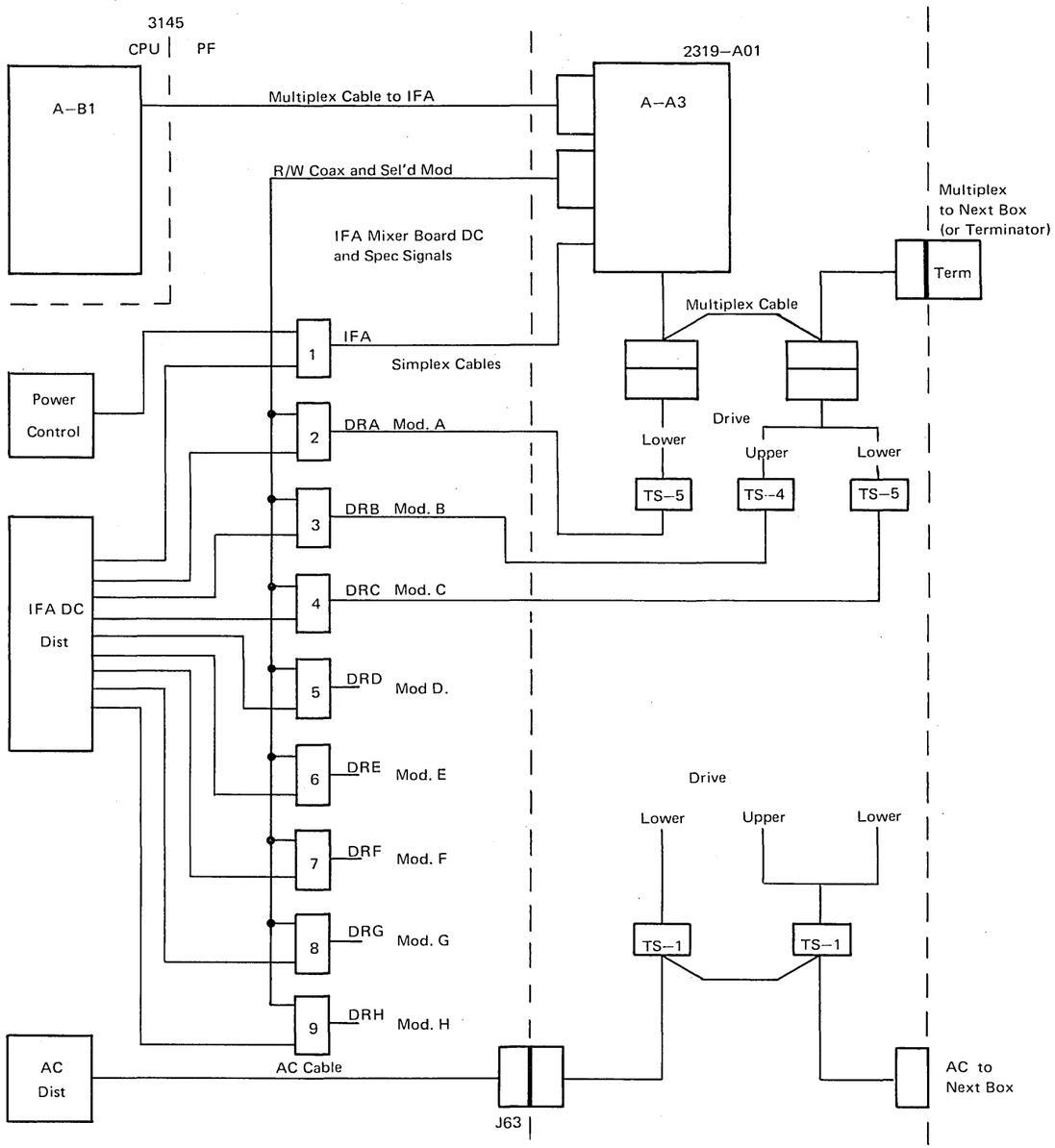
A cable originating at the 2319-A01 mixer board (A-A3) is plugged into the jack labeled IFA located above the DBA jack at the rear of the power frame. This cable carries the dc power to the mixer board and special signals used to control the power sequence.

R/W Coax and Selected Cable

This cable originates at the DB-jacks at the upper left end of the power frame and connect to the 2319-A01 mixer board (A-A3). It carries the R/W coax line and the selected module line from each DB-jack to the mixer board.

IFA Multiplex Cable

This cable originates in the IFA logic (CPU A-B1) and plugs into the 2319-A01 mixer board (A-A3). The cable carries the normal disk drive multiplex lines to the mixer board from which they are routed to the first drive and jumpered to the remaining drives. The IFA multiplex cable also carries the read and write data lines and the read (IFA) and write clock lines to the mixer board. These lines are all converted to SLT logic level of the 2319-A01. The multiplex lines to the drives must be terminated at the output of the last box.



IFA to 2319-A1 Interface

CHAPTER 11. POWER CONTROL AND DISTRIBUTION

REMEMBER

There is a Reader's Comment Form at the back of this publication.

GENERAL INFORMATION--STAGES 1 AND 2

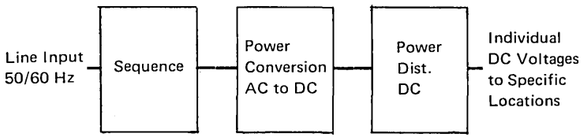
SAFETY

Safety cannot be overemphasized. To ensure your personal safety and the safety of co-workers, make it a practice to observe safety precautions at all times. You should be familiar with the general safety practices and procedures for performing artificial respiration outlined in IBM Form 229-1264.

Always use a reliable voltmeter to verify that power is actually off after using power-off switches.

Although all power supplies are provided with bleeder resistor to drain charges when power is turned off, check all capacitors with a meter before attempting maintenance. A defective bleeder resistor could create an unexpected hazard.

The Model 145 power system consists of three main parts: sequencing, power conversion, and distribution.



Input power for the Model 145 for domestic installation is 208/230V, ± 10%, 60 ± 0.5 Hz, 100 amperes maximum, via a three-phase, four-wire (fourth wire is equipment ground) shielded cable.

Input power for the Model 145 for World Trade installations, is 200V, 50 or 60 Hz; 220/380V, 235/408V, 50 Hz. The input voltage tolerance is ± 10%, and the frequency tolerance is ± 0.5 Hz. The input current is 100 amperes maximum. The input is provided by a three-phase, five-wire (fourth wire is neutral for a WYE system and not used on Delta, and the fifth wire is equipment ground) shielded cable.

Logic power is supplied by a motor generator (MG) set. The MG set converts 50/60 Hz, 3-phase ac input to 400 Hz, 3-phase power. High-current requirements for logic and storage are provided by phase-controlled rectifiers fed from the MG set. Other logic power is provided via MG set feeding 400 Hz. 3-phase transformer-rectifier assemblies, which in turn deliver power to series shunt regulators and series regulators.

Sequencing concerns the switches, relays, and contacts that control the input line voltage to the power conversion units. The sequencing unit also senses the outputs for overcurrent, and controls the input line voltage to protect the power system from damage.

Power conversion takes the input line voltage and converts it to a particular dc voltage at a particular amperage for distribution. The power conversion consists of two parts: first the input ac voltage is converted to some form of regulated dc voltage; second, the partially regulated dc voltage is converted to a particular dc voltage, and at a specific amperage.

The regulated dc from the dc (output of the power conversion) is distributed to the cards and boards through cables.

POWER SUPPLIES

The main types of regulators used with the Model 145 are:

- Middle-power package (Mid-Pac) regulators.
- Monolithic system technology (MST) regulators.
- Phase-control rectifiers.

Some regulators and associated components are used only with certain integrated I/O attachments. Where possible, these components are specified in feature groups and are installed only when the associated features are installed. A complete list of all power components and their function is contained on logic pages YE050 through YE055. Information concerning the regulators is in *Power Supplies, SLT, ASLT, MST,* Field Engineering Theory-Maintenance manual, SY22-2799. The power-supply manual does not cover the MG; therefore, a brief description of the MG used in the Model 145 is provided.

- Two kinds of power supplies are in the Model 145. The earliest is Stage 1; the latest is Stage 2. In the Stage 2 power package, seven MST-type dual supplies have replaced two phase-control dual regulators.

POWER WIRING DIAGRAMS and COMPONENT LABELING

The Model 145 power wiring diagrams are prefixed with YE and are assigned as follows.

- YE001–YE099 show references for component location charts, machine layout, timing charts, table of contents, etc.
- YE100–YE199 are the primary power and ac distribution diagrams.
- YE200–YE299 are the sequencing and control diagrams.
- YE300–YE399 are the regulators and dc distribution diagrams.

To identify power components and the frame at which they are located, the components within each frame are identified in this manner:

FRAME	COMPONENT ID
PF	001-099
CPU	101-199
MSF (if installed)	201-299
Power Unit (if installed)	xxx-xxx

The components (relays and contactors) are usually labeled in the order in which each becomes active during the power-on sequencing. For example, K1 would normally pick first, and K20 would be the twentieth relay to pick during the power-on sequence.

NOTE: Power to the logic boards within the CPU is now distributed by laminar buses, instead of flat wire buses, previously used in Stage 1 machines.

COOLING

Cooling in the system is achieved by blowers mounted below the T/Rs and logic gates, and above the phase-control regulators. The blowers are supplied from a 208V ac, 60 Hz source. The blowers in the 200/235/408V 50 Hz systems are powered through an auto transformer. Thermal sensors are installed above the logic gates, regulators, and T/Rs to detect an overtemperature condition. An overtemperature condition initiates a power-off sequence.

MAINTENANCE CONCEPTS

The Model 145 power system can detect failure in the power system and can protect itself and the CPU from failure that could cause permanent machine damage. Depending on the severity, malfunctions manifest themselves by either preventing the advance of the power-on sequence of by turning off the system and energizing an indicator to locate the malfunction.

Control and indicators help to isolate a malfunction accurately and rapidly. The indicators are located on the CE maintenance panel.

NOTE: Use ALDs YE and YD for Stage 1 power package. Use ALDs YB and YA for Stage 2 power package.

MOTOR GENERATOR, STAGES 1 AND 2

MOTOR GENERATOR

The brushless motor generator (MG) is the primary power source for the 3145 and is housed as a unit within the power frame in its own acoustic enclosure. The MG consists of four essential parts:

(1) drive motor
(2) 400 Hz generator
(3) exciter
(4) MG voltage regulator

The motor, 400 Hz generator, and exciter are contained in a single housing with their respective rotors mounted on one shaft that is supported by a ball bearing at each end. The shaft rotates from power derived from the drive motor. The MG voltage regulator, which is mounted remotely within the enclosure so that it is readily accessible, is electrically connected between the exciter field and the generator output.

Drive Motor

The drive motor is a two-pole, low-slip, induction motor designed either for 50 Hz or 60 Hz. The motor can operate from a range of input voltages through proper field-winding interconnections shown on YE170. When the motor is started from standstill, the initial inrush current may reach a value equal to six times the normal full load value. This surge can last for about two seconds but diminishes to the operating current of about 20-60 amperes when the motor reaches full speed. The generator load is always removed until the motor comes up to speed and also before the motor is shut off. Failure to observe this procedure may have an adverse effect on the amount of residual magnetism in the exciter field. Successful generator buildup depends entirely on the presence of some residual magnetism in the exciter field core. Loss of residual magnetism may require flashing the exciter field with a dc source to enable a subsequent generator buildup.

400 Hz Generator

The 395-415 Hz generators (50 Hz and 60 Hz inputs) are basically alike, having 14 poles on the rotating dc field of the generator for the 60 Hz units, and 16 poles for the 50 Hz units. The generator is designed for 3-phase output at 208V rms nominal. The generator armature (stator winding) has three coils in a WYE configuration with the neutral (or common) lead brought out to the terminal board TB2-4 (see logic, page YE170). The voltage from any phase to neutral is equal to 120V rms nominal. The generator output is controlled so that the output (if set at 208V) remains at 208V $\pm$ 1% under a variety of input power and load conditions. Control is accomplished through the exciter by the MG voltage regulator.

AC Exciter

The exciter is a low-power ac generator that supplies the varying dc field current required by the generator. The armature (rotor) of the exciter generates about 20V, 3-phase, 60 Hz ac in a WYE configuration. This voltage is half-wave, rectified by three diodes mounted on the rotating shaft to supply a varying pulsating dc current to the generator field. The degree of variation is controlled by the exciter's dc field strength (stator); the current to this field is supplied and controlled by a MG voltage regulator.

Motor Generator Regulator

A MG voltage regulator supplies and controls the dc to the exciter field. Initial voltage setting can by $\pm$ 10% of the rated generator output voltage with regulation maintained at $\pm$ 1% of the set value. The most common setting is 208V. The basic regulation action is divided into three steps; (1) sample the output voltage, (2) compare the sample voltage with a fixed reference and determine the magnitude and direction of error, and (3) cause a current-controlling device to change the exciter field in a direction so as to reduce the error to zero.

More than one type of regulator (see YE180) is available; however, the principal function of each is the same. The regulator depends upon the type of MG. A simplified drawing of the MG regulator is also shown on YE180. Briefly, the MG regulator operates as follows.

a. The stepdown rectifier samples the voltages at the generator armature and establishes an rms value.
b. The voltage divider network establishes a fixed value (over the Zener diode), and a voltage representing the rms value. The fixed value can be varied by the voltage-adjust potentiometer. The fixed voltage and the voltage representing the rms value are compared by the voltage compare circuit.
c. The voltage from the compare circuit is amplified to drive a power transistor.
d. The output from the power transistor varies the current through the field exciter. This field affects the indicator voltage of the generator.

The generator should build up to its rated voltage without need for any external power source provided that sufficient magnetism is present in the exciter field.

Overvoltage

The MG regulator is also provided with an overvoltage circuit that monitors the generator output voltage. Detection of an overvoltage causes the generator voltage to decrease to a safe level and provides an indication to the overvoltage detect circuit. MG regulator overvoltage conditions cause a power-down situation and are indicated by an MG check and a power check. (MG overvoltage is detected by the action of relay K60 (YE180) and its contents (YE072) error to zero.) The MG regulator contains a circuit breaker (CB1) that protects against damage due to an overcurrent condition within the regulator.

Motor Generator Enclosure

The MG enclosure not only reduces generator noise but also provides a calculated path for airflow to prevent excessive heat buildup within the MG set. The enclosure also provides some radio interference attenuation besides keeping the MG in a relatively clean environment and protecting it from the hazards of exposed power connections.

ELECTRICAL COMPONENT LOCATIONS

ELECTRICAL COMPONENTS--POWER FRAME, STAGE 1

Fuse	Function	Fuse	Function
1	Conv outlet protect	8	CF protect
2	Conv outlet protect	9	IFA/3210 +12V meter protect
3	24V Ctrl Tfrm protect	10	12V Seq protect
4	24V Ctrl Tfm protect	11	7.25V Ind lights protect
5	Conv outlet protect	12	Phase-rotation circuit protection
6	Conv outlet protect	13	3215 Input protect
7	W/T 3215 motor protect	14	3215 Input protect

(See "Primary Power Box", page 11-7 for internal functional units.)

(Front View)

PF/CPU Interface Connector

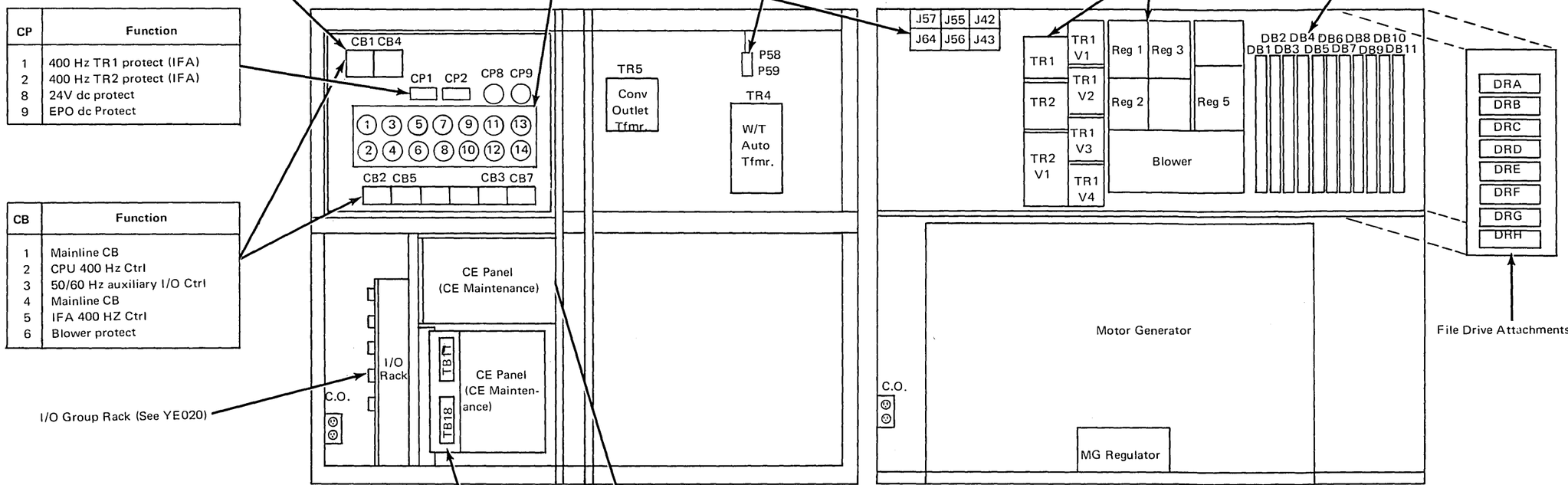
((Rear View)

Rectifier assemblies and Mid-Pac rectifiers for the IFA feature. The circuit protectors that protect the regulators against OV/OC are contained on the Mid-Pacs.

FA DC Distribution

CP	Function
1	400 Hz TR1 protect (IFA)
2	400 Hz TR2 protect (IFA)
8	24V dc protect
9	EPO dc Protect

CB	Function
1	Mainline CB
2	CPU 400 Hz Ctrl
3	50/60 Hz auxiliary I/O Ctrl
4	Mainline CB
5	IFA 400 HZ Ctrl
6	Blower protect

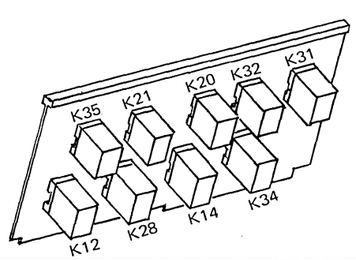


File Drive Attachments (See YE053)

I/O Group Rack (See YE020)

TB	Function
17	Voltage Detection Board
18	Voltage Detection Board

Power Sequence Relay Board (Located on back of sequence panel)



Relay	Function
K12	Power Check
K14	Power On
K20	Time Delay
K21	MG Hold
K28	Heads extended control
K31	Seq 1 detect
K32	Seq 2 detect
K34	Reg undervoltage control
K35	I/O power hold

ELECTRICAL COMPONENT LOCATIONS (Continued)

ELECTRICAL COMPONENTS--POWER FRAME, STAGE 2

Fuse	Function	Fuse	Function
1	Conv outlet protect	8	CF protect
2	Conv outlet protect	9	IFA/3210 +12V meter protect
3	24V Ctrl Tfrm protect	10	12V Seq protect
4	24V Ctrl Tfm protect	11	7.25V Ind lights protect
5	Conv outlet protect		
6	Conv outlet protect	13	3215 Input protect
7	W/T 3219 motor protect	14	3215 Input protect

(See "Primary Power Box", page 11-7 for internal functional units.)

CP	Function
1	400 Hz TR1 protect (IFA)
2	400 Hz TR2 protect (IFA)
8	24V dc protect
9	EPO dc Protect

CB	Function
1	Mainline CB
2	CPU 400 Hz Ctrl
3	50/60 Hz auxiliary I/O Ctrl
4	Mainline CB
5	IFA 400 HZ Ctrl
6	Blower protect

(Front View)

CB1 CB4

CP1 CP2 CP8 CP9

1 3 5 7 9 11 13
2 4 6 8 10 12 14

CB2 CB5 CB3 CB7

TR5 Conv Outlet Tfmr.

P58 P59

TR4 W/T Auto Tfmr.

CE Panel (CE Maintenance)

I/O Rack

TB11

TB18

CE Panel (CE Maintenance)

C.O.

I/O Group Rack (See YE020)

PF/CPU Interface Connector

((Rear View)

J66 J42 J55 J57
J68 J43 J56 J64

Rectifier assemblies and Mid-Pac rectifiers for the IFA feature. The circuit protectors that protect the regulators against OV/OC are contained on the Mid-Pacs.

TR1 TR1 V1
TR2 TR1 V2
TR2 V1 TR1 V3
TR1 V4

Reg 1 Reg 3
Reg 2 Reg 5
Blower

FA DC Distribution

DB2 DB4 DB6 DB8 DB10
DB1 DB3 DB5 DB7 DB9 DB11

DRA DRB DRC DRD DRE DRF DRG DRH

File Drive Attachments (See YB311-312)

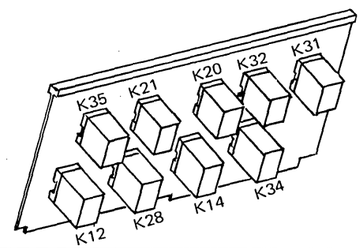
Motor Generator

C.O.

MG Regulator

TB	Function
17	Voltage Detection Board
18	Voltage Detection Board

Power Sequence Relay Board (Located on back of sequence panel)



Relay	Function
K12	Power Check
K14	Power On
K20	Time Delay
K21	MG Hold
K28	Heads extended control
K31	Seq 1 detect
K32	Seq 2 detect
K34	Reg undervoltage control
K35	I/O power hold

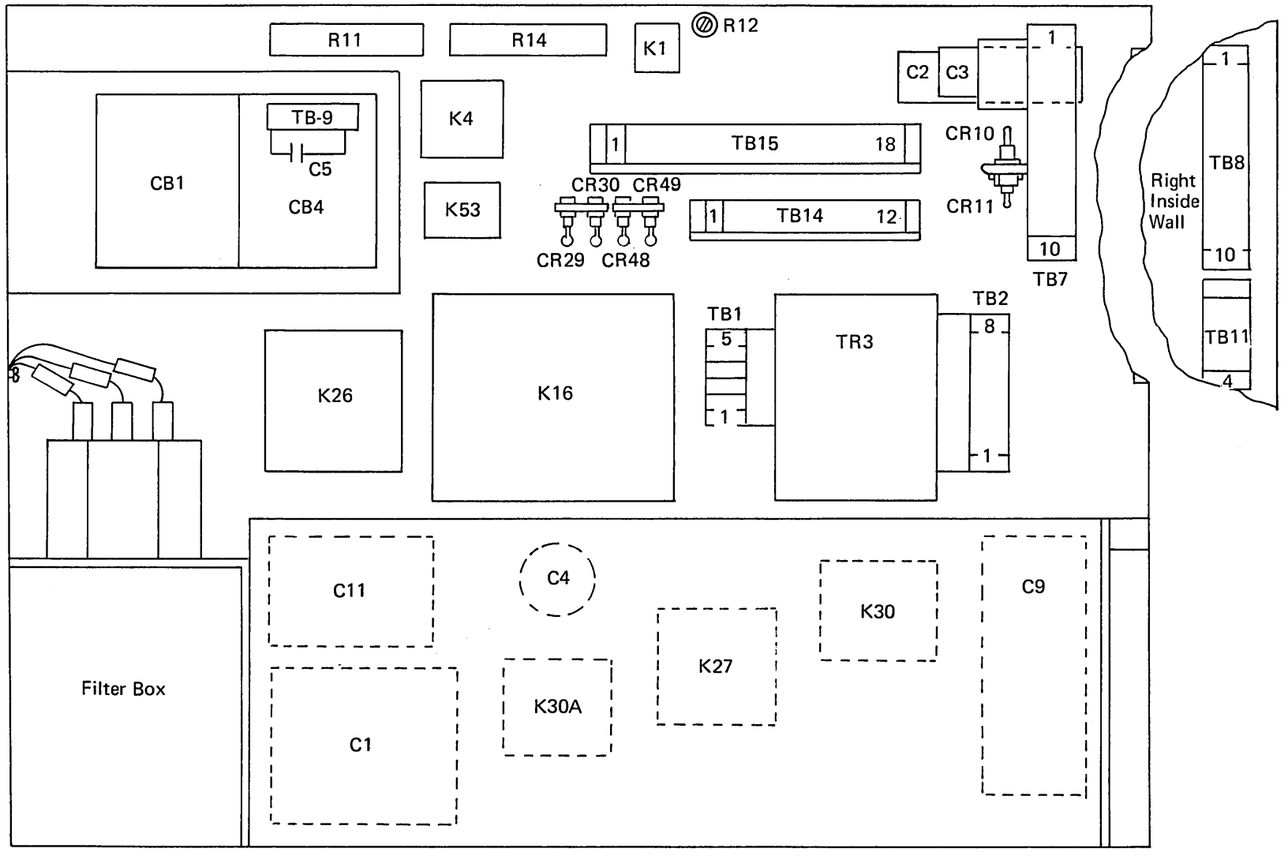
PRIMARY POWER BOX--INTERNAL COMPONENTS, Stages 1 and 2

RELAY	FUNCTION
K1	Phase Rotation (MG)
K2	Phase Rotation (3046 MG if installed)
K4	EPO Control
K16	MG Power Contactor
K26	400 Hz Contactor
K27	AC Contactor
*K29	60Hz 3210 Power Control
K30	Blower Ctrl Contactor
*K33	24V 2nd Seq. Control
K53	Console File AC

*Relays K29 and K33 are mounted on the inside of primary box door.

TB	FUNCTION
7	Conv Outlets Dist
8	Primary Pwr DC Dist.
11	Primary Pwr. AC Dist to PF
14	DC Control Voltage Dist.
15	Phase Rotation TB

CB	FUNCTION
1	Mainline CB
4	Mainline CB

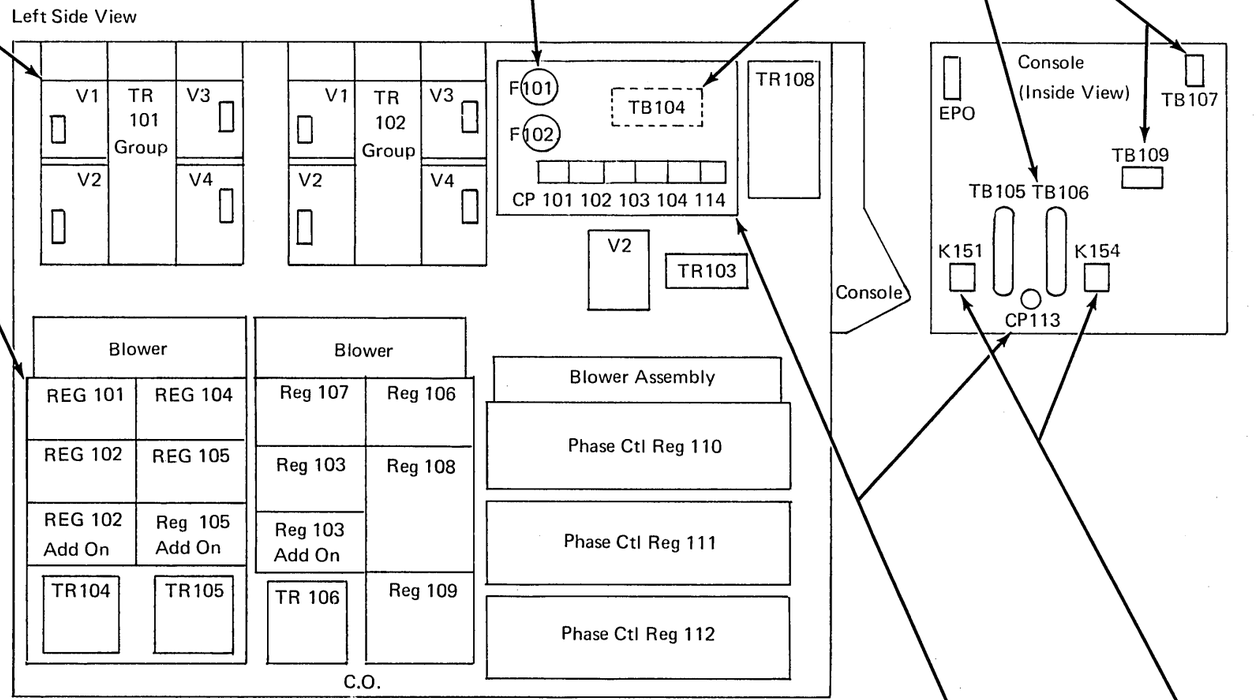


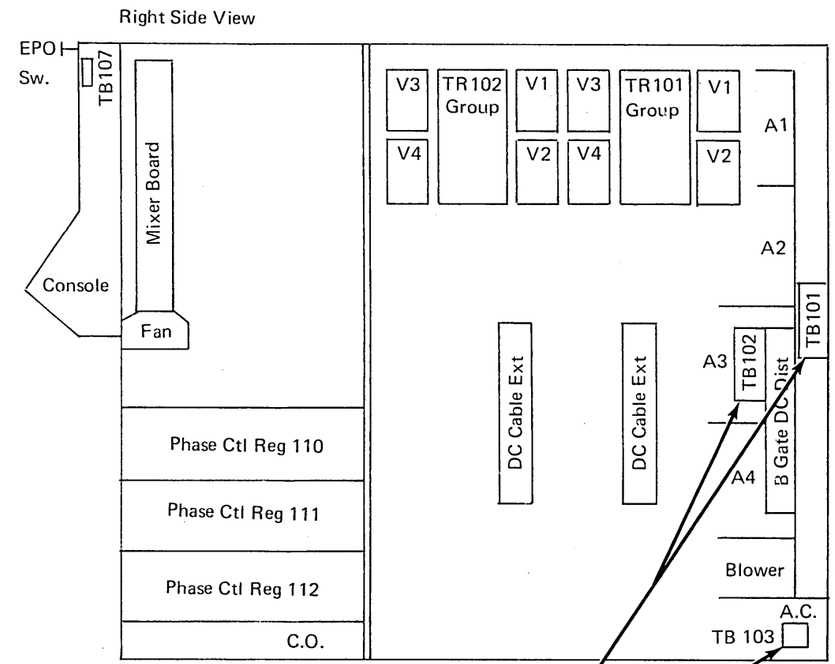
ELECTRICAL COMPONENTS--CPU FRAME, Stage 1

TR Pacs and Regulators: The TR Pacs and their associated regulators for the CPU logic gates. The circuit protectors that protect the regulators against OV/OC are contained on the TR Pacs. Forced air is used for cooling. All regulators are remote-sensed.

FUSE	FUNCTION
101	41V ac CE meter
102	TR108 protect

TB	FUNCTION
104	CPU 400 Hz distribution
105	Pwr reset to logic
106	24V dc return / IFA lamp test
107	7.25V ac 50 / 60 Hz
109	7.25V to Console





CP	FUNCTION
101	TR101 protect
102	TR102 protect
103	Phase control Reg protect
104	TR104,105,106 protect
113	-3V wire protect (CPU)
114	TR103 protect

RELAY	FUNCTION
K151	Power reset
K154	Console file dc

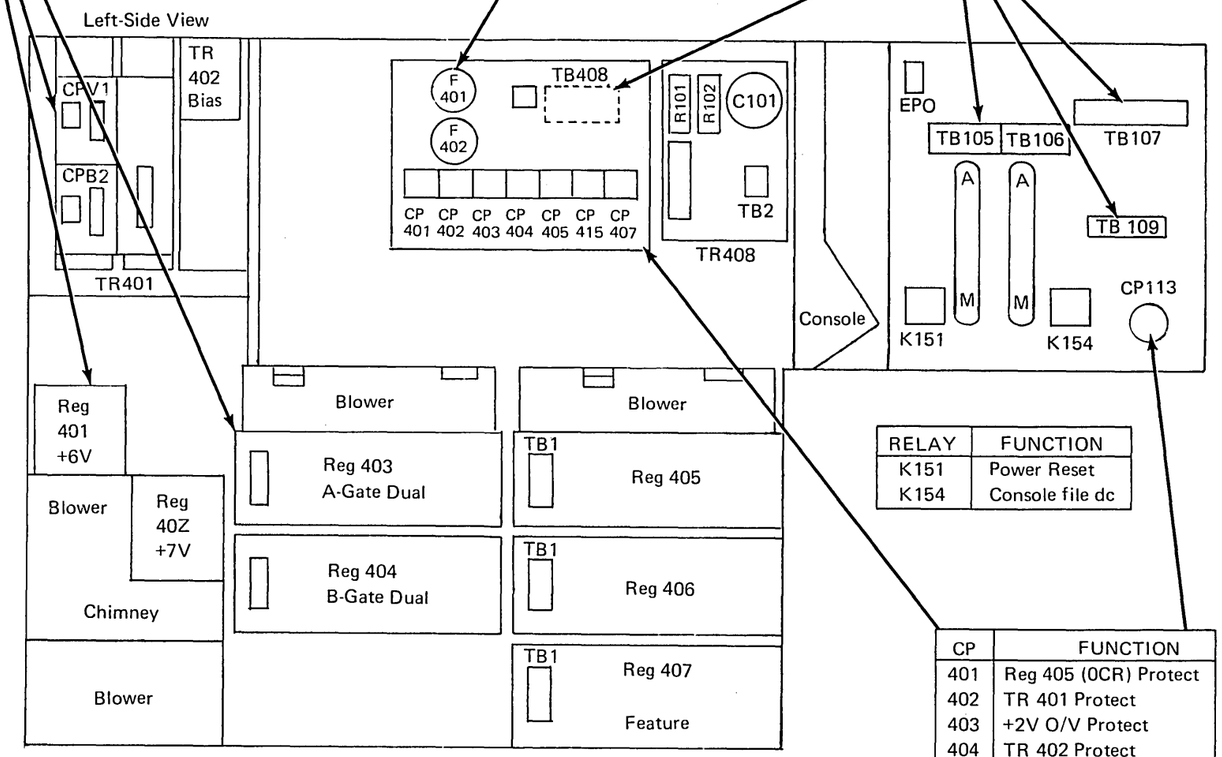
TB	FUNCTION
101	DC distribution
102	DC distribution
103	CPU 50/60 Hz

ELECTRICAL COMPONENTS--CPU FRAME, Stage 2

TR Pacs and Regulators: The TR Pacs and their associated regulators for the CPU logic gates. The circuit protectors that protect the regulators against OV/OC are contained on the TR Pacs. Forced air is used for cooling. All regulators are remote-sensed.

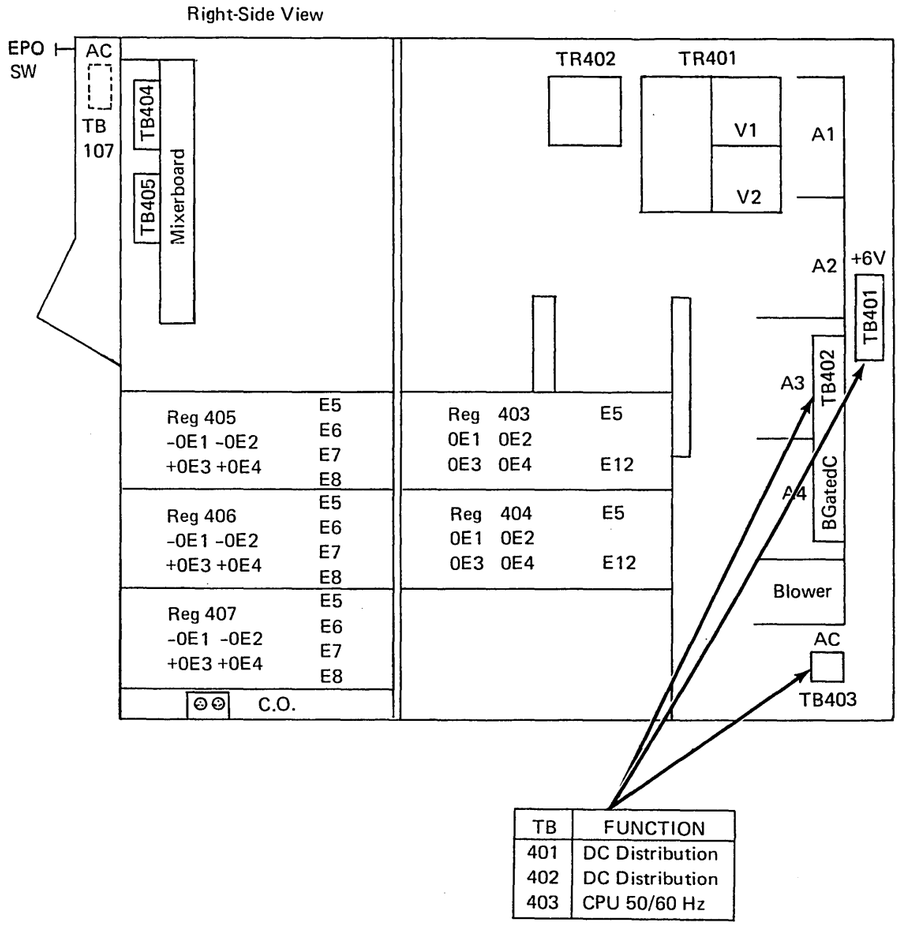
FUSE	FUNCTION
401	41V ac at CE Meter
402	TR408 Protect

TB	FUNCTION
408	CPU 400 Hz Distribution
105	Power reset to logic
106	24V dc Return IFA lamp
107	7.25V ac 50/60Hz
109	7.25V to console



RELAY	FUNCTION
K151	Power Reset
K154	Console file dc

CP	FUNCTION
401	Reg 405 (0CR) Protect
402	TR 401 Protect
403	+2V O/V Protect
404	TR 402 Protect
405	Reg 406 (0CR) Protect
406	Dual O/V Protect
407	Reg 407 (0/CR) Protect
414	Reg 403 (Dual) Protect
415	Reg 404 (Dual) Protect



TB	FUNCTION
401	DC Distribution
402	DC Distribution
403	CPU 50/60 Hz

POWER DISTRIBUTION, STAGE 1

The Model 145 power frame receives primary power input of 200/208/230V ac, 60 Hz or 200/220/235V ac volts, 380/408V ac, 50 Hz. Input distribution is three-phase and ground for 60 Hz; three-phase, neutral (for usage only), and ground for 50 Hz operation.

Power is routed from the customer ac service to a filter network (YE110). Each ac feed capacitor in the filter network has a resistor connected to ground. These resistors bleed off any charge remaining on the capacitors when the ac service is disconnected. Mainline circuit breakers for this system are CB1 and CB4.

After all circuit breakers are closed, initial turn-on brings:

1. AC voltage via fuses F3 and F4 to transformer TR3, which provides 24V dc to the power control circuit, 12V ac to the undervoltage detect circuits, and 7.25V to the sequence indicator lamps. The 24V dc is first routed to the system emerging power off (EPO) circuit via n/o contacts of K1 (and K3 if the 3046 is installed). If the EPO switch is closed and phase rotation to the MG is correct, relay K4 picks.
2. AC voltage to the convenience outlets. For 60 Hz systems, it is routed via fuses 1 and 2 to stepdown transformer TR5, which supplies 115V ac (for input voltage of 208V and 230V) to the convenience outlets. For 200V, 60 Hz systems, transform TR5 supplies 100V ac to the convenience outlets.

For Hz systems, the convenience outlet voltage is 220, 235, 380, or 408 (by Data-WYE) switching.

Further power distribution does not occur until one of the power switches is activated.

An auto transformer is provided for 200V 50 or 60 Hz; and 235/408 50 Hz inputs. Taps are provided on the transformer to obtain 220V output for the various input line voltages.

AC OUTPUTS

Convenience Outlets

Four duplex convenience outlets are provided with the system: CPU (2) and PF (2). For domestic machines, the outlets provide 115V ac, 60 Hz single-phase at a maximum of 15 amperes total for all outlets whose power is supplied by TR5. For World Trade, the outlets can provide either 100V ac, 60 Hz single-phase at a maximum of 15 amperes; or 200V, 220V, 235V, 50 Hz, at a maximum of 8 amperes. These currents are totals for all outlets.

Blowers

The system is cooled by blowers that require: 208/230V 60 Hz; or 220V, 50 Hz power.

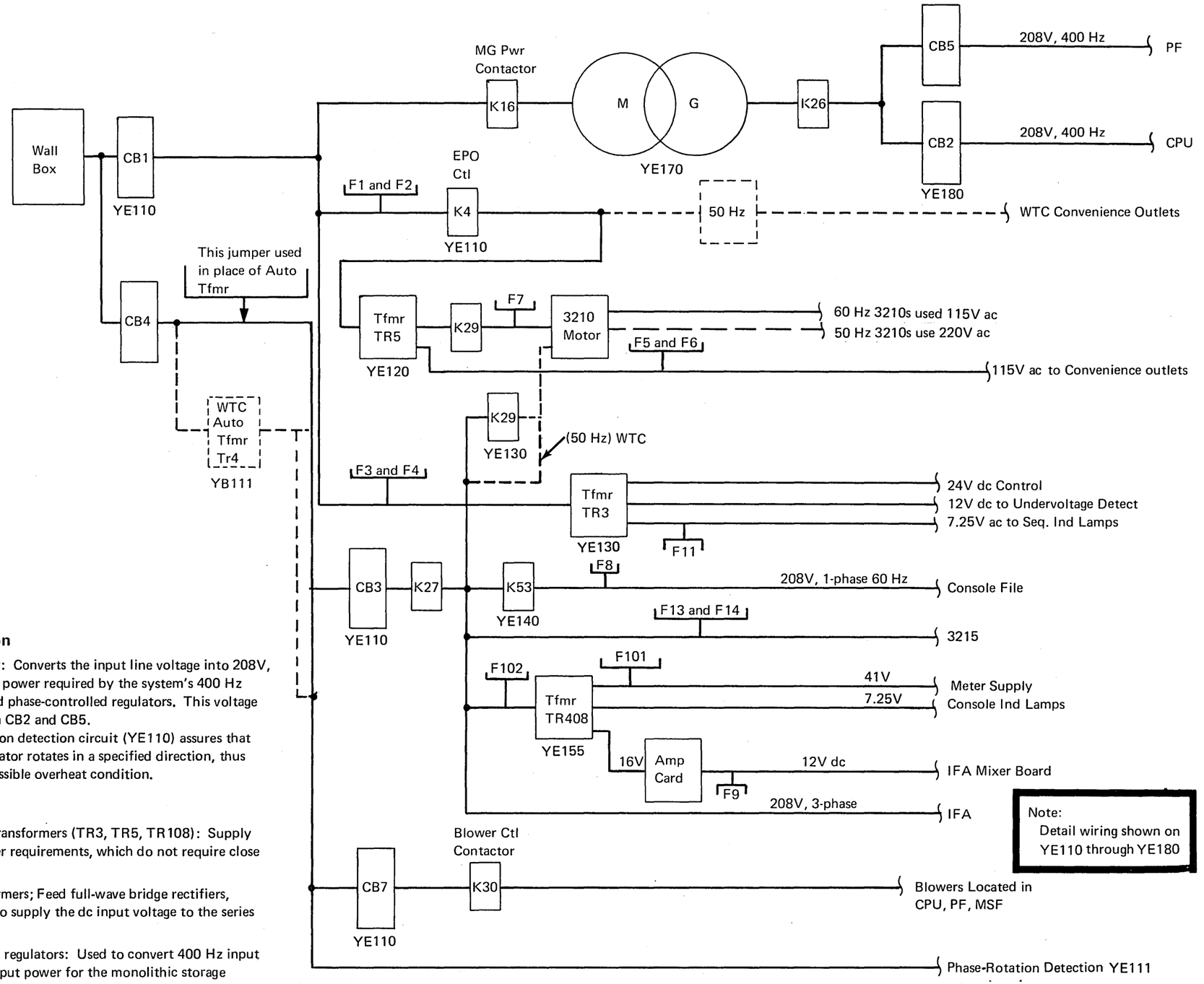
- The CPU uses two blowers for cooling the regulator stack, one for the phase-controlled regulator stack, three for the A-gate, six for the B-gate, and one for the mixer board.
- The power frame uses: one blower motor for the regulator stack.

Power Conversion

- Motor generator: Converts the input line voltage into 208V, 400 Hz, 3-phase power required by the system's 400 Hz transformers and phase-controlled regulators. This voltage is distributed via CB2 and CB5.

  A phase-rotation detection circuit (YE110) assures that the motor generator rotates in a specified direction, thus eliminating a possible overheat condition.

- Ferroresonant transformers (TR3, TR5, TR108): Supply the special power requirements, which do not require close regulation.
- 400 Hz Transformers; Feed full-wave bridge rectifiers, which are used to supply the dc input voltage to the series regulators.
- Phase-controlled regulators: Used to convert 400 Hz input regulated dc output power for the monolithic storage devices.



POWER DISTRIBUTION, STAGE 2

The Model 145 power frame receives primary power input of 200/208/230V ac, 60 Hz or 200/220/235V ac volts, 380/408V ac, 50 Hz. Input distribution is three-phase and ground for 60 Hz; three-phase, neutral (for usage only), and ground for 50 Hz operation.

Power is routed from the customer ac service to a filter network (YE110). Each ac feed capacitor in the filter network has a resistor connected to ground. These resistors bleed off any charge remaining on the capacitors when the ac service is disconnected. Mainline circuit breakers for this system are CB1 and CB4.

After all circuit breakers are closed, initial turn-on brings:

1. AC voltage via fuses F3 and F4 to transformer TR3, which provides 24V dc to the power control circuit, 12V ac to the undervoltage detect circuits, and 7.25V to the sequence indicator lamps. The 24V dc is first routed to the system emerging power off (EPO) circuit via n/o contacts of K1 (and K3 if the 3046 is installed). If the EPO switch is closed and phase rotation to the MG is correct, relay K4 picks.
2. AC voltage to the convenience outlets. For 60 Hz systems, it is routed via fuses 1 and 2 to stepdown transformer TR5, which supplies 115V ac (for input voltage of 208V and 230V) to the convenience outlets. For 200V, 60 Hz systems, transform TR5 supplies 100V ac to the convenience outlets.

For Hz systems, the convenience outlet voltage is 220, 235, 380, or 408 (by Data-WYE) switching.

Further power distribution does not occur until one of the power switches is activated.

An auto transformer is provided for 200V 50 or 60 Hz; and 235/408 50 Hz inputs. Taps are provided on the transformer to obtain 220V output for the various input line voltages.

AC OUTPUTS

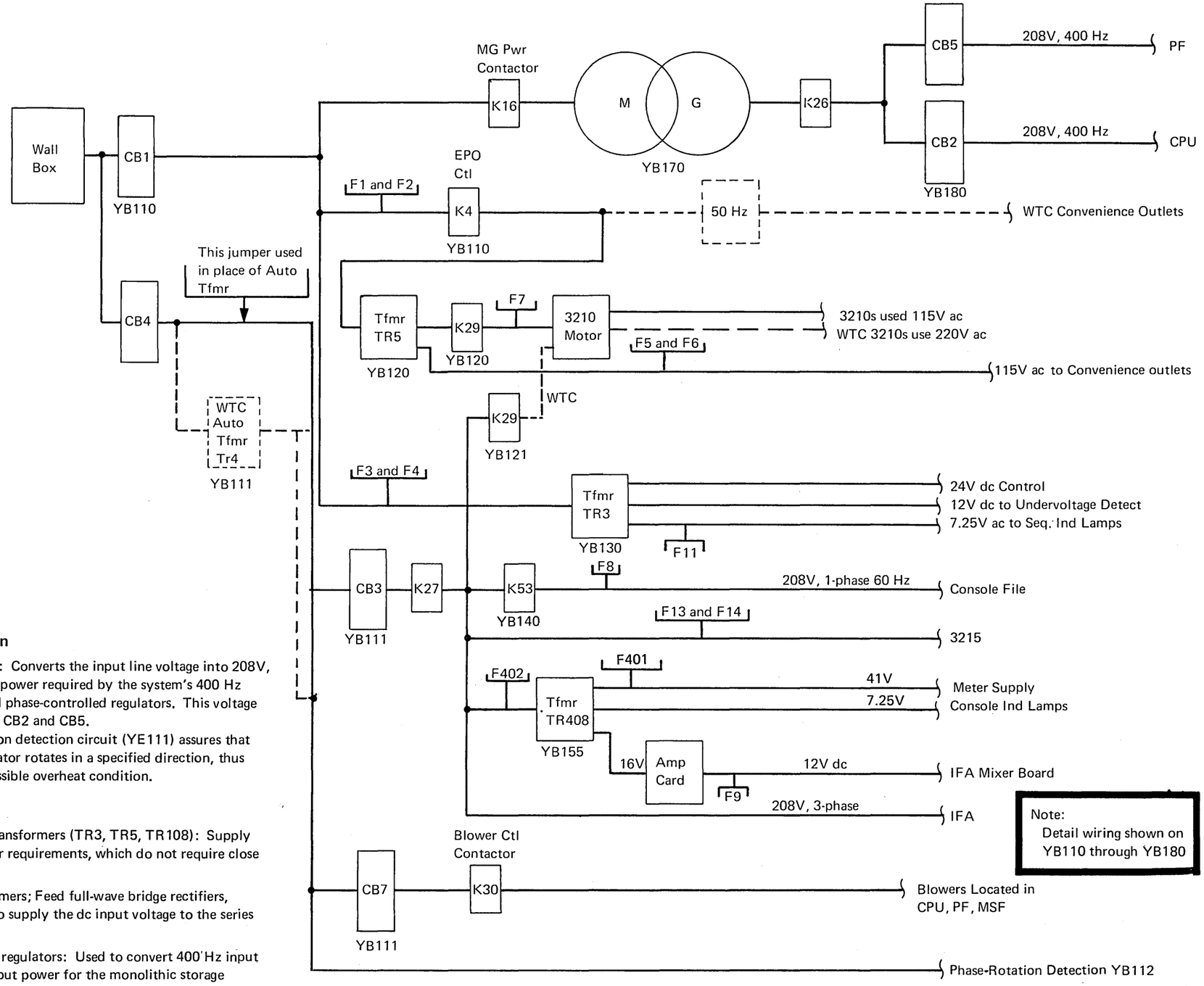
Convenience Outlets

Four duplex convenience outlets are provided with the system: CPU (2) and PF (2). For domestic machines, the outlets provide 115V ac, 60 Hz single-phase at a maximum of 15 amperes total for all outlets whose power is supplied by TR5. For World Trade, the outlets can provide either 100V ac, 60 Hz single-phase at a maximum of 15 amperes; or 200V, 220V, 235V, 50 Hz, at a maximum of 8 amperes. These currents are totals for all outlets.

Blowers

The system is cooled by blowers that require: 208/230V 60 Hz; or 220V, 50 Hz power.

- The CPU uses two blowers for cooling the regulator stack, six for the phase-controlled regulator stacks, three for the A-gate, six for the B-gate.
- The power frame uses: one blower motor for the regulator stack.



Power Conversion

- Motor generator: Converts the input line voltage into 208V, 400 Hz, 3-phase power required by the system's 400 Hz transformers and phase-controlled regulators. This voltage is distributed via CB2 and CB5.

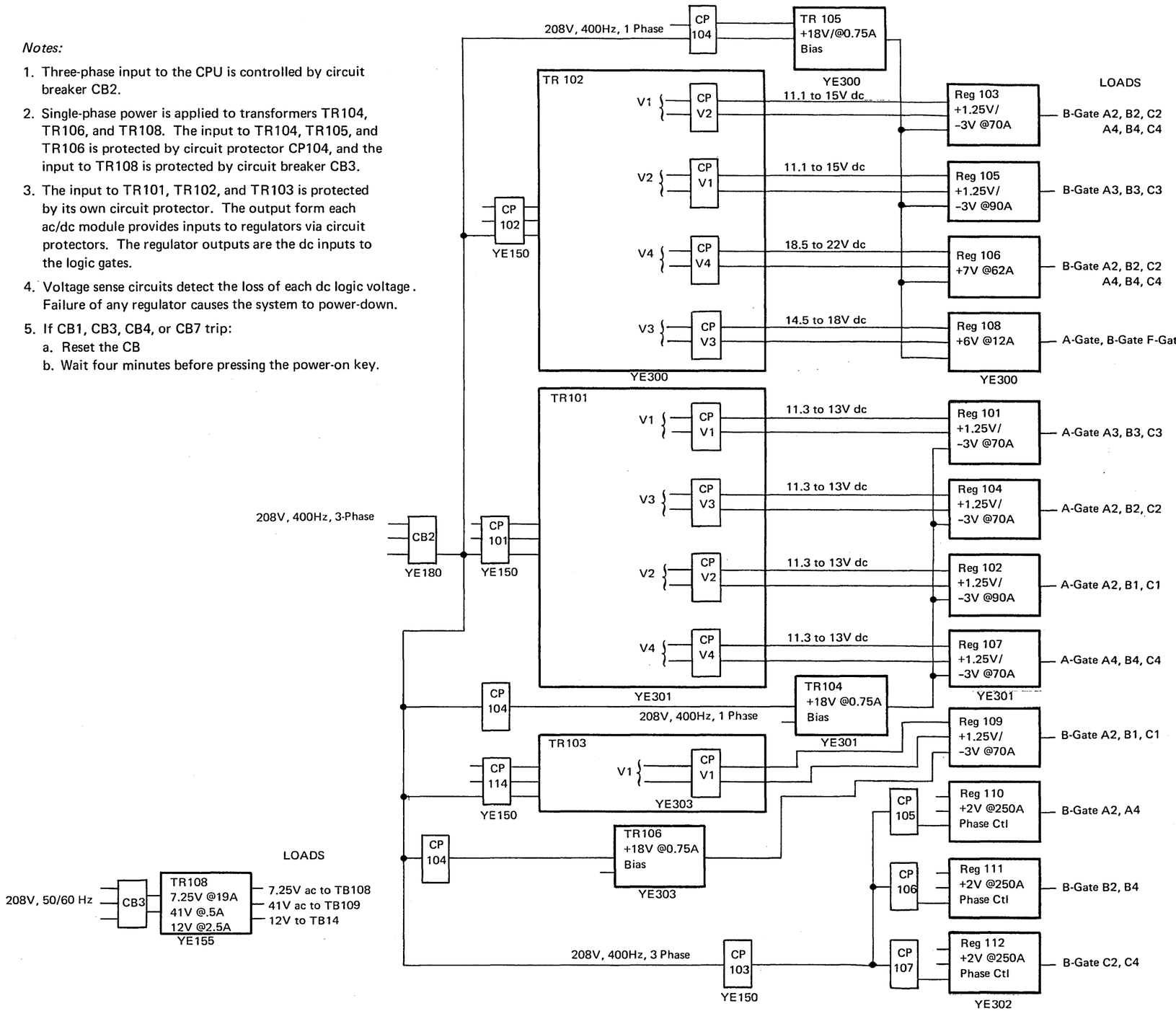
  A phase-rotation detection circuit (YE111) assures that the motor generator rotates in a specified direction, thus eliminating a possible overheat condition.

- Ferroresonant transformers (TR3, TR5, TR108): Supply the special power requirements, which do not require close regulation.
- 400 Hz Transformers; Feed full-wave bridge rectifiers, which are used to supply the dc input voltage to the series regulators.
- Phase-controlled regulators: Used to convert 400 Hz input regulated dc output power for the monolithic storage devices.

CPU POWER CONVERSION AND DISTRIBUTION, STAGE 1

Notes:

1. Three-phase input to the CPU is controlled by circuit breaker CB2.
2. Single-phase power is applied to transformers TR104, TR106, and TR108. The input to TR104, TR105, and TR106 is protected by circuit protector CP104, and the input to TR108 is protected by circuit breaker CB3.
3. The input to TR101, TR102, and TR103 is protected by its own circuit protector. The output form each ac/dc module provides inputs to regulators via circuit protectors. The regulator outputs are the dc inputs to the logic gates.
4. Voltage sense circuits detect the loss of each dc logic voltage. Failure of any regulator causes the system to power-down.
5. If CB1, CB3, CB4, or CB7 trip:
   a. Reset the CB
   b. Wait four minutes before pressing the power-on key.



BOARD ASSEMBLY LOCATION CHART

A-Gate (Card Side)

A1 Basic	B1 Feature Sel Ch. 1 or IFA	C1 Basic
A2 Sel Ch. 2	B2 Basic	C2 Basic
A3 Sel Ch. 3	B3 Basic	C3 Basic
A4 Feature Sel I/O	B4 Basic	C4 Basic

Hinge End

B-Gate (Wiring Side)
Main and Control Storage

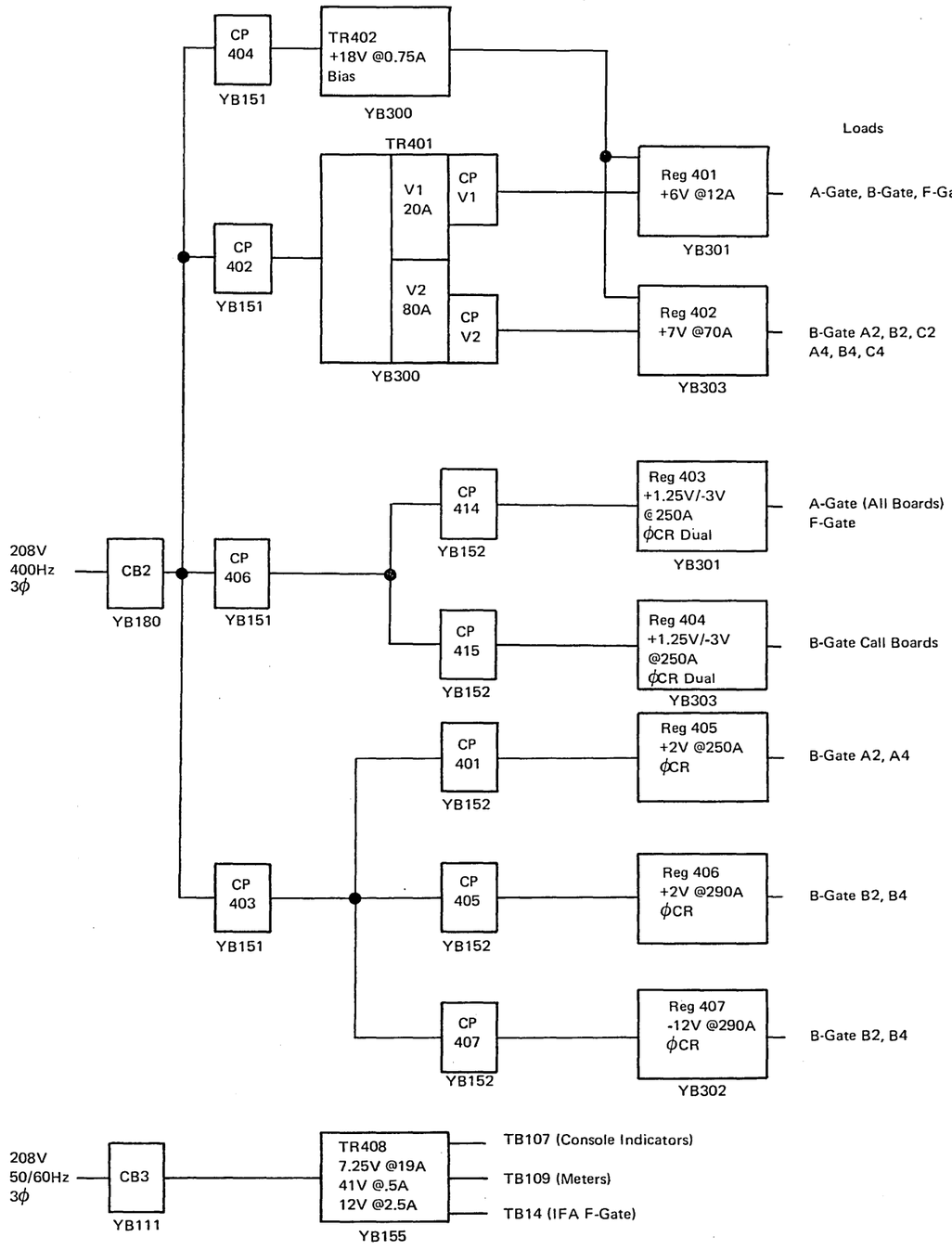
C1	B1	A1
C2	B2	A2
C3	B3	A3 ECC
C4	B4	A4

Hinge End

CPU POWER CONVERSION and DISTRIBUTION, STAGE 2

Notes

1. Three-phase input to the CPU is controlled by circuit breaker CB2.
2. The output from each ac/dc module provides inputs to regulators via the circuit protectors.
3. Voltage sense circuits detect the loss of each dc logic voltage. Failure of any regulator causes the system to power-down.
4. If CB1, CB3, CB4, or CB7 trip:
   a. Reset the CB.
   b. Wait four minutes before pressing the power-on key.



Board Assembly Location Chart
A-Gate (Card Side)

A1 Basic	B1 Feature Sel. Ch. 1 or IFA	C1 Basic
A2 Sel. Ch. 3	B2 Basic	C2 Basic
A3 Sel. Ch. 3	B3 Basic	C3 Basic
A4 Feature Sel. I/O	B4 Basic	C4 Basic

Hinge End

B-Gate (Wiring Side)
Main and Control Storage

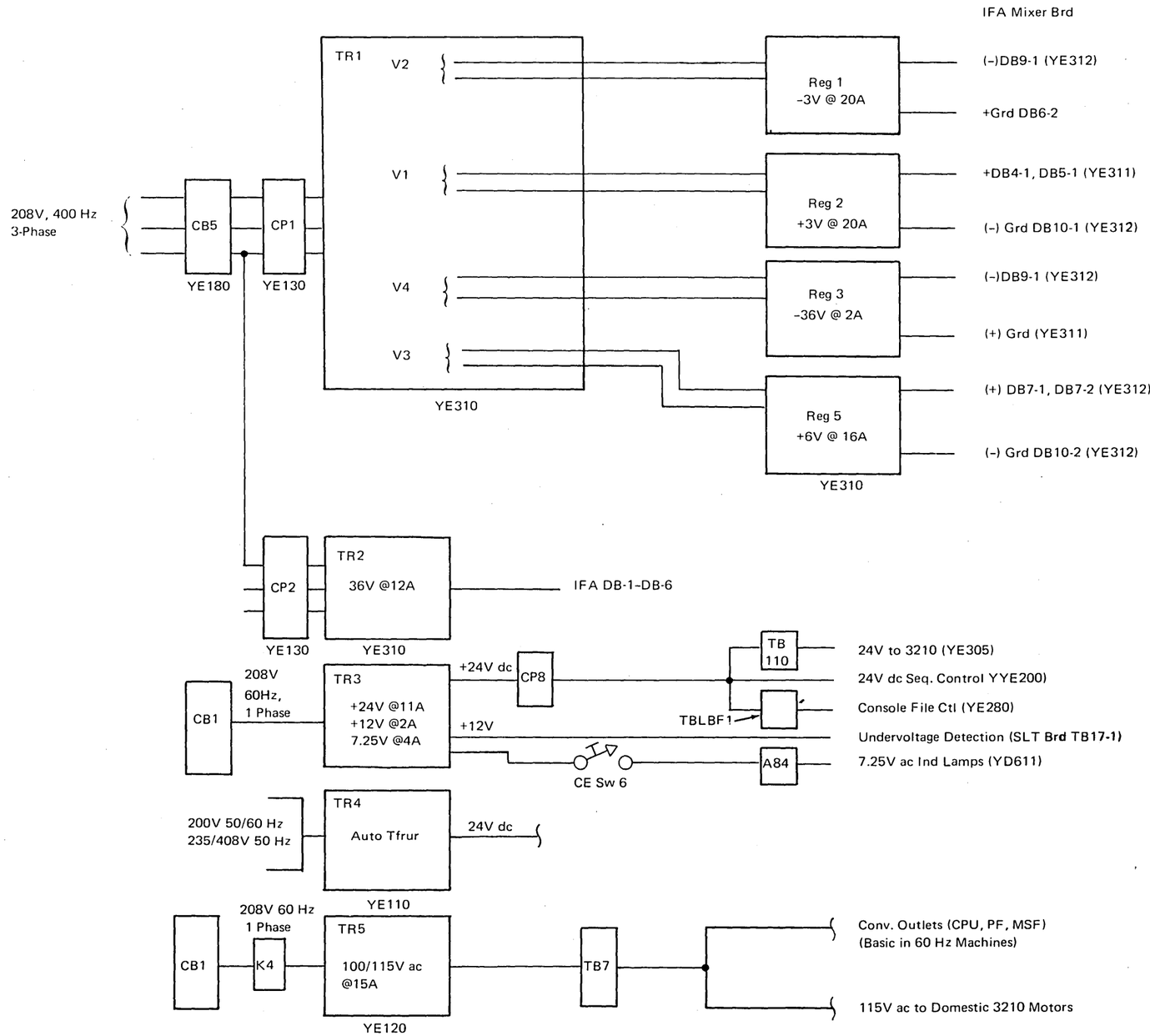
C1	B1	A1
C2	B2	A2
C3	B3	A3 ECC
C4	B4	A4

Hinge End

POWER FRAME POWER CONVERSION and DISTRIBUTION, STAGE 1

Notes:

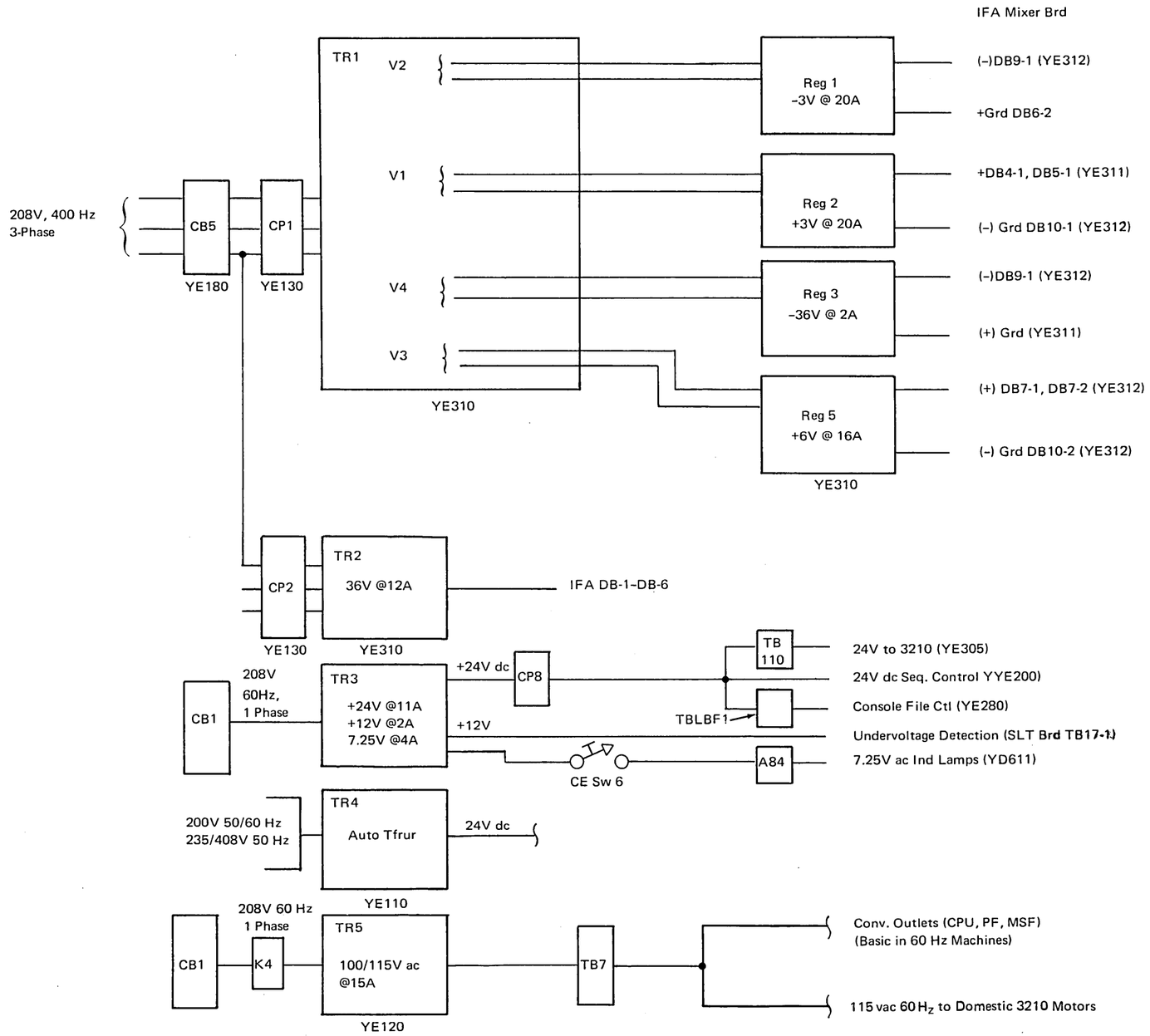
1. Three-phase power to TR1 and transformer TR2 is controlled by circuit breaker CB5. TR1 and TR2 are used exclusivley for the IFA feature. The input to each TR is protected by a circuit protector. The output from TR1 provides an input to four regulators. The regulator outputs and the output from TR2 supply dc voltages to the IFA distribution board DB1-DB11.
2. Circuit breaker CB1 controls the single-phase input to transformers TR3 (control transformer), TR4 (W/T auto transformer), and TR5 (convenience outlet transformer). The input to TR4 is also controlled by contacts of CPO relay K4.
4. Failure of the 24V dc output from TR3 causes a random shutdown of the entire system.



POWER FRAME POWER CONVERSION and DISTRIBUTION, STAGE 2

Notes:

1. Three-phase power to ac-dc module TR1 and transformer TR2 is controlled by a circuit breaker CB5. TR1 and TR2 are used exclusively for the IFA feature. The output to each TR is protected by a circuit protector. The output from TR1 provides an input to four regulators. The regulator outputs and the output from TR2 supply dc voltages to the IFA distribution board DB1-DB11.
2. Circuit breaker CB1 controls the single-phase input to transformers TR3 (control transformer), TR4 (WTC auto transformer), and TR5 (convenience outlet transformer). The input to TR4 is also controlled by contacts of CPO relay K4.
4. Failure of the 24V dc output from TR3 causes a random shutdown of the entire system.



CONTROL AND INDICATORS

POWER ON
START

This lamp is lit whenever a power-on sequence is in progress, also stays on for five-minute blower delay, after power-off.

POWER ON
COMPLETE

This lamp lights when the power-on sequence is complete and stays lit until power is turned off by means of the power-off switch or because of a power fault in any frame.
(exception: CE ERROR OVERRIDE ON)

HEADS
EXTENDED

This light is lit whenever a head-extend condition exists in IFA. If a power check is indicated during a normal power-down sequence while this is lit, a disk head remains extended in one of the disk-storage facilities.

POWER
CHECK
(R)

This indicator lights whenever any of the following conditions exists in any frame: power incomplete, to the system, CB trip, open thermal, MG check, CE2 switch at ERROR OVERRIDE, CE5 set at I/O HOLD.

MG CHECK
(R)

This indicator is on when a power problem (thermal or overvoltage) exists within the MG enclosure.

THERM
CHECK
(R)

This indicator is used with all thermal contacts in each frame. When this light is lit, at least one thermal switch has been opened as a result of a high ambient temperture.

CB TRIP
(R)

This indicator comes on when a circuit breaker trips in any of the three frames. (CPU, PF, MSF).

CPU
(R)

This indicator lights when a power failure or a thermal condition exists in the CPU frame.

PF
(R)

This indicator lights when a power failure or a thermal condition exists in the PF.

MSF
(R)

This indicator lights when a power failure or a thermal condition exists in the 3345 MSF (if attached).

CPU SEQ. 1		CPU SEQ. 2		MSF SEQ.1	MSF SEQ. 2	PF SEQ. 1
REG 101 ○ +1.25V	REG 101 ○ -3V	REG 102 ○ +1.25V	REG 103 ○ +1.25V	REG 205 ○ +2V	REG 201 ○ +1.25V	REG 1 ○ -3V
REG 102 ○ -3V	REG 104 ○ +1.25V	REG 104 ○ -3V	REG 103 ○ -3V	REG 206 ○ +2V	REG 201 ○ -3V	REG 2 ○ +3V
REG 105 ○ +1.25V	REG 105 ○ -3V	REG 107 ○ +1.25V	REG 106 ○ +7V	REG 207 ○ +2V	REG 202 ○ +7V	REG 3 ○ -36V
REG 107 ○ -3V	REG 109 ○ +1.25V	REG 109 ○ -3V	REG 108 ○ +6V	REG 208 ○ +2V		REG 5 ○ +6V
REG 110 ○ +2V	REG 111 ○ +2V	REG 112 ○ +2V				TR2 ○ +36V
TR 108 ○ 41V AC	TR 108 ○ +12V					

Each regulator lamp is marked for its level and lights when that regulator voltage reaches its designated level. These lights can be used, along with CE2 switch (ERROR OVERRIDE positio) and CE6 switch (REG TEST position), when an incompleté power-on sequence occurs to find which regulator(s) voltage has not reached its rated output.

CE SWITCHES

1	2	3	4	5	6
POWER ON	CK RESET	BLOWER OFF	NORMAL	I/O HOLD	LAMP TEST
○	○ ERROR OVERRIDE	○ MG HOLD	○	○ NORMAL	○ REG TEST
	NORMAL	MG PWR OFF CONTROLLED	POWER OFF	I/O OFF	NORMAL

CE PANEL, STAGE–1

SWITCHES	LABEL	FUNCTION
CE 1	POWER ON	Performs the same function as the console power-on key (initiates a power-on sequence).
CE 2	CK RESET	This is a momentary position of switch CE2 that resets the power check circuits (picks K12). The power check circuits must be reset after each power malfunction before power can be reapplied to the system.
	ERROR OVERRIDE	Bypasses all malfunctions that cause a power-check condition by providing a return path for relay, K12. The power-check conditions bypassed are: thermal trip, undervoltage detect, CB trip, and MG check. This position of CE2 switch can be used as a trouble shooting aid by enabling power turn-on under a power check condition. By using the error override capability along with CE6 switch (REG TEST position) when a CB trip is apparent, maintenance personnel can monitor the regulator sequence lights and find which regulator(s) are not supplying proper output voltages. When the switch is at ERROR OVERRIDE, the CE panel power check light is lit and the console POWER check light is lit to indicate that a CE switch is at a test mode position.
	NORMAL	Allows the power check circuits to function in their normal manner. CE2 switch should be maintained at this position during normal running operations.
CE 3	BLOWER OFF	Enables turning off the blowers prematurely after a power-down sequence has been started. (The blowers normally operate for five minutes after powering-down the system.)
	MG HOLD	Keeps the MG set running after the system has been powered down. (Bypasses the system power-off switch control over the MG set.)
	MG PWR OFF CONTROLLED	Allows the MG to be turned off with the system when the power-off switch is activated.
CE 4	NORMAL	Allows the power-on switches to control system power in the regular manner. CE4 switch must be at NORMAL in order for the system to operate.
	POWER OFF	Performs the same function as the console power0off key (intitiates a power-off sequence) Diasables the operation of both the console power-on and CE panel power-on switches. Anytime the CE 4 switch is at POWER OFF; the console power check light is lit. This switch can be used to prevent power from being applied to the system while it is being serviced.
CE 5	I/O HOLD	Allows the 24V dc control to the I/O units to be maintained after the system is powered-down. Anytime CE5 switch is at I/O HOLD, the CE panel power check light is lit and the console power light is turned on.
	NORMAL	Allows the I/O devices turn-on procedure to function under control of the system power-on operation. CE5 switch should be kept at NORMAL for all regular operations.
	I/O OFF	Inhibits the turn-on of I/O devices over the channel. Anytime CE5 is at I/O OFF, the power-on complete light does not light. If CE5 is switched to I/O OFF after power-on is complete, the channel I/Os will drop and the power-on complete light will turn off. To bring the I/Os up again, press POWER OFF; then POWER ON.
CE 6	LAMP TEST	Causes all CE panel lights to glow. The CE6 switch can be operated to LAMP TEST at any time without affecting system operation.
	REG TEST	Provides a means for checking all regulators outputs by means of the regulator sequencing lights. If a CB power check is indicated, one can use the CE2 switch in the ERROR OVERRIDE position along with REG TEST and detect the regulator(s) that are not providing the indicated output. (Associated regulator indicator is off.)
	NORMAL	Provides no function. CE6 switch should be kept at NORMAL during all regular system operations.

CPU CONSOLE (POWER CONTROL SWITCHES), STAGES 1 AND 2

There are three switches on the CPU console that are associated with power control: Emergency Pull switch, Power On key, and Power Off key.

1. **Emergency Pull Switch** (located on the upper right-hand section).

   Pulling this switch causes all system power to be removed within two seconds. However, primary power is present at both the entry and exit terminals of CB1, CB3, CB4, CB7, TR3, and TR4 (when auto transformer is used). Voltage is also present at the inputs to relays K4, K16, K27, and K30.

2. **Power On Key** (located on the lower left-hand section).

   The power on key, when pressed, initiates a power-on sequence for the CPU and on-line I/O units. The key turns red when pressed; and white when the power-on sequence is complete. The time required for a power-on sequence is determined by the number and type of I/O units online.

3. **Power Off Key** (located on the lower left-hand section).

   The power off key, when pressed, removes power to the CPU and online I/O units. Main- and control-storage information is lost.

   If power-supply failure or overtemperature condition occurs while the system is operating, a power-off sequence is initiated. After a power fault is corrected, press the power-off key, or the check reset switch on either the CE panel or the CPU console.

CONTROL AND INDICATORS (continued)

POWER ON START

This lamp is lit whenever a power-on sequence is in progress, also stays on for five-minute blower delay, after power-off.

POWER ON COMPLETE

This lamp lights when the power-on sequence is complete and stays lit until power is turned off by means of the power-off switch or because of a power fault in any frame.
(exception: CE ERROR OVERRIDE ON)

HEADS EXTENDED

This light is lit whenever a head-extend condition exists in IFA. If a power check is indicated during a normal power-down sequence while this is lit, a disk head remains extended in one of the disk-storage facilities.

POWER CHECK (R)

This indicator lights whenever any of the following conditions exists in any frame: power incomplete, to the system, CB trip, open thermal, MG check, CE2 switch at ERROR OVERRIDE, CE5 set at I/O HOLD.

MG CHECK (R)

This indicator is on when a power problem (thermal or overvoltage) exists within the MG enclosure.

THERM CHECK (R)

This indicator is used with all thermal contacts in each frame. When this light is lit, at least one thermal switch has been opened as a result of a high ambient temperture.

CB TRIP (R)

This indicator comes on when a circuit breaker trips in any of the three frames. (CPU, PF, MSF).

CPU (R)

This indicator lights when a power failure or a thermal condition exists in the CPU frame.

PF (R)

This indicator lights when a power failure or a thermal condition exists in the PF.

MSF (R)

This indicator lights when a power failure or a thermal condition exists in the 3345 MSF (if attached).

CPU SEQ. 1		CPU SEQ. 2	MSF SEQ.1	MSF SEQ. 2	PF SEQ. 1
REG 403 +1.25V	REG 404 +1.25V	REG 402 +7V	REG 205 +2V	REG 201 +1.25V	REG 1 -3V
REG 403 -3V	REG 404 -3V	REG 401 +6V	REG 206 +2V	REG 201 -3V	REG 2 +3V
			REG 207 +2V	REG 202 +7V	REG 3 -36V
			REG 208 +2V		REG 5 +6V
REG 405 +2V	REG 406 +2V	REG 407 +2V			TR2 +36V
TR 408 41V AC	TR 408 +12V	SPARE			

Each regulator lamp is marked for its level and lights when that regulator voltage reaches its designated level. These lights can be used, along with CE2 switch (ERROR OVERRIDE positio) and CE6 switch (REG TEST position), when an incomplete power-on sequence occurs to find which regulator(s) voltage has not reached its rated output.

Note:
CE Switch Function Chart same for Stage 1 and Stage 2.

CE SWITCHES

1	2	3	4	5	6
POWER ON	CK RESET	BLOWER OFF	NORMAL	I/O HOLD	LAMP TEST
	ERROR OVERRIDE	MG HOLD		NORMAL	REG TEST
	NORMAL	MG PWR OFF CONTROLLED	POWER OFF	I/O OFF	NORMAL

CE PANEL, STAGE–2

REMEMBER

There is a Reader's Comment Form at the back of this publication.

POWER SEQUENCING

POWER-ON SEQUENCE, Stage 1

- Before the power-on sequence can start, the EPO switch must be in the non-operated state so that the EPO relay can pick.
- Input power to the motor generator must have the proper phase rotation.
- Voltage, current and temperature indications must be normal, power sequence to all I/O channel related devices must be complete, and the voltage sense circuits must sense the correct voltage, before power-on sequence is complete.

For the following discussion, refer to the power-on flow on logic page YE011, the power-on sequence and timing chart (YE070), and the power sequencing and control illustration. With mainline power applied to the processing unit, but before the power-on key is pressed, three-phase input is routed through a phase detect circuit (YE111), and 208V ac is applied via fuses F3 and F4 to transformer TR3 that supplies: 24V dc via n/o contacts of CP8, 12V dc via fuse F10 and 7.25V ac via fuse F11, to the central sequencing and control circuits. The 24V dc is first routed to the emergency power-off (EPO) circuit. To energize the EPO control relay K4, two conditions must be satisfied; the EPO switch must be closed, and the three-phase input to the motor generator(s) must have proper phase orientation. If the input phase to the MG is correct, phase rotation relay K1 is energized. (If the 3046 Power Unit is connected to the system, a second relay K3 is used to assure proper phase relationship for the MG in the 3046.) The contacts of K1 and K3 are connected in series. This interlock makes sure that, whenever the 3046 is connected to the Model 145, the system will power up only when the main input source to each unit is connected in accordance with the installation planning data.

Note: Whenever the 3046 power unit is connected to the Model 145, certain interface control lines between the units directly affect system power. These interface control lines:

1. Assure that the 3046 motor generator rotation is correct.
2. Interlock the 3046 with the system EPO circuits.
3. Interlock the 3046 line cord with the line cord in the Model 145 power frame. If either cord is removed from the outlet or if the mainline circuit breaker in either frame is turned off, the entire system is turned off.
4. Provide an interface from the 3046 MG overvoltage and overheat detection circuits to the undervoltage board, located in the Model 145 power frame. If either or both of these conditions are detected, the entire system is turned off.
5. In order to reduce in-rush current, make sure that the 3046 is powered-up about 15 seconds after the generator within the Model 145 power frame has been powered.

From the contacts of K4, 208V ac is routed to the step down transformer TR5 when connected to a 60 Hz source, which provides 115V ac to the convenience outlets located on each frame. For a 50 Hz source, the voltage is routed directly to the convenience outlets as shown on logic page YE120. Because 24V dc is present, all I/O EPO relays (K5 through K9) pick (if installed) and signal to the attached I/O control units that no EPO condition exists. The MG check relay K11 also picks at this time.

Before the power supplies can be sequenced on, power check relay K12 must be picked, which occurs if no power fault is sensed, or by positioning CE switch 2 to the ERROR OVERRIDE. To initiate the system power-on sequence, CE4 switch must be at NORMAL, and the power-on switch on either the CPU console or the CE panel must be activated. Pressing the power-on key picks power-on relay K14. As soon as the K14 contacts transfer, MG start control relay K15 is picked and 24V dc is supplied to time-delay relay K20. A set of N/O contacts of K15 allow MG main power relay K16 to pick. The contacts of K16 are the mainline contactors that allow power to be applied to the MG set. Time-delay relay K20, which has a power-on delay of about 10 seconds, allows the MG to come up to speed under a no-load condition. The remaining power sequencing is as follows:

1. After the 10-second time-out of K20, its n/o contacts complete the circuit for MG hold relay K21. (A set of K21 contacts are used in conjunction with CE3 switch, position to MG HOLD, to keep the MG running even after the system is sequenced down.)
2. Relay K22 (400 Hz power control) is energized through K21 n/o contacts.
3. Relay K26 (400 Hz contactor), which controls the MG output, is picked through K22 n/o and K16 n/o contacts. The contactors of this relay allow 208V ac, 400 Hz, to pass to the 400 Hz transformer/rectifiers that supply the dc input voltage to the series regulators and also allow the 208V ac, 400 Hz, to pass to the phase-controlled regulators.
4. Relay K27 (AC contactor) is picked through K26 n/o contacts (YE110). With K27 picked, power is distributed as follows:
   - Heads extended relay K28 picks through n/o contacts of K15, K14, and K12, preventing a complete power shutdown and possible damage to the disk packs whenever a disk head is extended. The HEADS EXTENDED light, located on the CE panel, will always be lit whenever K28 is energized.
   - Relay K29 (60 Hz Citation power control) is picked, allowing power to be applied to the 3210 and 3215 (if installed).
   - Power is applied to transformer TR108 which supplies: 41V ac for the metering circuits, 7.25V ac to the console indicator lamps, and 12V dc to the CPU mixer board.
   - 208V ac is provided for the IFA.
5. Relay K30 (blower control contactor) is picked, providing ac power to all blowers.
6. After the MG output is distributed to the regulators and TR packs, first sequence detect relay K31 picks only when all of the following reach 70 per cent of their rated current.

CPU FRAME		POWER FRAME	MAIN STORAGE FRAME (If installed)
1.25V and 3V Reg		*Mid-Pack Reg*	*2V Reg*
101	105	Reg 1 (–3V)	204
102	107	Reg 2 (+3V)	205
104	108	Reg 3 (–36V)	206
2V Regs		Reg 5 (+6V)	
110, 111, 112			
TR108			
(12V, 36V, 41V)			

7. Second sequence relay K32 picks after K31 picks and only when all of the following reach 70 per cent of their rated current.

CPU FRAME	MAIN STORAGE FRAME (If installed)
Reg 103 (1.25V and 3V)	Reg 201 (1.25V and 3V)
Reg 106 (7V)	Reg 202 (7V)
Reg 108 (6V)	

8. Relay K34 picks through K20 n/o, K28 n/o, K14 n/o, and K12 n/o contacts. After the relay contacts transfer, K34 is held in a picked state through its own n/o contacts. The picking of this relay also enables a signal that allows the sensing of the phase-controlled rectifier outputs. The transfer of K34 n/c contacts removes the bypass for the undervoltage detection circuit to the power check relay, and causes K20 to drop.
9. Relay K33 (24V 2nd sequence control) picks through n/o contacts of K31, K32, n/c contacts of the CB auxiliary contacts, and n/c contacts of K14 and K12. With relay K33 picked, 24V dc is routed to the 3215 (if installed).
10. The attached I/O devices are now provided with 24V dc control voltage. A maximum of five I/O control groups can be installed, to control a maximum of 40 control units. There are five I/O stepper switches. If all are set to their home position, relay K35 (I/O power hold) picks through K12 n/o and K34 n/o contacts. (If any I/O stepper switch is not at the home position, resetting the switch is done by pressing the power-off key.)
11. For each installed I/O group, there is an associated I/O group power control relay (K38 through K42) and a stepper switch I/O group relay (K44 through K48). The 24V dc control voltage is sequenced to the I/O control units through a stepper-switch action, commencing with I/O group one (relay K44 picks). After the stepper switch for I/O group one is complete (switch returns to home), the sequence is repeated until all connected I/O groups receive their 24V dc control voltage, (see ALDs YE200 and YE290).
12. When the last device receives its 24V dc control voltage, relay K151 (power reset) is picked, causing the power-on key white backlight to turn on, indicating that the power-on sequence is complete.

POWER-ON SEQUENCE, Stage 2

- Before the power-on sequence can start, the EPO switch must be in the non-operated state so that the EPO relay can pick.
- Input power to the motor generator must have the proper phase orientation.
- Voltage, current, and temperature indicators must be normal, power sequence to all I/O channel-related devices must be complete, and the voltage sense circuits must sense the correct voltage, before power-on sequence is complete.

Refer to the power-on flow on logic page YB011, the power-on sequence and timing chart (YB070), and the power sequencing and control (Page 11-22). With mainline power applied to the processing unit, but before the power-on key is pressed, three-phase input is routed through a phase detect circuit (YB111), and 208V ac is applied via fuses F3 and F4 to transformer TR3 that supplies: 24V dc via n/o contacts of CP8, 12V dc via fuse F10 and 7.25V ac via fuse F11, to the central sequencing and control curcuits. The 24V dc is first routed to the emergency power-off (EPO) circuit. To energize the EPO control relay K4, two conditions must be satisfied: the EPO switch must be closed, and the three-phase input to the motor generator(s) must have proper phase orientation. If the input phase to the MG is correct, phase rotation relay K1 is energized. (If the 3046 power unit is connected to the system, a second relay K2 is used to assure proper phase relationship for the MG in the 3046.) The contacts of K1 and K2 are connected in series. This interlock makes sure that, whenever the 3046 is connected to the Model 145, the system will power-up only when the main input source to each unit is connected in accordance with the installation planning data.

Note: Whenever the 3046 power unit is connected to the Model 145, there are certain interface control lines between the units that directly affect system power. These interface control lines:

1. Assure that the 3046 motor generator rotation is correct.
2. Interlock the 3046 with the system EPO circuits.
3. Interlock the 3046 line cord with the line cord in the Model 145 power frame. If either cord is removed from the outlet or if the mainline circuit breaker in either frame is turned off, the entire system is turned off.
4. Provide an interface from the 3046 MG overvoltage and overheat detection circuits to the undervoltage board, located in the Model 145 power frame. If either or both of these conditions are detected, the entire system is turned off.
5. In order to reduce in-rush current, make sure that the 3046 is powered-up about 15 seconds after the generator within the Model 145 power frame has been powered.

From the contacts of K4, 208V ac is routed to the stepdown transformer TR5 when connected to a 60 Hz source, which provides 115V ac to the convenience outlets located on each frame. For a 50 Hz source, the voltage is routed directly to the convenience outlets as shown on logic page YB120. Because 24V dc is present, all I/O EPO relays (K5 through K9) pick (if installed) and signal to the attached I/O control units that no EPO condition exists. The MG check relay K11 also picks at this time.

Before the power supplies can be sequenced on, power check relay K12 must be picked, which occurs if no power fault is sensed, or by positioning CE switch 2 to the ERROR OVERRIDE. To initiate the system power-on sequence, CE4 switch must be at NORMAL, and the power-on switch on either the CPU console or the CE panel must be activated. Pressing the power-on key picks power-on relay K14. As soon as K14 contacts transfer, MG start control relay K15 is picked and 24V dc is supplied to time-delay relay K20. A set of n/o contacts of K15 allows MG main power relay K16 to pick. The contacts of K16 are the mainline contactors that allow power to be applied to the MG set. Time-delay relay K20 has a power-on delay of about 10 seconds to allow the MG to come up to speed under a no load condition. The remaining power sequencing is:

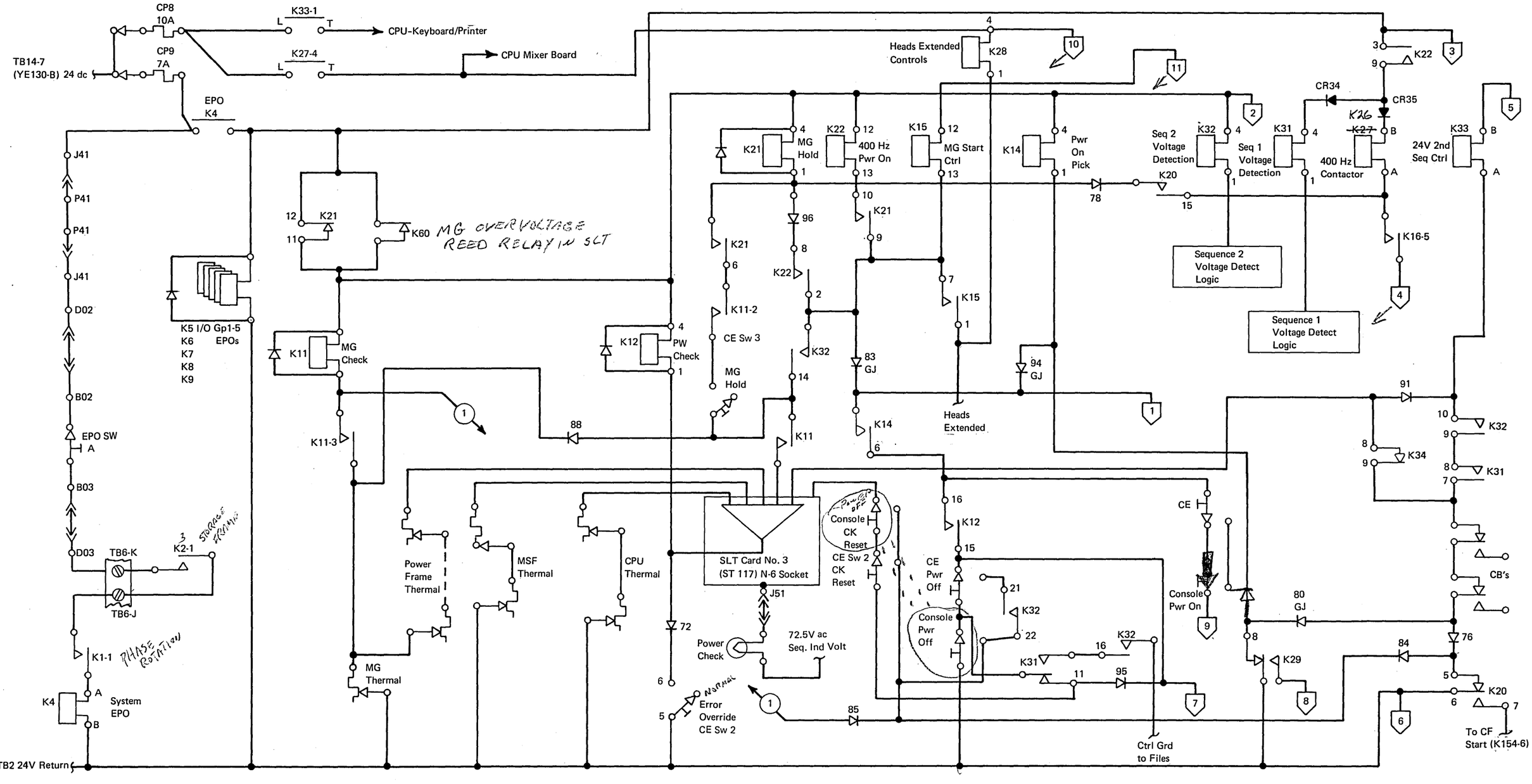
1. After the 15 second time-out of K20, its n/o contacts complete the circuit for MG hold relay K21. (A set of K21 contacts are used in conjunction with CE3 switch, positioned to MG HOLD, to keep the MG running even after the system is sequenced-down.)
2. Relay K22 (400 Hz power control) is energized through K21 n/o contacts.
3. Relay K26 (400 Hz contactor), which controls the MG output, is picked through K22 n/o and K16 n/o contacts. The contactors of this relay allow 208V ac, 400 Hz, to pass to the 400 Hz transformer/rectifiers that supply the dc input voltage to the series regulators and also allow the 208V ac, 400 Hz to pass to the phase-controlled regulators.
4. Relay K27 (AC contactor) is picked through K26 n/o contacts (YB110). With K27 picked, power is distributed as follows.
   - Heads-extended relay K28 picks through n/o contacts of K15, K14, and K12, preventing a complete power shutdown and possible damage to the disk packs whenever a disk head is extended. The heads-extended light located on the CE panel will always be lit whenever K28 is energized.
   - Relay K29 (60 Hz citation power control) is picked, allowing power to be applied to the 3210 and 3215 (if installed).
   - Power is applied to transformer TR408, which supplies: 41V ac for the metering circuits, 7.25V ac to the console indicator lamps, and 12V dc to the CPU mixer board.
   - 208V ac, is provided for the IFA.
5. Relay K30 (blower control contactor) is picked, providing ac power to all blowers.
6. After the MG output is distributed to the regulators and TR packs, sequence detect relay K31 picks only when all of the following reach 70 percent of their rated current.

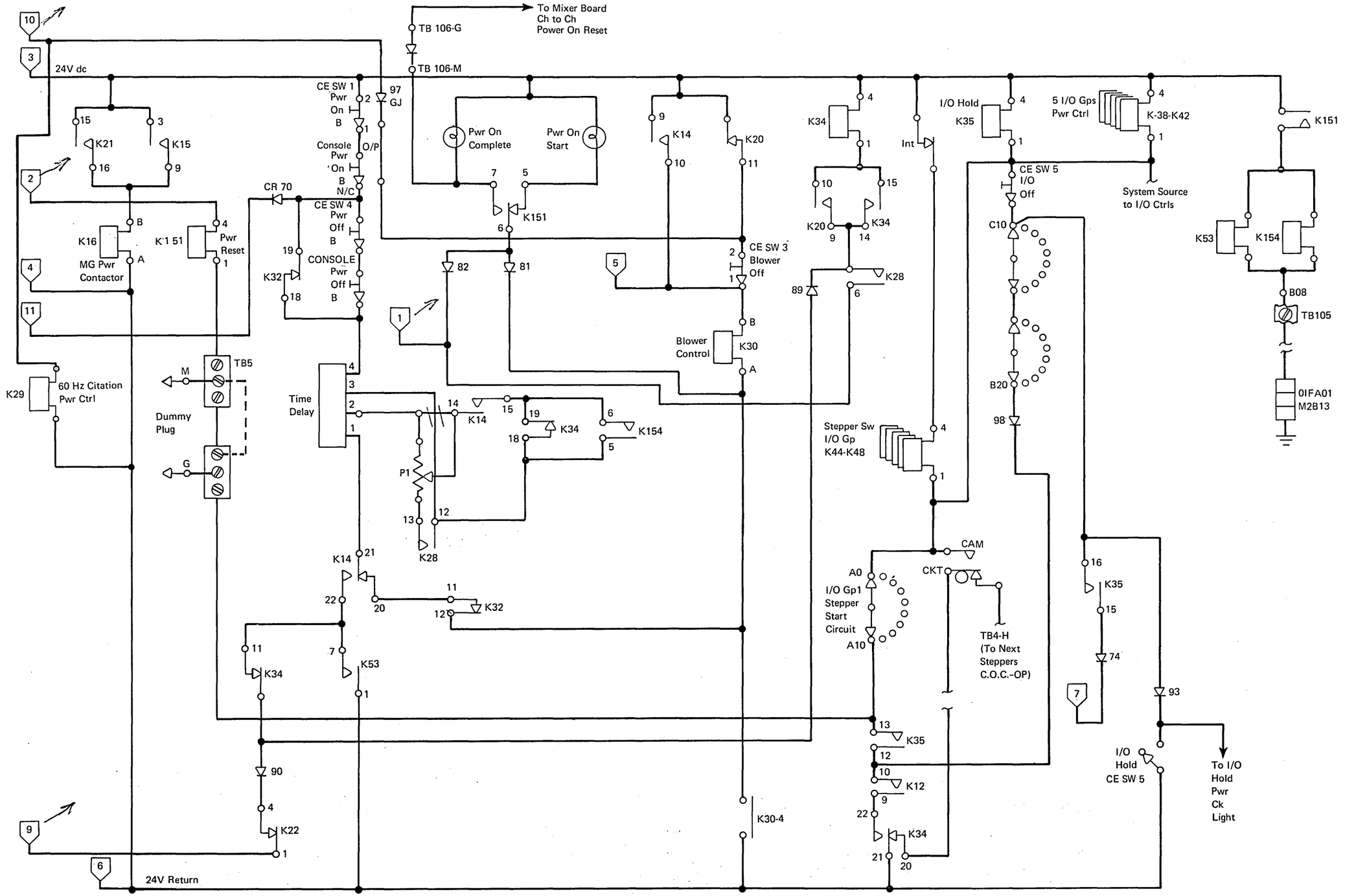
CPU FRAME	POWER FRAME	MAIN STORAGE FRAME (If installed)
1.25V and 3V Reg	*Mid-Pack Reg*	*2V Reg*
403 404	Reg 1 (–3V)	205
	Reg 2 (+3V)	206
	Reg 3 (–36V)	207
	Reg 5 (+6V)	208
2V Regs		
405, 406, 407		
TR408		
(12V, 36V, 41V)		

7. Second sequence relay K32 picks after K31 picks and only when all of the following reach 70 percent of their rated current.

CPU FRAME	MAIN STORAGE FRAME (If installed)
Reg 402 (7V)	Reg 201 (1.25V and 3V)
Reg 401 (6V)	Reg 202 (7V)

8. Relay K34 picks through K20 n/o, K28 n/o, K14 n/o, and K12 n/o contacts. After the relay contacts transfer, K34 is held in a picked state through its own n/o contacts. Picking this relay also enables a signal that allows the sensing of the phase-controlled rectifier outputs. Transfer of K34 n/c contacts removes the bypass for the undervoltage detection circuit to the power check relay and causes K20 to drop.
9. Relay K33 (24V 2nd sequence control) picks through n/o contacts of K31, K32, n/c contacts of the CB auxiliary contacts, and n/c contacts of K14 and K12. With relay K33 picked, 24V dc is routed to the 3215 (if installed).
10. The attached I/O devices are now provided with 24V dc control voltage. A maximum of five I/O control groups can be installed, to control a maximum of 40 control units. There are five I/O stepper switches. If all are set to their home position, relay K35 (I/O power hold) picks through K12 n/o and K34 n/o contacts. (If any I/O stepper switch is not at the home position, resetting the switch is done by pressing the power-off key.)
11. For each installed I/O group, there is an associated I/O group power control relay (K38 through K42) and a stepper switch I/O group relay (K44 through K48). The 24V dc control voltage is sequenced to the I/O control units through a stepper-switch action, commencing with I/O group one (relay K44 picks). After the stepper switch for I/O group one is complete (switch returns to home), the sequence is repeated until all connected I/O groups receive their 24V dc control voltage, (see logics YB200 and YB290).
12. When the last device receives its 24V dc control voltage, relay K151 (power reset) is picked, causing the power-on key white backlight to turn on, indicating that the power-on sequence is complete.


CP8
10A
CP9
7A
TB14-7
(YE130-B) 24 dc
K33-1
L
T
CPU-Keyboard/Printer
K27-4
CPU Mixer Board
EPO
K4
J41
P41
P41
J41
D02
B02
EPO SW
A
B03
D03
TB6-K
TB6-J
K2-1
K1-1
K4
A
B
System
EPO
TB2 24V Return
K5 I/O Gp1-5
K6
K7
K8
K9
EPOs
12
K21
11
K60
MG OVERVOLTAGE
REED RELAY IN SLT
K11
MG
Check
K11-3
MG
Thermal
Power
Frame
Thermal
MSF
Thermal
CPU
Thermal
K12
PW
Check
1
4
K11-2
CE Sw 3
MG
Hold
14
88
1
SLT Card No. 3
(ST 117) N-6 Socket
J51
72
Power
Check
72.5V ac
Seq. Ind Volt
6
5
Error
Override
CE Sw 2
85
Heads Extended
Controls
K28
4
1
10
11
K21
4
MG
Hold
1
K22
12
400 Hz
Pwr On
13
K15
12
MG Start
Ctrl
13
K14
4
Pwr
On
Pick
1
78
96
8
K22
2
K21
6
K21
10
K21
9
7
K15
1
K32
83
GJ
94
GJ
Heads
Extended
K14
6
1
Seq 2
Voltage
Detection
K20
15
K32
4
1
Seq 1
Voltage
Detection
K31
4
1
2
Sequence 2
Voltage Detect
Logic
Sequence 1
Voltage Detect
Logic
CR34
CR35
K26
K27
3
9
K22
3
400 Hz
Contactor
B
A
K33
B
A
24V 2nd
Seq Ctrl
5
K16-5
4
Console
CK
Reset
CE Sw 2
CK
Reset
Power
Off
16
K12
15
CE
Pwr
Off
21
K32
22
Console
Pwr
Off
K31
16
11
95
K32
7
CE
Console
Pwr On
9
8
80
GJ
K29
8
91
10
K32
9
8
K34
9
8
K31
7
84
76
5
K20
6
7
6
CB's
To CF
Start (K154-6)
Ctrl Grd
to Files


10
3
24V dc
2
4
11
15
3
K21
K15
16
9
B
K16
MG Pwr
Contactor
A
4
K151
Pwr
Reset
1
CR 70
19
K32
18
CE SW 1
Pwr
On
B
2
1
97
GJ
Console
Pwr
On
B
O/P
N/C
CE SW 4
Pwr
Off
B
CONSOLE
Pwr
Off
B
TB 106-G
To Mixer Board
Ch to Ch
Power On Reset
TB 106-M
Pwr On
Complete
Pwr On
Start
7
5
K151
6
82
81
1
9
K14
10
K20
11
5
2
CE SW 3
Blower
Off
1
B
Blower
Control
K30
A
K34
4
1
10
15
K20
9
14
K34
89
K28
6
Int
I/O Hold
K35
4
1
5 I/O Gps
Pwr Ctrl
K-38-K42
4
1
System Source
to I/O Ctrls
CE SW 5
I/O
Off
C10
B20
98
K151
K53
K154
B08
TB105
0IFA01
M2B13
K29
60 Hz Citation
Pwr Ctrl
TB5
M
Dummy
Plug
G
Time
Delay
4
3
2
1
14
K14
15
19
K34
18
6
K154
5
P1
13
12
K28
K14
21
22
20
11
12
K32
11
K34
7
K53
1
90
4
K22
1
9
6
24V Return
K30-4
Stepper Sw
I/O Gp
K44-K48
4
1
CAM
A0
I/O Gp1
Stepper
Start
Circuit
A10
CKT
TB4-H
(To Next
Steppers
C.O.C.-OP)
13
K35
12
10
K12
9
22
K34
21
20
16
K35
15
74
7
93
I/O
Hold
CE SW 5
To I/O
Hold
Pwr
Ck
Light

POWER SEQUENCING (Continued)

POWER-OFF SEQUENCE, Stages 1 and 2

- System power is turned off in varying ways, depending upon the power-off reason.
- Power can be turned off from the console (power-off key or EPO switch), from the CE panel (CE1 switch), or by one of several system protection devices (overcurrent, overvoltage, low voltage sense, over temperature).

NORMAL POWER-OFF SEQUENCE

Refer to the power-off timing chart (YE071) and the power-off flow chart (YE015) during the following discussion.

Note: After the power-off sequence, the 24V dc power control voltage and convenience outlet voltages are still available.

Pressing the power-off key breaks the circuit to relay K14 to start the power-off sequence. Activating the power-off key also breaks the paths to relays K28 (if all disk heads are retracted), K34, K35, K38 through K42, and disconnects the 24V dc system source to the I/O controls. Dropping of either relay K34 or K35 interrupts the return for relay K151, causing the white lamp in the console power-on key to turn off and the red portion to light. The power-on complete light on the CE panel is also turned off. Relay K14 n/o contacts 16 and 17, which are in series with K31 n/o contacts 18 and 19, interrupt the remote start line to regulator 108 (+6V). This action removes excitation from regulator 108; and when the regulator output drops to approximately 1.2V, sequence 2 detect relay K32 is dropped. The transfer of relay K32 n/o contacts now causes relay K33 (24V second sequence control) to drop, removing the 24V dc control to the CPU printer/keyboard. The return of K32 n/c contacts starts relay K20 for a five-minute time-out. This action allows the blowers to continue to operate after the power-down sequence is complete to allow adequate cooling for the system components.

The remaining power is sequenced off as follows.

1. If all head assemblies in the attached disk-storage devices are retracted, relay K15 (MG start control) and relay K28 (heads extended) are dropped via K32 n/o contacts. The heads must be retracted before the disk-storage device drive motors are stopped because the heads depend on surface air movement for proper head-to-disk distance. Further power-down sequencing is halted until relay K28 and relay K15 are dropped. If relay K28 remains on, n/o contacts of K28 cause relay K20 to pick prematurely. With K20 and K28 picked at the same time, one leg to the power check SLT card is opened, causing relay K12 (power check) to pick. This action causes the power-on key red backlight and the CE panel power check light to come on.
2. After K15 drops, relay K21 (MG hold) and relay K22 (400 Hz power control) are dropped due to n/o contacts of K32. If CE3 switch is at MG HOLD, relay K21 is maintained energized, and its n/o contacts keep relay K16 picked. This allows you to keep the MG set running even after the normal power-down sequence has been completed.
3. 400 Hz power control relay K16, whose contactors allow power to be applied to the MG, and 400 Hz contactor relay K26, whose contactors control the MG output, are dropped, removing power to all system power supplies.
4. Relay K31 (sequence 1 detect) is dropped because of the voltage loss to the power supplies.
5. Relay K27 (AC contactor) is dropped due to K26 n/o contacts, removing power from transformer TR108, console file, and the ac voltage to the WTC 3210.
6. Relay K29 is de-energized, removing the 115V ac from the 3210 motors.
7. After five minutes have elapsed since voltage has been applied to K20, this relay picks. K20 n/c contacts interrupt the 24V dc to relay K30 causing it to drop. When K30 drops, all system blowers are turned off.

All system power has now been removed except for primary ac input power, convenience outlet power, and power to transformer TR3. TR3 provides the 24V dc EPO power, 12V to the undervoltage detect logic, and 7.25V to the sequence indicator lamps.

EMERGENCY POWER-OFF (EPO), Stages 1 and 2

- Pulling the EPO switch removes all system ac and dc power.
- When pulled, the EPO switch latches in the OUT position and must be restored by a service representative.
- When two processing units are tied together, power for both systems can be removed by operation of either console EPO pull switch.

Operating the EPO pull switch on the console opens the circuit to relay K4 (YE200). When relay K4 drops, all convenience outlet power drops, and all the 24V control voltage is removed, causing all control relays to drop. Thus all power is removed from the system except for the voltage at both the entry and exit terminals of CB1, CB3, CB4, CB7, TR3, and TR4 (when the auto transformer is used). Voltage is also present at the inputs to relays K4, K16, K27, and K30.

The EPO pull switch is a two-pole switch. The second pole is connected to J41. A second system, when connected to J41, is controlled by this second switch pole. Likewise, this system is controlled by a second pole on the EPO switch of a second system. The second switch pole is connected in series with the first system EPO switch via J41.

When the EPO pull switch has been operated, a mechanical latch holds the switch in the transferred position. This mechanical latch must be restored by a service representative who must gain access to the rear of the console panel. If the switch has been pulled, learn the reason and repair; then reset the switch. Once the EPO switch is restored, power can be turned on by doing a power check reset and pressing the system power-on key.

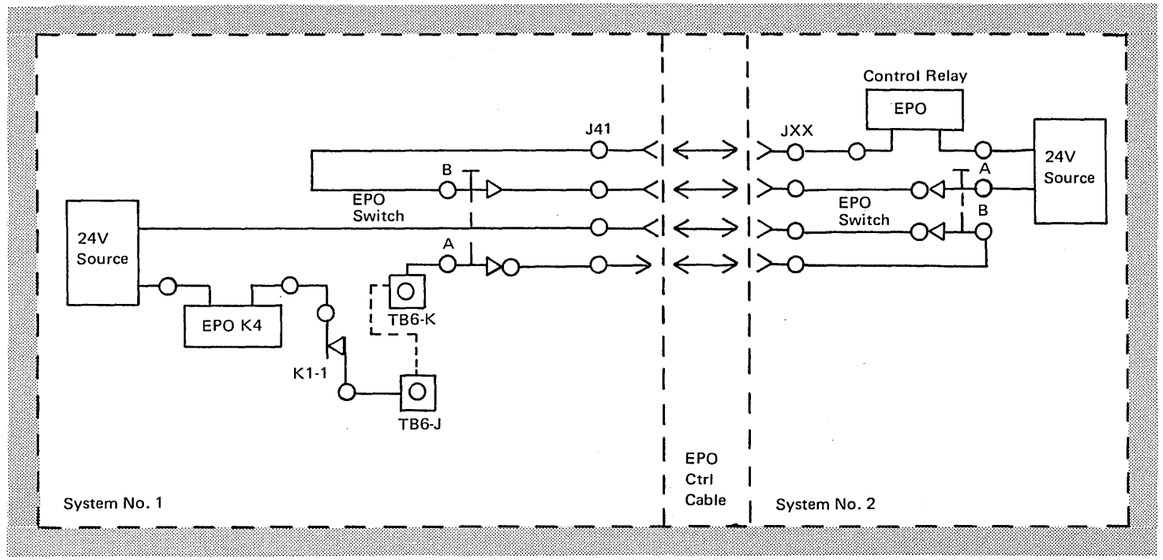
POWER-OFF RESULTING FROM a POWER CHECK, Stages 1 and 2

In the event of a power check (malfunction within the power or cooling system) a normal power-down sequence is initiated. Four conditions can cause a power check:

1. Overvoltage/overcurrent.
2. Undervoltage.
3. High temperature within the system frames, sensed by thermal devices.
4. Motor generator fault condition.

Whenever any one of these conditions is sensed, relay K12 (power check) is turned off. A set of n/o contacts of relay K12 are in series with the power-off switch; so, whenever K12 is turned off, a normal power-off sequence occurs. All power faults are indicated on the CE panel. By monitoring the lights, maintenance personnel can identify the power fault and its frame location. To assist in isolating a power failure, the CE panel is equipped with a switch (CE2). By placing the CE2 switch at ERROR OVERRIDE, all power fault circuits to relay K12 are bypassed, keeping this relay picked. Using the ERROR OVERRIDE position of CE2 switch along with CE6 switch (REG TEST position), one can observe the regulator lights to find which regulator is turned off.

Whenever power is sequenced off as a result of a power-check condition, the system power-off key must be pressed before the system can be restarted.



PROTECTION and CHECKING CIRCUITS

Protection and checking circuits are provided in this system to prevent equipment damage and to simplify maintenance. During normal system operation, the protection circuits operate in conjunction with the control circuits to apply power sequentially to the system. In the event of a malfunction (such as overvoltage, or thermal), the protection circuits initiate a sequential power-down operation.

Checking circuits allow detection of a circuit that is functioning marginally, before it becomes a failure.

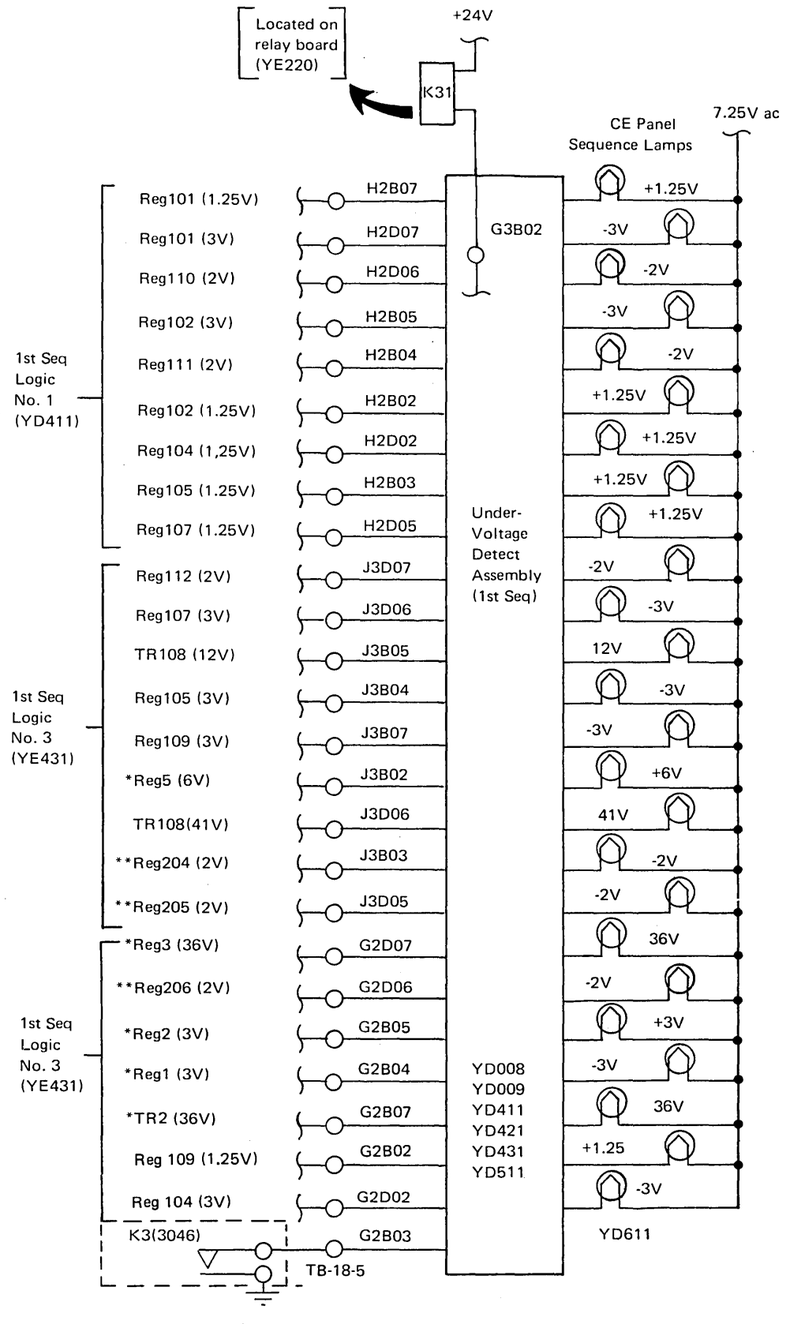
UNDERVOLTAGE SENSING SYSTEM, STAGE 1

- A voltage sensing system determines that all dc voltages are on and are supplying at least a certain minimum output.
- Failure in any dc supply causes power to be sequenced off and the power check light on the console and the CE panel to be turned on.

A special voltage sensing system monitors the output of all logic power supplies to determine that these supplies are on, and are supplying at least a certain minimum output. This system consists of an undervoltage detect card mounted on a small gate on the front-left side of the power frame, and two relays (K31 and K32) mounted on the relay sequence board that is located above the undervoltage detect card. (See YE016).

The logic supplies are divided into two groups: sequence 1 group, whose outputs are sensed by relay K31; and sequence 2 group, whose outputs are sensed by relay K32. Loss of any sensed voltage below the required minimum in either group causes the respective relay to drop (YE220). The transfer of K31 or K32 relay points causes one input to the fault detection AND circuit to open power check relay K12. Relay K14 de-energizes, which is the start of the power-down sequence. The console power check light and the CE panel power check light come on, but neither the red nor the white power-on light is on. To find which regulator is at fault, you can position CE2 switch at ERROR OVERRIDE, and CE6 switch to REG TEST. Turn power back on and monitor the regulator sequence lights. The absence of a lighted lamp represents a faulty regulator output. After the fault is detected, return all CE switches to their normal position.

To restore power after the fault is corrected, press the power-off key to reset the power check circuits. Then pressing power on causes a normal power-on sequence.



PROTECTION AND CHECKING CIRCUITS (CONTINUED)

Protection and checking circuits are provided in this system to prevent equipment damage and to simplify maintenance. During normal system operation, the protection circuits operate in conjunction with the control circuits to apply power sequentially to the system. In the event of a malfunction (such as overvoltage, or thermal), the protection circuits initiate a sequential power-down operation.

Checking circuits allow detection of a circuit that is functioning marginally, before it becomes a failure.

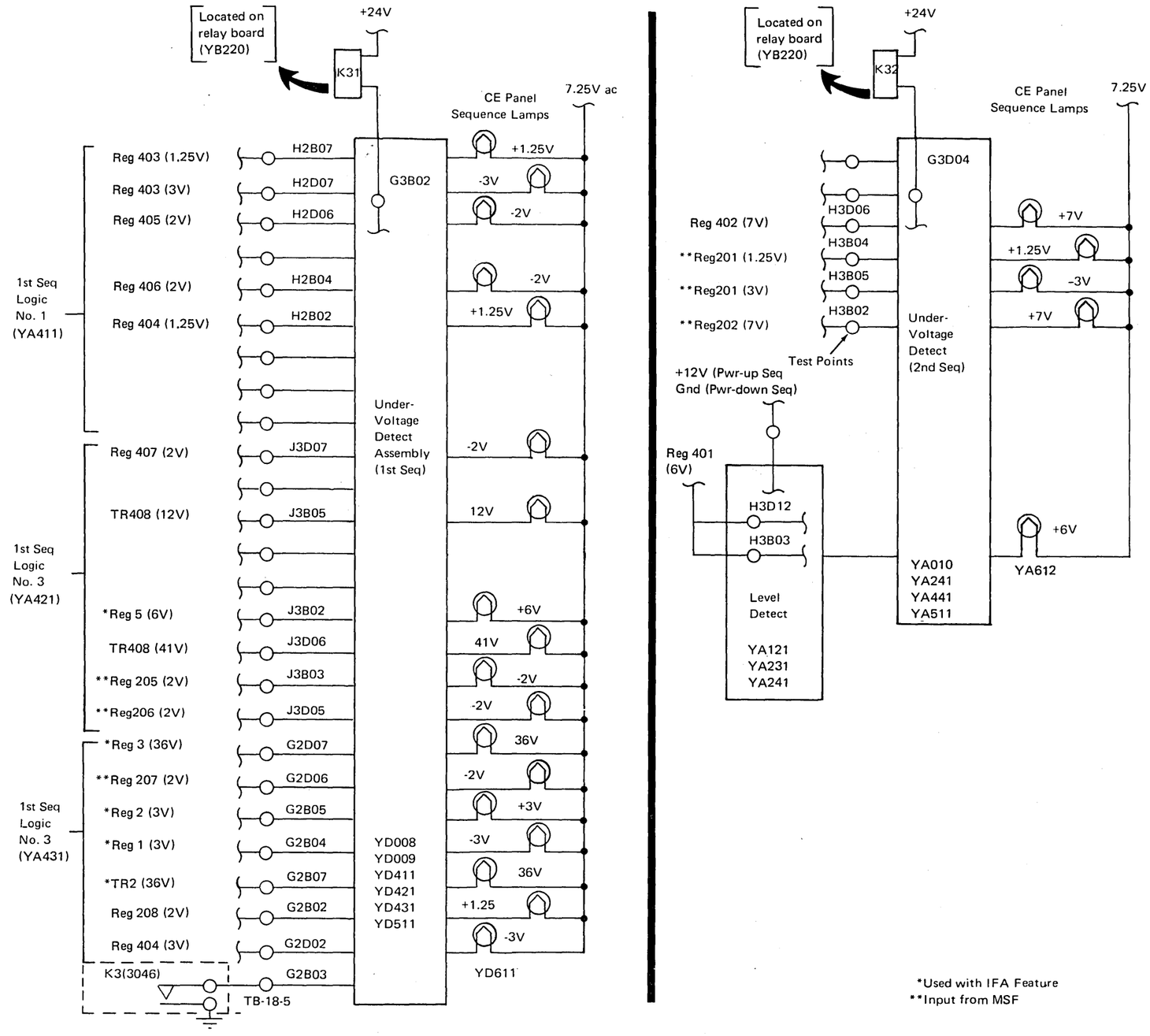
UNDERVOLTAGE SENSING SYSTEM, Stage 2

- A voltage sensing system determines that all dc voltages are on and are supplying at least a certain minimum output.
- Failure in any dc supply causes power to be sequenced off and the power check light on the console and the CE panel to be turned on.

A special voltage sensing system monitors the output of all logic power supplies to determine that these supplies are on, and are supplying at least a certain minimum output. This system consists of an undervoltage detect card mounted on a small gate on the front-left side of the power frame, and two relays (K31 and K32) mounted on the relay sequence board that is located above the undervoltage detect card. (See YE016).

The logic supplies are divided into two groups: sequence 1 group, whose outputs are sensed by relay K31; and sequence 2 group, whose outputs are sensed by relay K32. Loss of any sensed voltage below the required minimum in either group causes the respective relay to drop (YE220). The transfer of K31 or K32 relay points causes one input to the fault detection AND circuit to open power check relay K12. Relay K14 de-energizes, which is the start of the power-down sequence. The console power check light and the CE panel power check light come on, but neither the red nor the white power-on light is on. To find which regulator is at fault, you can position CE2 switch at ERROR OVERRIDE, and CE6 switch to REG TEST. Turn power back on and monitor the regulator sequence lights. The absence of a lighted lamp represents a faulty regulator output. After the fault is detected, return all CE switches to their normal position.

To restore power after the fault is corrected, press the power-off key to reset the power check circuits. Then pressing power on causes a normal power-on sequence.



OVERCURRENT, OVERVOLTAGE DETECTION, Stage 1

- Regulators used on this system have protection circuits that disable the regulator either by shorting the input (SCR circuit) or by an internal electronic shutoff circuit.
- Either protection device causes the system to power-down.

The MST regulators (101-109) are protected with an electronic shutoff circuit, and in some cases by an additional axe (SCR) circuit that disables the regulator for overvoltage/overcurrent conditions. The electronic shutoff circuit protects the regulator from short-duration current spikes whose amplitude is greater than the circuit breaker rating. The loss of voltage from a regulator due to an electronic shutoff is detected by the system undervoltage detect circuit. The axe (SCR) circuit, usually on the regulator input, shorts the bulk supply, which trips the bulk circuit breaker.

The Mid-Pac power supply (1, 2, 3, 5) is disabled for overvoltage/overcurrent conditions by removing the input bulk voltage by tripping the associated circuit breaker. An overvoltage condition in regulators 1, 2, and 5 causes an SCR to fire. The fired SCR provides a shorted load as the output from the regulator, causing excessive current drain. No overvoltage protection exists for regulator 3.

The phase-control regulators are protected for overvoltage conditions by circuit protector CP103. Overvoltage is sensed at the regulator outputs. An overvoltage condition fires an SCR that is connected to one pole of a three-pole circuit breaker. This pole contains a trip coil, causing CP103 to open. The trip coil is mechanically linked to the remaining two poles. The phase-control regulators are not internally protected against overcurrent. Overcurrent protection is provided by the primary input circuit protector for each regulator (CP105, 106, 107).

When a power fault is present as a result of an overvoltage/overcurrent condition, an input to the fault detection AND circuit is opened, turning off power check relay K12. Relay K12 drops power-on relay K14, which starts the power-down sequence. The console power check light and the CE panel power check light come on, but neither the red nor the white power-on light is on.

If the regulator is shut down as a result of a CB trip, the CB TRIP light and the light representing the frame at which the regulator is located illuminate. If the regulator output is turned off by the electronic shutoff, you can find which regulator turned off by placing CE2 switch at ERROR OVERRIDE. CE6 switch at REG TEST, restoring power to the system by pressing the power-on key, and monitoring the regulator sequence lights. The absence of a lighted lamp represents a faulty regulator output. The CE switches must be positioned to their normal position before the system is placed back into operation.

To restore power, correct the error condition and, if tripped, reset the CB only when power is off. Pressing the console power-off key or operating one of the check reset switches resets the power check circuits. Normal power-on can now be accomplished by pressing the power-on key.

POWER-SUPPLY RESPONSE to OVERVOLTAGE/ OVERCURRENT CONDITIONS, Stage 1

Regulator	Part No.	Voltage Level	Power Response to OV/OC Condition (See Note 1)	
			O/V	O/C
1	5762000	-3V	A	A
2	5762000	+3V	A	A
3	5762320	-36V	None	A
5	5761710	+6V	A	A
102	2557550	1.25V	C	A or B
		-3V	A(axe)	A or B
101	5797470	1.25V	B	A or B
		-3V	A	A or B
103	25727730	1.25V	C	A or B
		-3V	A(axe)	A or B
104	5797470	1.25V	B	A or B
		-3V	A	A or B
105	2557550	1.25V	C	A or B
		-3V	A(axe)	A or B
106	2572740	7V	A(axe)	A or B
107	2557550	1.25V	C	A or B
		-3V	A(axe)	A or B
108	2557470	6V	A(axe)	A or B
109	5797470	1.25V	B	A or B
		-3V	A	A or B
110	2572750	2V	A	A
111	2572750	2V	A	A

Note 1: Definitions of Power Supply Response to OC/OV Conditions

A— Regulator disabled by bulk circuit breaker

B— Regulator disabled by electronic shutoff

A or B— CB trips when current is above CB rating and below rating of electronic shutoff. Electronic shutoff disables regulator when current spikes are of short duration and greater than CB rating.

C— Regulator disabled by bulk circuit breaker and electronic shutoff.

OVERCURRENT, OVERVOLTAGE DETECTION, Stage 2

- Regulators used on this system have protection circuits that disable the regulator either by shorting the output (SCR circuit) or by an internal electronic shutoff circuit.
- Either protection device causes the system to power-down and results in a power-check indication.

The MST regulators (401, 402) are protected with an electronic shutoff circuit, and by an additional axe (SCR) circuit that disables the regulator for overvoltage/overcurrent conditions. The electronic shutoff circuit protects the regulator from short-duration-current spikes having amplitude greater than the circuit breaker rating. The loss of voltage from a regulator due to an electronic shutoff is detected by the system undervoltage detect circuit and results in a power-check indication.

The Mid-Pac power supply (1, 2, 3, 5) is disabled for overvoltage/overcurrent conditions by removing the input bulk voltage by tripping the associated circuit breaker. An overvoltage condition in regulators 1, 2 and 5 causes an SCR to fire. The fired SCR provides a shorted load as the output from the regulator, causing excessive current drain. No overvoltage protection exists for regulator 3.

Phase-control regulators 403 and 404 are protected by circuit protector CP406 for overvoltage conditions.

Phase-control regulators 405, 406, 407 are protected for overvoltage by CP403. Overvoltage is sensed at the regulator outputs. An overvoltage condition fires an SCR that is connected to one pole of a three-pole circuit breaker. This pole contains a trip coil, causing CP403 or CP406 to open. The trip coil is mechanically linked to the remaining two poles. The phase-control regulators are not internally protected against overcurrent. Overcurrent protection is provided by a primary input circuit protector to each phase-controlled regulator.

When a power fault is present as a result of an overvoltage/overcurrent condition, an input to the fault-detection AND circuit is opened, turning off power check relay K12. Relay K12 turns off the power-on relay K14, which starts the power-down sequence. The console power check light and the CE panel power check light come on, but neither the red nor the white power-on light is on.

If the regulator is shut down as a result of a CB trip, the CB TRIP light and the light representing the frame at which the regulator is located illuminate. If the regulator output is turned off by the electronic shutoff, you can find which regulator turned off by placing CE2 switch at ERROR OVERRIDE, CE6 switch at REG TEST, restoring power to the system by pressing the power-on key, and monitoring the regulator sequence lights. The absence of a lighted lamp represents a faulty regulator output. The CE switches must be positioned to their normal position before the system is placed back into operation.

To restore power, correct the error condition and reset the CB (if tripped). Pressing the console power-off key resets the power-check circuits. Normal power-on can now be accomplished by pressing the power-on key.

POWER-SUPPLY RESPONSE to OVERVOLTAGE/ OVERCURRENT CONDITIONS, Stage 2

Regulator	Part No.	Voltage Level	Power Response to OV/OC Condition (See Note 1)	
			O/V	O/C
1	5762000	-3V	A	A
2	5762000	+3V	A	A
3	5762320	-36V	None	A
5	5761710	+6V	A	A
403	2610400	1.25V	A	A
		-3V	A	A
404	2610400	1.25V	A	A
		-3V	A	A
402	2572740	7V	B	B
401	2557470	6V	B	B
405	2572750	2V	A	A
406	2610410	2V	A	A
407	2610410	2V	A	A

Note 1: Definitions of Power-Supply Response to OC/OV Conditions

A: Regulator disabled by bulk circuit breaker

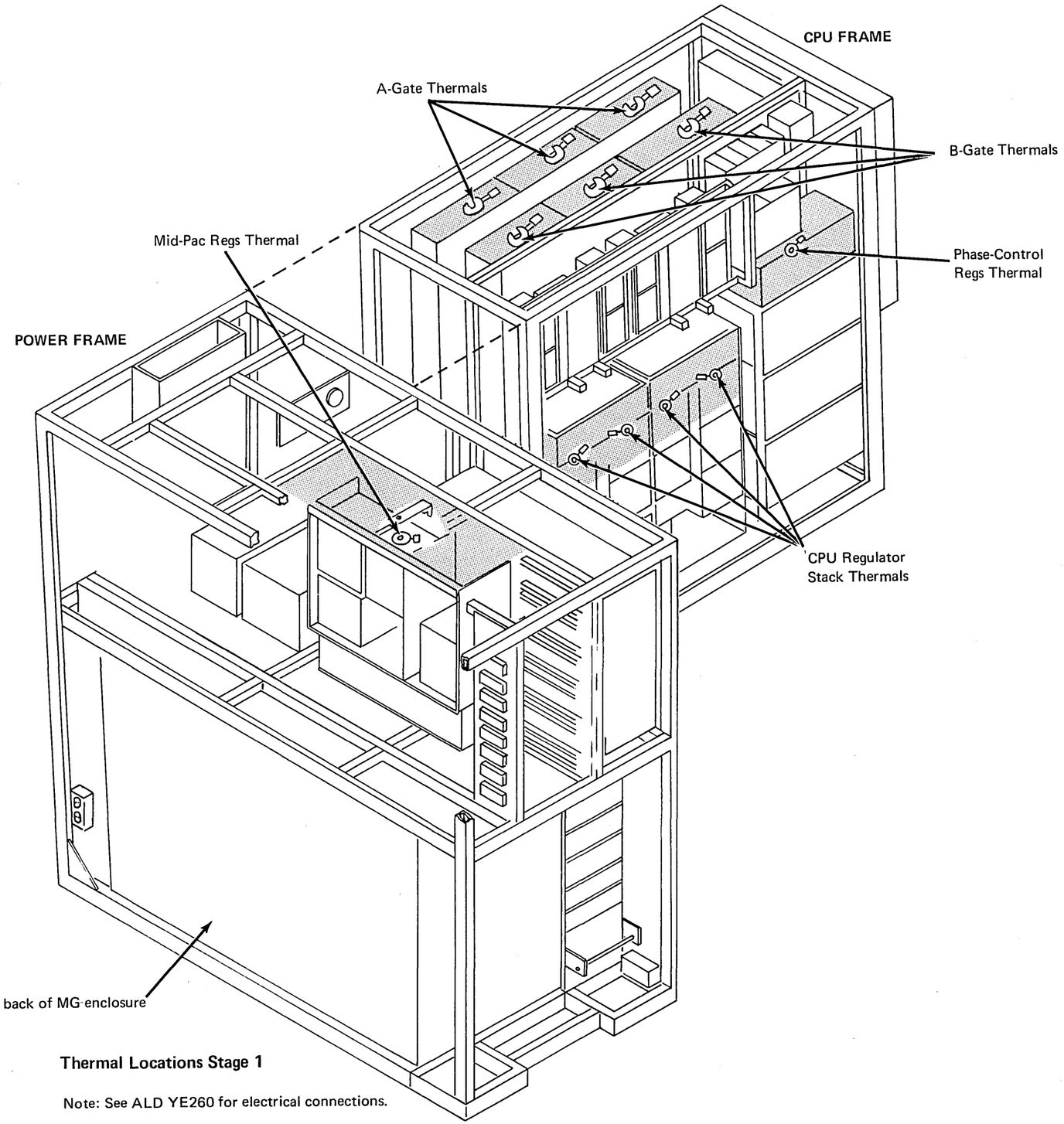
B: Regulator disabled by electronic shutoff

THERMAL SENSING, Stage 1

Thermal sensing switches provide overtemperature protection in both the CPU and the power frames. An overtemperature condition causes an associated overtemperature switch to transfer open. With a thermal switch open, relay K12 (power check) is dropped and a power-off sequence is initiated. A thermal condition is indicated by the illumination of the power check and Therm check lights on the CE panel and CPU console lights. When a thermal condition exists, check whether there is proper flow of air through the area (fans are running, filters are clean and not obstructed, and the exhaust area is not blocked). If the airflow is proper, locate and replace the component that is overheating. Also check the accuracy of the sensing elements; one may have changed the point at which it opens. To restore power after a thermal trip, the condition causing the thermal must be corrected, and the power-off key must be pressed in order to reset the power-check circuit.

Following is a list of the thermal locations and their high-temperature limits.

THERMAL LOCATION	TEMPERATURE LIMITS FAHRENHEIT	CENTIGRADE
PF-Above Mid-Pac regulators (1)	155°	68°
MG enclosure(1)	180°	82°
CPU-Top of B-gate(3)	155°	68°
CPU-Top of A-gate(3)	134°	56°
CPU Regulator stack assembly(4)	180°	82°
CPU-Phase-control blower assembly (1)	180°	82°



Thermal Locations Stage 1

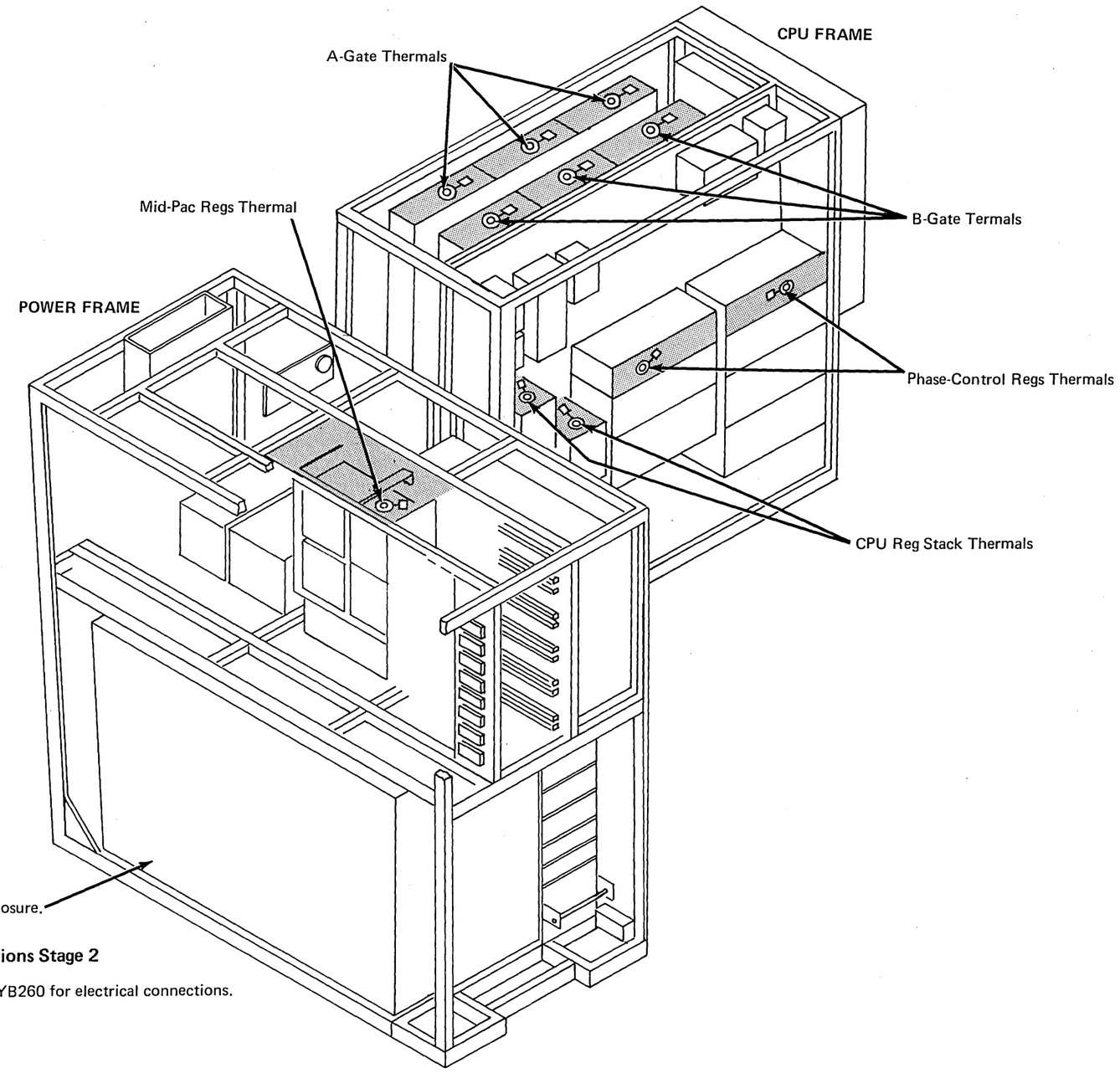
Note: See ALD YE260 for electrical connections.

THERMAL SENSING, STAGE 2

Thermal sensing switches provide overtemperature protection in both the CPU and the power frames. An overtemperature condition causes an associated overtemperature switch to transfer open. With a thermal switch open, relay K12 (power check) is turned off and a power-off sequence is initiated. A thermal condition is indicated by the illumination of the power check and therm check lights on the CE panel and CPU console lights. When a thermal condition exists, check whether there is proper flow of air through the area (fans are running, filters are clean and not obstructed, and the exhaust area is not blocked). If the airflow is proper, locate and replace the component that is overheating. Also check the accuracy of the sensing elements; one may have changed the point at which it opens. To restore power after a thermal trip, the condition causing the thermal must be corrected, and the power-off key must be pressed in order to reset the power-check circuit.

Following is a list of the thermal locations and their high-temperature limits.

THERMAL LOCATION	TEMPERATURE LIMITS FAHRENHEIT	CENTIGRADE
PF–Above Mid-Pac regulators (1)	155°	68°
MG enclosure(1)	180°	82°
CPU–Top of B-gate(3)	155°	68°
CPU–Top of A-gate(3)	134°	56°
CPU Regulator stack assembly(2)	180°	82°
CPU–Phase-control blower assembly(2)	180°	82°



Thermal Locations Stage 2

Note: See ALD YB260 for electrical connections.

POWER SERVICE CHECKS

When checking power, it is presumed that the primary power is within ±10 percent of the voltage specified on the voltage plate, and that the room temperature and relative humidity is within the specified tolerances.

Before making changes, resistance measurements, or replacements, be sure that all power is off and that all capacitors are fully discharged; do not rely on bleeder resistors. Remember that normal power-off does not turn off the convenience outlet power, EPO power, or power to transformer TR3. Set CB1 and CB4 (and CB2 in the 3046–if installed) off or turn off the primary wall power switch for both the 3145 and 3046 to remove these voltages.

VOLTAGE MEASUREMENTS and ADJUSTMENTS, Stage 1

This section contains tables of the dc voltages supplied within the system, test points at which the voltages can be monitored, and any applicable adjustment procedure. The tables provided are:

A CPU dc outputs for B-gate main-storage logic.

B CPU dc outputs for A- and B-gates.

C Power frame regulators.

D T/R special power requirements.

E Console-file voltages (supplied by power frame).

F IFA voltages (supplied by power frame).

G Printer/keyboard voltages (supplied by power frame).

H MG Regulator Adjustment.

I Motor generator output voltage adjustment.

Set all regulated voltages within the specified tolerances at the designated locations. With a digital voltmeter DIGITEC† 251-1 or equivalent, make all dc voltage measurements and adjustments that apply to CPU logic boards. Turn on the digital voltmeter for twenty minutes before taking any voltage measurements. For all AC voltage measurements, you can use a Weston†† 904 or equivalent.

A CPU DC Outputs for B-Gate Main Storage, Stage 1

Regulator	Logic Page	Rating	Nominal Volt Setting (Note 1)	Test Point (For Voltage Setting) (+) Lead	(-) Lead	*Tolerance Range at any Storage Board K2 Socket
103	YE300	1.25V @ 69A	+1.294 ± 1 mv	TB102-11	TB102-12	1.222 to 1.305
		-3V @ 69A	-3.087 ± 2 mv	TB102-7	TB102-8	-2.928 to -3.127
106	YE300	+7V @ 70A	+7.063 ± 5 mv	TB102-13	TB102-14	6.776 to 7.280
110	YE302	+2V @ 250A	Note 2	A2/A4 board K4B04 to D08		2.077 to 2.218
111	YE302	+2V @ 250A	Note 2	B2/B4 board K4B04 to D08		2.077 to 2.218
112	YE302	+2V @ 250A	Note 2	C2/C4 board K4B04 to D08		2.077 to 2.218

*Tolerance range includes the dc voltage measured with a digital voltmeter (Digitec 251-1 or equivalent), plus the ac ripple measured with an oscilloscope. The ac ripple measurement is valid only when the pulse width at 50% amplitude exceeds 20 nanoseconds.

NOTE 1: Set all voltages with storage in standby mode (not addressed).

NOTE 2: +2 Volt Setting

a. Measure +2V level on upper and lower storage boards (A, B, or C) at designated test-point location.
b. Determine the average for the two voltage readings. The result should be from 2.168 to 2.172 volts.
c. If the average reading exceeds 2.172 volts, slightly decrease the voltage at the regulator (use lower potentiometer on regularo card--R110-R112). If the average reading is less than 2.168 volts, slightly increase regulator voltage.
d. Repeat Steps b and c until the voltage falls into the desired range.

†Trademark of United Systems Corporation
††Made by Weston Instrument Division of Daystrom Inc., Daystrom Inc.

B CPU DC OUTPUTS for A- and B-GATES, Stage 1

Regulator	Logic Page	Rating	Nominal Volt Setting	Test Point (For Voltage Setting) Gate	Brd	Socket	Pin	Gnd	Feature	*Tolerance Range at any Board Pin
102	YE310	+1.25V @ 69A	Note 1	A	B1	M4	D03	D08	Basic	+1.212 to 1.278V
		-3V @ 69A	Note 1	A	B1	M4	B06	D08	Basic	-2.910V to -3.090V
101	YE310	+1.25V @ 69A	Note 1	A	B3	M4	D03	D08	Basic	+1.212V to 1.287V
		-3V @ 69A	Note 1	A	B3	M4	B06	D08	Basic	-2.910V to -3.090V
104	YE300	+1.25V @ 69A	Note 1	A	B2	M4	D03	D08	Basic	+1.212V to 1.287V
		-3V @ 69A	Note 1	A	B2	M4	B06	D08	Basic	-2.910V to -3.090V
107	YE301	+1.25V @ 69A	Note 1	A	B4	M4	D03	D08	Basic	+1.212V to 1.287V
		-3V @ 69A	Note 1	A	B4	M4	B06	D08	Basic	-2.910V to -3.090V
108	YE300	+6V @12A	6.000V ± 6 mv	A	*	TB101	5(+)	6(-)	Basic	+5.982V to 6.018V
109	YE303	+1.25V @ 69A	Note 1	B	A1	M4	D03	D08	SX4, CH-CH	+1.212V to 1.287V
		-3V @ 69A	Note 1	B	A2	M4	B06	D08	SX4,CH-CH	-2.910V to 3.090V
105	YE300	+1.25V @ 69A	Note 1	B	B3	M4	D03	D08	Basic	+1.212 to +1.287
	YE300	-3V @ 69A	Note 1	B	B3	M4	B06	D08	Basic	-2.910 to -3.090

*Located on hinge side of A-gate. To gain access to TB101, open both A- and B-gates.
**Tolerance range includes the dc voltage measured with a *digital voltmeter* (DIGITEC 252-2 or equivalent), plus the ac ripple measured with an oscilloscope.

Note 1: 1.25V AND 3.00V MEASUREMENT AND ADJUSTMENT PROCEDURE

The nominal value for the 1.25V and 3.00V regulators (101, 102, 105, and 109) is 1.250V ± 2 mv and -3.000 ±3 mv, respectively. The measured dc level at each board must be within ± 12 mv for the 1.25V regulator, and ± 30 mv for the -3.00V regulator. To comply with these requirements, do the following for each regulator.

1. Connect digital voltmeter to the regulator test points listed in the table.
2. Adjust associate regulator card located on the supply regulator, to the nominal value(s) (1.250V ± 2 mv and -3V ± 3 mv).
3. Measure dc differential as follows:
   a. Measure center of boards A, B, and C at proper row location.
      A Board A (row) L4D03; D08 Grd (1.25V) (VA)
      A (row) L4B06; D08 Grd (-3V) (VA)
      B Board B (row) L4D03; D08 Grd (1.25V) (VB)
      B (row) L4B06; D08 Grd (-3V) (VB)
      C Board C (row) L4D03; D08 Grd (1.25V) (VC)
      C (row) L4B06; D08 Grd (-3V) (VC)
4. Determine the difference between VB and VA, and BV and VC, for both the 1.25V and -3V outputs.
5. If the differential between VB and VA, or BV and VC, exceed 12 mv for the 1.25V regulator or 30 mv for the -3.00V regulator, proceed with step 6. If the difference is within the allowable tolerance, proceed to Step 10.
6. Determine voltage setting on board B for both the 1.25V and 3V by using the following formula:

$$VB^1 = VNOM + \frac{VDIFF}{2}$$

where: VB^1 = Adjusted nominal voltage setting at Board B
VNOM = 1.250V or -3.000V
VDIFF = VB-VA or VB-VC, whichever is greater.

Note: Observe the polarities at VA, VB, VC, VDIFF, and VNOM while solving equation to assure a correct voltage setting.

7. Connect digital voltmeter to test points located at Board B.
8. Adjust regualtor for both the 1.25V and -3V to the VB^1 value determined for each voltage level.
9. Repeat Steps 3 through 5.
10. Disconnect test equipment, secure the CPU gates, and close the cover over the voltage regulator cards.

POWER SERVICE CHECKS (Continued)

It is presumed that the primary power is within ±10 percent of the voltage specified on the voltage plate, and that the room temperature and relative humidity is within the specified tolerances.

Before making changes, resistance measurements, or replacements, be sure that all power is off and that all capacitors are fully discharged; do not rely on bleeder resistors. Remember that normal power-off does not turn off the convenience outlet power, EPO power, or power to transformer TR3. Set CB4 and CB1 off or turn off the primary wall power switch to remove these voltages.

VOLTAGE MEASUREMENTS and ADJUSTMENTS, Stage 2

This section contains tables of the dc voltages supplied within the system, test points at which the voltages can be monitored, and any applicable adjustment procedure. The tables provided are:

A CPU dc outputs for B-gate main-storage logic.

B CPU dc outputs for A-gate.

C Power frame regulators.

D T/R special power requirements.

E Console-file voltages (supplied by power frame).

F IFA voltages (supplied by power frame).

G Printer/keyboard voltages (supplied by power frame).

H MG regulator adjustment.

I Motor generator output voltage adjustment.

Set all regulated voltages within the specified tolerances at the designated locations. With a digital voltmeter (DIGITEC 251-1 or equivalent), make all dc voltage measurements and adjustments that apply to CPU logic boards. Turn on the digital voltmeter for twenty minutes before taking any voltage measurements. For all ac voltage measurements, you can use a Weston 904 or equivalent.

A CPU DC OUTPUTS for B-GATE MAIN-STORAGE LOGIC, Stage 2

Regulator	Logic Page	Rating	Nominal Voltage Setting (Note 1)	Test Point (for Voltage Setting)		*Tolerance Range at K2 socket on any storage board	*Tolerance Range at any Logic board
402	YB303	7V @ 70A	7.063 ± 5mv	TB402-13	TB402-14	6.776 to 7.280	NA
404	YB303	1.25V @250A	1.294 ± 1mv	TB-402-9	TB402-10	1.222 to 1.305	1.212 to 1.287
404	YB303	-3V @250A	-3.087 ±2mv	TB402-7	TB402-8	-2.928 to -3.127	2.910 to 3.090
405	YB302	2V @ 250A	Note 2	A2/A4 board K4B04 to D08		2.077 to 2.218	NA
406	YB302	2V @ 290A	Note 2	B2/B4 board K4B04 to D08		2.077 to 2.218	NA
407	YB302	2V @ 290A	Note 2	C2/C4 board K4B04 to D08		2.077 to 2.218	NA

*Tolerance ranges include supply drift, power supply noise, system differentials, and ac variations due to load changes. AC variations with pulse widths less than 20 ns at 50% of the peak amplitude are ignored on all memory boards. Supplies must be adjusted to setting tolerance but are allowed to drift 1%, at the test point.

Note: CP406 may trip on a U/V condition of regulators 405, 406, 407.

Note 1: Set all voltages with with storage in standby mode (not addressed).

Note 2: +2V setting.

A. Measure +2V level on upper and lower storage boards (A, B, or C) at designated test-point location.

B. Determine the average for the two voltage readings. The result should be from 2.168 to 2.172 volts.

C. If the average reading exceeds 2.172 volts, slightly decrease the voltage at the regulator (via voltage-adjust potentiometer). If average reading is less than 2.168V, slightly increases the regulator voltage.

D. Repeat Step B and C until the voltage falls into desired range.

B CPU DC OUTPUTS for A-GATE, Stage 2

Regulator	Logic Page	Rating	Nominal Voltage Setting	Test Point (for Voltage (+) lead	(-) lead	*Tolerance Range at any board pin
403	YB301	6V @ 12A	6.000 ±6mv	TB401-5	TB401-6	5.760 to 6.240V
403	YB301	1.25V @ 250A	1.283 ±1mv	TBA1-C1	TBA1-B1	1.212 to 1.287V
403	YB301	-3V @ 250A	3.061 ±2mv	TBA1-A1	TBA1-B1	-2.910 to 3.090V

*Tolerance range includes the dc voltage measured with a digital voltmeter (DIGITEC 251-1 or equivalent), plus the AC ripple measured with an oscilloscope. Supplies must be adjusted to setting tolerance but are allowed to drift 1%, at the test point (6V supply has 2% drift).

C POWER-FRAME REGULATORS, Stage 1

Regulator	Logic Page	Rating	Nominal Volt Setting	Test Point		Tolerance Range at Board Pins	Feature
				Plus	Gnd		
1	YE310	-3V @ 20A	-3.000V ±3mv	DB9-9	DB6-9	-2.88V to -3.12V	IFA
2	YE310	+3V @ 20A	+3.000V ±3mv	DB4-9	DB8-9	2.88V to 3.12V	IFA
3	YE310	-36V @ 2A	-36V ±36mv	DE11-9	DB6-9	34.36V to 37.44V	IFA
5	YE310	+6V @ 16A	+6.000V ±6mv	DB7-9	BB8-9	4.76V to 6.24V	IFA

D T/R SPECIAL DC POWER REQUIREMENTS, Stage 1

T/R	Logic Page	Rating	Nominal Volt Setting	Test Point (Power Frame)		Tolerance at Board Pins	Feature
				Plus	Gnd		
2	YE310	36V @ 12A	Not Adjustable	DB1-9	DB2-9	32.40V to 39.60V	IFA
*3	YE130	24V @ 11A	Not Adjustable	TB14-7	TB14-8	21.60V to 26.40V	Basic
		12V @ 3A	Not Adjustable	C3(+)	C3(-)	10.80V to 13.20V	Basic
*108	YE155	12V @ 2.5A	12.0V ±12mv	TB14-11	TB14-11	10.80V to 13.30V	Basic

NOTE: *Test points for TR3 and TR108 are located in the primary power compartment.

E CONSOLE-FILE VOLTAGES (SUPPLIED by POWER FRAME), Stages 1 and 2

To measure voltage to the console file, place the diagnostic console file control rotary switch to CE MODE. For voltage locations on the console file, refer to *Theory-Maintenance Manual, IBM 23FD Disk Drive,* SY26-4175.

(Measure voltages with DIGITEC 251-1 or equivalent).

Cable Connector: J62

PIN	VOLTAGE
5	+24V dc
4	24V Return
2	-3V dc
6	-3V Return
1	+6V dc
3	+6V Return

NOTE: There is no adjustment for ac or dc voltages to the console file.

F IFA VOLTAGES (SUPPLIED by POWER FRAME), Stages 1 and 2

(Measure ac voltages with Weston 904 or equivalent).

Cable Connector: DRA through DRH

PIN	VOLTAGE
2	-3V
3	-3V Return
13&14	+3V
3&4	+3V Return
15	-36V
3	-36V Return
1	+6V
3&4	+6V Return
19&20	+36V
10&11	+36V Return

Cable Connector: J63

PIN	VOLTAGE
1-3	208/230V ac, 60 Hz, 3-Phase (220/380V ac, 50 Hz, 3-] hase)
4	Frame Grd
5	W/T Neutral

C POWER-FRAME REGULATORS Stage 2

Regulator	Logic Page	Rating	Nominal Volt Setting	Test Point		Tolerance Range at Board Pins	Feature
				Plus	Gnd		
1	YB310	-3V @ 20A	-3.000V ±3mv	DB9-9	DB6-9	-2.88V to -3.12V	IFA
2	YB310	+3V @20A	+3.000V ±3mv	DB4-9	DB8-9	2.88V to 3.12V	IFA
3	YB310	-36V @ 2A	-36V ±36mv	DE11-9	DB6-9	34.36V to 37.44V	IFA
5	YB310	+6V @ 16A	+6.000V ±6mv	DB7-9	BB8-9	4.76V to 6.24V	IFA

D T/R SPECIAL DC POWER REQUIREMENTS, Stage 2

T/R	Logic Page	Rating	Nominal Volt Setting	Test Point (Power Frame)		Tolerance at Board Pins	Feature
				Plus	Gnd		
2	YE310	36V @ 12A	Not Adjustable	DB1-9	DB2-9	32.40V to 39.60V	IFA
*3	YE130	24V @ 11A	Not Adjustable	TB14-7	TB14-8	21.60V to 26.40V	Basic
		12V @ 3A	Not Adjustable	C3(+)	C3(-)	10.80V to 13.20V	Basic
*408	YE155	12V @ 2.5A	12.0V ±12mv	TB14-11	TB14-11	10.80V to 13.30V	Basic

NOTE: *Test points for TR3 and TR408 are located in the primary power compartment.

G PRINTER-KEYBOARD VOLTAGES (SUPPLIED by POWER FRAME), Stages 1 and 2

(Measure ac voltages with Weston 904 or equivalent).

3210 Mode 1

Cable Connector: J60

PIN	VOLTAGES
1&2	115V ac 60 Hz (220V ac, 50 Hz)

3215 Printer

Cable Connector: J1

PINS	VOLTAGE
1	+24V
2	24V RETURN
3	+24V
4	24V RETURN
8	-3V
9	+6V
10	6V & -3V RETURN
11	+6V
12	6V RETURN

3215 Printer

Keyboard Power

Cable Connector: KC

PIN	VOLTAGE
R	+6V
T	24V RET
U	6V RET
V	-3V RET
W	+24V
X	-3V

Note: There is no adjustment for ac or dc voltages to the printer/keyboard(s).

MOTOR GENERATOR REGULATOR OVER–VOLTAGE ADJUSTMENT, Stages 1 and 2

NOTES:

1. Apply power to the CPU a minimum of 30 minutes before performing this adjustment.
2. Plug digital voltmeter into receptacle on power frame and allow at least a 20-minute warm up time.

1. Open the covers on left side of CPU frame.
2. Open the covers on power frame and remove MG regulator cover.
3. Connect a Weston 904 (or equivalent) to any two input lines on the phase-control regulator input terminal block.
4. Bring up system power.
5. Place digital voltmeter (DIGITEC–Model 251A/251-1 or equivalent) alongside of the power frame so that the voltage reading can be observed while adjusting the MG voltage-adjust potentiometer.
6. Connect banana leads from MG meter converter (Part 2637491) to input terminals (Lo-Hi) on digital voltmeter.
7. Plug 208V, 3-phase, 400 Hz input plug from MG meter converter to receptacle labeled MG TEST POINT on MG voltage tester adapter. (The adapter is on the upper right side of the CPU frame.) Wait a minimum of five minutes to allow the conversion circuits to stabilize.
8. Check the digital voltmeter calibration pins' zero settings. Make any required adjustments and report checks.
9. Set digital voltmeter range to 10V.
10. Adjust MG voltage-adjust potentiometer until the DIGITEC meter reading is within the voltage range specified on the MG meter converter.
11. Adjust the Weston 904 to 208V ac, by needle-zero adjusting screw on face of meter. You have now calibrated your ac meter.
12. Turn off MG power.
13. Disconnect Weston 904 from phase-control regulator input terminal block.
14. Turn O/V ADJ (overvoltage trip) control fully clockwise.
15. Set CB2 and CB5 off. This disconnects the MG output.
16. Position the CE switch No. 2 to ERROR OVERRIDE; and CE switch No. 3 to MG HOLD.
17. Connect the Weston 904 to input terminals of K-26, located within primary power box for phase-to-phase measurements.
18. Apply ac power to motor generator.
19. Record the voltage reading on the Weston 904 at this time.
20. Advance VOLT ADJ control so that phase-to-phase voltage reads 220V ac.
21. Slowly turn the O/V ADJ counterclockwise until relay K-60 contacts open. At this point, system will power-down.
22. Remove power from MG to reset K-60. Allow two to three minutes for the O/V circuit to reset.
23. Turn VOLT ADJ control about four to five turns counterclockwise.
24. Apply AC power to motor generator.
25. With VOLT ADJ control, adjust MG output for the value recorded in Step 19 (measured at K-26 input terminals.)
26. Power down.
27. Set CB2 and CB5 on, and restore CE switch 2 to NORMAL, and CE switch 3 to MG PWR OFF controlled.
28. Remove voltmeter leads from K-26 terminals; re-zero meter.
29. Perform motor generator output voltage adjustment.

MOTOR GENERATOR OUTPUT VOLTAGE ADJUSTMENTS

NOTE: Apply power to the CPU for a minimum of 30 minutes before performing this adjustment.

1. Open the covers on the left side of CPU Frame.
2. Open the covers on the power frame and remove the MG regulator cover.
3. Place digital voltmeter (DIGITEC model 251A/251-1 or equivalent) alongside of power frame so that the voltage from the voltmeter can be observed while you adjust the MG voltage-adjust potentiometer.
4. Connect banana leads from MG Meter converter (part 2637491) to input terminal (Lo-Hi) on digital voltmeter.
5. Plug 208V, 3-phase, 400 Hz input plug from MG meter converter to receptacle labeled MG TEST POINT on MG voltage tester adapter. (The adapter is located on upper right side of CPU Frame.) Wait a minimum of five minutes to allow conversion circuits to stabilize.
6. Plug the digital voltmeter into the receptacle on the power frame and allow at least a five-minute warm-up time.
7. Check digital voltmeter calibration and zero settings. Make any required adjustments and repeat check.
8. Set digital voltmeter range to 10V.
9. Compare the voltage reading on the voltmeter with the dc voltage stamped on the MG meter converter.
10. If the measured voltage is within the voltage tolerance specified, proceed to Step 15. If the measured voltage is not within the voltage tolerance specified, proceed with Step 11.
11. Adjust the MG voltage-adjust potentiometer until the digital voltmeter reading is within the voltage range specified on the MG meter converter. If the MG output voltage is adjusted within the voltage range specified, proceed to Step 15. If the MG output voltage cannot be adjusted within the voltage range specified, proceed with Step 12.
12. Check fuses in MG voltage tester adapter. If any of the fuses are blown, replace and repeat Steps 9 through 11. If the fuses are not blown, continue with Step 13.
13. Connect Weston 904 or equivalent to any two input lines on the phase-control regulator input terminal block.
14. Monitor the Weston 904 and vary the MG voltage-adjust potentiometer for a motor generator output of 208V. If 208V is measured on the Weston voltmeter, either the MG meter converter or the digital voltmeter is out of tolerance and requires calibration. Proceed to Step 15 for termination of test. If 208V is not measured on the Weston 904 voltmeter, a problem may exist in the MG regulator. Refer to MG troubleshooting guide.
15. Disconnect the plug from the MG voltage tester adapter, disconnect remaining test equipment from system, and secure all doors.

REGULATORS, STAGES 1 and 2

There are various kinds of regulators used on the Model 145 that rectify and regulate the input voltage to a particular dc voltage at the needed amperage for distribution.

If possible, limit maintenance to the following.

1. Card replacement.
2. Output-voltage adjustment.
3. Cleaning and checking for loose connections.
4. Airflow checks.

Be sure that all power is off and that capacitors are fully discharged; do not rely on the bleeder resistors.

Following is a suggested sequence to use when a voltage regulator is causing a problem.

1. Replace the overcurrent and regulator cards.
2. Check for proper bias on the voltage regulator.
3. Check for proper connections of remote sensing.
4. Check input voltage.
5. Make sure that all terminals are tight.
6. Replace the voltage regulator.

REGULATOR REMOVAL and REPLACEMENT

DUAL-LEVEL SUPPLIED (REGULATORS 101, 102, 104, 105, 107, and 109), Stage 1

1. All power on the frame must be off before a supply can be removed. Press power-off key and open circuit breakers CB1 and CB4 and CB2 within the 3046 (if installed).
2. Open the covers on left side of CPU frame.
3. Remove T-shaped plastic retainer from above and below the supply.
4. Remove voltage leads from E1 through E4, E7 through E14, and E15. Make sure that you mark the leads for proper reconnection.
5. Slide out the supply.
6. To replace the supply, insert the base of the supply into mounted channels and slide the supply into place.
7. Reconnect leads that have been removed in Step 4 and replace plastic retainers.
8. Close CB1 and CB4 and turn on the power.
9. Check the output voltages for the changed supply and adjust if necessary. (See the voltage measurement charts.)

DUAL-LEVEL SUPPLIES (REGULATOR 103), Stage 1

1. All power on the frame must be off before a supply can be removed. Press power-off key and open circuit breaker CB1 and CB4.
2. Open the covers on left side of CPU frame.
3. Remove T-shaped plastic retainers from above and below supply.
4. Remove voltage leads from E1 through E4, E7 through E11, E13, and E15. Make sure that you mark the leads for proper reconnection.
5. Remove wires from TB1, 3 and 4.
6. Slide out the supply.
7. To replace the supply, insert base of the supply into the mounted channels and slide the supply into place.
8. Reconnect leads that have been removed in Steps 4 and 5 and replace plastic retainers.
9. Close CB1 and CB4, and turn on the power.
10. Check the output voltage for the supply and adjust if necessary. (See voltage measurement charts.)

SINGLE-LEVEL SUPPLY (REGULATOR 106 and 108), Stage 1

The procedure for removing and replacing a single-level supply is the same as that for the dual-level supplies except for the removal of the voltage leads. Therefore, follow the procedure for the dual-level supply but replace Step 4 with the following.

4. Remove voltage leads from E1 through E4, E9, E10, E12 through E14.

Phase-Control Regulators (Regulators 110, 111, 112)

Replaceable Parts (Field) Regulators 110, 111, 112

1. Control card
2. Control assembly
3. Entire supply

Replacement of Control Assembly (Regulators 110, 111, 112)

1. Remove cover (over control card).
2. Remove two mounting screws on front of supply, one above TB1 and one below E14.
3. Remove cabling from TB1 and slip-ons from rear of supply.
4. Retain SLT control card.
5. Install SLT control card in new control assembly.
6. Install control assembly and secure with mounting screws.
7. Check the active cap adjustment and readjust if necessary.

Replacement of Control Card (Regulators 110, 111, 112)

1. Loosen two screws and remove the card cover.
2. Open the card retainer.
3. Replace card.
4. See "Active Cap Adjust Procedure."

Replacement of Entire Supply (Regulators 110,111, 112)

1. Remove the cabling to and from the supply.
2. Remove four mounting screws.
3. Install the new supply.
4. Attach cables.
5. Adjust output voltage.

REGULATOR CARD REPLACEMENT, Stage 1

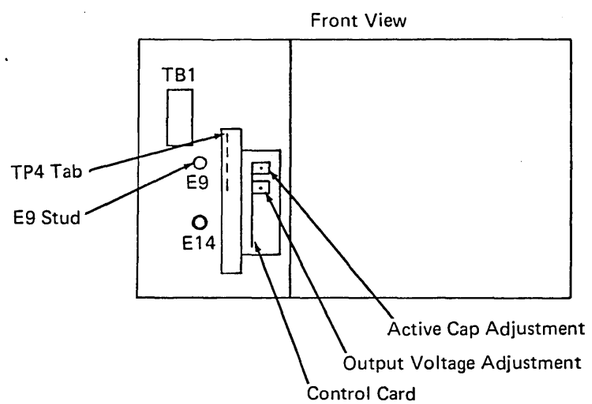
1. To replace a card within a regulator, press the power-off key and remove power to the regulator. This may be done by turning off power to the entire system (open CB1 and CB4) or by turning off power to the one supply (by opening the input circuit protector for the supply). See Power Conversion and Distribution charts for regulators and its associated circuit protector.
2. Open the cover of the frame in which the regulator is located.
3. Release the card and retainer (if applicable) and remove with a card puller remove the card from the regulator.
4. Insert new card.
5. If complete power was turned off, close CB1 and CB4 and bring power back up using one of the power-on switches.
6. If power was turned off only to the power supply, close the input circuit protector.
7. If the CE panel power check light is on, press the power-off key on the control console, or the check reset on the console or CE panel.
8. Turn on the power by using one of the power-on switches.
9. Check the voltage controlled by the replaced card and adjust if necessary. (See the voltage measurement charts.)

MID-PAC REGULATORS (REGULATORS 1, 2, 3, and 5), Stages 1 and 2

1. All power on the frame must be dropped before a supply can be removed. Press power-off key and open the circuit breaker CB1 and CB4.
2. Open covers at back side of power frame.
3. Remove screws (two) above and below regulator, and remove plastic shield.
4. Remove wires from TB1–1 through 9, and mark.
5. Slide the regulator out and replace with a new unit.
6. Reconnect the lead removed in Step 4.
7. Replace the plastic shield and secure the regulator to channel with two screws removed in Step 3.
8. Close CB1 and CB4, and turn on the power.
9. Check the output voltage from the changed regulator and make any necessary adjustments. (See voltage measurement charts.)

ACTIVE CAP ADJUSTMENT PROCEDURE (REGULATORS 110-112), Stage 1

1. Adjust E out of supply to nominal.
2. Attach digital voltmeter (+) lead to TP4 (see sketch) and (-) lead to E9. Meter should read 200 mv±6 mv; if not remove adhesive from active cap pot on SLT card and readjust.
3. To prevent accidental readjustment of the active cap apply Epoxy, part 483002 to potentiometer.
4. Adjust output to nominal. (Refer to illustration at right.)



REGULATOR REMOVAL AND REPLACEMENT, Stage 2

PHASE-CONTROL REGULATORS (REGULATORS 405, 406, 407), Stage 2

REPLACEABLE PARTS (FIELD) (REGULATORS 405-407)

1. Control card
2. Control assembly
3. Entire supply

Replacement of Control Assembly (Regulators 405-407

1. Remove cover (over control card).
2. Remove two mounting screws on the front of the supply, one above TB1 and one below E14.
3. Remove cabling from TB1 and slip-ons from rear of supply.
4. Retain SLT control card.
5. Install SLT control card in new control assembly.
6. Install control assembly and secure with mounting screws.
7. Check active cap adjustment and readjust if neccessary.

Replacement of Control Card (Regulators 405-407)

1. Loosen two screws and remove the card cover.
2. Open card retainer.
3. Replace card.
4. See "Active Cap Adjust Procedure."

Replacement of Entire Supply (Regulator 405)

1. Remove the cabling to and from supply.
2. Remove four mounting screws.
3. Install new supply.
4. Attach cables.
5. Adjust output voltage.

Replacement of Entire Supply (Regulator 406)

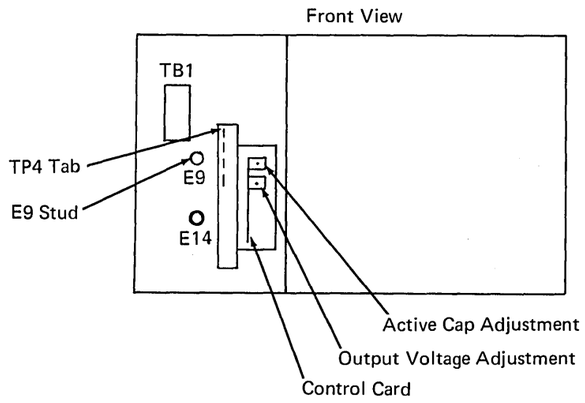
1. Remove cabling.
2. Remove four mounting screws.
3. Install new supply.
4. The new supply is wired for high current (165-290A) output. If the machine has 12K memory installed, the supply must be rewired for low current output (90-175A). (See "Current Tap Adjust Procedure"; Page 11-36.)

Replacement of Regulator 407

1. Same procedure as for regulator 406 entire replacement.
2. The new supply is wired for high current (165-290A) output. If the machine has 208K memory, the installed supply must be rewired for low current output (90-175A). (See "Procedure to Change Current Taps.")

Active Cap Adjustment Procedure (Regulators 405-407), Stage 2

1. Adjust E out of supply to nominal.
2. Attach digital voltmeter (+) lead to TP4 and (-) lead to E9. Meter should read 200 mv±6 mv; if not remove adhesive from active cap pot on SLT card and readjust.
3. To prevent accidental readjustment of the active cap, apply RTV 3.45 selastic adhesive (Part 2557523) to potentiometer.
4. Adjust output to nominal.



PHASE-CONTROL REGULATOR DUALS (REGULATORS 403 and 404), Stage 2

Replacement Parts

1. Control card
2. Control assembly
3. Entire supply

Replacement of Control Assembly

1. Remove wiring from TS1 and slip-ons from rear of supply.
2. Remove and save control card.
3. Remove four mounting screws from control assembly and remove.
4. Replace with new control assembly and replace control card.
5. Adjust output voltage if necessary.

Replacement of Control Card, Stage 2

1. Remove (2) screws retaining plastic cover over control card.
2. Remove card and replace.
3. Check output voltage and adjust if necessary.

Replacement of Regulator 403

1. Remove four mounting screws on front of supply.
2. Remove wiring to front and back of supply.
3. Remove supply and replace.
4. Attach cables removed in Step 1.
5. Adjust supply output voltage if necessary.

Replacement of Regulator 404

1. Follow steps 11A through C.
2. The replacement supply is wired for high current output (130A-260A). The supply must be rewired for low current output (60-190A). See the procedure for changing current tops.
3. Attach cables removed in Step A.
4. Check the output voltage setting and readjust if required.

REGULATOR 404 HIGH AND LOW OUTPUT CURRENT TAPS, STAGE 2

A. Regulator 404 should always be wired for low current output (60-190A).

1. Remove four screws securing the large plastic cover over the front of supply.
2. Locate TB4 mounted inside regulator.
3. Locate cable wiring between TB1 and TB4 inside the supply. It is the leads of this cable that are to be moved.
4. Move leads as follows.
   TB4-2 move to TB4-1
   TB4-5 move to TB4-4
   TB4-8 move to TB4-7
5. Reinstall the large plastic cover.
6. Install label, Part 2631505, over old label on large plastic cover. With a black marking pen indicate the current range of supply to be (60A-190A).
7. Check the output voltage of supply and adjust if required.

REGULATORS 406 AND 407 HIGH AND LOW OUTPUT CURRENT TAPS, STAGE 2

A. Regulator 406 is wired for high current output except when a 112K memory is installed.

B. Regulator 407 is wired for high current output except when a 208K memory is installed.

1. To change the supply to a low current output, use same steps as for regulator 404.
2. Install label, Part 2631505, over old label on large plastic cover. With black marking pen, indicate the current range of supply to be wired for (90A-175A).
3. Check the output voltage of supply and adjust if required.

NOTE: Regulators 404, 406, and 407 current taps (see ALD YB309).

Regulators 401 and 402 Card Replacement, Stage 2

1. To replace a card within a regulator, press the power-off key.
2. Open the cover of the frame in which the regulator is located.
3. Release the card and retainer (if applicable), and with a card puller remove the card from the regulator.
4. Insert a new card.
5. If the CE panel power check light is on, press the power-off key on the control console, or the CHECK RESET on the CE panel.
6. Press POWER ON.
7. Readjust the voltage controlled by the replaced card. (See "Voltage Measurements and Adjustments.")

SINGLE-LEVEL SUPPLIES (REGULATORS 401 and 402)

1. All power on the frame must be turned off before a supply can be removed. Press the power-off key and open the circuit breakers CB1 and CB4 and CB2 within the 3046 (if installed).
2. Open the covers on the left side of the CPU frame.
3. Remove the T-shaped plastic retainer from above and below the supply.
4. Remove the voltage leads from E1 through E4, E9, E10, E12 through E14. Make sure that you mark the leads for proper reconnection.
5. Slide out the supply.
6. To replace the supply, insert the base of the supply into the mounted channels and slide the supply into place.
7. Reconnect leads that have been removed in Step 4 and replace the plastic retainers.
8. Close CB1 and CB4 and turn on the power.
9. Check the output voltages for the changed supply and adjust if necessary. (See the voltage measurement charts.)

AC/DC MODULE REMOVAL and REPLACEMENT, Stage 1

1. Remove all power to the system by opening the circuit breakers CB1 and CB4.
2. Open the cover on the left side of the CPU frame.
3. Remove wires from:
   a. TB1-1 and mark
   b. TB2-1, 2, and 3 and mark
   c. E2 and mark
4. Open the covers and gates A and B on the right side of CPU frame.
5. Remove back panel located behind ac/dc modules (upper right hand side of frame).
6. Remove two leads from E1 and E2 and mark.
7. Slide ac/dc module out on left side of CPU frame and replace with new unit.
8. Reconnect leads that have been removed in Steps 3 and 6.
9. Replace back panel located behind the ac/dc module, and close gates and covers.
10. Close CB1 and CB4 and bring up power.

BLOWERS and FILTERS

Blowers and filters are located as follows:

Location	*Blower Assembly*
CPU-A Gate	3 w/filters
CPU-B Gate	3 w/filters
CPU Regulator Stack	2—no filter
CPU Phase Ctl Reg	1—no filter
CPU F Gate (mixer board)	1 w/filter
PF Mid Pac Reg Stack	1 w/filter

The blower assemblies require no lubrication. Inspect periodically for noisy operation or other malfunctions and replace when found defective. Inspect filters for dirt and replace when dirty.

CPU BLOWER ASSEMBLY and FILTER REMOVAL and REPLACEMENT, Stage 1

Blower Assembly

1. Open the cover on the right side of the CPU frame and release the gate.
2. From the pin side of the gate, disconnect the electrical connector.
3. From the card side of gate, remove two screws on each end of blower.
4. Secure blower with one hand (hold bottom of blower) and remove center screw.
5. Tilt blower down slightly, and pull toward you.

To replace blower, reverse the removal procedure.

Filter

The filter is located at the bottom of the blower housing and is removed from the card side of gate.

1. Place the thumb of each hand through the finger cutouts, provided in the housing, against the filter.
2. With equal pressure, push filter toward the pin side of the gate until clear of housing.
3. Pivot the filter down and pull toward you.
4. To replace the filter, reverse the removal procedure.

CPU REGULATOR STACK BLOWER ASSEMBLY REMOVAL and REPLACEMENT, Stage 1

1. Open the cover on the left side of the CPU frame.
2. Disconnect the electrical connector located at front center of housing.
3. Remove two screws on the right side of the blower housing.
4. Lift from right side of blower housing to disengage housing from slot, and lift out.

To replace the blower, reverse the removal procedure.

CPU PHASE-CONTROL REGULATOR BLOWER REMOVAL and REPLACEMENT, Stage 1

1. Open the cover on the left side of the CPU frame.
2. Disconnect the electrical connector located at the front-left corner.
3. Disconnect the thermal leads from the blower at the knife connector.
4. Remove four screws holding the blower assembly to the frame (one screw is located at each corner).
5. Lift the blower assembly out of CPU frame.

To replace the blower, reverse the removal procedure.

CPU F-GATE (MIXER BOARD) BLOWER ASSEMBLY and FILTER REMOVAL and REPLACEMENT, Stage 1

Blower Assembly

NOTE: The blower assembly is secured to frame via a cam-action-type latch.

1. Open the cover on the right side of the CPU frame and release the gate latch. Pull the gates away from frame.
2. Disconnect the electrical connector located on opposite side of latch.
3. Release the latch and pull out the blower assembly and away from latch.

Filter

NOTE: The filter fits into a channel on one end of the blower housing and is secured by a bullet-type catch. To remove:

1. Place finger in cutout at each side of the housing.
2. Pull filter straight down until the housing is cleared.
3. To replace, push filter up into housing until the bullet-type catch secures filter.

POWER FRAME MID-PAC REGULATOR STACK BLOWER and FILTER REMOVAL and REPLACEMENT, Stage 1

Blower Assembly

1. Open the cover on the back side of the power frame.
2. Disconnect electrical connection located at the center of blower assembly.
3. Loosen four mounting screws (two on each side of assembly) that secure blower to the center panel.
4. Support the blower so that it cannot drop on the frame and remove the four mounting screws.

To replace, reverse the removal procedure.

Filter

Remove the filter by first disconnecting the electrical connection at the center of the blower assembly and then sliding the filter out away from the assembly.

MOTOR GENERATOR REMOVAL and REPLACEMENT, Stages 1 and 2

1. Remove all power to the system.
2. Open the cover on the back side of power frame.
3. Remove two bolts on each side of MG that secure the MG enclosure to the power frame.
4. Back the MG enclosure from power frame until the power cables become accessible.
5. Remove safety cover from cable connectors.
6. Remove three-400 Hz input cables, three-50/60 Hz cables, and 1 frame ground cable. Mark all cables.
7. Remove two wires from the overvoltage detect input terminals, and two wires from the overtemperature terminals.
8. Back MG away from power frame and replace with new MG unit.
9. Replace wires and cables removed in Steps 6 and 7.
10. Replace safety cover over cable connectors and slide MG enclosure back into power frame.
11. Align MG channel with power frame and secure MG with bolts removed in Step 3.
12. Return power to system.

NOTE: If power does not cycle up after five seconds, check MG input leads for proper plus rotation. (Relay K1 will not pick if phase rotation is incorrect.) If relay K1 is not energized, reverse the phasing to MG.

13. Perform the motor-generator-output-voltage adjustment.

AC/DC MODULE REMOVAL and REPLACEMENT, Stage 2

1. Press the power-off key.
2. Remove all power to system by opening circuit breakers CB1 and CB4.
3. Open the cover on the left side of CPU frame.
4. Remove wires from:
   a. TB2-1, 2, and 3 and mark.
   b. E2 and mark.
5. Open covers and gates A and B on the right side of CPU frame.
6. Remove the back panel located behind ac/dc modules (upper right-hand side of frame).
7. Remove two leads from E1 and E2, and mark.
8. Slide ac/dc module out on left side of CPU frame and replace with new unit.
9. Reconnect leads that have been removed in Steps 3 and 6.
10. Replace the back panel located behind the ac/dc module and close gates and covers.
11. Close CB1 and CB4, and bring power up.

BLOWERS and FILTERS

Blowers and filters are located as follows.

Location	*Blower Assembly*
CPU A-Gate	3 w/filters
CPU B-Gate	3 w/filters
CPU Regulator Stack	1 w/filter
CPU 2V Phase Ctl Reg	1—no filter
PF Mid-Pac Reg Stack	1 w/filter
CPU Dual Phase CR Stack	1—no filter

The blower assemblies require no lubrication. Inspect them for noisy operation or other malfunctions and replace when found defective. Inspect filters for dirt and replace when dirty.

CPU GATE BLOWER ASSEMBLY and FILTER REMOVAL and REPALCEMENT, Stage 2

Blower Assembly

1. Open the cover on the right side of the CPU frame and release the gate.
2. From the pin side of the gate, disconnect the electrical connector.
3. From the card side of the gate, remove screws on each end of blower.
4. Secure the blower with one hand (hold bottom of blower) and remove center screw.
5. Tilt the blower down slightly, and pull toward you.

To replace blower, reverse the removal procedure.

Filter

The filter is located at the bottom of the blower housing and is removed from the card side of gate.

1. Place the thumb of each hand through the finger cutouts, provided in the housing, against the filter.
2. With equal pressure, push the filter toward the pin side of the gate until clear of housing.
3. Pivot the filter down and pull toward you.
4. To replace filter, reverse the removal procedure.

CPU REGULATOR STACK BLOWER ASSEMBLY REMOVAL and REPLACEMENT (UNDER REGULATORS 401 and 402), Stage 2

1. Open the cover on left side of CPU frame.
2. Disconnect electrical connector located at the front center of housing.
3. Remove screws on each end of blower housing.
4. Follow Steps 4 and 5 of CPU Blower and Filter Removal and Replacement procedure.

To replace the blower, reverse the removal procedure.

CPU PHASE-CONTROL REGULATOR BLOWER REMOVAL and REPLACEMENT (OVER REGULATORS 403 and 405), Stage 2

1. Open cover on left side of CPU frame.
2. Disconnect two electrical connectors located on front of blower.
3. Disconnect the thermal leads from the blower at the knife connector.
4. Remove two screws holding the blower assembly to the frame.
5. Lift the blower assembly out of CPU frame.

To replace blower, reverse the removal procedure.

POWER FRAME MID-PAC REGUALTOR STACK BLOWER and REMOVAL and REPLACEMENT, Stage 2

Blower Assembly

1. Open the cover on the back side of power frame.
2. Disconnect the electrical connection located at the center of blower assembly.
3. Loosen four mounting screws (two on each side of assembly) that secure blower to center panel.
4. Support blower so that it cannot drop on frame, and pull it toward you.

To replace, reverse the removal procedure.

Filter

Same procedure as CPU Gate Blower Filter Replacement.

PREVENTIVE MAINTENANCE, STAGES 1 AND 2

Preventive maintenance for the Model 145 power section consists of scheduled equipment checks and adjustments.

Equipment checks and adjustments involve general care and appearance, mechanical inspection, and electrical checks and adjustments. Repair of chipped paint, checking for cracked or loose connections, keeping the machine area clean, and similar items should be performed as deferred maintenance.

Preventive maintenance for the power section involves the following.

Filters and blowers: Check that each blower motor or fan is running. Examine filters for accumulation of dust. Replace dirty filters (schedule every 26 weeks).

Lamp test: All lamps on the CE panel light when CE No. 6 switch is positioned to LAMP TEST. Lamp test for the CE panel can be done anytime the system is on.

Visual inspection and cleaning: Visual inspection of the power assemblies, power supply modules, heat sinks, capacitors (check vent plugs), harness wiring and cables, terminal boards (charred boards or loose connections), relay and contactor contacts (burned or pitted), and safety covers (missing or damaged).

Cleaning of the machine is best done with a brush and a vacuum cleaner. Pay particular attention to the areas near the blower fans and air filters. If the air filters will not pass light after being vacuumed, replacement is necessary.

MOTOR GENERATOR PREVENTIVE MAINTENANCE

The preventive maintenance schedule provides a checkout routine to keep the MG set performance at original specification. Some of these procedures permit aniticipation of failure, thus providing ample warning for replacement planning.

MOTOR GENERATOR PREVENTIVE MAINTENANCE-POWER OFF

WARNING
Remove all power before making these service checks. With light check rotor for no motion.

Frequency (Months)	Item	Procedure
12	Lubrication	Lubricate with IBM 20 grease, part 17397, using the following procedure. 1. Remove drain plugs at bottom of bearing housing at each end of MG. 2. Force grease at each fitting until excess appears at drain opening. 3. Replace drain plugs and grease-fitting protection. CAUTION Use IBM 20 synthesized grease or equivalent. A lower-temperature grease may melt and flow into the motor, causing a short circuit and fire. Too frequent lubrication, or lubrication while the unit is moving, may destroy both the bearing and the grease seal. Once the seal has been fractured, lubricant will leak through to the winding and cause premature failure. Do not check and relubricate the MG set during installation. The manufacturer lubricates the bearings with a metered charge of grease that should last for 52 weeks of three-shift operation. Becuase the metered charge of grease may not fill the cavity, no grease may appear on the relief plug.
3 for first year;every 6 thereafter	Cleaning	Clean dirt and obstructions in the airflow path, preferably with a vacuum cleaner.
	Electrical connections	Check the electrical connections as follows. 1. Check all terminals, grounds, etc; for loose electric connections. Tighten all terminals as needed. 2. Check for loose crimps on wire terminals and tighten as needed. 3. Check for charred or molten insulation on wires; exposing the conductors. Apply tape to protect the conductor. Order new cable(s) for replacement on next inspection.
	Grease fitting Shock mounts	Check for dripping grease at two end bearings. Rock MG by hand on shock mounts. Tighten any loose mechanical parts.

MOTOR GENERATOR TROUBLESHOOTING GUIDE, STAGES 1 AND 2

The MG troubleshooting guide exclusively applies to the MG set (including the MG regulator). Use the guide only after the power problem has been identified as an MG-set problem. Complete phase-to-phase line voltages must be present at the motor terminal block at power-on time.

Before proceeding to use the troubleshooting guides, perform the Power Off Preventive Maintenance routine.

MOTOR GENERATOR DRIVE MOTOR FAULT-ISOLATION AIDS (MG LOAD OFF)

TROUBLE: MOTOR WILL NOT TURN: NO GROWLING NOISE

Probable Cause	Isolation Procedure	Remedy
Two or more open circuits in motor field.	Remove all input power and the three input leads to motor at TB1. Make resistance check of field winding. Also check for loose lug crimps	If evidence of internal open circuits, replace MG set. If any leads within MG frame are chared, replace MG set.

TROUBLE: MOTOR STARTS: CPU CB1 TRIPS

Probable Cause	Isolation Procedure	Remedy
Loss of any one phase to motor (broken wire in motor).	Remove MG load (set CB2, CB5 and CB6 off). Apply ac power to motor. Check phase-to-phase voltage using VOM. If results are satisfactory, proceed with the following: Remove all power and the 3 input leads to motor at TB1. Make resistance check of field winding. Also check for loose lug crimps.	If evidence of internal open circuits, replace MG set. If any lead within MG set is charred, replace MG set.

TROUBLE: MOTOR DOES NOT COME UP TO SPEED. EMITS GROWLING NOISE

Probable Cause	Isolation Procedure	Remedy
Excessive voltage drops to motor.	Remove MG load (set CB2, CB5 and CB6 off). Measure voltage at TB1. Remove power to regulator by tripping CB1 on regulator. If input voltage ok and motor comes up to speed after power to regulator has been removed faulty regulator.	Replace MG regulator.
	If motor continues to turn slowly after regulator has been disconnected, possible shunted diode in rotor (also check for order of insulation).	Replace MG set.
Seized bearing	Trip CB1 on regulator. Excessive bearing heat and noise being generated.	Replace MG.

MG GENERATOR AND REGULATOR FAULT-ISOLATION AIDS
(MG previously checked to assure normal operating speed)

TROUBLE: ZERO VOLTAGE AT ANY PHASE-TO-PHASE AT GEN OUTPUT TB2

Probable Cause	Isolation Procedure	Remedy
Loss of residual magnetism	Connect VOM to 400 Hz output TB2. Apply ac power to motor. VOM registers zero volts.	Flash field winding as follows: Apply 6V dc, momentarily to the field. Make sure that the plus (+) side is connected to TB3 of the regulator terminal board. (Have a 10-ohm resistor in series with the 6V dc source) or 24V dc with suitable resistor.
Two or more open circuits to generator armature, or all three diodes to generator rotor.	Remove all power, and the output leads from generator at TB2. Make resistance check of generator windings.	If evidence of internal open circuit, replace MG set.

TROUBLE: LOW OUTPUT VOLTAGE (LESS THAN 30V)

Probable Cause	Isolation Procedure	Remedy
CB1 tripped	Reset CB1; CB1 opens second time.	Replace regulator.
Regulator	Check generator output voltage at TB2. If voltage exceeds the over voltage trip setting (about 225V) trouble is in regulator.	Replace regulator.
Regulator-O.V. trip circuit 2K	Disconnect cable T† & T3 at regulator. Monitor 2K point (N/C) If output voltage comes up to over 208 and then drops back raise over-voltage setting to maximum and check generator output at TB2. If O.V. trips at any setting, check overvoltage setting by raising generator output via voltage-adjust potentiometer until K2 picks. Repeat until desired overvoltage setting is ovtained.	Replace regulator.

MG TROUBLESHOOTING GUIDE (Continued)

MG GENERATOR AND REGULATOR FAULT-ISOLATION AIDS
(MG previously checked to assure normal operating speed)

TROUBLE: LOW OUTPUT VOLTAGE (LESS THAN 30V)

Probable Cause	Isolation Procedure	Remedy
Regulator (cont)	If O.V. relay trips regardless of the overvoltage setting, trouble is in regulator.	Replace regulator.
Loss of residual magnetism	Check generator output voltage at TB2 if voltage is less than 15. This may be insufficient to enable generator buildup. (See ALD YE180) or YB180.	Flash field winding using a 6V battery. Make sure the plus (+) side is connected to TB3-1. (Have a 10-ohm resistor in series with 6V dc source). Connect minus (-) side to TB3-2.
Shorted diode or open in exciter	If voltage does not rise above 30 after flashing the field winding, trouble is in exciter field or other internal windings on rotor.	Replace MG.

TROUBLE: GENERATOR OUTPUT GOES BEYOND 260V

Probable Cause	Isolation Procedure	Remedy
Regulator and O.V. circuit failure.	Adjust voltage potentiometer over-voltage potentiometer to minimum setting. If voltage still soars beyond 208, trouble is in regulator.	Replace regulator.

TROUBLE: UNSTABLE OUTPUT VOLTAGE (WITH OR WITHOUT LOAD)

Probable Cause	Isolation Procedure	Remedy
Regulator	Vary the Volt-Adjust Potentio-meter to determine whether any voltage control can be achieved. (Varying the voltage-adjust control sometimes will clean the potentio-meter and re-establish good control.)	Replace regulator at earliest opportunity.
Regulator	As load is applied, trouble is in regulator.	Replace regulator

MG GENERATOR AND REGULATOR FAULT-ISOLATION AIDS
(MG previously checked to assure normal operating speed)

TROUBLE: EXCESSIVE MG HEATING (AIR PASSAGES CLEAR)

Probable Cause	Isolation Procedure	Remedy
Excessive current drawn by motor MG not overloaded.	Check current at each leg of motor (TB1). Current for 16KW at 208V/60 Hz for example should read approximately 23 amperes. At no load, motor currents are not necessarily balanced. At medium to full load, the currents are approximately equal and are stated on name-plate.	Replace MG
Shorted turns on generator	Measure phase-to-phase generator voltages. The voltages must balance within two percent. An unbalance that exceeds two per cent may indicate possible shorted turns in generator armature windings.	Replace MG
Overload of MG	Check MG max current on nameplate. Measure generator output current with clamp-on armature. (That is, circuit breakers may not trip).	Reduce to rated load. Faulty CBs

TROUBLE: HIGH EXCITER FIELD VOLTAGE

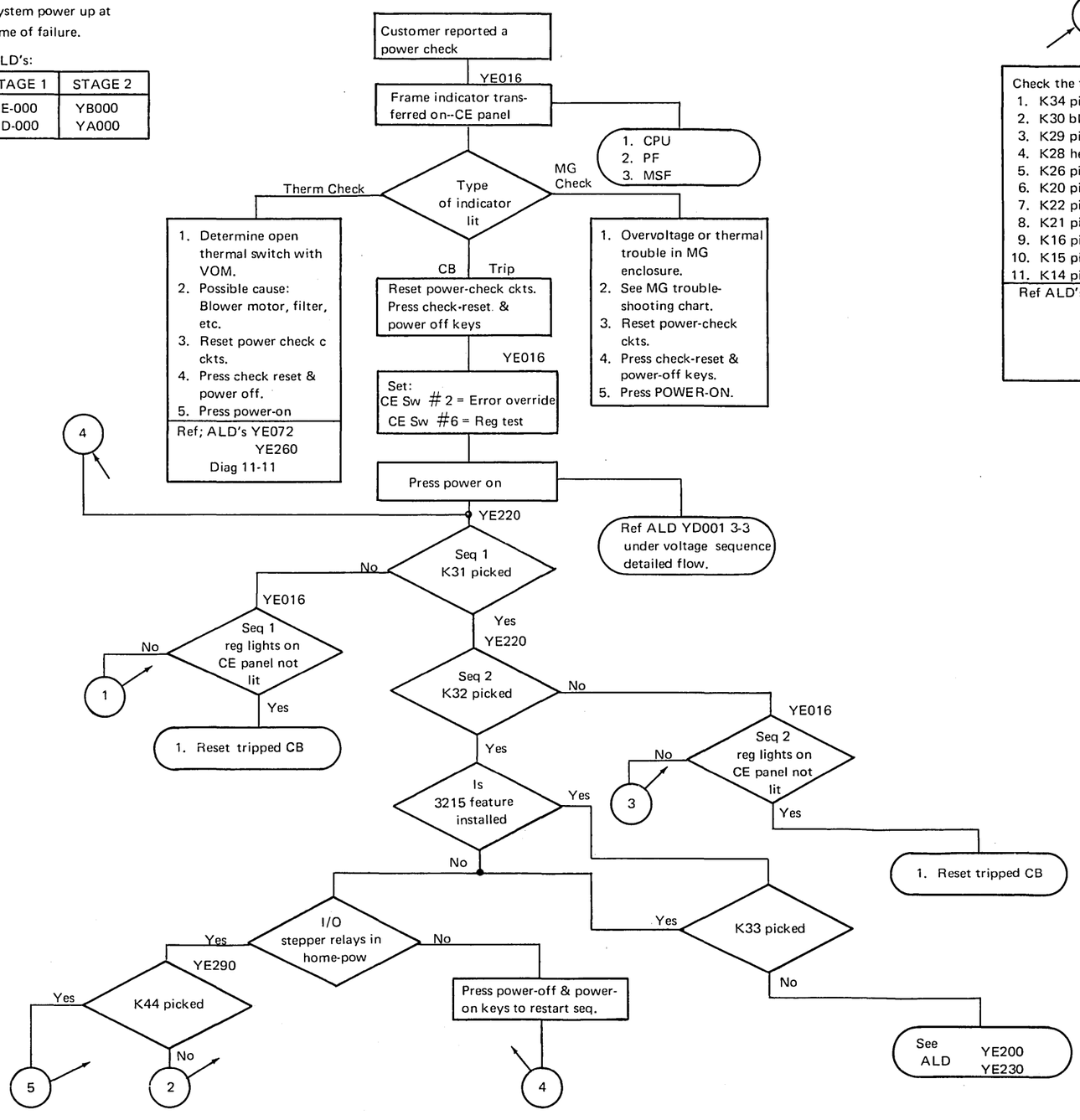
Probable Cause	Isolation Procedure	Remedy
Open diode on rotor	Measure voltage at exciter field TB3. If voltage reading is significantly beyond 12, only two phases are supplying current to generator field.	Replace MG at earliest opportunity.

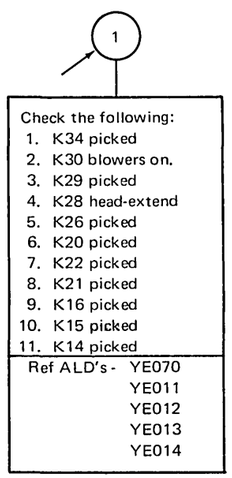
POWER CHECK TROUBLESHOOTING DATA FLOW, Stages 1 and 2

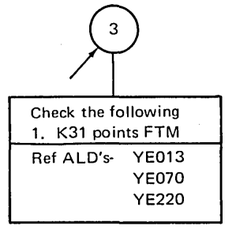
NOTE:

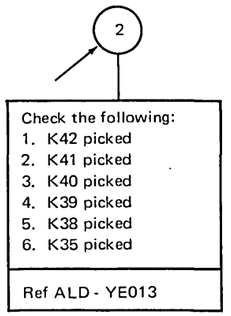
1. System power up at time of failure.
2. ALD's:

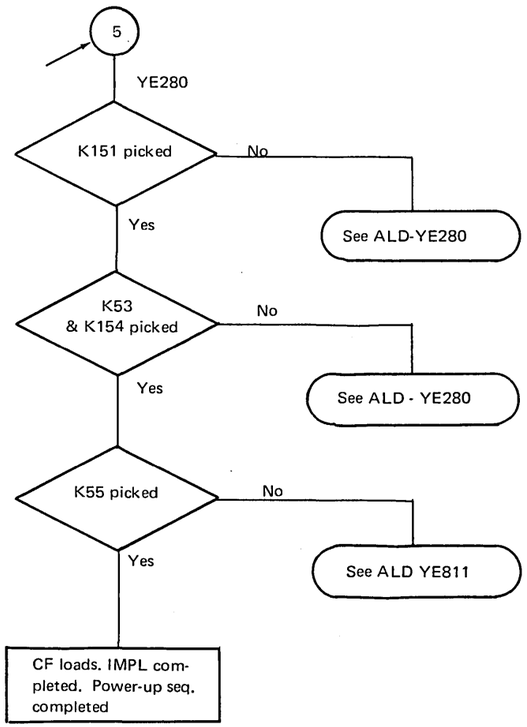
STAGE 1	STAGE 2
YE-000 YD-000	YB000 YA000











REMEMBER

There is a Reader's Comment Form at the back of this publication.

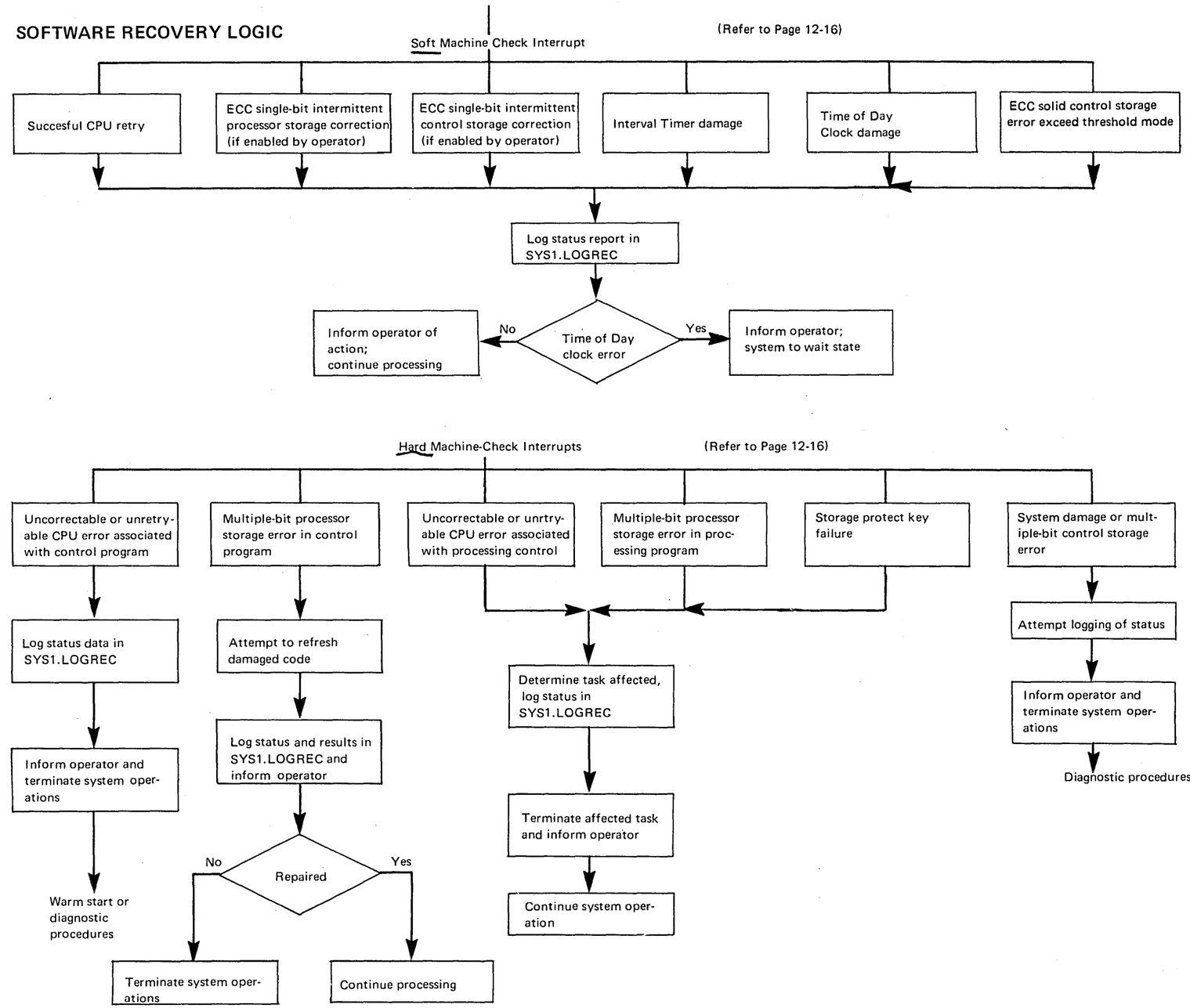
CHAPTER 12. ERROR HANDLING

INTRODUCTION

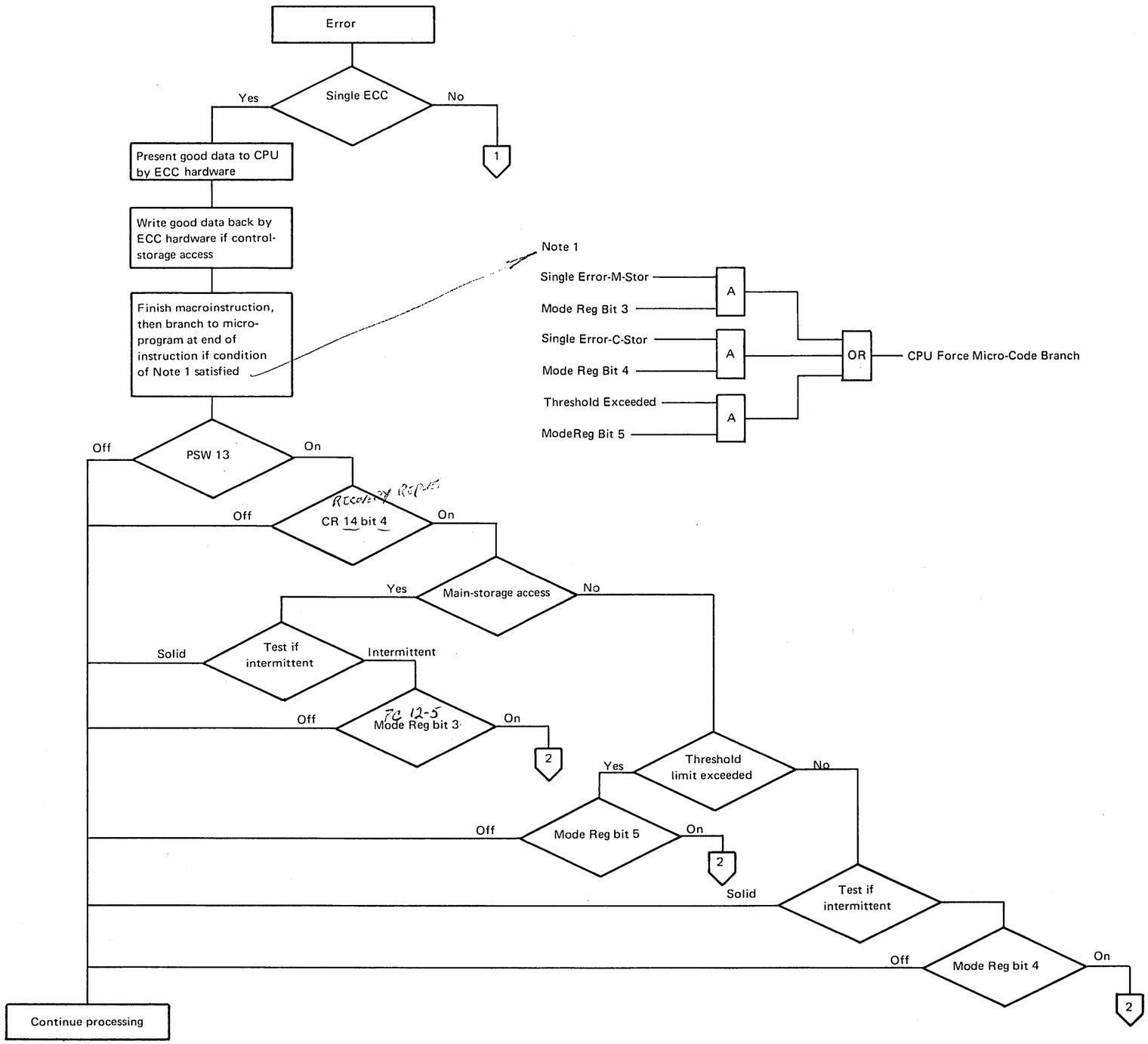
The System/360 Model 145 incorporates several types of error-detection and correction mechanisms, to provide optimum machine availability to the customer. These mechanisms include: single-bit storage error correction (ECC), machine-check error retry, machine-check error interrupts, and software recovery.

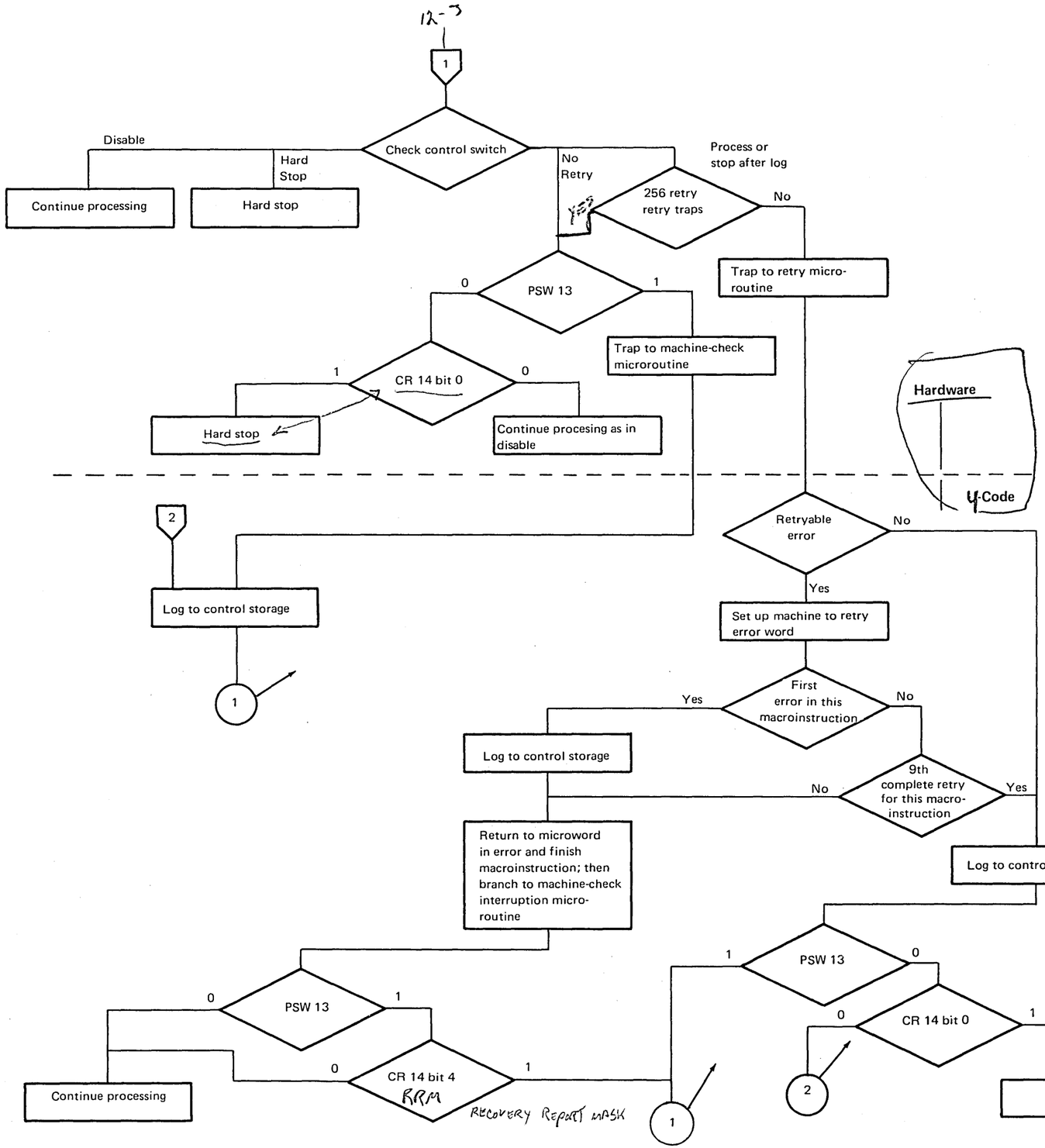
- Single-bit storage errors are detected and corrected by the Error Check and Correction (ECC) logic. Single ECC errors are not normally presented to the system control program. Double or excessive single ECC errors are considered a machine check.
- Machine-check errors initiate a hardware retry (except when CHECK CONTROL is in NO RETRY or DISABLE), which repeats the failing operation. Up to eight complete retries, and up to 255 retry traps are allowed. If the retry operation is unsuccessful in eight attempts or if the nature of the error is unretryable (for example, clock sync check), a machine-check interrupt is presented to the system control program.
- Machine-check interruptions are used to notify the system control program that an uncorrected machine error has occurred, or to log corrected intermittent failures. If the current PSW bit 13 equals 1, an old PSW is stored at location 48, containing the information necessary for the system control program to perform a software recovery if desired. An additional eight bytes of data are stored at location 232 (MCIC, pages 12-18) for system control programmer's use.
- Software recovery consists of the Recovery Management Support (RMS) package contained in OS and DOS. The RMS program analyzes the machine-check interruption code (Main Storage Loc. 232) and determines whether the failing operation can be reconstructed or must be terminated. It also stored the logout information on the SYS1 LOGREC disk pack for further analysis by maintenance personnel.

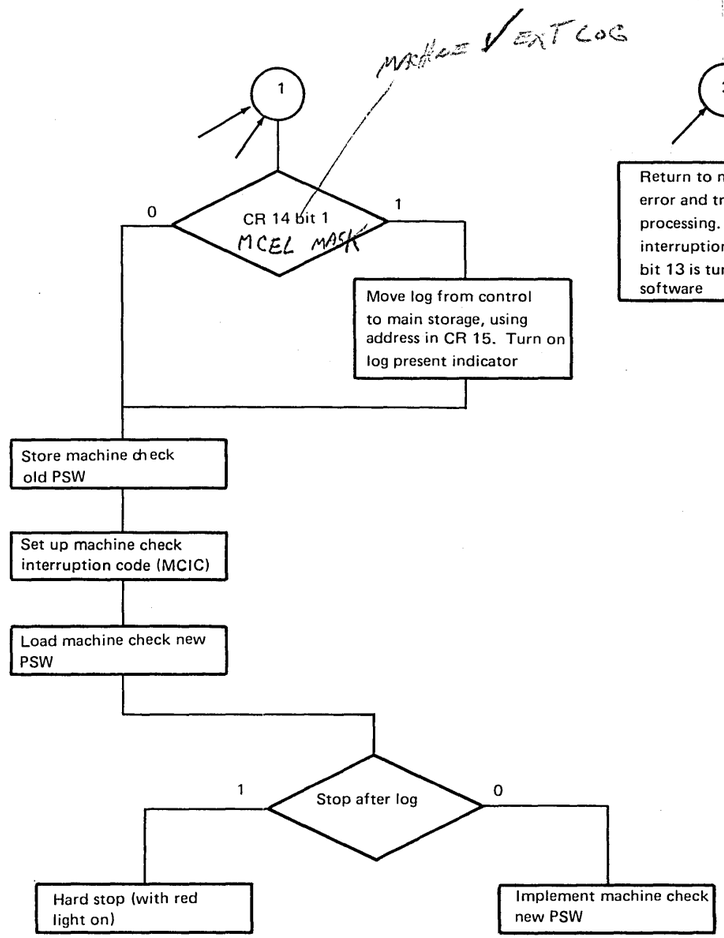
SOFTWARE RECOVERY LOGIC



HARDWARE RECOVERY LOGIC




Disable
Check control switch
Hard Stop
Continue processing
Hard stop
No Retry
Process or stop after log
256 retry retry traps
No
0
PSW 13
1
Trap to retry micro-routine
Trap to machine-check microroutine
1
CR 14 bit 0
0
Hard stop
Continue procesing as in disable
Hardware
Y-Code
2
Retryable error
No
Yes
Log to control storage
Set up machine to retry error word
1
First error in this macroinstruction
Yes
No
Log to control storage
9th complete retry for this macro-instruction
No
Yes
Return to microword in error and finish macroinstruction; then branch to machine-check interruption micro-routine
Log to control storage
0
PSW 13
1
Continue processing
0
CR 14 bit 4
1
RRM
RECOVERY REPORT MASK
1
PSW 13
0
0
CR 14 bit 0
1
2
Hard stop


MACHINE CHK EXT COG
1
0
CR 14 bit 1
MCEL MASK
1
Move log from control to main storage, using address in CR 15. Turn on log present indicator
Store machine check old PSW
Set up machine check interruption code (MCIC)
Load machine check new PSW
1
Stop after log
0
Hard stop (with red light on)
Implement machine check new PSW
2
Return to microword in error and try to continue processing. No machine interruption until PSW bit 13 is turned on by software

RECORDING MODES

- The mode register (Ext. word 08 byte 0) is used to determine the action to be taken for single-bit storage errors.
- The mode register is set by the diagnose command (83--BDDD).
- Operator commands to alter the mode register are incorporated under OS and DOS.

Byte 0 **MODE Register (External Address 08)**

- Bit 0 Hard Stop Latch-Control Register 14 Bit 0
- Bit 1 Enable I-Cycle and Adr/Adj Ctrl and Expanded Local Storage.
- Bit 2 Enable Hardware Retry
- Bit 3 Full Recording Mode for Single-Bit Failures in Main Storage
- Bit 4 Full Recording Mode for Single-Bit Failures in Control Storage
- Bit 5 Threshold Mode for Single-Bit Failures in Control Storage
- Bit 6 Reserved
- Bit 7 Reserved

The diagnose instruction (83--BDDD) sets the mode register to determine what action is to be taken when a single-bit storage error occurs. The mode register is also set by a power-on reset or system reset. Mode register bits 3, 4, and 5 determine the mode the system is in:

When the system is in record mode for either main or control storage, intermittent single-bit errors result in a machine-check interruption. Solid single-bit storage failures do not result in a machine-check interruption.

When the system is in threshold mode, both solid and intermittent single-bit corrections in control storage are counted by a hardware counter. If the counter reaches 256 before it is reset by a pulse generated from the Time-of-Day Clock, a machine-check interruption is taken. This reset pulse occurs approximately every two milliseconds.

Table 1	Control-Storage Mode	Main-Storage Mode	CR 14 Bit 4	PSW Bit 13
Initialized by Pwr On Reset or System Reset with Enable Clear	Threshold	Quiet	0	0
Initialized by System Reset	Threshold	Quiet	No Change	No Change
Initialized by Recovery Management Support (RMS)	No Change	No Change	1	1
Hardware Threshold count exceeded for Control Storage	Set to QUIET by hardware. Operator message presented by RMS and a record Logged to SYS1. LOGREC	No Change	No Change	No Change

Table 2 Instruction Format (83--BDDD)	Control-Storage Mode	Main-Storage Mode	CR 14 Bit 4	PSW Bit 13
MODE ECC, RECORD, MAIN B+D=000008	No Change	Record	No Change	No Change
MODE ECC, QUIET, MAIN B+D=00000C	No Change	Quiet	No Change	No Change
MODE ECC, RECORD, CONT B+D=000010	Record	No Change	No Change	No Change
MODE ECC, QUIET, CONT B+D=000018	Quiet	No Change	No Change	No Change
MODE ECC, THRHOLD, CONT B+D=000014	Threshold	No Change	No Change	No Change

Table 1 outlines the modes of taking machine-check interruptions for recovery reports without taking into account the operator mode commands.

Table 2 outlines how the mode commands affect the modes of taking machine-check interruptions for recovery reports. Note that in record mode for main and control storage, only intermittent single-bit failures result in a machine-check interruption. The microprogram tests the failing storage location (main or control) to see whether the single-bit failure is solid or intermittent; if the failure is solid, the machine returns to processing instructions and no machine-check interruption is taken.

REMEMBER

There is a Reader's Comment Form at the back of this publication.

ERROR CHECK AND CORRECTION (ECC)

For details of ECC circuits, refer to Chapter 3, Main and Control Storage.

- Error-Check and Correction circuitry for main and control storage automatically corrects single-bit errors.
- Double-bit errors are detected and cause a machine check (MCKB byte 2 bit 5).
- The hardware retry procedure is not executed for single ECC errors.
- In control storage, a store is done to attempt to put corrected data back in storage; however, in main storage no store of corrected data is made.

When a single-bit error occurs, the ECC hardware presents corrected data to the CPU. In control storage, one machine cycle of degradation results; a restore is done by the ECC hardware in an attempt to correct the bad bit in control storage. In main storage no degradation results from a single-bit failure; no attempt at restore is done. This action is taken regardless of the settings of the mode register bits 3, 4, and 5. These bits provide another level of masking below CR 14 bit 4 (recovery mask bit) in order to control the machine-check interruption action taken for single-bit failures.

The operating values of these bits are:

3 4 5

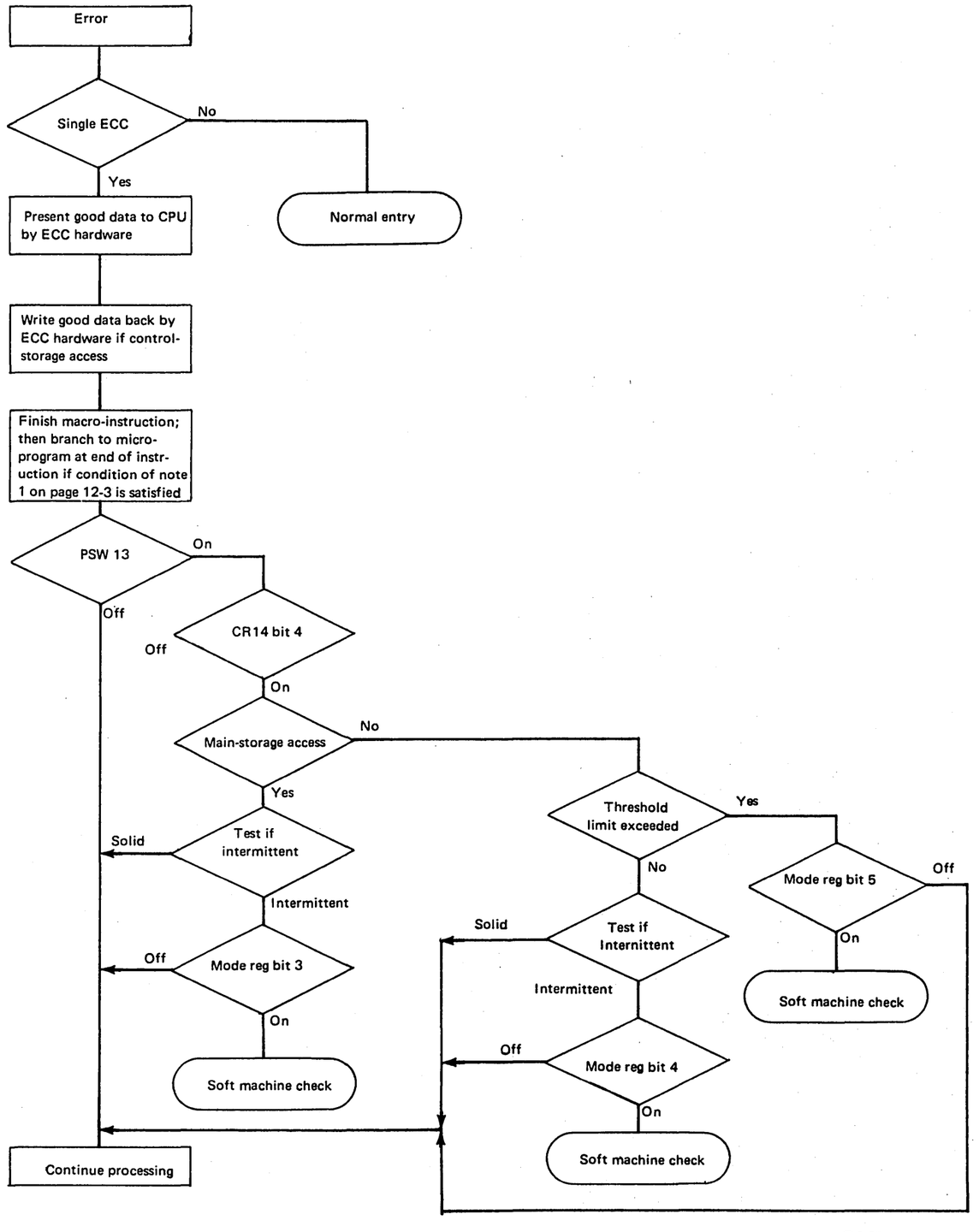
X 0 1 Every single-bit failure in control storage is counted in a counter (external word 00 byte 2 ECNT). If 256 errors occur within two milliseconds, the retry registers are frozen and at the end of the current macroinstruction a soft machine interruption is initiated.

X 1 0 When a single-bit failure in control storage occurs, the retry registers are frozen, containing the needed information to refetch the control word. At the end of the current macroinstruction, a special microprogram is entered and the control-storage location is again accessed. If no single-bit failure occurs, the error is considered to be intermittent and a soft machine-check interruption is initiated.

1 X X When a single-bit failure in main storage occurs, the retry registers are frozen, containing the needed information to refetch this main-storage location. At the end of the current macroinstruction, a special microprogram is entered and a restore is attempted by fetching the failing location and then storing back the same data. A second fetch is done; and if no single-bit failure occurs, the error is considered to be intermittent, and a soft machine-check interruption is initiated.

0 0 0 Single-bit errors are not relayed to the microcode and thus no machine-check interruptions can be initiated.

Note that both PSW bit 13 and the recovery mask (CR 14 bit 4) must be on for any of the single-bit failures to result in machine-check interruptions. If the microprogram determines for any reason that a machine-check interruption should not be taken, normal System/370 processing is continued.



MACHINE CHECKS AND ERROR RETRY

- Any error detected within 3145 hardware circuits causes a bit to be set in the machine-check registers (MCKA, MCKB).
- A bit being on in the machine-check register (except MCKB2 bit 6-7 and MCKB3 bit 0-7) activates the line 'any machine check', which causes a trap to the retry microroutine (GHRT).
- After a retry trap occurs, the microroutine determines the type of error that has occurred (Type 1, 2, or 3). If the error is a Type 1, 2, LS compare, or FTC error, the hardware registers are refreshed and the word in error is retried. If the error is a Type 3, it is unretryable and a hard machine-check interrupt is initiated.

A flow chart of the retry microroutine is on page 12-12.

When a machine malfunction occurs in the Model 145, the operation is retried by hardware and microprogram, in an attempt to produce a valid result. Such equipment malfunctions are referred to as *machine-check conditions.* The retry is always executed except when the nature of the error is such that recovery is deemed impossible, (for example, CLOCK SYNC CHECK or CHECK CONTROL is set to NO RETRY.)

When a retry results in a correct operation, the condition is considered a soft machine check and is reported as a recovery report if CR14 bit 4 and PSW bit 13 = 11. This information is stored for later use by the service representative.

When the machine-check condition is uncorrectable or when the retry does not result in proper operation, the condition is considered a hard machine check and the incident is reported as a damage report. The internal damage condition results in a machine-check interruption if the machine-check mask bit (PSW bit 13) is set to one. Operation under these conditions is described under "Machine Check Interruptions," page 12-16.

After a machine-check interrupt has occurred, the system program tries to recover from, or repair, the condition through programming. The sequence of the operation and the possible outcome depend on the ability to retrieve suitable data and effect an alternate method of operation.

BIT	MCKA0		MCKA1		MCKA2		MCKA3	
0	LSA Source Address Check	1	ACB Register Parity Check	2	ALU2 Half-Sum Check	2	Ext Destine X Comp Check	3
1	LSB Source Address Check	1	LS Compare Check		ALU3 Half-Sum Check	2	Ext Destine Y Comp Check	3
2	LSA Destine Address Check	2	Flush-Through Check		ALU Logical Check	2	Ext Source Y Check	1
3	LSB Destine Address Check	2	H-Register Parity Check	2	B-Register Shift Check	2	Ext Control Asmbl Parity	1
4	Destine Byte Control Check	3			A-Register Parity Check	1	Interval Time Parity Check	3
5	LSA,B Destine Address Comp	3	P-Register Parity Check	2	B-Register Parity Check	1	S-Register Dup Check	2
6	LS Control Asmblr Check	1	T-Register Parity Check	2	Z-Register Parity Check	2	Time-of-Day Clock Check	3
7	C-Register Parity Check	1	L-Register Parity Check	2	D-Register Parity Check	2	Control Stg Address Check	2

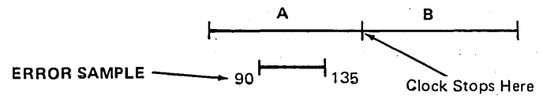
These indicators are shown in position 8 of the upper roller.

BIT	MCKB0		MCKB1		MCKB2		MCKB3
0	Storage Address Check	1-2	M-Register Comp Check A	1	I-Cycle Hardware		CT Error Correction
1	SDBI Parity Check	2	M-Register Comp Check B	1	Double ECC Error-SDC	2	C32 Data Bit Corrected
2	SDBO Parity Check	2	M-Register Comp Check C	1	Control-Line Parity Check	2	C16 Data Bit Corrected
3	Store Parity Check	2	M-Register Comp Check D	1	ECC Busy Check	2	C8 Data Bit Corrected
4	Spare		Addr X-Late No Match	2	ECC Hardware Check	2	C4 Data Bit Corrected
5	Store Protect Parity Check	2	Addr X-Late Multi Match	2	Double ECC Error	2	C2 Data Bit Corrected
6	Clock Sync Check A	3	Addr X-Late LRU Invalid	2	Single ECC Error Correct		C1 Data Bit Corrected
7	Clock Sync Check B	3	Any Machine Check		Single Data Bit Correct		C0 Data Bit Corrected

These indicators are located in the System Checks portion of the console. The indicators on the console are grouped into functional areas, not as they appear in MCKB. MCKB is an external register and can be displayed by setting address 06 in switches F, G and STORAGE SELECT to EXT REGS.

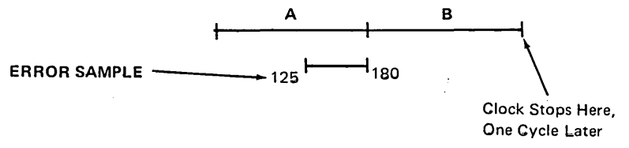
If the machine is operated with CHECK control in HARDSTOP the following timings apply.

Type 1 - Clock stops in same cycle.



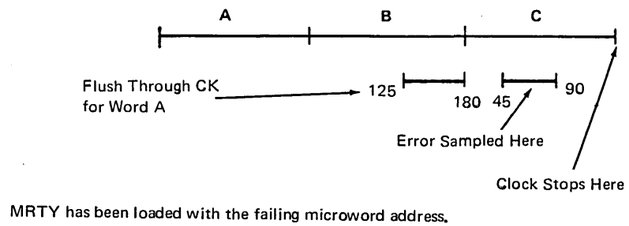
Note: In this type of error, pressing START once loads MRTY with the failing microword address.

Type 2 Clock stops in the following cycle.



MRTY has been loaded with the failing microword address.

FTC (Flush-Through Check), LS Comp (Local Store Compare)
Clock stops two cycles later.



MRTY has been loaded with the failing microword address.

Type 3 errors: These are uncorrectable errors. The retry hardware cannot attempt to reconstruct and repeat the failing microword.

RETRY OPERATION

Microprogram Instruction Retry

The ability to recover from most intermittent failures is provided by retry techniques. CPU retry is done by microprogram routines that save the source data before it is altered by the operation. When an error is detected, a microprogram routine returns the CPU to the beginning of the operation (or to a point during the operation that was executed correctly), and the operation is repeated.

The retry routine is entered through the retry priority operation (trap). The retry priority operation occurs when any machine check occurs if the retry counter is not full, retries are not masked off, and system register byte 2 bit 6 (indicates stop word error) is off. Depending upon the nature of the error and the word type, the error may be detected during execution of the failing microprogram word (Type 1), during execution of the following word (Type 2), or may be detectable but uncorrectable (Type 3),

The information necessary to retry an error is retained in backup registers that contain the contents (at the time the error occurred) of:

- A-register or B-register (whichever was used as the destination).
- SPTL register
- M2- and M3-registers
- H-register
- External and local-storage destination latches (six bits for X and Y-addressing plus 1 bit for expanded local storage.
- Retry flag register (bit 0 on for storage word cycle 2; bit 1 on for error Type 1; bit 2 on for error Type 2; bit 3 on for error Type 3; bit 4 on if the destination was an external, or off if destination was a local-storage location; bit 5 on for a storage word; bit 6 on for a trap 2; bit 7 on for an unretryable error).

A return to the microprogram word to which the check is attributed occurs, and by use of information in backup registers, that microprogram word is retried. Up to 8 retries are attempted. If the retry operation is successful, a machine-check interruption may be performed to notify the supervisor that a machine-check occurred. This occurs if the PSW machine-check bit and the recovery report mask bit (CR 14 bit 4) allows the interruption.

During the machine-check-interruption microprogram routine, the logout information is moved from control storage to main storage so that the supervisor routines can determine the cause or error. Action is then dependent upon the supervisor program.

An unsuccessful retry procedure results in a machine-check interruption, unless the machine-check bit in the PSW prevents it.

Channel Retry

This feature has been implemented to ensure that most failing channel operations can be retried by error-handling routines. Both a limited and an extended channel logout are implemented. When a channel error or a CPU error associated with a channel operation occurs, the channel status word (CSW) and a new extended channel status word (ECSW) are stored in the fixed lower storage area (I/O communications area) during the I/O interrupt. The ECSW, or limited channel logout data, provides additional, more exacting status information about the channel failure. This data is formatted by the Channel Check Handler (CCH) routine and passed to a device-dependent error-recovery routine to be used in the retry of the failing I/O operation. For a detailed description of Channel Retry, refer to Chapter 8, Channels.

Command Retry

Command retry is a control-unit-initiated procedure between the channel and the control unit. (Not all control units have this capability.) No I/O interruption is required. The number of retries is device-dependent. For a detailed description of Command Retry, refer to Chapter 8, Channels.

RETRY HARDWARE

When an error is detected because of a machine malfunction in the system, instead of repeating the whole macroinstruction concerned, only the microword in error is repeated, and the operation is continued from that point. Hardware backup registers are used to provide the backup system status information needed to perform the retry.

In general, the backup registers contain system status information pertaining to the machine cycle preceding the cycle in process. Each register contains one word (four bytes) of information. Every machine cycle (that is, microword) the backup registers are updated to reflect the status of the machine.

When a machine malfunction occurs, the registers are blocked (prevented from changing) and thus contain the status information of the system just before the microword in error. Upon detecting the malfunction, the machine traps to the retry microroutine (H1=1 trap ADDR D200). In the retry routine, the information in the hardware retry registers is used to refresh the machine, after which the microword in error is re-executed.

Retry and Backup Registers

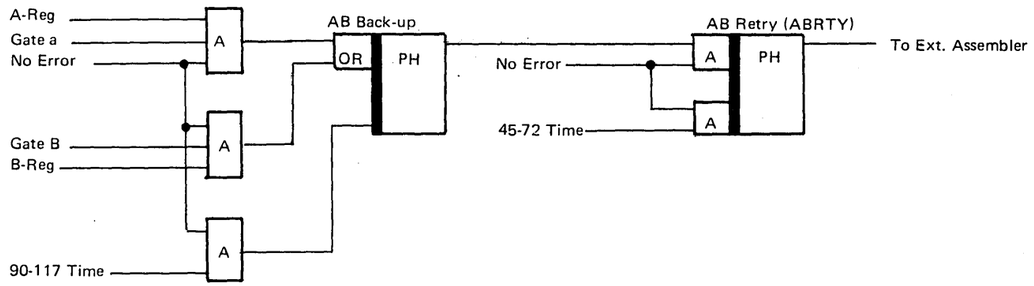
Because of differences in the kinds of errors that may occur, two levels of backup registers are required for some of the information. The retry registers are set to the information for the cycle before the one being executed. The backup registers are set to the current cycle information.

Unless blocked, each register is updated every machine cycle. At 45-72 time, the retry registers are set with the information for the cycle immediately preceding the one in process. At 90-117 time, the backup registers are set with the information for the cycle in process. Each register retains the information until the new information (next cycle) is set or, in the case of a detected malfunction, until the retry trap has been taken and the block is removed by the microprogram. (See timing chart, page 12-14).

The ABRTY, SPTLB, HMRTY, and CPURTY registers enter the data flow through the external assembler. When the contents of these registers are blocked because of a detected failure, they will contain the information to be used by the retry microprogram. The information in the registers is as follows.

ABRTY

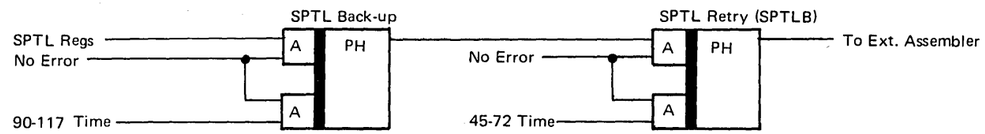
This register contains the source data read into the A- or B-register during the microword in error. If the word type was A=A/B, it contains the A source data; if the word type was B=A/B, it contains the B source data.



SPTLB

This register contains the contents of SPTL immediately prior to starting the microword in error.

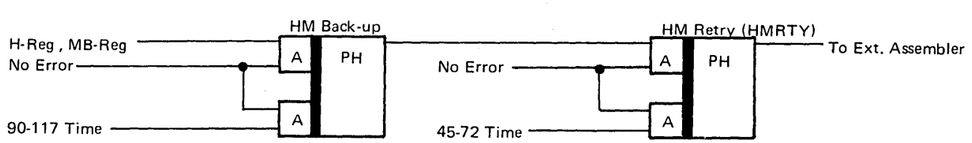
Byte 0	1	2	3
S	P	T	L
SRTY	PRTY	TRTY	LRTY



HMRTY

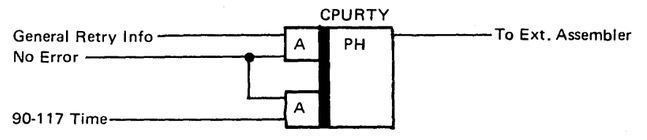
Byte 1 of this register contains the contents of the H-register immediately prior to starting the microword in error. Bytes 2 and 3 contain the address of the microword in error.

Byte 0	1	2	3
Spare	H	MB2	MB3
	(HRTY)	(MRTY2)	(MRTY3) (MRTY2)



CPURTY

This register contains various information relative to the operation being performed. The information contained is described in the following paragraphs.



Byte 0	1	2	3
BYTDST	RTYFLG	LSDST	EXTDST

BYTDST (Byte 0): Bits 0-3 of BYTDST are spares. Bits 4-7 of BYTDST contain the destination byte lines for the microword in error. These lines tell which bytes of the destination register (local storage or external) specified in the error cycle were changed.

RTYFLG (byte 1): This register contains information about the malfunction and is used to support the retry microprogram.

Bit 0 (storage 2 cycle): This bit is set when the system malfunction is attributed to the second cycle of a storage microword. These bits are turned on by hardware latches (MCKA or MCKB registers).

Bit 1 (Type 1 error): A Type 1 error is a system malfunction that is detected in the same cycle that initiated the operation in error. The error is detected early enough so that the normal destination of the word in error is blocked. Thus the source data is not refreshed by the microprogram.

Bit 2 (Type 2 error): A Type 2 error is a system malfunction that is detected in the next cycle after the cycle that initiated the operation in error.

Bit 3 (Type 3 error): A Type 3 error is a system malfunction that cannot be corrected by the retry microprogram.

Bit 4 (external or local-storage destination): This bit is set to one if an external register was being destined by the microword in error. It is zero if a local-storage location was being destined by the microword in error.

Bit 5 (storage word): This bit is set to one if the system malfunction is attributed to a storage microword.

Bit 6 (stop word): This bit is set to one when the system malfunction is attributed to a Trap 2 Cycle.

Bit 7

LSDST (byte 2): This byte contains the address of the local-storage location that was the destination in the microword in error. This byte has meaning only if bit 4 of RTYFLG is zero. The format of the bits in the register correspond to the format of the direct address in the word-move microword.

Bit 0: Zero.

Bits 1-3: These bits define the local storage Y-lines.

Bits 4-6: These bits define the local-storage X-lines.

Bit 7: One expanded local storage.

EXTDST (byte 3): This byte contains the address of the external register that was the destination in the microword in error. This byte has meaning only if bit 4 of RTYFLG is one. The format of the bits in this register corresponds to the format of the direct address in the word-move microword.

Bit 0: One

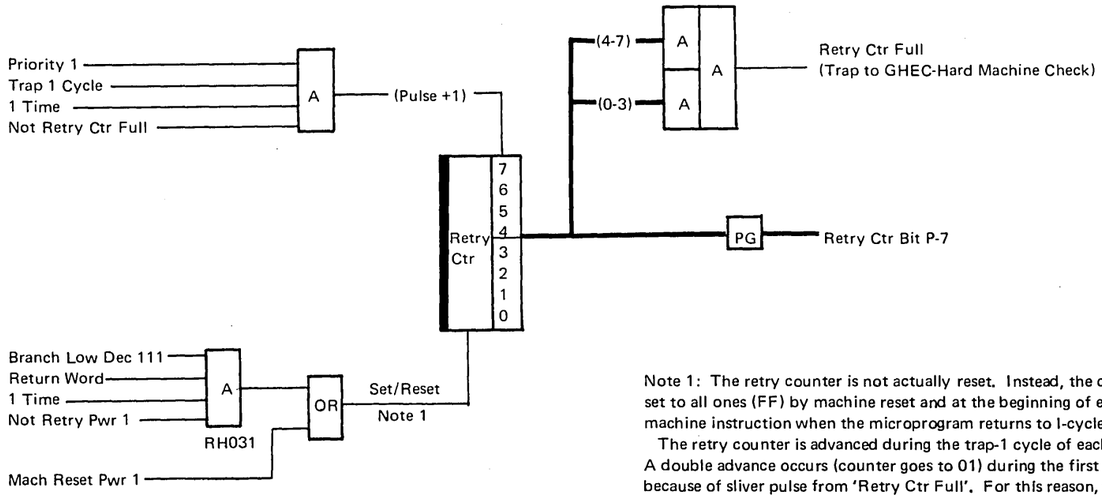
Bits 1-3: These bits define the external Y-lines.

Bits 4-6: These bits define the external X-lines.

Bit 7: Zero.

Retry Counter

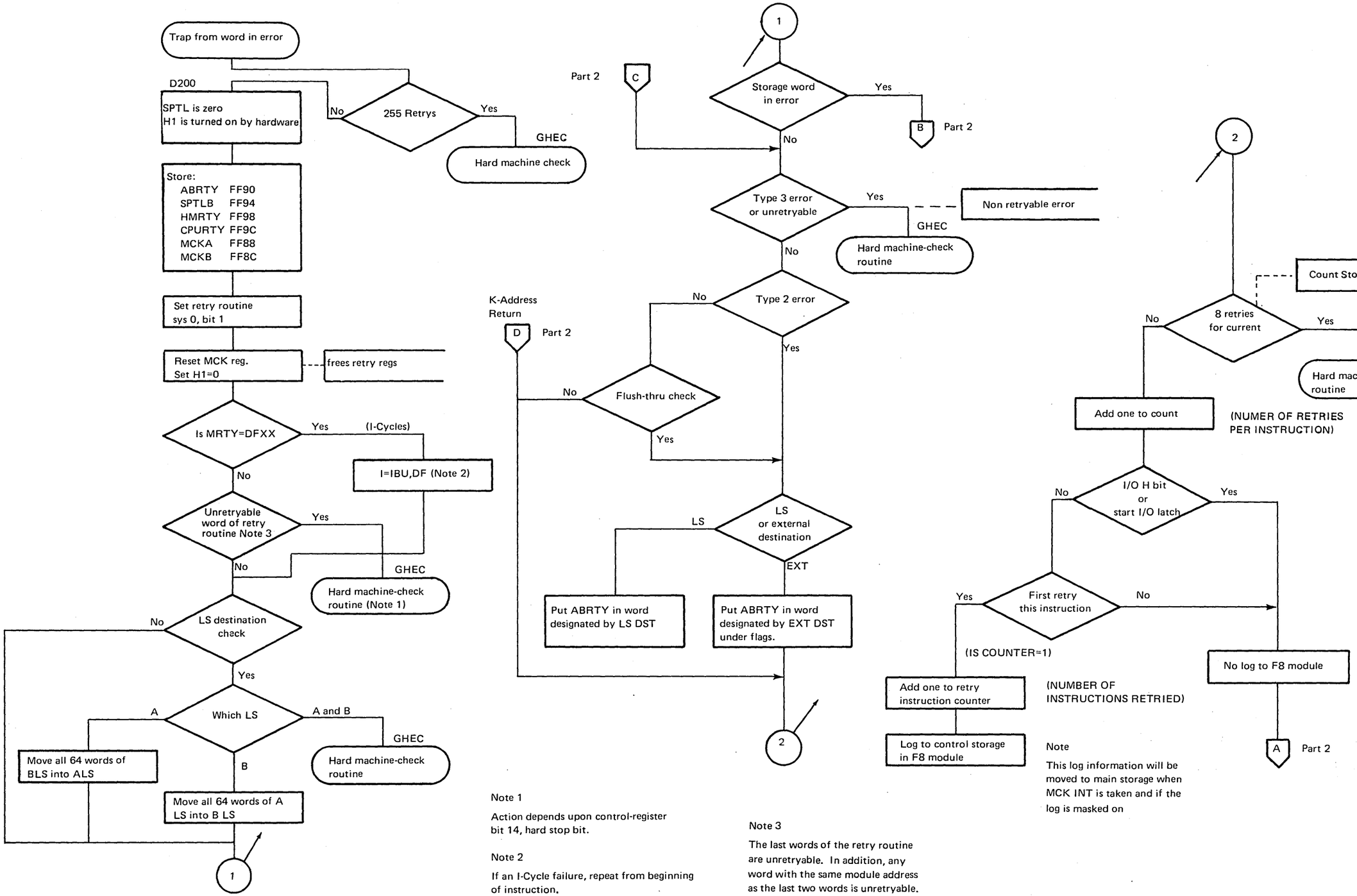
- The retry counter is an eight-position hardware counter that is used to regulate the number of retry traps that are allowed.
- The counter is set to FF when the microprogram returns to I-cycles, and is advanced by 1 for each priority 1 trap that occurs. (*Note*: The first retry causes a double-advance pulse.)
- If the error is solid enough to occur 256 times during one machine instruction, on the 256th time a trap is taken to the GHEC (machine check) microroutine.
- The counter is stored at external location 00 byte 3 (RCNT).



Note 1: The retry counter is not actually reset. Instead, the counter is set to all ones (FF) by machine reset and at the beginning of each machine instruction when the microprogram returns to I-cycles.

The retry counter is advanced during the trap-1 cycle of each retry. A double advance occurs (counter goes to 01) during the first retry because of sliver pulse from 'Retry Ctr Full'. For this reason, the maximum count is 255.

Retry Microroutine



Note 1

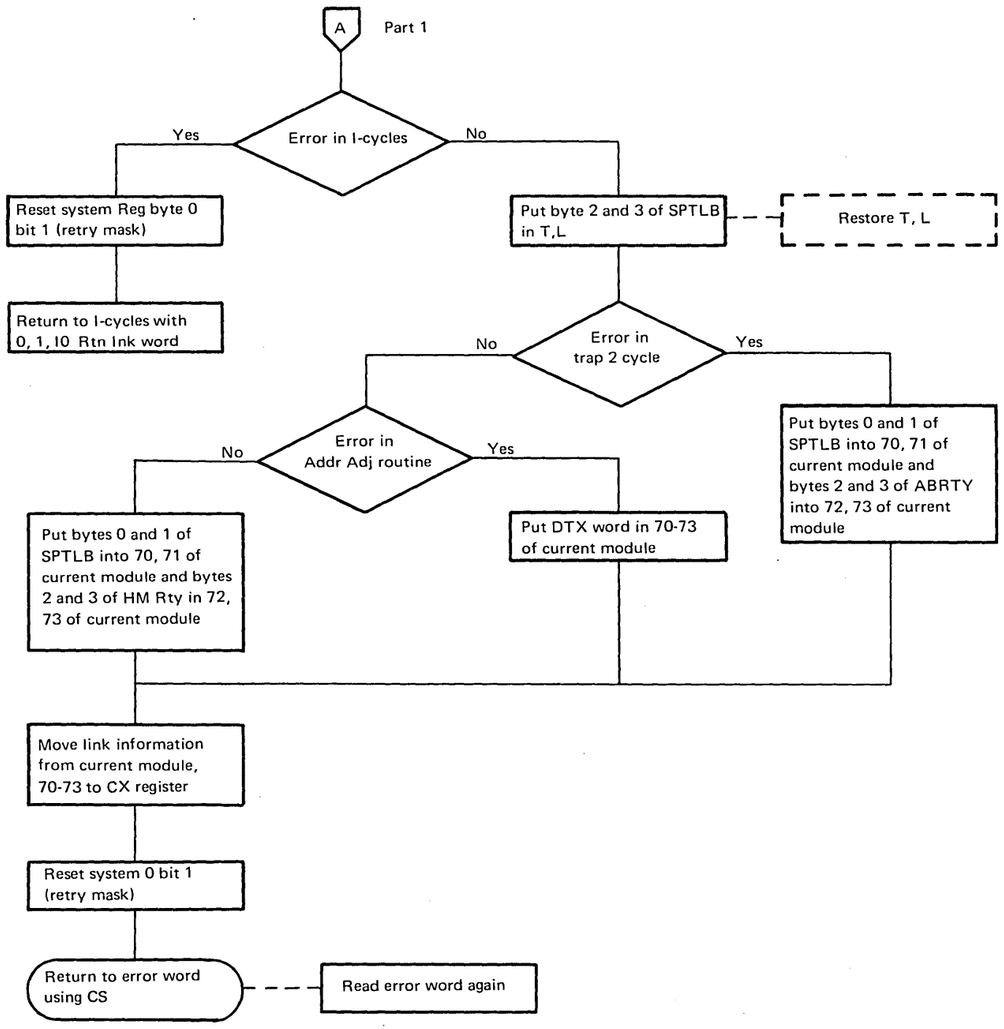
Action depends upon control-register bit 14, hard stop bit.

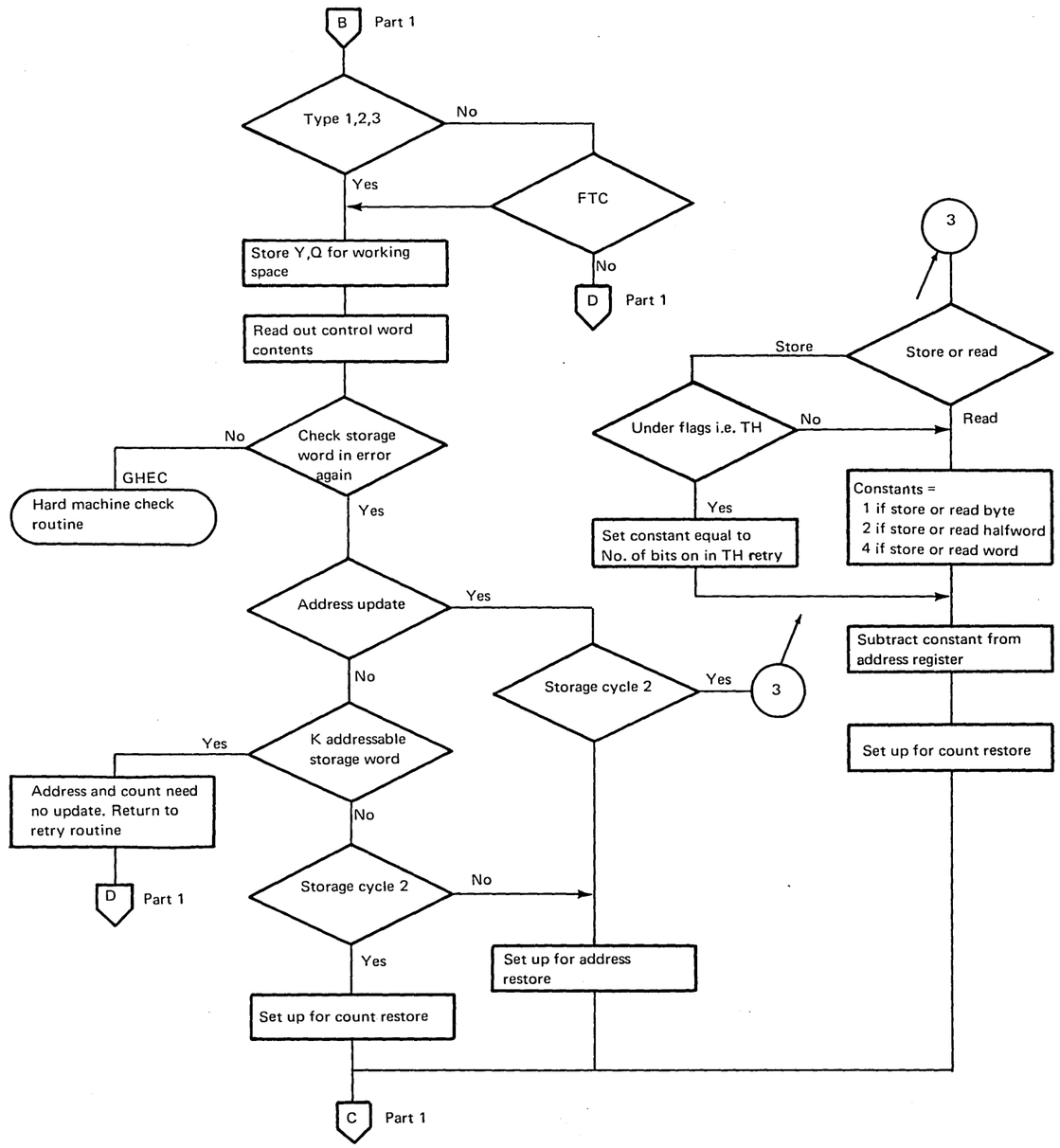
Note 2

If an I-Cycle failure, repeat from beginning of instruction.

Note 3

The last words of the retry routine are unretryable. In addition, any word with the same module address as the last two words is unretryable.


A
Part 1
Error in I-cycles
Yes
No
Reset system Reg byte 0 bit 1 (retry mask)
Return to I-cycles with 0, 1, I0 Rtn Ink word
Put byte 2 and 3 of SPTLB in T,L
Restore T, L
Error in trap 2 cycle
No
Yes
Error in Addr Adj routine
No
Yes
Put bytes 0 and 1 of SPTLB into 70, 71 of current module and bytes 2 and 3 of HM Rty in 72, 73 of current module
Put DTX word in 70-73 of current module
Put bytes 0 and 1 of SPTLB into 70, 71 of current module and bytes 2 and 3 of ABRTY into 72, 73 of current module
Move link information from current module, 70-73 to CX register
Reset system 0 bit 1 (retry mask)
Return to error word using CS
Read error word again

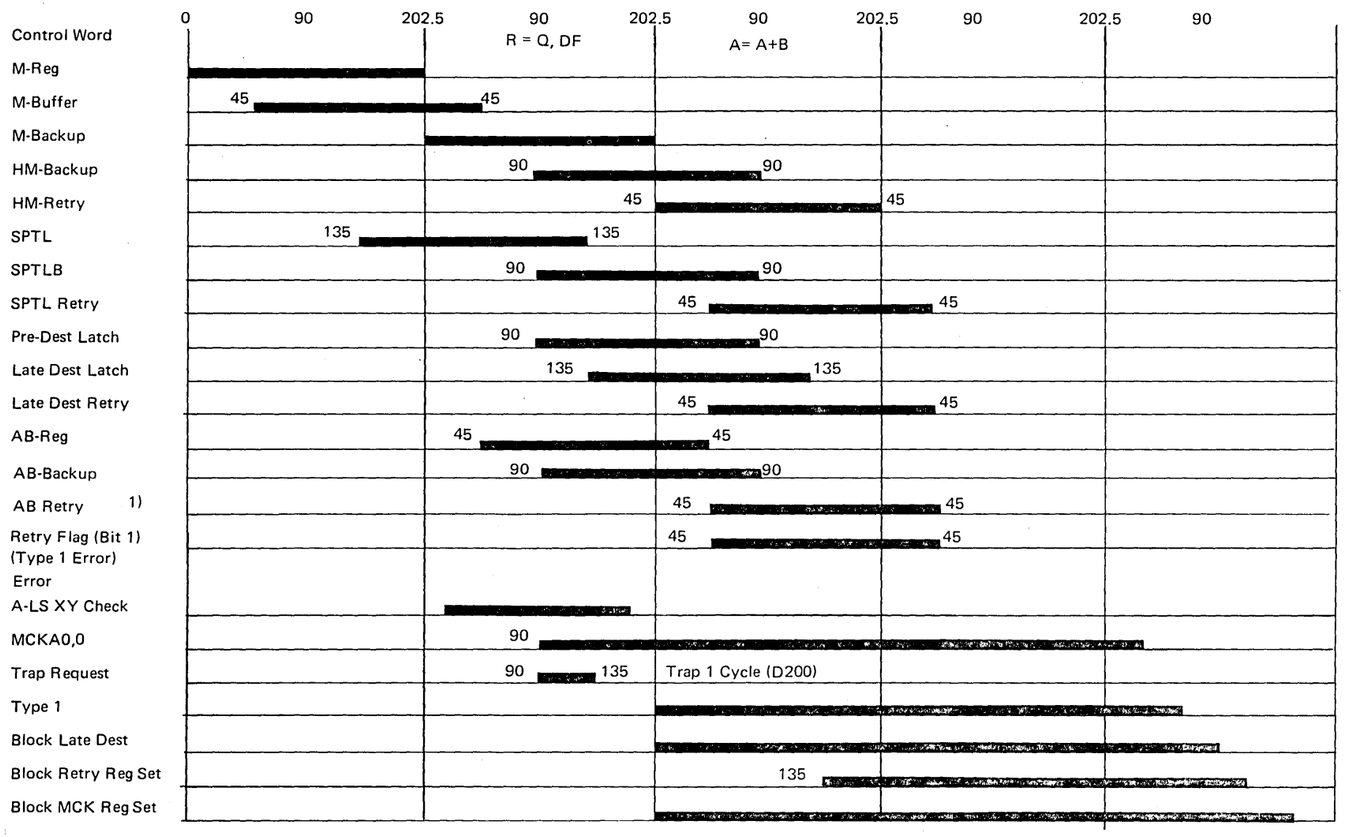

B
Part 1
Type 1,2,3
No
Yes
FTC
No
D
Part 1
Store Y,Q for working space
Read out control word contents
Check storage word in error again
No
GHEC
Hard machine check routine
Yes
Address update
Yes
No
K addressable storage word
Yes
Address and count need no update. Return to retry routine
D
Part 1
No
Storage cycle 2
No
Yes
Set up for count restore
C
Part 1
Storage cycle 2
Yes
3
Set up for address restore
3
Store or read
Store
Read
Under flags i.e. TH
No
Yes
Set constant equal to No. of bits on in TH retry
Constants =
1 if store or read byte
2 if store or read halfword
4 if store or read word
Subtract constant from address register
Set up for count restore

RETRY TIMINGS

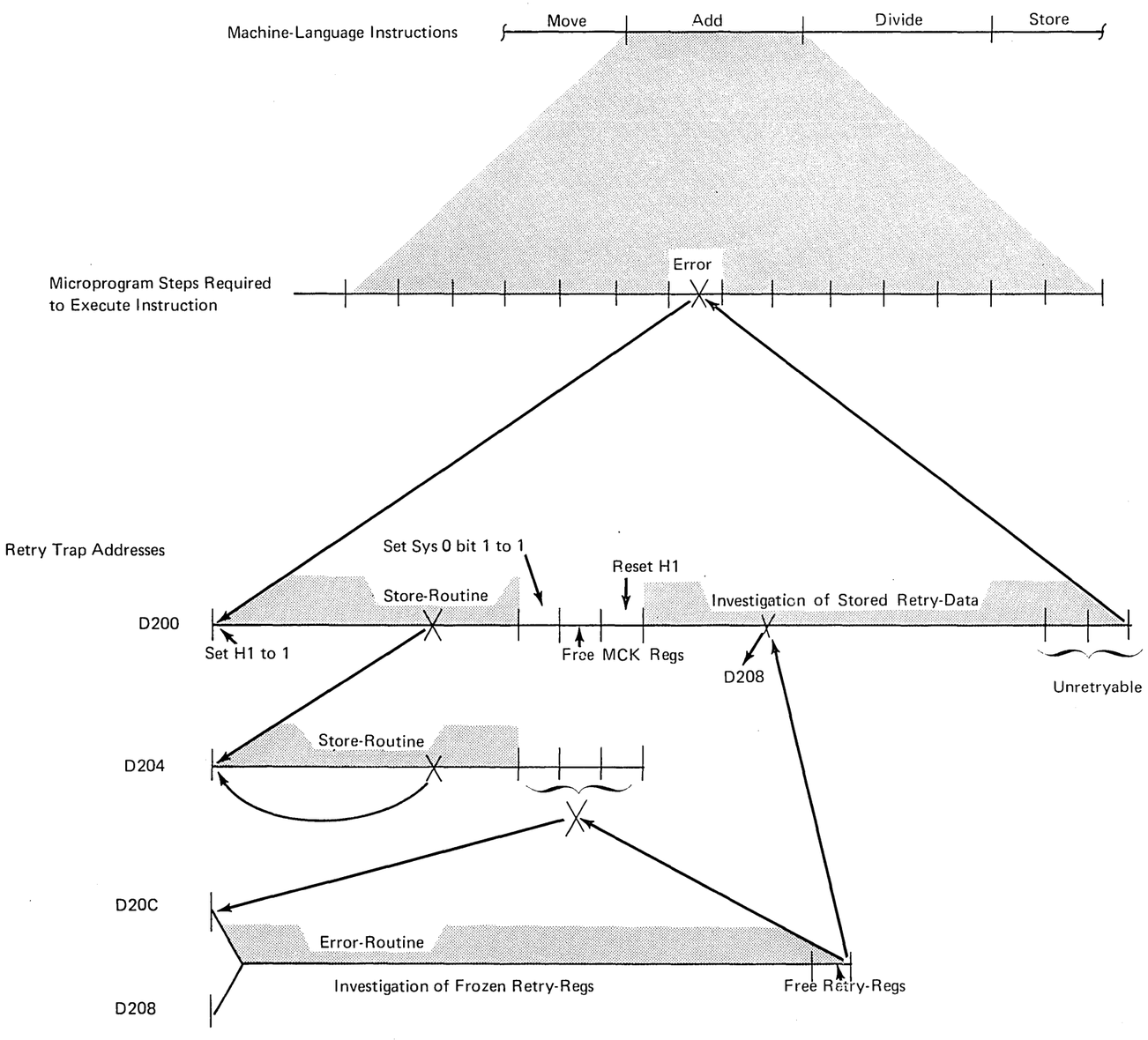
During normal operation, the retry registers save information from the previous cycle to allow retry of a failing control word. Also, retry flag bits are set to determine destination (local storage or external) in the event of error.

In the control-word example shown, R=Q,DF (Word Move—Version 0), the contents of local-storage Q-register are read out to the B-register. The contents of local-storage R-register are read out to the A-register and also gated to the AB-retry. The original contents of the destination address (R-register) must be saved in case of error.

The example shows an error occurring during the readout of the A-source data as indicated by A-LS XY Check (Type 1 Error). The next control word is read out and executed; however, the data is not destined (except in the case of SPTL or H-registers). The next cycle becomes a trap-1 cycle. The normal destination of the contents of the Z-register to the local-storage R-register is blocked. Also, the setting of the retry registers is blocked to retain information pertaining to the word in error. However, SPTL retry contains values relating to the control word prior to the one in error. This is necessary in case the control word in error used SPTL to generate source or destination addresses.



Errors During Retry



TRAP ADDRESSES

D200 The microprogram traps to this address when the first error is recognized. The retry and machine-check registers are frozen, and their contents stored in control storage. The microword is again performed. If the problem is circumvented, it returns to the normal microprogram routine.

D204 The microprogram traps to this address if another error occurs during the store routine of a previous error. It consists of identical microprogram steps as the routine for D200; however, it is stored in a different area of control storage. This is done in case the routine at D200 got damaged along with the original microprogram (noise burst).

D208 The microprogram traps to this address if another error occurs during the investigation of the stored retry data. The routine operates with the information contained in the retry registers.

D20C The microprogram traps to this address if another error occurs after the store routine but before the investigation of the stored retry data. It is handled the same as the D208 traps.

Note: A hardware counter keeps track of the number of traps taken within a given machine-language instruction. After 256 attempts, a machine-check trap occurs to prevent the machine from continuously retrying an error.

MACHINE-CHECK INTERRUPTIONS

- Machine-check interruptions are used to notify the system control program that a machine error has occurred.
- A soft machine check indicates that the error has been retried and corrected.
- A hard machine check indicates that the error was unretryable or that the number of retries exceeded the retry count.
- Information needed to perform a software recovery log to main storage starting at location 232.

HARD MACHINE CHECK

A hard machine-check condition results in a machine-check interruption, only when both PSW bit 13, and the associated subclass mask bit (control Reg 14 bits 4-7) are set to one. The PSW reflecting the point of interruption is stored in location 48 as the old PSW and information needed for performing the software recovery is stored beginning at main-storage location 232. This information includes all control registers, general registers, floating-point registers, region code, the failing storage address, and the machine-check interruption code (MCIC) consisting of eight bytes. In addition, when allowed by the Machine Check Extended Log (MCEL) mask bit (CR 14, bit 1) the machine-dependent extended logout occurs starting at the address indicated by the machine-check log pointer (control register 15, bits 8 through 31).

The interrupt terminates execution of the current instruction and eliminates the program and supervisor-call instructions that would occur as a result of the instruction. The extent and accuracy of the resulting action depends on the nature of the malfunction. Normally the old PSW is stored, and the machine-dependent and independent logouts are stored. When such damage condition occurs during the execution of a system function, such as an interruption or a timer update, the sequence is not completed.

If the machine-check bit (PSW bit 13) is set to zero, a hard machine-check condition does not result in a machine-check interruption. The subsequent action depends on the setting of the hard stop bit (control register 14, bit 0) as follows.

1. When the hard stop bit is set to zero, the machine-check condition is accumulated and processing continues at the point of error.
2. When the hard stop bit is set to one, processing stops immediately and the CPU enters the hard stop state.

A soft machine-check condition never causes the CPU to enter a hard stop state.

SOFT MACHINE CHECK

A soft machine-check condition results in a machine-check interruption only when both PSW bit 13, and the associated subclass mask bit (control Reg 14 bits 4-7) are set to one. Soft machine-check interruptions do not terminate the current instruction. Such interrupts are delayed until the current instruction comes to a normal ending and any associated program or supervisor-call interruptions have been taken. Program or supervisor-call interruptions are never ignored or eliminated by a soft machine-check interruption. When a soft machine-check condition is caused by a system function, the machine-check interruption takes place after the system function has been completed. When a soft machine-check condition is detected during execution of the interruption procedure of a previous soft or hard machine check, the soft condition detected is not accumulated.

Machine checks that occur in a Wait state or instruction step are handled the same way that they are handled in a Run state.

An equipment malfunction detected while the CPU is in the stopped state causes a pending machine-check condition. If the malfunction affects an I/O operation that is being executed with the CPU in the stopped state, the condition is either (1) accumulated as a pending machine-check condition or (2) indicated in the status information associated with the termination of the I/O operation.

Mask Bit(s)	Interrupt Type and Cause	Machine Check
PSW 13 and R **	System Recovery ● CPU error corrected by retry ● Intermittent single-bit processor or control-storage error corrected.	Soft
PSW 13 and E **	Interval Timer Damage	Hard *
PSW 13 and E **	Time of Day Clock Damage	Hard *
PSW 13	System Damage ● Irreparable hardware malfunction.	Hard
PSW 13	Instruction Processing Damage One of the following occurs during instruction execution: ● Unretryable CPU error. ● Uncorrectable CPU error. ● Multiple-bit processor or control storage error. ● Storage-protect key failure.	Hard

* Occurs after current instruction is completed.

** R = Control Reg. 14 bit 4
E = Control Reg. 14 bit 6

MACHINE-CHECK LOGOUT

- Storing information into main storage as a result of a machine error is referred to as a machine-check logout. When a machine-check logout occurs during the machine-check interruption, it is called synchronous. The Model 145 performs only synchronous logouts. (Note: System/370's perform asynchronous logouts as well as synchronous.)
- Machine-check logout is stored at the location specified by control register 15 (system reset to 512 decimal).

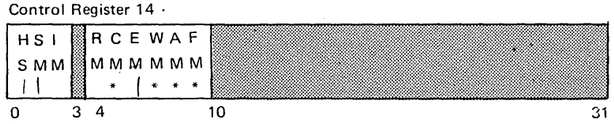
MACHINE-CHECK CONTROL REGISTERS

- Two registers are provided to control machine-check logouts. (Control Registers 14 and 15).
- The registers reside in the F4 module of control storage. Control registers can be displayed on the PRKB by the mnemonic DC.
- Two instructions are provided: Load Control and Store Control. Load Control moves control information from main storage to the control registers. Store Control moves control information from the control registers to main storage. The registers are also set by system reset.
- On a machine-check logout, the registers are stored in the control-register save area (Location 448-512).

Control Register 14

Control register 14 contains mask bits that specify the conditions that will cause machine-check interruptions. It also contains mask bits to control the conditions under which a logout may occur. Bits 3 and 10 to 31 are reserved.

Control Register 14



HS (Hard Stop)

Bit 0 controls the system action taken when a hard machine check occurs during the machine-check microroutine, or when PSW bit 13 is off. If the hard stop bit is on, the machine comes to a hard stop. If the bit is off, the machine will continue to run. The hard stop bit is set on by system reset.

SM (Synchronous Machine-Check Extended Logout Mask)

Bit 1, if on, allows extended logout information to be placed into the main-storage location specified by control register 15. If the bit is off, no extended logout takes place. The SM bit is set on by system reset.

IM (Input/Output Extended Logout Mask)

Bit 2, if on, allows channel logout into the I/O extended logout area as part of an I/O interruption. When bit 2 is off, I/O extended logouts cannot occur. The IM bit is set to zero by system reset.

Bit 3 is reserved.

RM (Recovery Report Mask)

Bit 4, if on, allows a soft machine-check interruption to occur after the first retry of a hardware error or after a single ECC error in record mode. If the bit is off, no interruption occurs. The RM bit is set off by system reset.

CM (Configuration Report Mask)

Bit 5 is set off by system reset and is not used on the Model 145.

EM (External Damage Report Mask)

Bit 6, if on, allows a hard machine-check interruption at the end of the current instruction if interval timer damage, time-of-day clock damage, or external damage occurs. If this bit is off, no interruption occurs. The EM bit is set on by system reset.

WM (Warning Mask)

Bit 7 is set off by system reset and is not used on the Model 145.

AM (Asynchronous Machine-Check Logout Mask)

Bit 8 is set off by system reset and is not used on the Model 145.

FM (Asynchronous Fixed Logout Mask)

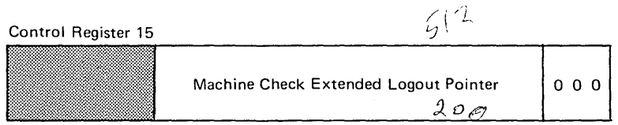
Bit 9 is set off by system reset and is not used on the Model 145.

Bits 10-31 are reserved.

Control Register 15

Bits 8-28 of control register 15, with three low-order zeros appended, specify the starting location of the machine-check extended logout area. Bits 0-7 and 29-31 are reserved. The contents of control register 15 are set to 512 (decimal) by System Reset.

Control Register 15



LOGOUT AREAS

CPU-Independent Logout

Locations 232-511 of main storage are reserved for logout. A logout to this area occurs when any type of machine-check interruption is taken.

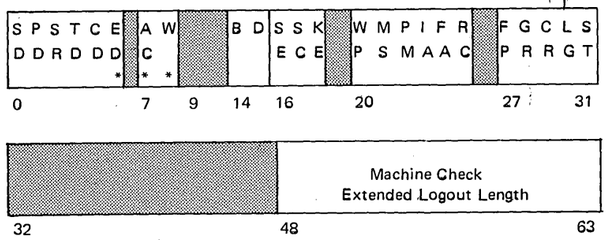
Dec.	Hex.		
232 236	E8 EC	**Machine-Check Interruption Code**	A
240 244	F0 F4		
248	F8	00000000 **Failing-Storage Address**	B
252	FC	**Region Code**	
256 260 264 268	100 104 108 10C	**Fixed Logout Area** Note: When CHECK CONTROL is set to STOP AFTER LOG, the I/OEL pointer is ignored and channel logouts are stored, starting at 256.	
340 344 348	154 158 15C		
352 356 360 364 368 372 376 380	160 164 168 16C 270 174 178 17C	Floating-Point Register Save Area	C
384 388 392 396	180 184 188 18C	General-Register Save Area	C
436 440 444	1B4 1B8 1BC		
448 452 456 460	1C0 1C4 1C8 1CC	Control-Register Save Area	C
500 504 508	1F4 1F8 1FC		
512 516	200 204	Machine-Dependent Logout (192 bytes)	

Machine-Check Interruption Code

Programming systems support of machine checks is enhanced by the additional information given to the programmer by Machine-Check Interruption Code (MCIC).

A

Machine-Check Interruption Code



Note: Bits 0-8 Subclass
Bits 14-15 Time of Interruption Occurrence
Bits 16-18 Storage Error Type
Bits 20-31 Machine-Check Code Validity Bits
* These bits are not used on the Model 145.

SUBCLASS

Bits 0-8 identify the machine-check conditions causing the interruption. At least one bit will be stored as a "one" in the subclass field. When multiple errors have occurred, several bits may be set to one.

SD (System Damage): Bit 0, when one, indicates that interruptions may have been lost, or that damage has occurred that cannot be isolated to one or more of the less severe machine-check damage subclasses (EX clock sync check).

PD (Instruction Processing Damage): Bit 1, when one, indicates that a malfunction has been detected that may have caused incorrect resutls in the processing of instructions.

SR (System Recovery): Bit 2, when one, indicates that errors were detected but have been successfully corrected or circumvented without loss of system integrity. The indication of system recovery does not imply storage logical validity, or that the fields stored as a result of the machine-check interruption are valid.

TD (Timer Damage): Bit 3, when one, indicates that damage has occurred to the timer or to location 80.

CD (Time-of-Day Clock Damage): Bit 4, when one, indicates that damage has occurred to the time-of-day clock.

ED (External Damage): The Model 145 does not set the external damage bit.

AC (Automatic Configuration): The Model 145 does not set the automatic configuration bit.

W (Warning): The Model 145 does not set the warning bit.

TIME OF INTERRUPTION OCCURRENCE

Bits 14 and 15 of the machine-check interruption code indicate when the interruption occurred in relation to the error.

B (Backup): Bit 14, when one, indicates that the point of interruption is at a hardware checkpoint before the point of error. This bit is set only when retry is unsuccessful. When the backup bit is one, a valid instruction address stored in the machine-check old PSW points to the instruction in which the error occurred. If the backup bit is zero, a valid instruction

IA (Instruction Address Validity): Bit 23, when one, indicates that the instruction address in the old PSW accurately reflects the point in the instruction sequence at which the interruption occurred. If the backup bit is one, a valid instruction address points to an instruction in error. If the backup bit is zero, a valid instruction address points to an instruction following the error.

FA (Failing-Storage Address Valid): Bit 24 when one, indicates that the failing-storage address is valid.

RC (Region Code Valid): Bit 25 indicates that a valid region code has been stored.

FP (Floating-Point Registers Valid): Bit 27, when one, indicates that the contents in the floating-point register save area accurately reflect the state of the floating-point registers at the interruption point.

GR (General Registers Valid): Bit 28, when one, indicates that the contents stored in the general register save area accurately reflect the state of the general registers at the interruption point.

CR (Control Registers Valid): Bit 29, when one, indicates that the contents stored in the control register save area accurately reflect the state of the control registers at the interruption point.

LG (Logout Valid): Bit 30 when one, indicates that the CPU extended logout information was correctly stored.

ST (Storage Logical Validity): Bit 31, when one, indicates that the contents of those storage locations that are modified by the instruction processing stream contain the correct information relative to the point of interruption.

MACHINE-CHECK EXTENDED LOGOUT LENGTH

Bits 48-63 of the machine-check interruption code indicate the length in bytes of the information stored in the extended logout area starting at the location specified by the machine-check extended logout pointer. When no extended logout has occurred, this field is set to zero.
address points to an instruction beyond the error.

D (Delayed): Bit 15, when one, indicates that some or all of the information stored as a result of the interruption was delayed in being reported because the CPU was disabled for that type of interruption at the time the error was detected.

STORAGE ERROR TYPE

Bits 16-18 of the machine-check interruption code are used to indicate errors that occurred in main storage or in a key in storage as a result of requests. The failing-storage address field, when indicated as valid, identifies the area in storage found to be in error. The portion of the system affected by the storage or protection error is indicated in the subclass field of the machine-check interruption code. Storage or protection errors that occur on prefetched or unused data are not indicated.

SE (Storage Error Uncorrected): Bit 16, when one, indicates that a reference to storage caused or resulted in, the detection of damaged data that could not be corrected.

SC (Storage Error Corrected): Bit 17, when one, indicates that a reference to storage caused, or resulted in, the detection of an error that was corrected.

KE (Key in Storage Error Uncorrected): Bit 18, when one, indicates that a reference to a key in storage caused or resulted in the detection of an uncorrectable error in the key in storage. The keys in storage are not checked for errors during storage references when the PSW key or channel key referring to storage is all zeros.

MACHINE-CHECK CODE VALIDITY BITS

Bits 20-31 of the machine-check interruption code are validity bits. With the exception of the storage logical validity bit (bit 30), each bit indicates the validity of a particular field stored during the machine-check interruption. When a validity bit is one, it indicates that the error conditions did not affect the original information and that no error was detected when the data was stored. When the bit is zero, one or more of the following conditions may have occurred; the original information was incorrect, the original information had parity errors, errors were detected during the storing of the information, or none or only part of the information was stored. Even though the information is unpredictable, the machine will attempt, when possible, to ensure that both the information store and the original register contents will have good parity and thus reduce the possibility of triggering additional errors.

WP (PSW MWP Validity): Bit 20, when one, indicates that bits 12-15 of the machine-check old PSW are valid.

MS (PSW Masks and Key Validity): Bit 21, when one indicates that all PSW bits other than interruption code, ILC, MWP, instruction address, condition code, and program mask of the machine-check old PSW are valid.

PM (Program Mask and Condition Code Validity): Bit 22, when one, indicates that the program mask and condition code in the machine-check old PSW are valid.

B

MACHINE-CHECK EXTENDED INTERRUPTION INFORMATION

The machine-check extended interruption information consists of five fields, which are stored at machine-check interruption time. Each of these fields has a validity bit associated with it in the machine-check interruption code. If for any reason the machine cannot store one of these fields or cannot store the field validity, the associated validity bit is set to zero.

Failing-Storage Address: When a storage error uncorrected, storage error corrected, or key-in storage error uncorrected has been indicated, the failing-storage address is stored in bits 8-31 of the word at location 248. Bits 0-7 of the word are set to zero. In the case of storage errors, the failing-storage address may point to any address within the ECC block. For key-in storage error uncorrected, the failing-storage address may point to any address within the 2048-byte block of storage associated with the key-in storage that is in error.

Region Code: The word at location 252 contains information that more specifically defines the location of the error.

Bytes 0-1 Control-storage address

When:

1. Control-storage single-bit failure (these bytes not logged in threshold mode).
2. Control-storage double-bit failure.

Byte 2 ECC syndrome bits

When:

1. Control-storage single-bit failure (this byte not logged in threshold mode).
2. Main-storage single-bit failure.

Byte 3

Bits 0-3 not used

Bit 4 =1 control-storage double-bit failure.

Bit 5=1 Interruption due to control-storage threshold limit being exceeded (bit 7 of this byte should also be 1).

Bit 6=1 Intermittent single-bit failure in control or main storage.

Bit 7=0 Main-storage intermittent single-bit failure (in conjunction with bit 6).

Bit 7 = 1 Control-storage intermittent single-bit failure (in conjunction with bit 6).

C

REGISTER SAVE AREA

On all machine-check interruptions, the addressable registers are saved sequentially in storage. Floating-point registers 0, 2, 4, and 6 are stored starting at location 352, general registers 0-15 are stored starting at location 384, and control registers 0-15 are stored starting at location 448. Locations assigned to control registers that are not implemented are set to zero.

CPU-Dependent Logout

The model-dependent CPU extended-logout area begins at the address specified in control register 15, which is set by the hardware to decimal location 512* by an IPL or on a system reset. The length of this extended-logout area on the Model 145 is 192 bytes. A logout to the extended area occurs for all kinds of machine-check interrupts if allowed by the Synchronous Machine-Check Extended Logout Mask (MCEL-Control Register 14, bit 1).

* Dec 512 = Hex 200

Model 145 Machine Dependent Log

Word Name	Byte 0	Byte 1	Byte 2	Byte 3	Original Location
Retry Counts	Note 1				CS FF84
MCKA	MCKA0	MCKA1	MCKA2	MCKA3	EXT 07
MCKB	MCKB0	MCKB1	MCKB2	MCKB3	EXT 06
ABRTY	ABRTY0	ABRTY1	ABRTY2	ABRTY3	EXT 18
SPTLB	SRTY	PRTY	TRTY	LRTY	EXT 19
HMRTY		HRTY	MRTY2	MRTY3	EXT 1A
CPURTY	BYTDST	RTYFLG	LSDST	EXTDST	EXT 1B
Control	(Control Word in Error)				
SYSREG	SYS0	SYS1	SYS2	H-REG	EXT 05
I-REG	KEY REG	Instruction Counter			EXP LS 50
U-REG	ILC CC Prog Mask	AMWP	Op Code	Immed Byte	EXP LS 53
W-REG	(First Operand Address)				EXP LS 52
V-REG	(Second Operand Address)				EXP LS 51
X-REG	(CPU Working Area)				LS 11
R-REG	(CPU Working Area)				LS 15
Y-REG	(CPU Working Area)				LS 16
Q-REG	(CPU Working Area)				LS 17
IBU-REG					EXP LS 54
TR-REG					EXP LS 55
(Spare)					
SN-Reg					EXP LS 78
PN-REG					EXP LS 79
WK-REG					EXP LS 7A
DM-REG	(Adjustment Factor)				LS 3A
RW-REG	(Address Adjustment Working)				LS 3B
CPU-REG					
PSWCTL-REG			MSKA	MSKB	EXT 10

Note 1 — Stop After Log ID for SEREP

Retry Counts — Interruption Code in old mck. PSW=145F=CPU

Byte 0 — Log present and CPU check light

Bits 0-3 — Zero

Bits 4-7 — Retry Attempts for Current Instruction

Bytes 1 and 2 — Zeros

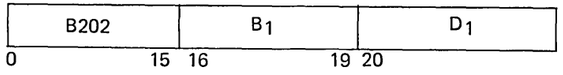
Byte 3 — Number of Instructions Retried

CPU Identification

The instruction STORE CPU ID specifies the identity of the CPU and the amount of storage that must be allocated for the machine-check extended logout.

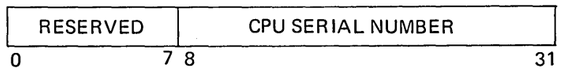
STORE CPU ID

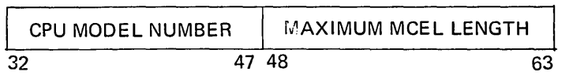
STIDP D_1 (B_1)



Information identifying the CPU is stored in the eight-byte field designated by the operand address. The instruction STORE CPU ID is executed only in the supervisor state.

The format of the information is as follows:





Bits 0-7 are reserved and are set to zeros.

Bits 8-31 contain the CPU serial number stored in packed decimal format.

Bits 32-47 contain the CPU model number identification stored in packed decimal format. (0145).

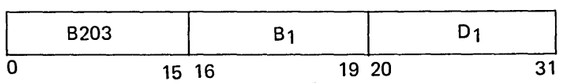
Bits 48-63 contain the length in bytes of the longest machine-check extended logout that can be stored by the machine. (00C0).

Channel Identification

The instruction STORE CHANNEL ID specifies the identity of the designated channel.

STORE CHANNEL ID

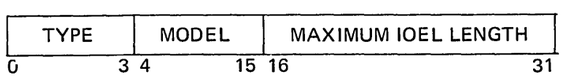
STIDC D_1 (B_1)



Information identifying the addressed channel is stored in the four-byte field at location 168. The instruction STORE CHANNEL ID is executed only when the CPU is in the supervisor state.

Bit positions 16-23 of the sum formed by the addition of the contents of register B_1 and the contents of the D_1 field identify the channel to which the instruction applies.

The format of the information stored at location 168:



Bits 0-3 identify the channel type. When a channel can operate as more than one type, the code stored identifies the channel type at the time the instruction is executed. The following codes are assigned.

0000 Selector
0001 Byte Multiplex
0010 Block Multiplex

Bits 4-15 identify the model.
Note: On Model 145, IFA stores 00A.

Bits 16-31 contain the length in bytes of the longest IOEL that can be stored by the channel during an I/O interruption. If the channel never stores logout information using the IOEL pointer, this field is set to zero.

Note: (16-31) is 0060 for all channels.

Channel-Dependent Logouts

The channel logs are located at the address designated by the I/O extended logout pointer in storage location 172.* A logout to this area is allowed if the I/O extended logout mask (control register 14 bit 2) is one. The length of the channel logout area is 96 bytes. If the pointer is set such that any part of the log would be out of storage, no part of the log is stored.

The channel attempts to log anytime a CSW is stored with an interface-control check or a channel-control check. The logging is, therefore, done both at I/O interruption time and CC1 time to an I/O instruction under control of CR14 bit 2. A channel error does not cause a machine-check interrupt.

Channel logs are lost under the following circumstances:

1. CR 14 bit 2 is set to zero.
2. The IOEL pointer is pointing outside the limits of storage.
3. A second channel error occurs before the first is cleared. In this case, the first log is preserved.
4. A machine-check log is pending or a machine check occurs before the I/O log is cleared. In this case, the first ten log words are valid, but the fourteen CPU words that may be appended will be taken by the machine check.

Words 4 and 5 contain the machine-check registers that give the status of the processor error-detection circuitry. Words 11 through 19 are the processor registers that are stored in case of unsuccessful microretry in the channel.

* If the check control switch is set to the Stop After Log position, ignore the I/OEL Pointer (Loc 172) and store into 256.

MPX Channel Machine Dependent Log

Word Name	Byte 0	Byte 1	Byte 2	Byte 3	Original Location	
MA	Note 1	Unit Addr	UCW Address		LS 18	
MBS	Int. Buffer ADDR	Status	Seq No.		LS 19	
MC	Key	Next CCW Addr			LS 1B	
MCKA	MCKA0	MCKA1	MCKA2	MCKA3	EXT 07	
MCKB	MCKB0	MCKB1	MCKB2	MCKB3	EXT 06	MPX
MPX	MTO	MTI	MBI	MBO	EXT 0E	
DOC	TI	TA	TT	TE	EXT 0F	
MD			Catalog No.		LS 1C	
MF	Flags, Ops	UCW/CHAN STATUS	Count		LS 1D	
(Spare)						
Retry Counts	Note 2				CS FF84	
ABRTY	ABRTY0	ABRTY1	ABRTY2	ABRTY3	EXT 18	
SPTLB	PRTY	DRTY	TRTY	LRTY	EXT 19	
HMRTY		HRTY	MRTY2	MRTY3	EXT 1A	
CPURTY	BYTDST	RTYFLG	LSDST	EXTDST	EXT 1B	
Control Word	(Control Word in Error)					
SYS REG	SYS0	SYS1	SYS2	H-REG	EXT 05	
I REG	KEY REG	Instruction Counter			EXP LS 50	CPU
U REG	ILC CC Prog Mask	AMWP	Op Code	Immed Byte	EXP LS 53	
(Spare)						
(Spare)						
(Spare)						
(Spare)						
(Spare)						

Catalog Numbers

01	No Op-in time-out
02	Status-in time-out
03	No Addr-in time-out
04	Addr. in bad parity/no addr. match
05	No status-in time-out
06	Op-in time-out
07	Bad parity status-in
08	Op-in failed to fall in data loop
09	Disconnect-in during Init. selection
0A	Bad parity on status (not initial)
0B	Service-in but expected status-in
13	No Addr-in or Sel-in on share request
1C	False share request from Doc Cons
2F	Sel-in during chaining

Stop After Log ID for SEREP

Interruption code in old Mck. PSW = 1450 = multiplex. (145F-CPU)
Log present and CPU check light

I/O Log Start Addresses

Stop after log - MS 100 (Hex) or IOEL pointer = MS AD-AF.

Note 1 MA-REG
Byte 0
Bit 0 = 1 MPX Log Valid
Bit 0 = 0 MPX Log Invalid

Note 2 Retry Counts
Byte 0
Bits 0-3 Zero
Bits 4-7 Retry Attempts for Current Instruction
Byte 1 and 2 Zeros
Byte 3 Number of Instructions Retried

IFA Extended Logout

Word Name	Byte 0	Byte 1	Byte 2	Byte 3	Original Location	
FBAK	Log Length Note 1	FCH	FCL	FOP	EXT 20	IFA
FSTAT	FFL	FCS	FST	FGL	EXT 22	
FTAG	FTO	FTI	FBO	FDR	EXT 23	
MCKA	MCKA0	MCKA1	MCKA2	MCKA3	EXT 07	
MCKB	MCKB0	MCKB1	MCKB2	MCKB3	EXT 06	
FDRL						
FD	Protect	Data Address			LS28	
FC			Count		LS29	
FM	Protect	CCW Address			LS2A	
FW	Unit Address				LS2B	
Retry Counts	Note 2					CPU
ABRTY	ABRTY0	ABRTY1	ABRTY2	ABRTY3	EXT 18	
SPTLB	SRTY	PRTY	TRTY	LRTY	EXT 19	
HMRTY		HRTY	MRTY2	MRTY3	EXT 1A	
CPURTY	BYTDST	RTYFLG	LSDST	EXTDST	EXT 1B	
Control Word	(Control Word In Error)					
SYS REG	SYS0	SYS1	SYS2	H-REG	EXT 05	
I REG	KEY REG	Instruction Counter			EXP LS 50	
U REG	ILC CC Prog Mask	AMWP	Op Code	Immed Byte	EXP LS 53	
(Spare)						
(Spare)						
(Spare)						
(Spare)						
(Spare)						

Note 1

FBAK Byte 0 = 28 Only IFA Log Valid
Byte 0 = 4C IFA and CPU Log Valid

Note 2

Retry Counts
Byte 0
Bits 0-3 Zero
Bits 4-7 Retry Attempts for Current Instruction
Bytes 1 and 2 Zeros
Byte 3 Number of Instructions Retried

Selector Channel /Block Multiplex Channel Dependent Log

Word Name	Byte 0	Byte 1	Byte 2	Byte 3	Original Location	
GBS	Log Length Note 1	GBF	GCT	GBD	EXT 31 (SX2)	
GSTAT	GF	GE	GS	GL	EXT 32 (SX2)	SX
GTAG	GTO	GTI	GO	GR	EXT 33 (SX2)	
MCKA	MCKA0	MCKA1	MCKA2	MCKA3	EXT 07	
MCKB	MCKB0	MCKB1	MCKB2	MCKB3	EXT 06	
GDRL						
GD	CHANNEL WORKING REGISTERS				LS20 (SX2)	
GC	CHANNEL WORKING REGISTERS				LS21 (SX2)	
GM	CHANNEL WORKING REGISTERS				LS22 (SX2)	
GW			Catalog No.		LS23 (SX2)	
Retry Counts	Note 2				CS FF84	CPU
ABRTY	ABRTY0	ABRTY1	ABRTY2	ABRTY3	EXT 18	
SPTLB	SRTY	PRTY	TRTY	LRTY	EXT 19	
HMRTY		HRTY	MRTY2	MRTY3	EXT 1A	
CPURTY	BYTDST	RTYFLG	LSDST	EXTDST	EXT 1B	
Control Word	(Control Word in Error)					
SYS REG	SYS 0	SYS 1	SYS 2	H-REG	EXT 05	
I REG	KEY REG	Instruction Counter			EXP LS 50	
U REG	ILC CC Prog Mask	AMWP	Op-Code	Immed Byte	EXP LS 53	
(Spare)						
(Spare)						
(Spare)						
(Spare)						
(Spare)						

Note 1

GBS

Byte 0 = 28 Only SX Log Valid
Byte 0 = 4C SX and CPU Log Valid

Note 2

Retry Counts
Byte 0
Bits 0-3 Zero
Bits 4-7 Retry Attempts for Current Instruction
Bytes 1 and 2 Zeros
Byte 3 Number of Instructions Retried

Catalog Numbers

00 Indeterminate
01 Automatic selection failed
02 Halt stop will not set
03 Microcode select failed on halt
04 Interface disconnect failed
05 Selective reset failed
06 Address miscompare on selection
07 Unused
08 Poll control will not set
09 Address parity error
0A Disconnect-in received
0B Hard set of poll control failed
0C Poll control failed on Halt I/O
0D Unsuccessful Microprogram retry

Stop After Log ID For SEREP

Interruption Code in old MCK PSW

1451 = SX1
1452 = SX2
1453 = SX3
1454 = SX4

Log Present, SX-check and CPU checklight

I/O Log Start Address:

Stop After Log : MS 100 (Hex)
Or IOEL Pointer : MS AD-AF

SUMMARY OF THE EXTENDED MACHINE-CHECK HANDLING FEATURE

Following is a summary of the major extensions and changes to the machine-check handling architecture of the System/360.

1. An I/O extended logout pointer is defined at location Hex AC (Decimal 172).
2. An eight-byte machine-check interruption code is defined at location 232.
3. Additional fields are defined that are stored as an extension to the machine-check interruption code.
   a. A four-byte field at location 248 to indicate the address of failing storage or failing key in storage.
   b. A four-byte field at location 252 is allocated for model-dependent indication of the physical location of the error.
   c. A 32-byte field is defined at location 352 in which the floating-point registers are automatically saved during a machine-check interruption.
   d. A 64-byte field is defined at location 384 in which the general registers are automaticaly saved during a machine-check interruption.
   e. A 64-byte field is defined at location 448 in which the control registers are automatically saved during a machine-check interruption.
4. A 96-byte field starting at location 256 is reserved as a fixed logout area.
5. No logout occurs in locations beyond 511 unless so designated by the machine-check extended logout pointer. Control register 15 is assigned as the machine-check extended logout pointer. The contents of the register are set to 512 upon reset.
6. Nine bits are assigned in control register 14 for error-recovery control.
7. A new instruction, STORE CPU ID, is introduced. The instruction causes model identification to be stored at the designated storage location.

STOP AFTER LOG

The System/370 does some error logging that is not done by the System/360. When programs written for the System/360 are used on the System/370, an error log may overlay either the instructions or data of System/360.

The Stop After Log mode of operation avoids this situation by stopping the system after logging takes place. In this mode of operation, the customer has the full capabilities of the retry mechanism to correct errors, but is not exposed to erroneous execution because of logging. After using Stop After Log, the SEREP program may be used to retrieve the log for analysis. The Stop After Log mode (under control of the Stop After Log switch) is not needed for running System/370 programs.

CPU errors are handled the same in the Stop After Log position as in the Process position of the check control switch. This means that machine-check interruptions and logging are done under control of the current PSW and control-register settings. The sequence of events for taking the machine-check interruption is:

1. Put machine-dependent log in main storage at MCEL pointer under control of control register 14 bit 1. After the log is finished, the log present indicator on the console is turned on.
2. Store machine check old PSW.
3. Store machine-independent log and machine-check interruption code (also the failing storage address and region code if appropriate).
4. Load machine check new PSW.
5. Enter check stop state.

To distinguish between a machine-check interruption and an I/O stop, a coded halfword is stored in the machine-check old PSW interruption code (hex locations 32 and 33). The code values are:

1450	Channel 0
1451	Channel 1
1452	Channel 2
1453	Channel 3
1454	Channel 4
145F	CPU

Note that the interruption code is the only portion of the machine-check old PSW that is stored in the case of I/O stops.

Because System Reset sets the control registers to prevent the extended channel logging, the log would not normally be available for analysis when running old (System/360) programs. The Stop After Log switch overrides the control switch and allows extended channel logging.

Stop After Log causes the following action regardless of the system mode or control-register setting.

The IOEL pointer is loaded with the address of the fixed logout area (hex 100). The ECSW is stored and the extended channel log is stored at hex 100. Both the ECSW and extended channel log are always available when running in Stop After Log mode. The CPU places an identifier in hex 32 and 33 to identify the failing channel. The system then stops with at least one red light and the log pending indication.

In Stop After Log mode, the I/O old PSW, the channel status word, and the I/O address are not stored. All information necessary to construct the CSW and I/O address is present in the extended channel log. When a channel error occurs, the Stop After Log may take place immediately, even though channel interruptions are masked off.

Results are unpredictable if the system is switched to Stop After Log mode between the time an error occurs and the error is cleared.

SOFTWARE RECOVERY

The machine-check interruption furnishes a means of reporting equipment malfunction and certain external distrubances, and it supplies the program with the information about the location and the nature of the cause. In some cases, depending on the nature of the malfunction, the system may either take correcting action or circumvent the failing components.

RECOVERY MANAGEMENT SUPPORT (RMS) FOR OS MFT AND MVT

The two RMS routines, machine-check handler (MCH) to handle machine-check interruptions and channel check handler (CCH) to handle certain channel errors are included in MFT and MVT control programs generated for the Model 145.

The two primary objectives of RMS are (1) to reduce the number of system terminations that result from machine malfunctions, and (2) to minimize the impact of such incidents. These objectives are accomplished by programmed recovery to allow system operations to continue whenever possible and by the recording of system status for both transient (corrected) and permanent (uncorrected) hardware errors.

Machine-Check Handler

After IPL of a control program containing Model 145 RMS routines, the recovery mask bit (CR 14, Bit 4) is enabled to permit recording of CPU retry corrections, quiet mode is established for single-bit processor storage corrections, threshold mode is established for single-bit control-storage corrections, and external interruptions are enabled as are CPU and I/O extended logouts. MCH receives control after the occurrence of both soft and hard machine-check interruptions.

SOFT MACHINE CHECKS

When a system recovery soft machine check occurs to indicate a successful CPU retry, MCH formats a recovery report record to be written in the system error recording data set SYS1 . LOGREC. This record contains pertinent information about the error, including the data in the fixed logout area, an indication of the recovery that occurred, identification of the job, job step, and program involved in the error, the date, and the time of day. MCH schedules the writing of the recovery report record and informs the operator that a soft machine check has occurred.

MCH supports an operator mode command that can be used to enable interruptions after intermittent single-bit error corrections. The operator can establish full recording mode for intermittent single-bit processor and/or control-storage ECC corrections. These corrected errors will then be recorded, and the operator will be notified. The operator can also re-establish quiet mode for processor-storage corrections, and threshold mode for control-storage corrections. (The operator is notified when a switch from threshold to quiet mode is made for control-storage corrections.)

A capability for the operator to switch to quiet mode for successful CPU retries is not provided, as is discussed for the DOS MCAR routine; thus, recording always occurs for these errors.

The operator also is informed of the occurrence of a time-of-day-clock damage or an interval-timer damage machine-check interruption. Error recording is performed, after which the system is placed in a wait state if a time-of-day clock error occurred. System operation continues after an interval timer error.

HARD MACHINE CHECKS

When an instruction processing damage hard machine check occurs (uncorrectable or unretryable CPU error, double ECC error, or storage-protection key failure), MCH determines whether the error is one that is correctable by programming, such as a double ECC error or a storage-protection key failure.

The program damage assessment and repair (PDAR) routine of MCH can repair damaged control program storage areas by loading a new copy of the affected module if the module is marked refreshable (it has been written in a read-only manner that allows reloading of the module at any time without altering execution results). Only damaged refreshable modules residing in the control program nucleus, the link pack area, or an SYC transient area are refreshed, if possible. Processing program modules are not refreshed. PDAR also attempts to repair storage-protection keys.

If PDAR cannot correct the error or if the error is an uncorrectable type, PDAR attempts to identify the task associated with the error so that the task can be terminated abnormally. A damage report record that contains both the fixed and CPU extended-logout area data, the recovery action taken, the program identification, the date, and the time of day, is prepared and logged. System operation continues if the error is corrected or if the error task associated with an uncorrectable error can be identified and terminated. System operation halts, and a re-IPL is required if the error involves control storage (a double-bit error), if an uncorrectable error damages a portion of the control program, or if the error cannot be associated with a specific task. The operator is informed of whatever action is taken.

When a system damage hard machine check occurs, programmed recovery is not possible, and MCH places the system in a wait state after a logout and termination procedures are attempted.

MCH for the Model 145 contains model-dependent routines and will not execute correctly on System/360 models or another System/370 model.

Channel-Check Handler

CCH receives control after a channel error causes an I/O interruption. CCH formats both an error information block (containing the limited channel logout data) for use by an ERP routine and a CCH error record for recoding in SYS1.LOGREC. The latter contains status information from the logout area, the ECSW, program identification, date, and time of day.

If CCH determines that operating system integrity has been impaired by the channel error, control is given to MCH for error recording, and system operations are terminated. Otherwise, the error information block and error record are passed to the appropriate device-dependent error-recovery procedure (ERP) that logs the error record and retries the failing I/O operation, using status information from the error information block. If a successful retry occurs, system operation continues. If the error is deemed permanent (uncorrectable), another error record is prepared and recorded by the outboard recorder routine (OBR), and the task involved is abnormally terminated (unless I/O RMS or a user-written permanent error-handling routine is present). The operator is informed of the abnormal termination, and system operation continues.

The CCH routine is structured in a manner that makes it model-independent. A channel/model-independent module resides in the operating system nucleus. The required channel-dependent modules for the Model 145 included in the operating system at system generation time are loaded during the IPL procedure. The nucleus initialization program (NIP), using channel configuration data specified by the user at system generation time and the STORE CHANNEL ID instruction, determines the types of channels present in the system. OS CCH routines are, therefore, compatible for System/370 Models, for System/360 Models 65 and up, and for MP/65 systems.

RECOVERY MANAGEMENT SUPPORT (RMS) FOR DOS

Machine-check analysis and recording (MCAR), channel-check handler (CCH) routines, and the I/O error-recording routines OBR and SDR will be included in a DOS supervisor generated for a Model 145.

Machine-Check Analysis and Recording

After IPL of a control program containing Model 145 RMS routines, mask bits are enabled and control-register values are set to permit machine-check interruptions and logouts to occur as discussed for OS MCH.

When a system recovery soft machine check occurs to indicate a successful CPU retry, an environment record (recovery report), containing pertinent status information from the fixed area, recovery action, program identification, data, and time of day, is constructed by MCAR and written in the environmental recording data set (ERDS), whose symbolic unit name is SYSREC (corresponding to the SYS1.LOGREC recording data set of OS). The operator is informed that a soft machine check has occurred.

Before relinquishing CPU control, MCAR determines whether or not an automatic mode switch from recording mode to quiet (nonrecording) mode should be made for CPU retry recoveries. Quiet mode is established by MCAR (the system recovery mask bit is disabled) if the number of CPU retry corrections that occur during system operation exceeds the established error-count threshold value for these corrections. The IBM-supplied threshold value in the program can be altered during system generation or by the operator mode command during system operation. The operator is informed of any mode switch made by MCAR and can switch back to recording mode any time thereafter. Quiet mode can be used to prevent SYSREC from being filled with CPU recovery reports when a large number of transient errors are occurring.

As described for the OS MCH routine, MCAR also supports an operator mode command to permit the operator to enable interruptions after single-bit intermittent processor and control-storage corrections so that these errors can be logged.

An interruption because of an error in the time-of-day clock or interval timer causes error recording to occur, and system operation continues.

When an Instruction Processing Damage hard machine check occurs (uncorrectable or unretryable CPU error, double-bit storage error, or storage protection key error) during the execution of supervisor (or any privileged) code, the system is placed in a hard wait state after an attempt is made to prepare and record a damage report record. MCAR does not attempt to refresh damaged supervisor code. The occurrence of an Instruction Processing Damage interruption during processing program execution always results in termination of the task involved after error recording occurs.

MCAR performs repair procedures if a storage-protection key failure or multiple-bit processor storage error occurs in a processing program partition. Validation of damaged processor storage is attempted by moving a doubleword of binary zeros and then ones into the area.

A System Damage hard machine check interruption results in an attempt to record the error, followed by system termination (a hard wait state). The operator is informed of whatever action is taken after a hard machine check occurs, including whether or not error recording was successful.

Channel-Check Handler

CCH receives control after a channel error occurs. It records the error in SYSREC and passes the ECSW and other pertinent status information to the appropriate I/O error-recovery routine (ERP) unless analysis of the error indicates that system operation cannot continue (the error involved SYSRES, for example). If the ERP can correct the error, operations continue. If a permanent channel error exists, CCH records the error and cancels the partition affected. The operator is notified. System termination occurs, (1) if a hard channel error occurs during the access of program phases or critical data contained on SYSRES, (2) if two channels are damaged at the same time, or (3) if more than four channel errors are outstanding concurrently.

The recovery support provided by the MCAR and CCH routines represents an extension of the facilities provided by the optional MCRR routine of DOS, which is available for System/360 models and which does not contain any repair or channel-retry procedures.

ERROR-RECOVERY PROCEDURES (ERPS)--OS AND DOS

These device-dependent error routines are a standard part of the control program generated for any OS or DOS environment. OS ERPs are modified to accept and use limited channel logout ECSW data formatted by the CCH routine after a channel error. The ECSW provided by the DOS CCH routine is handled by a set of completely new CCH ERP routines.

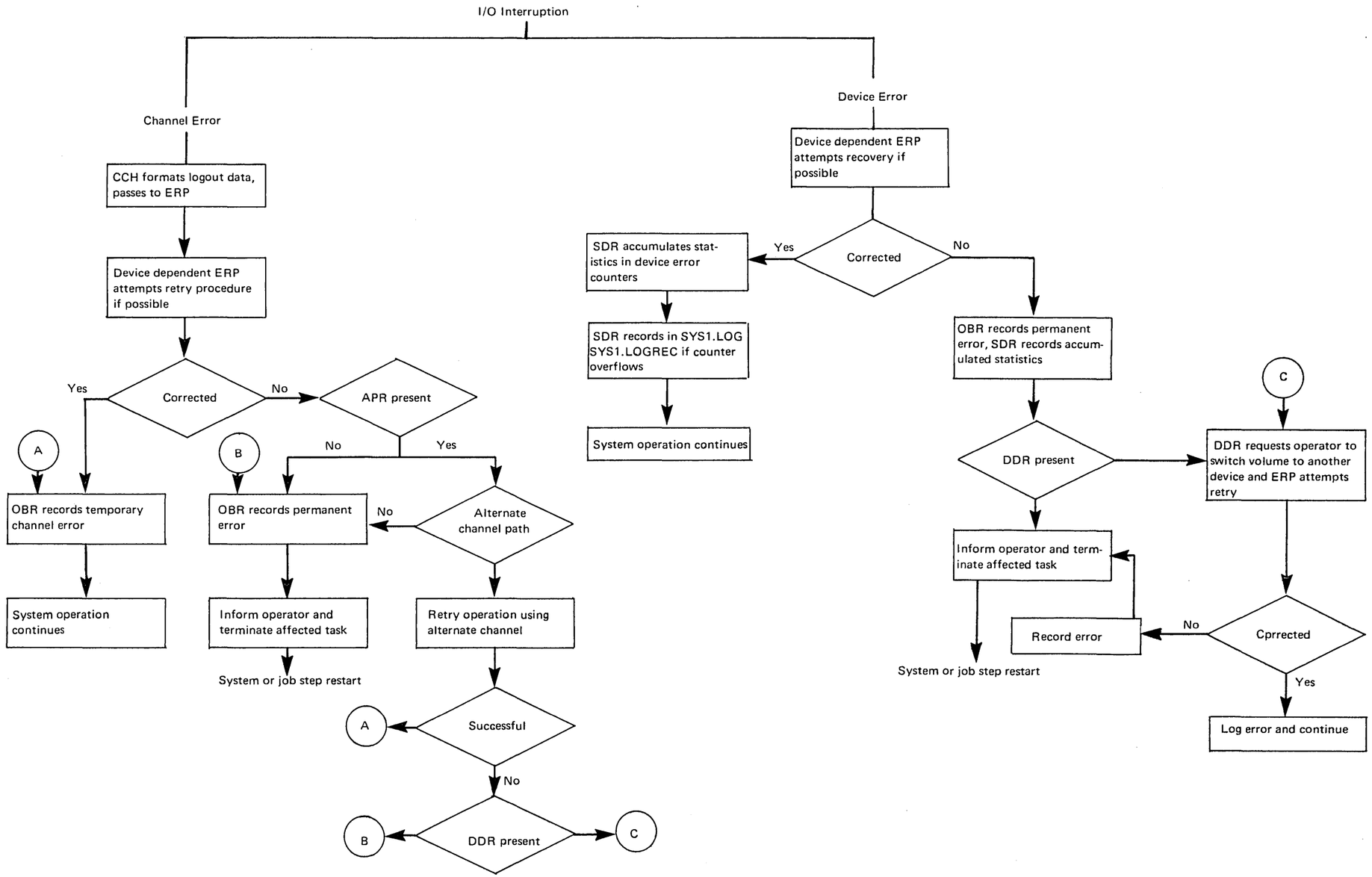
OS ERP routines written for the 3211 Printer, the 3330 facility, and the 2305 facility include support of the larger number of sense bytes provided by the control units of these devices. The DOS ERP routines for the 3211 Printer also support these sense bytes.

When a channel or an I/O device error occurs on a Model 145, the appropriate ERP is scheduled to perform recovery procedures. If the error is corrected, operations continue normally. If the error cannot be corrected (it is permanent), error recording occurs. If I/O RMS or a user-written permanent error-handling routine is not present, the affected OS or DOS task is abnormally terminated. The operator is notified of permanent I/O errors.

STATISTICAL DATA RECORDER (SDR) AND OUTBOARD RECORDER (OBR)--OS AND DOS

OBR and SDR routines are included in all OS control programs. OBR and SDR routines are included in any DOS supervisor generated for a Model 145.

These routines are requested by the ERP routines during their processsing of error conditions. The SDR routine is requested when one of the error statistics counters become full. Counters are maintained in the resident control program storage area for each I/O device in the system configuration. SDR records these statistics in the appropriate SDR summary record for that device contained in the error log data set (SYS1.LOGREC for OS, SYSREC for DOS). This ensures recording of temporary I/O device error data. The OBR routine of OS records both temporary and permanent channel errors (handled by the CCH routine in DOS) and writes an outboard record containing pertinent status data whenever a permanent error occurs for a device. SDR is also executed to write accumulated statistics for that device when a permanent error occurs.



ENVIRONMENT RECORDING, EDIT, AND PRINT PROGRAM (EREP)--OS AND DOS

OS EREP is a standard system utility that can be initiated as a job step via standard job control statements at any time. It contains model-dependent routines and will be extended to handle the status records written by System/370 OS RMS routines. It performs the following:

1. Edits and prints all error records contained in SYS1.LOGREC. These records have been constructed and written by MCH, CCH, OBR, and SDR routines.
2. Accumulates a history of specified record types from SYS1.LOGREC by creating or updating an accumulation data set.
3. Edits and prints a summary of selected records from SYS1 LOGREC or an accumulation data set.

The currently available EREP routine of DOS is a special-purpose utility that can be initiated as a job step via job control statements in the input stream or by an operator command entered via the console. Its function is to edit and print all error records contained in the SYSREC recorder file. EREP will be extended to handle all status records written by DOS Model 145 recovery routines (MCAR, CCH, OBR, SDR) and will be included in all DOS operating systems generated for the Model 145. Modifications to the current EREP will enable it to perform the three functions previously discussed.

I/O RMS FOR OS

I/O RMS routines are optional, model-independent routines supported in MFT and MVT environments. These reconfiguration procedures attempt to minimize the number of abnormal job terminations and unscheduled system halts that occur because of errors on channels or I/O devices.

The alternate path retry (APR) routine provides for the retry of a failing I/O operation on another channel path to the device involved, if one is available, when an uncorrectable channel error occurs. Thus APR, if present, is entered from a device-dependent ERP when a permanent error is deemed to exist after retry procedures have been attempted. If the I/O error is corrected using the alternate channel path, operations continue. If a permanent error still exists, the task is abnormally terminated unless the dynamic device reconfiguration routine is present. A malfunctioning channel path can be varied offline by the operator if necessary.

The dynamic device reconfiguration (DDR) routine permits the operator to move a demountable volume from one device to another of the same type when a permanent hardware error occurs and provides repositioning of the volume so that the failing I/O operation can be retried. A volume can also be demounted so that device cleaning procedures can be performed, and it can then be remounted on the same device. The DDR option also supports demountable system residence devices and unit record equipment. DDR is entered from a device-dependent ERP after a permanent channel or device error occurs on a demountable device. Task termination occurs if the error cannot be corrected and a user-written permanent error-handling routine is not present.

I/O RMS is not included in DOS support that handles only channel paths for tape-unit switching, and does not provide dynamic I/O device allocation by the supervisor.

ADVANCED CHECKPOINT/RESTART AND WARM START FACILITIES FOR OS

If the RMS and I/O RMS routines fail in their attempt to correct a hardware error and the error is one that causes program or system termination, the automatically provided advanced checkpoint/restart and warm start facilities of OS can be employed to minimize the impact of the termination on system operation. The automatic restart facility can be used to cause terminated programs to be rescheduled immediately without resubmission of their job control, so that a minimum of operator intervention is required. The operator must authorize all automatic job step restarts. If a permanent I/O device or channel failure caused the program termination, the device or channel can be varied offline. This ensures allocation of a different device when the program step is reinitiated.

The warm start facilities of the control program provide automatic saving of SYSIN and SYSOUT data sets and input and output work queues so that processed work is not lost when a system termination occurs. The operator is informed of the status of jobs in execution when the system terminated and these jobs should be restarted automatically from the beginning or from a checkpoint if the type of processing involved permits such a restart. System design should include planned restart procedures for unscheduled terminations of individual programs and the system.

CHECKPOINT/RESTART FACILITIES FOR DOS

Programs terminated because of an I/O device or channel error or as a result of a system termination can be restarted from a checkpoint or from the beginning of the job step if their job control is resubmitted with the appropriate restart control statements included. Malfunctioning I/O devices can be removed from the table of available devices by the operator, and different devices of the same type can be assigned to job steps via their job control or by the operator. Warm start facilities are not required because DOS does not build work queues. (DOS power, that builds input and output queues, does provide a warm start facility.)

CHAPTER 13. DIAGNOSTIC HARDWARE

TESTING PHILOSOPHY

Microdiagnostics (micros) are loaded from the console file. The micros are in two groups: basics and extendeds. The basics must be loaded first to ensure that the basic data paths are functional. The extendeds follow the basics and are executed under a micromonitor control program.

When the micros are loaded into control storage, the System/370 microprogram is destroyed. For example: a SI/O, TI/O loop to an I/O device could not be executed while the micros are loaded in control storage.

The microdiagnostic servicing handbook (part 2641601 Program I. D. MIC1—*) contains all the necessary information on run procedures for the basics and extended diagnostics. It contains a fault dictionary of fault-locating data for the basics and has the card part-number reference list and multiple card-usage reference list. These reference lists are used with the fault-locating information obtained from the basics or extended diagnostics.

This section of the manual covers the diagnostic hardware used in the Model 145.

BASIC TESTS

Load the basic disk. (Use the data flow chart to follow this description.) Errors turn on the diagnostic stop light.

The path from the console file to the C-register is checked first. The sequence of testing is:

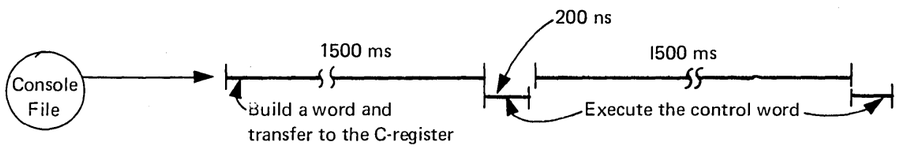
Check the console-file attachment hardware and the console-file commands.

Check that the C-register can be loaded properly.

Check the basic word types, using a minimum of CPU circuitry.

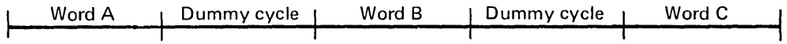
Check that local storage can be addressed and loaded properly.

All of these tests are executed directly from the console file. That is, the C-register is loaded directly from the console file and executed one word at a time.



After the basic CPU data paths have been checked, small microprograms are loaded into local storage and executed in LSCS mode (local store, control store mode). Control words are read from the disk and loaded into A local storage. Diagnostic hardware forces A Local Storage to act as control storage. In LSCS mode a control word is read out of local storage, loaded into the C register and executed. The CPU is tested more intensively with control words operating at normal machine speeds, except that a dummy cycle occurs after each word is executed. This dummy cycle is necessary to allow normal destination of local storage to occur in the next cycle. Operation is:

Execute control words A, B, C



Each cycle is approximately 200 ns

The Phase 2I storage and the ECC board are tested in LSCS mode. First the addressing lines are checked, then the microprogram attempts to find one error-free doubleword in main storage. Once this doubleword is located, the ECC board can be checked out. Then control storage is scanned for double errors. Single ECC errors are not checked until later.

Now that control storage has been checked out, the microprogram can be loaded into control storage and executed in the normal manner.

The SDBO path to the C-register is checked first.

Address lines from SDBO to local storage are checked.

Word types 4-7 (read and store words) are checked.

The last area to check before the extended tests can be loaded is the console PRT/KBD(s) and its native attachment. Once this area is tested, and any failures are corrected, the extended diagnostics can be loaded and communications between the service representative and the micromonitor can be via the console PRT/KBD.

EXTENDED TESTS

The extended tests at EC 128657 or above occupy four disks. This is subject to change on future EC levels. The micromonitor is located on the first disk (*EX1) and the fourth disk (*SA0). The disks must be run sequentially for the fault-locating data to be effective. All errors are printed out on the console PRT/KBD. The sequence of testing is:

The external registers and the machine-check register.

I-cycles and address-adjustment hardware.	
Multiplex channel	Disk 1 (*EX1)
Trapping and priority controls.	
Selector channels group 1	
Selector channels group 2	
Storage protect	Disk 2 (*EX2)
Retry hardware	
Interval timer and TOD clock.	
Manual tests.	Disk 3 (*EX3)
IFA (Integrated File Adapter)	Disk 4 (*SA0)

All of these tests are executed automatically and require no operator intervention except for the manual and IFA tests. The manual tests are run only if requested by the operator.

MD0: Console PRT/KBD—Checks out the keyboard functions.

ME0: I/O Exerciser—allows service representative to exercise tapes, printers, readers, and punches on either multiplexor or selector channels.

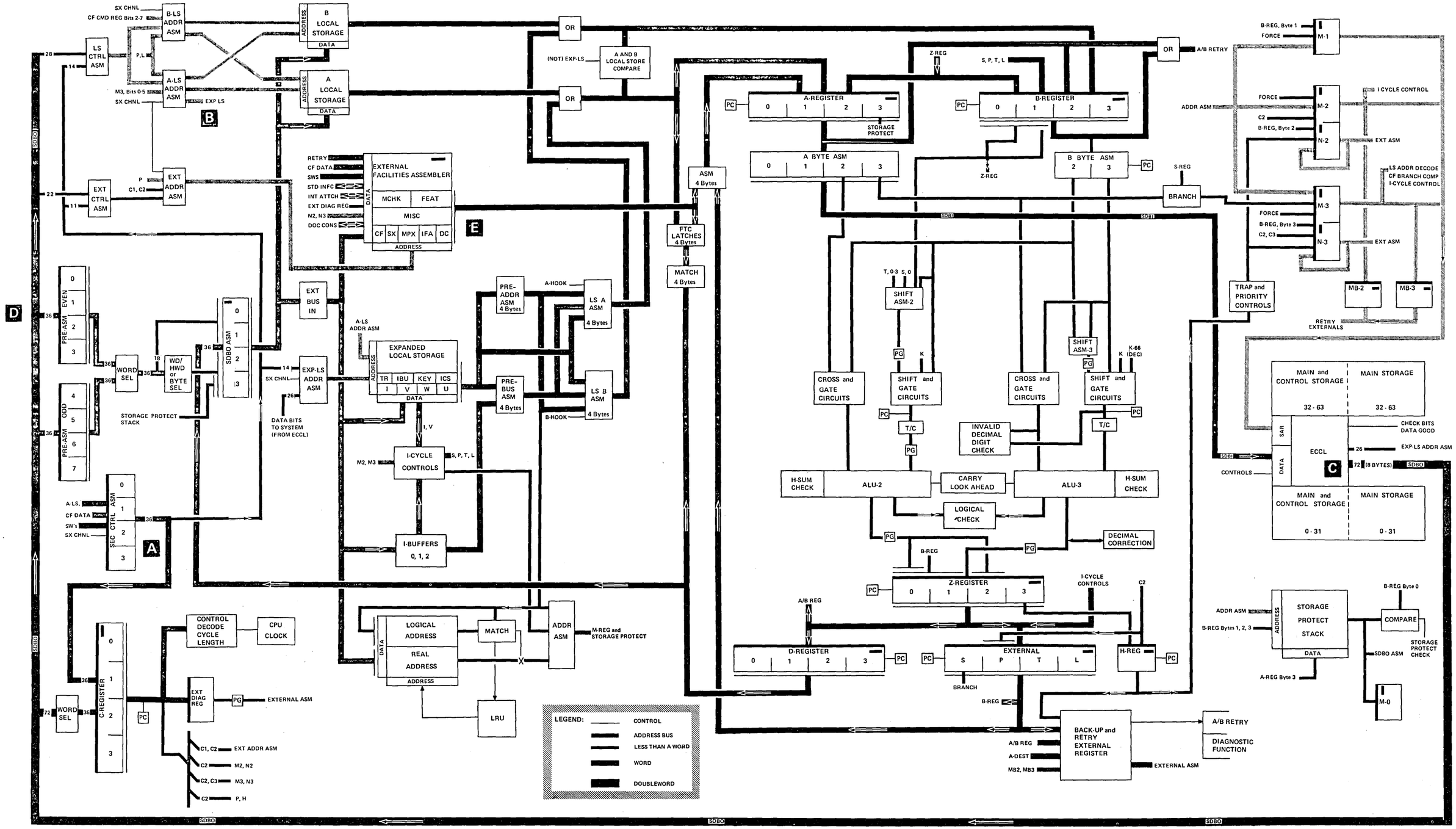
MA0: CPU Console—Checks out all the switch functions on the console.

MB0: Storage Analysis—Checks the Phase 2I main storage for single and double ECC errors; checks control storage for single ECC errors.

Notes 1: All basic tests begin with the letter B.
All extended tests begin with the letter E.
All manual tests begin with the letter M.
All IFA tests begin with the letter S.

2: Usually the first test of a group contains a description and useful data about the section under test. For example: EGA0—(Priority Controls) contains a description, flow charts and timing sequences used while running the EG tests.

DETAILED DATA FLOW



BASIC DIAGNOSTIC TROUBLESHOOTING

BAB5 ROUTINE (EXAMPLE)

The BAB5 routine is an example of troubleshooting problems in the console-file mode of operation. (Use the listing on page 13-5 to follow this description.)

The first thing to do when a stop occurs is to use the fault dictionary (section 7) and the card part-number reference list (section 8) in the microdiagnostic servicing handbook to locate the failing card indicated by the stop word. If the failing card cannot be located by using the fault dictionary and reference list, follow the instructions of section 2.0.3 for using the micro-diagnostic listing to determine the failure. The following examples illustrate how to use the listings when the failure is detected but not located. A test description of the microcode is in every routine.

```
**************************************************************************

*** TEST BAB5   VALIDATE WORD MOVE VERSION 1 MASK CONTROLS.
                FLUSH ZEROES & ONES THRU ALU 2 & ALU3, TRUE ADD.
                VALIDATE Z-REG INPUT BYTE GATING

                * DESCRIPTION *

                SUBTEST 1 - -

                SPTL REGISTERS ARE INITIALIZED WITH ONES.  THE
                WORD MOVE VERSION 1 WORD IS USED TO MOVE A BYTE
                OF ZEROS TO ONE OF THE FOUR REGS - S,P,T, OR L.
                THE OTHER THREE REGS ARE TESTED FOR ONES AND THE
                DESTINATION REG TESTED FOR ZEROS.  THIS PROCEDURE
                IS REPEATED FOR EACH REGISTER.  IN EACH WORD MOVE
                A BRANCH HIGH AND LOW DECODE IS TESTED.

                NOTE - WHEN BYTE 2 IS MOVED TO THE T REG, THE
                DATA PATH IS THRU ALU2.  THIS IS THE FIRST USAGE
                OF ALU2.  THE TRUE BINARY ADD OPERATION IS PER-
                FORMED.  BYTE 3 IS MOVED THRU ALU3 TO THE L REG.
```

A This is the track/sector address: T/S = 01-2. The 0A to the left of sequence No. 0049 is the Hex conversion of the track/sector address.

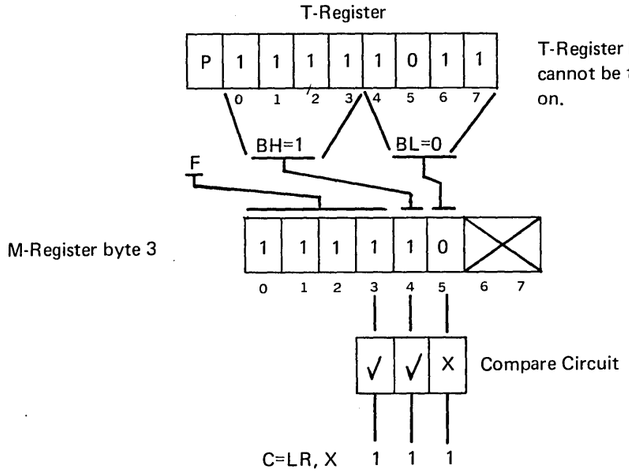
Track					Sector		
0	1	2	3	4	5	6	7
0	0	0	0	1	0	1	0
0					A		

This Hex conversion is useful when setting switches A, b to recycle from the console file.

B CNT: The CNT stands for the byte-counter value. For example, if the byte-counter value set in switches C, D was 51, the console file would execute from the 'BAL NOREG P=10' word up through and including the word 'P, OR, KFF'. By changing the byte-counter value to 56, the next time the console-file start push button is pressed, the console file would execute from the 'BAL NOREG P=10' up through and including the word SPTL = NOREG, S1; therefore, the byte counter can be used to single-cycle through a microprogram written in the console-file mode format. A complete description of the byte-counter operation is in Chapter 5 following the write-up on the read and recycle positions on the diagnostic/file control switch.

C The first word read from the file is 'BAL NO REG P=K10'. The file command that causes this to happen is a 70 command; the mnemonic for this command is C=LR,X which states "load the C-register from the console-file data register and execute the C-register." A description of all console-file commands is in Chapter 6. If the byte counter was used to stop after this command (SW'sC, D = 06), the C-register contents would be 20901000. When it is determined which word causes the failure to occur, copy the failing word from the listing and put it in the switches A-H and use the 'Execute Control Word' position of the diagnostic/file control switch to troubleshoot the problem.

D Let us assume that there is a problem on the machine and that T-register, bit 5 cannot be set on. Seq. No. 60 should set the T-register to all ones (T=T,OR,KFF). The T-register is tested for all ones at Seq. No. 65 (F,BH,BL, T).



The console-file command 7F--C=LR, X111 states "load the C-register from the console-file data register, execute the C-register, and do a compare of M-register byte 3 bits 3, 4, 5." If a no-compare occurs, turn on the diagnostic stop indicator and stop executing from the console file.

The best way to troubleshoot this problem is to use the byte counter to stop at a CNT of 1A to see whether the T-register was set properly. If it was not, put the word 88C8FF00 in the switches and use the Execute Control Word function. If the T-register was set properly, advance the byte-counter value while continuing to observe the T-register, bit 5.

E The last two console-file commands

05 STP = S-DUP
14 STP = M-DUP

These commands turn on the diagnostic stop indicator if either an S-register or an M-register duplicate check occurs.

```
CNT CMD  WORD      NEXT  SEQ  I FILE CMD    NEXT T/S, MOD, OR HEX  BX,BH,BL STAT STATEMENT            COMMENTS                WA1-WA2
--- ---  --------  ----  ---- - ----------  ---------------------  -------- ---- -------------------------------------------
                   0A    0049 C T/S=01-2     A
                         0050 *
                         0051 *                  ************************************************************
 B                       0052 *
                         0053 *                   - - - - - - - - - - - - - - - - - - - - S U B T E S T  1 - -
                         0054 *
 01 70    20901000        0055 W C=LR,X                                C     BAL NOREG P=K10       POINT P TO MCK REGS.    E01
 06 70    3890FF00        0056 W C=LR,X                                      MCKA=NOREG,SF         RESET MCKA & MCKB.      E07 E01
                         0057 *
 0B 70    88C0FF00        0058 W C=LR,X                          0,0,0       S=S,OR,KFF            INITIALIZE SPTL=1'S.    E04
 10 70    88C4FF00        0059 W C=LR,X                          0,0,0       P=P,OR,KFF                                    E04
 15 70    88C8FF00        0060 W C=LR,X                          0,0,0     D T=T,OR,KFF                                    E04
 1A 70    88CCFF00        0061 W C=LR,X                          0,0,0       L=L,OR,KFF                                    E04
 1F 7F    3A90C812        0062 W C=LR,X111                       1,S1,Z0     SPTL=NOREG,S8         MOVE 0'S TO S REG.      E04 E01
                         0063 *                                                                  TEST BR DCD S1,Z0=11.
 24 7B    07C40027        0064 W C=LR,X011  NEXTMOD=00            2,BH,BL     P                     TEST P=FF.              E04
 29 7F    07C800F7        0065 W C=LR,X111  NEXTMOD=00            F,BH,BL     T                     TEST T=FF.              E04
 2E 7F    07CC00F7        0066 W C=LR,X111  NEXTMOD=00            F,BH,BL     L                     TEST L=FF.              E04
 33 7C    17C30F37        0067 W C=LR,X100                       3,BH,BL     S,OR,KFF              TEST S=00. SET S=FF.    E04
                         0068 *
 38 7B    3C90C444        0069 W C=LR,X011                       4,S2,S3     SPTL=NOREG,S4         MOVE 0'S TO P REG.      E04 E01
                         0070 *                                                                  TEST BR DCD S2,S3=11.
 3D 7F    07C00057        0071 W C=LR,X111  NEXTMOD=00            5,BH,BL     S                     TEST S=FF.              E04
 42 7F    07C800B7        0072 W C=LR,X111  NEXTMOD=00            B,BH,BL     T                     TEST T=FF.              E04
 47 7F    07CC00B7        0073 W C=LR,X111  NEXTMOD=00            B,BH,BL     L                     TEST L=FF.              E04
 4C 78    17C70F67        0074 W C=LR,X000                       6,BH,BL     P,OR,KFF              TEST P=00. SET P=FF.    E04
                         0075 *
 51 7B    3D90C1C5        0076 W C=LR,X011                       C,S4,S5     SPTL=NOREG,S1         MOVE 0'S TO L REG.      E04 E01
                         0077 *                                                                  TEST BR DCD S4,S5=11.
 56 7F    07C00077        0078 W C=LR,X111  NEXTMOD=00            7,BH,BL     S                     TEST S=FF.              E04
 5B 7F    07C400D7        0079 W C=LR,X111  NEXTMOD=00            D,BH,BL     P                     TEST P=FF.              E04
 60 7F    07C800D7        0080 W C=LR,X111  NEXTMOD=00            D,BH,BL     T                     TEST T=FF.              E04
 65 78    17CF0F87        0081 W C=LR,X000                       8,BH,BL     L,OR,KFF              TEST L=00. SET L=FF.    E04
                         0082 *
 6A 7B    3E90C2E6        0083 W C=LR,X011                       E,S6,S7     SPTL=NOREG,S2         MOVE 0'S TO T REG.      E04 E01
                         0084 *                                                                  TEST BR DCD S6,S7=11.
 6F 7F    07C00097        0085 W C=LR,X111  NEXTMOD=00            9,BH,BL     S                     TEST S=FF.              E04
 74 7F    07C40097        0086 W C=LR,X111  NEXTMOD=00            9,BH,BL     P                     TEST P=FF.              E04
 79 7F    07CC0097        0087 W C=LR,X111  NEXTMOD=00            9,BH,BL     L                     TEST L=FF.              E04
 7E 78    17CB0FA7        0088 W C=LR,X000                       A,BH,BL     T,OR,KFF              TEST T=00.  SET T=FF.   E04
                         0089 *
 83 05                    0090 C STP=S-DUP  E                                                     STOP IF S-REG DUP CHK
 84 14                    0091 C STP=M-DUP                                                        STOP IF M-REG DUP CHK
                         0092 *
                         0093 *                   - - - - - - - - - - - - - - - - - - - - S U B T E S T  2 - -
                         0094 *
 85 70    88C0FF00        0095 W C=LR,X                          0,0,0       S=S,OR,KFF            INITIALIZE S=FF.        E04
```

BEA6 ROUTINE (EXAMPLE)

The BEA6 routine is an example of troubleshooting problems in LSCS mode (local store, control store mode). Always swap the indicated MST card(s) before proceeding into the listing to analyze the failure.

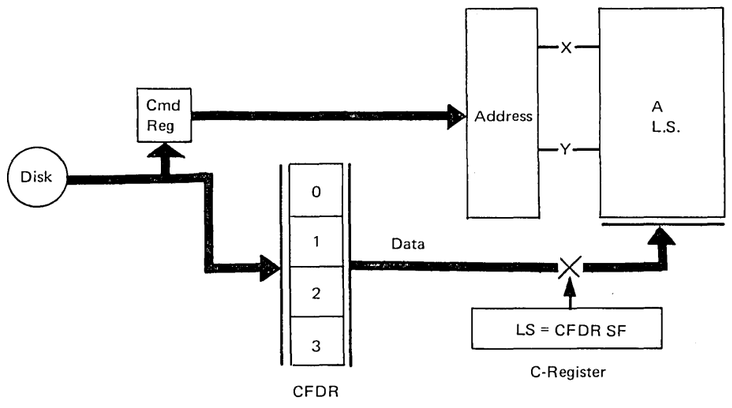
This is the test description of BEA6.

```
*****************************************************************

*** TEST BEA6   VALIDATE THE ABILITY TO CREATE AND PROPAGATE
                A CARRY CONDITION.

                * DESCRIPTION *

                VALIDATE THE ABILITY TO GENERATE A CARRY FROM
                BYTE 2 TO BYTE 1 ON A FULLWORD ADD OPERATION

*****************************************************************
```

A This test starts at track/sector = 08-5. Set switches A, B to 45 to start reading or recycling this test.

B The first thing that must be done is to load A L.S. with subtests from the disk. The Format = LOC STG is an instruction given to the microassembler program that builds the disk. The 60 command, C=LR, loads the C-register with the microword LS00=CFDR,SF. (The microword is not executed now.) The next command C1--LR,X,LS01--loads four bytes (00 00 0F FF) from the console-file data register (CFDR) into L.S. 01 under control of the microword in the C-register. The words that follow are loaded into L.S. in the same manner. The L.S. address is supplied by the disk command byte. (This command byte is not shown on the listings.)

Example:



The C-register controls loading L.S. with the address supplied by bits 2-7 of the command register.

C Now that the entire microprogram has been loaded into L.S., more control instructions must be read from the disk.

Before execution of the microprogram in local storage can begin, diagnostic controls must be turned on and LSCS mode must be established. This is done in Seq. Nos. 0146–0148. At Seq. No. 0149, a branch word is executed to address 00 F0. This word transfers control to Seq. No. 0024, and execution of the microprogram in L.S. begins.

D For convenience when troubleshooting, both the M-register value and the L.S. address are shown in the listing.

Example:

ADDR
M/LS — For single-cycle operation or address matching, the M-register byte 3 value would be stepped.
F0/3C
18/06
A4/29 — To display the local-storage value, set 3C, 06, or 29 in switches F,G.

Remember that a dummy cycle occurs after each word is executed.

E Assume that a failure occurred and that L.S. 00 byte 1 was not equal to zero. The diagnostic stop would be turned on and the machine would stop at Seq. No. 043 with the C-register = 04 00 00 11 and the M-register = 00 A0. The diagnostic stop was forced because of word executed at Seq. No. 147 - DIAG=0,OE, K90 Stop On Z-Reg not zero. A scope loop is set up by simply pressing the start push button. The sequence numbers of the scope loop are:

	Seq.	
	0043	
	0044	Note that the evaluation word is not executed again. The diagnostic stop light remains on, and the microprogram cycles from Seq. 0025 through Seq. 0030.
Loop	0025	
	0026	
	0027	
	0029	
	0030	

F The last command from the file on this sector is a file wait command. It sets the track/sector address to 08-6 or Hex 46. This command is executed just before entering LSCS mode. If a failure occurs in test BEA6, the console-file display would indicate 46.

The last subtest executed in BEA6 would reset file wait, and reading from the disk would begin again at track/sector 08-6.

```
D  ADDR     WORD       NEXT   SEQ   I LABEL   LEG  NEXT SEQUENCE     NEXT LABEL        STAT  STATEMENT              COMMENTS                    WA1-WA2
   M/LS     --------   ----   ----  - ------  ---  --------------    ---------------   ----  ---------------------  --------------------------  -------
                       45     0002  W T/S=08-5      A
                              0003
                              0015  FORMAT=LOCSTG
   60       38920F00    B     0016  W C-LR                                              LS00=CFDR.SF                                       L00 E09
                              0017  RESERVE    FROM 0000 THRU 000C
                              0018  *
   C1       00000FFF           0019  C LR,X,LS01    00000FFF                             LS01=  00 00 0F FF
                              0020  *
                              0021  *                                                  - - - - - - - - - - - - - - - - - - - - - - - - - S U B T E S T - - - -
                              0022  *
                              0023  A WORD      ADDR=00F0
   F0/3C    C0CC0018   0018    0024                        0025          SUBT1  1,0          L=0,0E,K00             ADDRESSING CYCLE            E04
   18/06    3020C8A1   00A4    0025    SUBT1  10                                             LS02=SPTL,D8           SAVE S-REG LOOP CTRL        L02 E04
   A4/29    39900FA0   00A8    0026                                                          LS00=NOPEG,SF          LS00=  00 00 00 00          L00 E01
   A8/2A    C0C010AC   00AC    0027                                                          S=0,0E,K10             SET S-3 FOR CARRY IN        E04
   AC/2B    A20C1CB0   00B0    0028                                                          LS00C=LS00+-LS01+C     LS00=  00 00 10 00          L00 L01
   B0/2C    3820C8B1   00B4    0029                                                          SPTL=LS02,S8           RSTOR S-REG LOOP CTRL       E04 L02
   B4/2D    C0CC0011   0010    0030                                  SUBT1  S2,S3          L=0,0E,K00             EXPECTED RESULT             E04
   *******             00-0036       01-0049   10-0025    11-0044
                              0031  *                                                                                SCOPE LOOP IF S2=1
                              0032  *
                              0033  *                                                  - - - - - - - - - - - - - E V A L U A T E - - -
                              0034  *
                              0035  A  WORD    ADDR=0010                                                             FIX STOP IDENTIFIER
   10/04    934104A0   00A0    0036     SUBT1  00                                     S12   Z=0,0E,LS001          TEST FOR ANSN Z STOP        L   L00
                              0037  *                                                                                LS001 SHUD BE 00
                              0038  *                          **************       E  LS00 CONTAINING     00 00 00 00  IS ADDED TO
                              0039  *                          * STOP WORD *          LS01 CONTAINING     00 00 0F FF  PLUS
                              0040  *                          **************         A CARRY IN 0F       -- -- -- -1. THE RESULT
                              0041  *                                                 IS PUT IN LS00=     00 00 10 00.
                              0042  *
   A0/28    04000011   0014    0043                                  0049,0044     SUBT1 S2,1                                                 S2=1 IF PREV FAILED
   1C/07    01000010   0018    0044     SUBT1  11                    0025          SUBT1 1,0                                                  BRANCH TO SCOPE LOOP
                              0045  *
                              0046  *
```

Other subtests are included between Seq. 46 and 146.

```
   70       C0C41000           0146  W  C=LR,X   C                                    P=0,0E,K10             SET P-REG                   E04
   70       C0A09000           0147  W  C=LR,X                                        DIAG0=0,0E,K90         STOP ON Z NOT ZERO          E02
   70       10D18100           0148  W  C=LR,X                                        SYS0,0P,K01 SDK        SET DIAG KEY/LSCS MOD       E05
   70       009100F0           0149  W  C=LR,X  NEXTMOD=00              F,0,0       S=NOREG0               START AT LS3C/S-REG=0       E01
   41                  46      0150  W  FILEWAIT LT/S=08-6
                               F
```

EXTENDED TESTS TROUBLESHOOTING

EGE7 ROUTINE (EXAMPLE)

The extended tests are loaded from the console file into control storage and executed from control storage. The micromonitor controls the operation of each extended test much like DMA4 or DMA8 controlled sections when testing a 360 System. Errors are indicated by a printout on the console PRT/KBD.

A

```
*EGE7          D4 80 D4 D40000
Actual
Error                    **
Expected       D4 80 D4 D40200
Card List      RH03--
Cable Net      RH031 AM2
```

The error shows that byte 5, H-register bit 6 was not set properly. Always swap the indicated MST card(s) first. Use the card part number reference list in the Microdiagnostic Servicing Handbook to locate the part number and location of the failing card. Use the Multiple Card-Usage Reference List to see whether the indicated card(s) is swappable within the machine. If a tri-lead cable net is involved, the logic page and net ID is given for reference.

Test EGE7 checks IFA Trap Request.

B Seq. No. 0031: This Bal LS3C P=K12 is used to tell the micromonitor where the test starts. The actual test begins with Seq. No. 0035 when the feature plug card is checked to see whether the IFA feature is installed. (Misplug of the feature cards could result in false errors.)

C Seq. No. 0037 through 0056 are not performing any diagnostic testing. These microsteps are initializing the IFA circuits only and turning on the diagnostic controls that are necessary to force a trap. The actual test is done by branching to the common routine (EGA0) at Seq. No. 0058. In the common routine, EGA0, the results are generated and stored away in L.S. The best troubleshooting method is to find out where in the common routine the result was stored away and work backward.

D The fault-record information is used only by the micro-programmer. This information is interpreted by the micromonitor. It will eventually be removed from the listing.

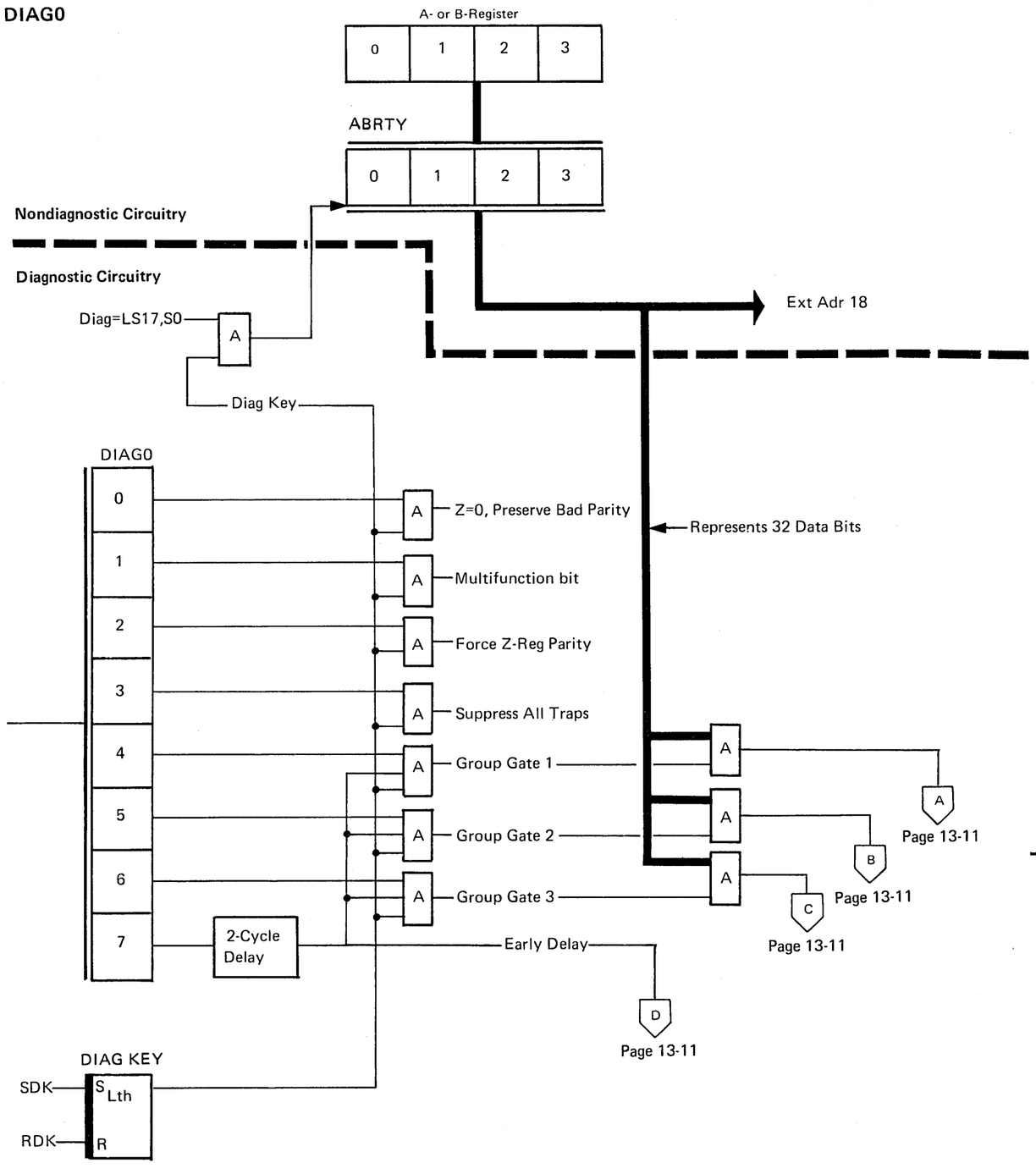
NOTES:

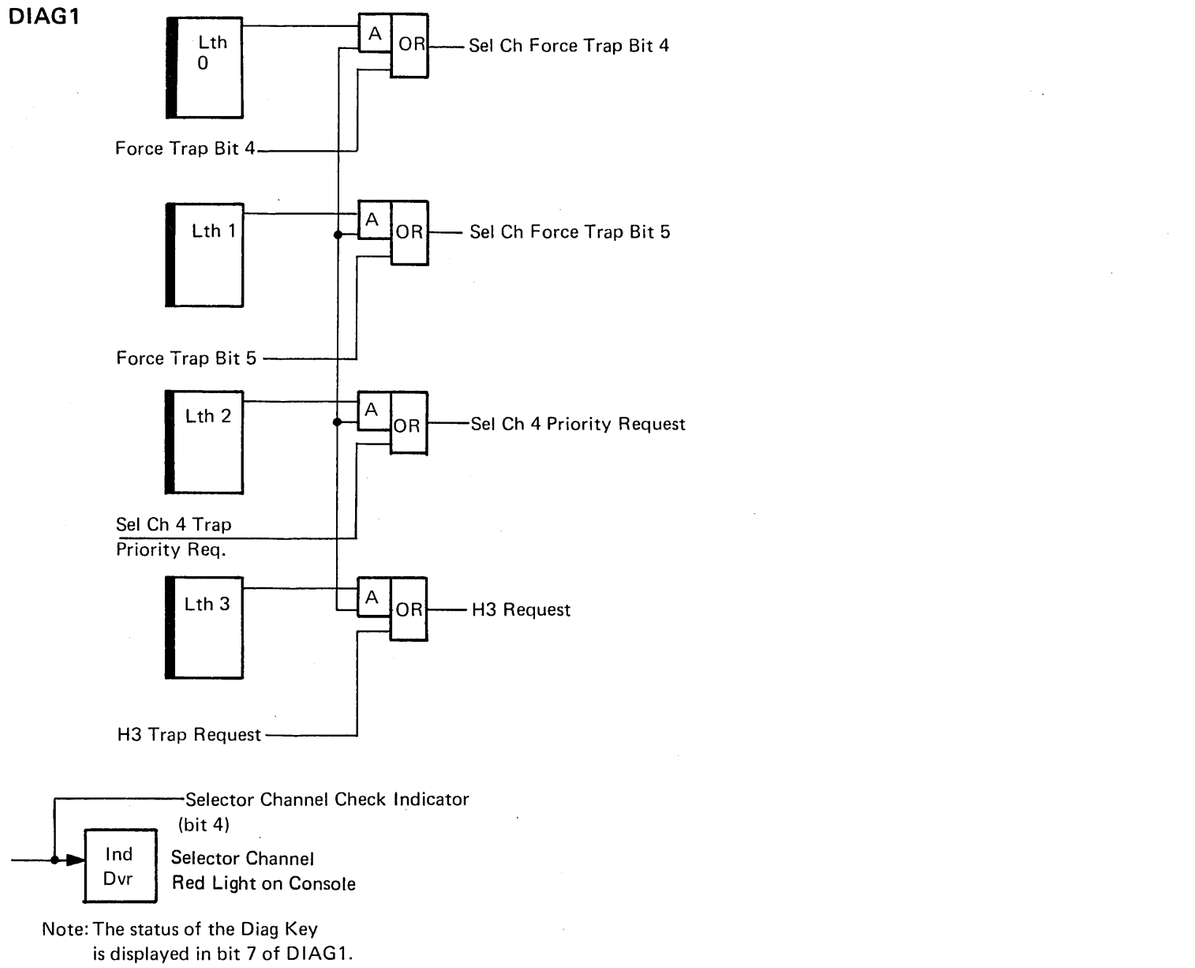
1. If the expected results equal the actual results and asterisks still appear in the error row, this means the parity bit did not match.
2. Not all extended tests use a common routine. If there is a common routine used, it is always the first routine in a section.
3. If there is no fault-locating data in a test, the machine does not stop after the error printout. Read section 3 of the Microdiagnostic Servicing Handbook for looping options.
4. MCKA=NOREG, SF--This microword resets both machine-check A- and machine-check B-registers.
5. Selector-channel tests are EJXX-SX1; EKXX-SX2; ELXX-SX3; EMXX-SX4. The second letter identifies which channel is being tested. Use microroutines labeled EJXX when using the microlisting. There are no microroutines labeled EK, EL or EM.
6. Tests ENXX are common tests to all selector channels.

```
**********************************************************************

*** TEST EGE7  PRIORITY CONTROL TEST OF REQUESTS 6 AND 8

        * TEST FORM - NORMAL
        * RSLT FORM - NORMAL

             *  DESCRIPTION *

             TWO REQUESTS WILL BE GENERATED, THEN SUPPRESS ALL
             TRAPS WILL BE TURNED OFF. THE HIGHER REQUEST
             SHOULD HAVE PRIORITY.
             REQ LOW    ADDR CHECK              ADDR D808
             REQ HIGH   IFA LOW TRAP            ADDR D480
             REQ LOW    ADDR CHECK              ADDR D80C
```

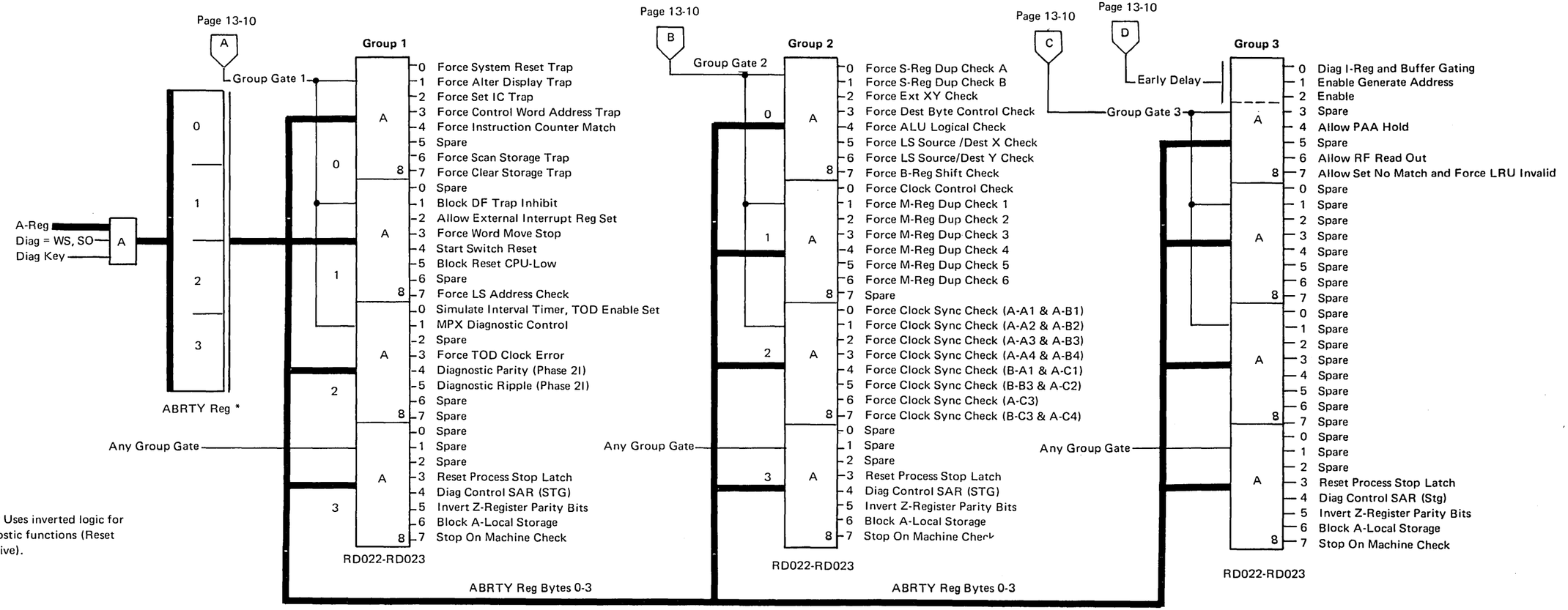
```
ADDR     WORD_____ NEXT  SEQ  I LABEL_  LEG  NEXT__SEQUENCE_   NEXT LABEL_____ STAT STATEMENT_____ COMMENTS_____ WA1-WA2
                         0020 *                                                  *  EXPECTED RESULT *
                         0021 *
                         0022 *                                                  BYTE 1 SHOULD BE D4 - ADDR BYTE 2 OF TRAP 2 CYCLE
                         0023 *                                                A BYTE 2 SHOULD BE 80 - ADDR BYTE 3 OF TRAP 2 CYCLE
                         0024 *                                                  BYTE 3 SHOULD BE D4 - ADDR BYTE 2 OF TRAP 3 CYCLE
                         0025 *                                                  BYTE 4 SHOULD BE D4 - ADDR BYTE 2 OF TRAP 4 CYCLE
                         0026 *                                                  BYTE 5 SHOULD BE 02 - H REG AFTER TRAP
                         0027 *                                                  BYTE 6 SHOULD BE 00 - MCKB1 AFTER TRAP
                         0028 *
                         0029 *                                  ********************************************************************************
                         0030 *
ECF4     214E12F6  ECF8  0031     START              0035,0034   LINK BR S7      B BAL LS3C P=K12      *LINK TEST START ADDR L3C
                         0032 *                                                                          0 BR EXEC THIS TEST
                         0033 *                                                                          1 BR GO TO NEXT TEST
ECFC     0091E650  E650  0034     LINKBR 1     EGE8 0029          START             S-NOREGO          GO TO NEXT TEST S=0 E01
ECF8     00A8E56C  E560  0035     LINKBR 0          0036,0037     FEATBR B4         FEAT2             BR ON IFA INSTALLED E02
E560     282E0089  0004  0036     FEATBR 0                       8,0,1             RTN LS3A          0 BR GO TO MONITOR  L3A
E564     C0C49574  E574  0037     FEATBR 1   C                                     P=0,0E,K95        *LS28 THRU LS2F DIR E04
                         0038 *                                                                          EX20 THRU EX3F
E574     10023170  E570  0039                                                     GA,OR,K10         SET IFA CHANNEL GATE
E570     11027360  E568  0040                                                     GA,A-,K30         ISSUE MACHINE RESET
E568     11017E61  E56C  0041                                                     GA,A-,K0E         SET ALLOW IFA TRAPS
E56C     0000E280  E280  0042                                                                       DUMMY CYCLE
E280     0100E271  E27C  0043                                                                       DUMMY CYCLE
E27C     0000E851  E854  0044                                                                       DUMMY CYCLE
E854     39908730  E838  0045                                                     FBAK-NOREG,S7     RST BACKUP OP AND CNT E20 E01
E838     11013D31  E83C  0046                                                     GA,OR,K0D         *SET LOW PRIORITY SPF
                         0047 *                                                                          REQUEST LATCH
E83C     C0C41240  E840  0048                                                     P=0,0E,K12        *LS10 THRU LS17 DIR E04
                         0049 *                                                                          EX00 THRU EX07
E840     10754441  E844  0050                                                     LS171,A-,K04      *SET DIAG GRP 1 MSK L17
                         0051 *                                                                          BLOCK RESET OF CPU
                         0052 *                                                                          TRAP REQUEST 8
E844     3974A040  E848  0053                                                     DIAG-LS17,S0      *DIAG GRP 1 TO ABRTY E18 L17
E848     C0A0184C  E84C  0054                                                     DIAG0=0,0E,K18    *SET DIAG GRP 1 GATE E02
                         0055 *                                                                          AND SUPPRESS TRAPS
E84C     C0600050  E850  0056                                                     LS160=0           *GENERATE INVALIDE   L16
                         0057 *                                                                          STG ADDR.(00FFFFFF)
E850     0000E001  E004  0058          EGA0 0389          CR0011                                     *BR TO COMMON RTNE
                         0059 *                                                                          REQ GENERATION RDW
                         0060 *
                         0061 *   D                ****          F A U L T  R E C O R D
                         0062 *
PGM      0273            0063 K                    XPGM'C006'                 KEY-C0,CARDS-0,RESULT-6
PGM      0275            0064 K                    XPGM'F8'                   ZERO MASK-F8
PGM      0276            0065 K                    XPGM'D480D4D402'           CONDENSED EXPECTED RSLT
```

CPU DIAGNOSTIC HARDWARE





Note: Use these two pages as reference drawings for the material that follows.



***Note:** Uses inverted logic for Diagnostic functions (Reset bit-active).

DIAGNOSTIC KEY

The diagnostic key is the master gate to the diagnostic circuitry. The diagnostic functions are active only when the diagnostic key is on.

The diagnostic key is set by a branch word (word type 1) using the OR function and having C2 bit 0 = 1.

SYS0,OR,K01 SDK (set the diagnostic key)

The diagnostic key is reset with the A– function of the branch word.

P,A–,KFF RDK (reset the diagnostic key)

Besides a set and reset to the diagnostic key, the two examples show that the other functions of the branch word are also performed.

The status of the diagnostic key can be displayed in the external word DIAG 1 bit 7, (external address 02).

DIAGNOSTIC REGISTERS

DIAG 0 and DIAG 1 are the one-byte registers that control the diagnostic lines in the machine.

These registers are set directly from the C-register K-field.

The functions of the diagnostic register bits are not active unless the diagnostic key is set.

The diagnostic capabilities are further expanded by using an external register called ABRTY (A, B retry). The ABRTY register, external address 18, is normally used by the System/370 microprogram for error-recovery purposes. When control storage is loaded with the microdiagnostics, error recovery and microinstruction retry are not needed; therefore, this register can be used for diagnostic purposes.

Loading and control of ABRTY will be explained later.

All diagnostic hardware used in the system, is illustrated in 3145 Processing Unit Maintenance Diagrams, SY24-3580.

DIAG 0 Register (Refer to Page 13-10)

Diag 0 Bit	Function
0	Z ≠ 0 Stop or Preserve Bad Parity
1	Multifunction Bit
2	Force Z-Reg Parity Bits
3	Suppress all Traps
4	Diagnostic Group Gate 1
5	Diagnostic Group Gate 2
6	Diagnostic Group Gate 3
7	Diagnostic Group Gate Delayed Pulse

Bit 0 Z ≠ 0 or Preserve Bad Parity

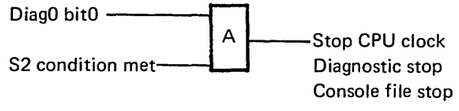
This bit provides two functions:

1. Stop if the Z-register output is not zero (Z ≠ 0). If the Z-register output is not zero, the CPU clock is stopped, the console-file stop latch is set and the diagnostic stop indicator is turned on.

Microprogram example: (Note: In all examples, assume that the Diag key is on.)

DIAGO=0,OE,K80	Turn on Z ≠ 0
T=0,OE,K02	Put 02 in the T-register
LS00=NOREG,SF	Clear LS00
LS00=0+K02	Add 02 to LS00
S12 Z=LS003,OE,T	Check Results

When the results are checked, L. S. 00 byte 3 should be = 02 and the OE against the T-register should cause the Z-register byte 3 to be 00. If the Z-register output was not zero, the CPU would stop with the next sequential word in the C-register and the diagnostic stop indicator would be on.



The status set conditions of S12 or Z24 must be met before the stop condition becomes active.

Byte Operation:	Only byte 3 of the Z bus in checked for zeros.
Fullword Operation:	All four bytes of the Z bus must be all zeros.
Z24 Condition:	The three low-order Z-register bytes are checked for all-zero condition.

2. Preserve Bad Parity

The other function of this bit allows the transfer of data with bad parity from an external register to L. S. Normally when data is transferred through the ALU, the parity (if it is bad) is corrected and sent to the destination register in good parity. This diagnostic function allows bad parity to be shipped to the destination register.

Microprogram example: (Checks MCKA parity bits)

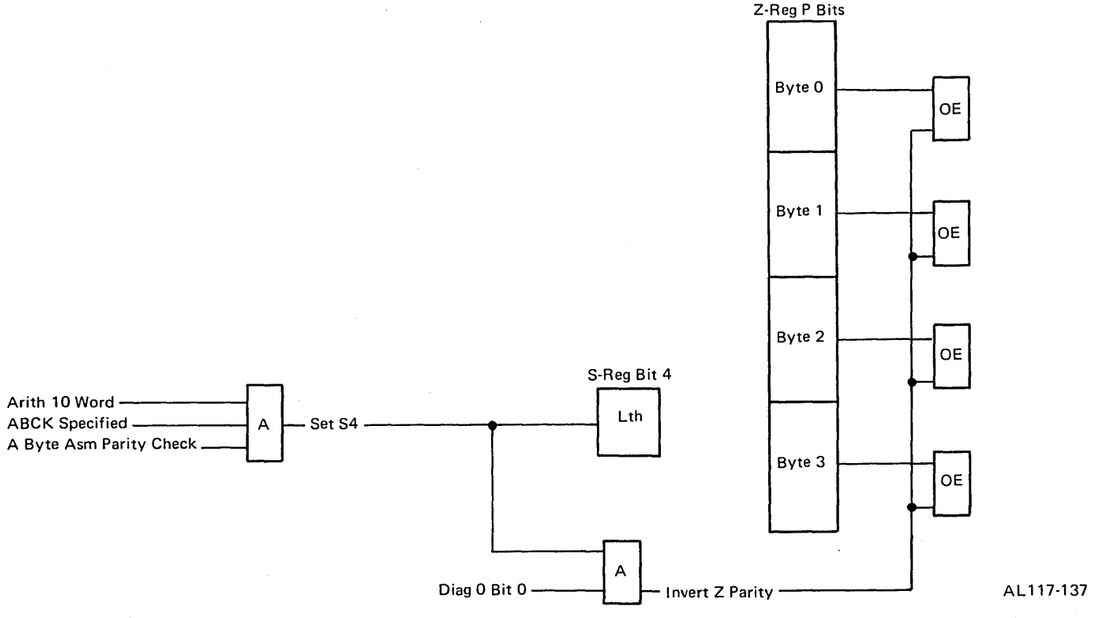
DIAG0=0,OE,K80	Set preserve Bad Parity
MCKA=NOREG,SF	Reset the Machine Check Reg. Parity bits shou ld be on.
LS18=MCKA,SC	Transfer bytes 0, and 1 to LS 18. Preserve bad parity.
LS182=MCKA2, OE,O ABCK	Transfer byte 2. Preserve bad parity.
LS183=MCKA3,OE, O ABCK	Transfer byte 3. Preserve bad parity.

Now the result stored away in LS18 can be compared against expected results. If the parity in MCKA (machine check A-register) was not correct, the diagnostic printout would indicate an error.

The microprogram example shows both word types that allow the Preserve Bad Parity to occur.

LS18 = MCKA, SC. This word-move word transfers bytes 0 and 1 directly from the B-register to the Z-register; thus, bad parity is maintained. Bytes 2 and 3 of a word-move word go through ALU 2 and 3 and generate good parity coming out of the ALU. Therefore, to preserve bad parity on bytes 2 and 3, a different word type is required.

The other word type is the arithmetic 10 word with ABCK (A, B Check) specified, LS182=MCKA2, OE,O ABCK. When ABCK is specified, S4 is set if the A source has bad parity, and S5 is set if the B source has bad parity.



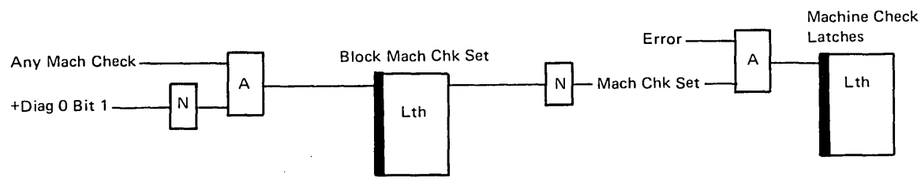
If S4 condition is met, the output of the Z-Reg parity bits are inverted. Byte 3 of the Z-Reg is transferred to L. S. 182 in bad parity. The next word in the example transfers MCKA3 to LS183 in the same manner, preserving bad parity.

Bit 1 Multifunction Bit

As the name implies, this bit is used for several different functions depending on the microdiagnostic program in operation.

1. Allow multiple machine-check setting. During normal operation of the machine (System/370 microprogram), the detection of a machine check between 135 time and the end of the cycle suppresses the setting of any more machine-check bits. This suppression becomes active at 0 time of the next cycle and remains active until the machine-check register is reset. This is necessary so that the System/370 microprogram can attempt to retry the first error that occurred.

   For diagnostic purposes, it is sometimes necessary to accumulate all machine checks over a certain period of time. DIAG 0 bit 1 provides this function by blocking the suppression of machine-check settings.



The multifunction bit does not allow the block mach chk set latch to set, therefore, the Mach Chk set gate is always active.

2. The multifunction bit is used by the I-cycle extended test (EDXX) to disable the normal operation of the I-cycle controls. (Refer to Maintenance Diagrams, page 1-70, Coordinate A-2.)

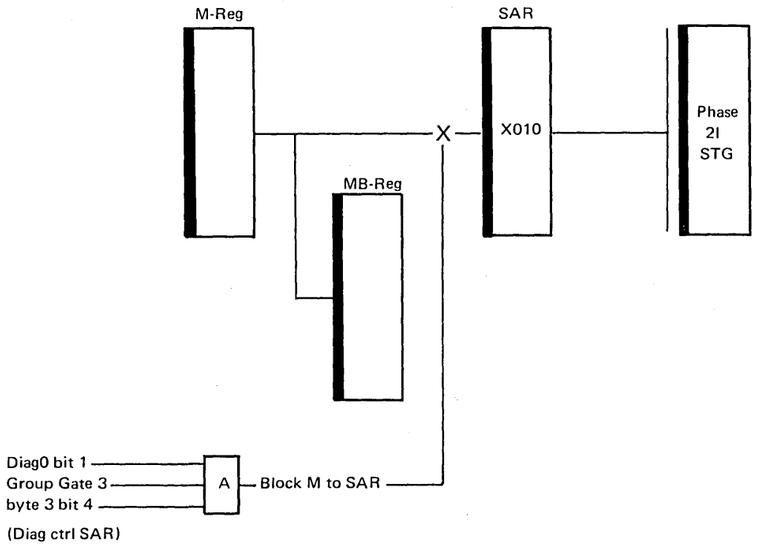
   It prevents a set/reset pulse to the control register except during a storage-1 cycle. This gives the micro-programmer positive control of the control-register functions.

   It blocks the normal turn-on of the generate address and generate controls latches. This allows other diagnostic lines to set these latches under control of the I-Cycles diagnostic tests.

   Along with group gate 3 byte 0 bit 0, it allows a diagnostic control of setting the I-buffers, the U-Reg, and Op and Immediate Byte Regs.

   Extended diagnostic test EDA0 gives a comprehensive write-up and timing chart of the I-cycle diagnostic controls. A detailed explanation of the diagnostic group gate 3 functions follows later in this chapter.

3. Another function of the multifunction bit is to allow a freeze on the storage address. This is used primarily by the priority tests (EGXX) and the I-cycles tests (EDXX).



Normal operation is to transfer the M-register to the SAR (Storage Address Register) and to the MB-Register. This diagnostic control allows the M-Register to be gated only to the MB-register and not to SAR. The address in SAR is forced to X010. The microword read out at X010 stores away the address sent to the MB-Register for evaluation. Therefore, forced addresses, such as trap addresses, can be checked out without losing control of the microprogram. An example is given with the explanation of the group gate 3 byte 3 bit 4 line (diagnostic control SAR).

Bit 2 Force Z-Register Parity Bits

This bit forces all the Z-register parity bits to be on by ORing a 1 into the Z-register parity bit latches.

Microprogram example:

LS113=0,OE,KAD	Load L.S. 11 byte 3 to AD
MCKA=NOREG,SF	Clear MCKA, B
DIAG0=0,OE,K20	Force bad parity
LS11=LS11,SF	LS 11 = 00 00 00 AD Byte 3 has bad parity and should cause Z-Register, D-register checks.
LS183=MCKA2,OE, K00	Save the machine-check register. Z- and D-Checks should be on.

This example checks the ability of Z-register and D-register parity checking circuits and the error latches in the machine-check register (MCKA).

Bit 3 Suppress All Traps

This bit performs several functions:

1. It prevents the trap 1 latch from coming on and, consequently suppresses all traps. Problems in the trapping and priority area would cause the microprogram to lose control by gating an incorrect address into the M-register. The Suppress All Traps is set by every diagnostic test up until the trap, and priority diagnostic test is started (extended test EGA0), thereby preventing false traps from causing problems in the earlier tests.
2. It degates the Any Machine Check Or signal (Maintenance Diagrams 1-30 D3). This allows the machine-check registers to be set but prevents the machine from a hardstop condition even if the check stop switch is in Hardstop.
3. It kills the Block Destination signal. During normal machine operation, if an error occurs, destination gating is blocked so that possible bad data is not stored away in local storage or an external register. In diagnostic mode, it is often necessary to store away data during a forced error condition. The Suppress All Traps allows destination control lines to remain active.
4. It blocks the cancel function in the Phase 2I storage. Normal operation blocks writing in Phase 2I storage during an error: Suppress All Traps allows a write into storage with possible bad data.

Note: The Suppress All Traps function is also active when the console-file power is on and the file wait latch is off (used by the basic tests).

Bits 4, 5, 6 Diagnostic Group Gates 1, 2, and 3

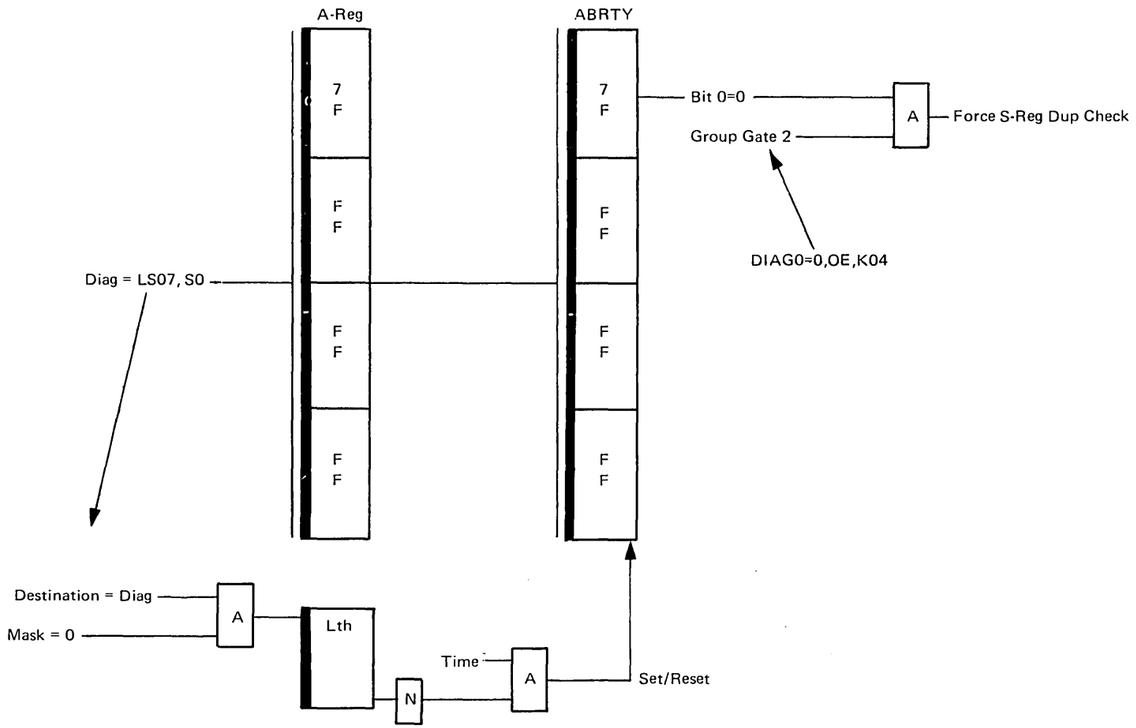
These three Diag 0 bits cause the data bit(s) stored in the ABRTY register to perform some diagnostic function.

Microprogram example: Using group gate 2 byte 0, Force S-register Dup Checks A.

A. LS07=0-K01+1	Store FFFFFFFF in LS 07
B. LS070,A-,K80	Force S-Reg DUP (duplicate) Check A
C. DIAG=LS07,S0	Set diagnostic mask in ABRTY
D. DIAG0=0,OE,K04	Turn on group gate 2

(The diagnostic now performs a test on the S-Reg.)
(Reference to page 13-11.)

Word A stores all F's in L. S. 07. Word B resets L. S. 07 byte 0 bit 0. These two instructions perform the setup to load ABRTY. The output of ABRTY uses negative logic. Therefore, any 0 bit initiates a diagnostic function. Word C is a special word-move word with a source of zero (S0). This statement causes the setup register LS07 (which = 7FFFFFFF), to be gated to the A-Reg. The S0 is the special decode that freezes the A-Reg data into the ABRTY Reg. When this microword is completed, ABRTY is loaded with 7FFFFFFF. ABRTY is held with this data until another DIAG = XX, S0 or the diagnostic key is reset. Word D turns on group gate 2 and the diagnostic function, Force S-Reg Dup Check A, is active.

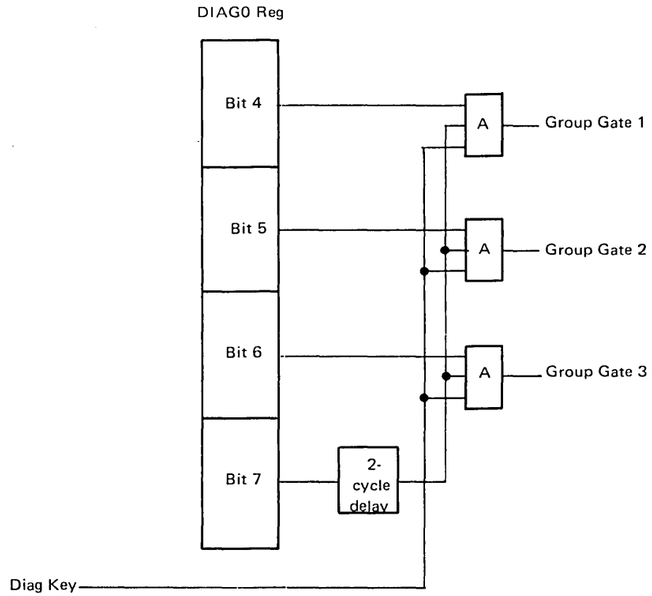


Normally, ABRTY is loaded every cycle from either the A-Reg or the B-Reg. The special word-move word with a destination of DIAG and a Mask of 0 blocks the set/reset pulse to the ABRTY Reg so that the ABRTY Reg is held frozen. (Refer to Maintenance Diagrams 1-35.) Bit 0 being off and group gate 2 active forces the S-Reg Dup Check.

The group gates are turned off by resetting DIAG0 bits 4, 5, 6. Group gate 1 is also reset by the trap-2 cycle latch. Note: Externals addresses 19 (SPTLB), 1A (HMRTY), 1B (CPURTY) cannot be displayed when ABRTY is frozen. If these locations are displayed, the contents of ABRTY will be displayed instead.

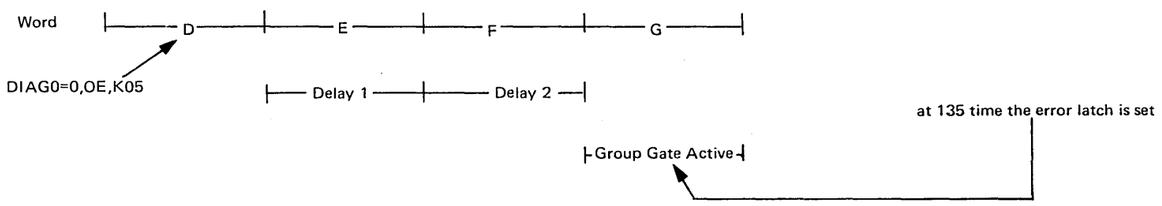
Bit 7 Diagnostic Group Gate Delayed Pulse

This bit delays the conditioning of the group gates for two cycles and allows the group gate to be active for one cycle only.



Microprogram example: (Expanding on the previous examples)

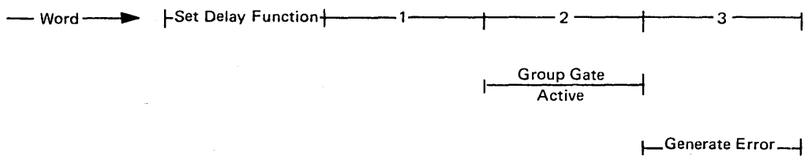
A. LS07=0-K01+1	Store FFFFFFFF in LS 07
B. LS070,A-,K80	Force S-Reg Dup check A
C. DIAG=LS07,S0	Set diagnostic mask in ABRTY
D. DIAG0=0,OE,K05	Turn on group gate 2 and group gate delayed.
E. MCKA=NOREG,SF	Reset the machine-check Regs
F. S,OR,KFF	Set S to all ones
G. (Dummy word)	Error is forced in this cycle
H. LS18=MCKA3, OE,K00	Save results. S-Reg Dup check should be set in MCKA3.



Word E makes sure that the machine-check registers are reset. Word F sets the S-Reg to all ones so that when one of the duplicate circuits is compared in the next cycle, an error is forced. Word G is just a branch word to word H. It is during cycle G that one of the S-Reg duplicate circuits is blocked and the error gets turned on. Word H saves the machine-check Reg to compare against expected results. The S-Reg duplicate check latch should be on.

Microwords A, B, C set up ABRTY as explained in the previous example. Word D turns on group gate 2 and also the group gate delayed function. Three cycles later, during the dummy word, the S-Reg Dup Check should be turned on in the machine check Reg.

Note: There is one exception for the group gate delayed pulse. Group gate 3 byte 0 bits 1, 2, 3 are diagnostic functions used in the I-cycles 0 to 0 time, but instead, operates on a one-half cycle earlier basis. This necessitates generating an early delayed pulse.



I-cycles diagnostic controls are set up one cycle early so that the actual error can occur on the third or normal delayed function word.

DIAG 1 Register (Refer to Page 13-10)

DIAG 1 Bit	Function
0	Selector-channel force trap bit 4
1	Selector-channel force trap bit 5
2	Selector-channel 4 priority request (H4 request)
3	H3 trap request
4	Selector-channel check indicator
5	Spare
6	Spare
7	Diagnostic key (display only)

The function of the diagnostic lines in DIAG 1 is to allow the trap and priority tests to check out the selector channel and IFA trap circuits in the CPU. The priority tests (EGXX) are run before the selector-channel tests. No selector-channel circuits have been tested yet. The diagnostics need some way to generate the trap and priority request lines from the channel. Because none of the basic channel hardware has been tested, normal trap request signals from the channel circuits cannot be used. DIAG 1 provides the necessary diagnostic lines to enable the CPU trap and priority circuits to be checked.

In addition to this, when the selector-channel and IFA tests are run, it is not necessary to check out the trap functions again; the force trap bits and request lines are also gated to the external assembler for DIAG 1. It is necessary only to read out DIAG 1 to check the channel trap lines.

Bits 0 and 1: Selector-channel Force Trap Bits 4 and 5

These lines force trap address byte 3 bits 4 and 5 on, which eventually gate to M-Reg byte 3 bits 4 and 5.

Microprogram Example (EGB9)

```
DIAG1=0,OE,KD0  Set H3 request for D12X
                Set force bit 4    XXX8
                Set force bit 5    XXX4
```

(The microprogram branches to the common routine, which removes the Suppress All Trap function so that the forced M-Reg address can be tested.)

The actual forced M-Reg address is D12C. The D12 is a result of turning on the H3 request. The C is a result of forcing bits 4 and 5 on.

Bit 2: Selector-Channel 4 trap priority request (H4 Request)

a. If the IFA feature is not installed, this bit will be used for a selector channel 4 request.
b. If the IFA feature is installed, this bit will be used for selector channel 2 and 3 requests.

The purpose of this bit is to turn on the priority 4 request latch to establish a trapping priority for the selector channel.

Bit 3: H3 Request

a. If the IFA feature is not installed, this bit is used for channels 1, 2, and 3 requests.
b. If the IFA feature is installed, this bit is used for IFA request.

The purpose of this bit is to turn on the priority 3 request latch.

Bit 4: Selector Channels 1, 2, 3, 4 Check

The selector channel check red light on the console is not part of the machine-check A- or B-registers. When the selector-channel diagnostics force an error, they need to determine whether an error really occurred. The error line, besides being gated to the indicator driver, is also sent to DIAG 1 bit 4. DIAG 1 can be read out and analyzed by the diagnostic.

Bits 5 and 6: Spares

These bits are spares; there are no latches in the DIAG 1 reg for these positions.

Bit 7: Diagnostic Key

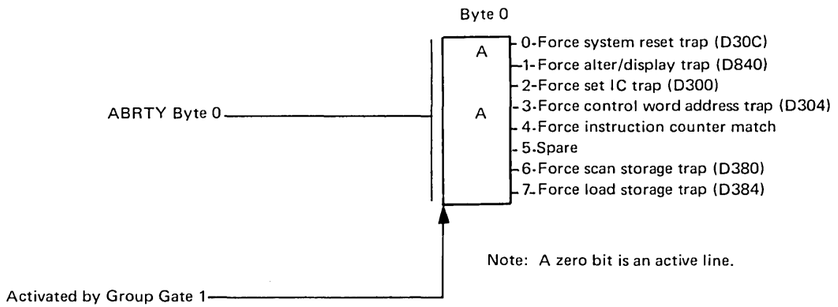
The diagnostic key can be displayed via this position. The key can be turned on only by the word type 1 – SDK (set diagnostic key). It cannot be turned on by destining to DIAG 1 Bit 7. Because bit 7 is not used in generating the parity bit, DIAG1 should contain bad (even) parity.

ABRTY GROUP 1

The bits in this group are active when:
A. The diagnostic key is on.
B. DIAG0 bit 4, Diagnostic Group Gate 1 is on.

ABRTY is loaded and frozen with the microstatement DIAG=LSXX,S0. A group bit is active when it is a zero (inverted).

ABRTY Group Gate 1, Byte 0



The bits in byte zero are primarily used by the trap and priority tests (EGXX) to test the trapping circuits. They are used to force, by microprogram, traps that would normally only be forced by a manual operation.

Microprogram example: (EGB6)

The diagnostic key was turned on by micromonitor.

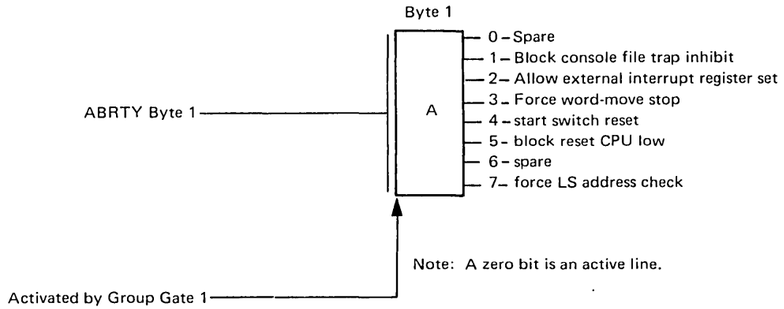
A. LS170,A-,K80	*Set Diag Grp 1 mask L17 force Sys reset trap
B. DIAG0=0,OE, K19	*Set Diag Grp 1 gate E02 and Grp gate delay
C. DIAG=LS17,S0	*Diag Grp 1 to ABRTY E18 L17
D. LS170,OR,K80	*Res DIAG Grp 1 mask L17 force Sys Res trap Sys Res Lth will be set in next Cyc.
E. (Branch word)	*Br to common Rtne diagnostic trap

Local storage addresses 16 and 17 are always set to ones by the micromonitor. These two locations are used to load ABRTY. So, in statement A, turning off bit zero of LS 17, byte 0 causes the Force System Reset Trap to become active when statement C (DIAG=LS17 S0) is issued. ABRTY gets loaded with 7FFFFFFF. (Statement B turns on the controls, Group Gate 1, necessary to force the system reset trap.) In word C, ABRTY is held frozen because of the special decoding of this word-move word. The destination is DIAG, and the mask is zero. Microword D restores LS17 to all ones because this LS location will be used again by the common routine. The next microword (E) branches to the common routine (EGA0). It is during this word (E) that the system reset latch is turned on by the diagnostic hardware. In the common routine, the Suppress All Traps function is lifted and allows the system reset trap address (D30C) to be forced into the M-register.

After the common routine is executed, the actual results should equal the expected results.

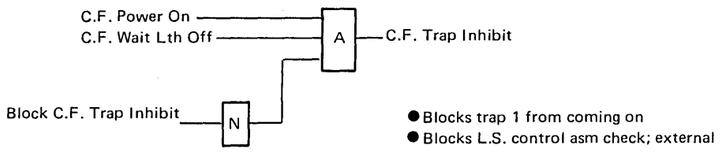
If there is an error and card swapping fails to resolve the problem, it will be necessary to analyze the failure using the common routine (EGA0).

ABRTY Group Gate 1, Byte 1



Bit 0: Spare

Bit 1: Block Console File Trap Inhibit



This bit is used only by the early machines (M2I/C40s). When executing the basic tests from the console file, errors are blocked from stopping the machine and traps are prevented by the 'CF trap inhibit' line. The priorities test (EGXX) on M2I/C40 machines checks out the trap mechanism while reading from the console file. This necessitates a diagnostic blocking function to disable the 'CF trap inhibit' line so that trapping will be allowed while reading from the console file.

Service hint for priorities tests on M2I/C40 machines: By opening the console-file door (make the CF not ready) while single-cycling through the common routine (EGA0), it is possible to observe the forced trap address in the M-Reg.

Bit 2: Allow External Interrupt Register Set

The EXTINT register, external address 12, byte 0, cannot be used as the destination register of a microword. This diagnostic function allows data to be gated from EBI (external bus-in) to the EXTINT Reg.

Bit 3: Force Word-Move Stop

This bit is used to check the alter/display trap request. Normal operation of the alter/display function of the consolePR-KB requires the cycling of a word-move stop word (38000008). This is normally accomplished by pressing the STOP push button. To prevent manual intervention while running diagnostics, the word-move stop function is forced by diagnostic hardware.

Microprogram example:

LS170,A-,K40	Force the Alter/Display Trap
LS171,A-,K10	Force Word-Move Stop Condition
DIAG=LS17,S0	Load ABRTY
DIAG0=0,OE,K18	Turn on Group Gate 1 and Suppress all Traps

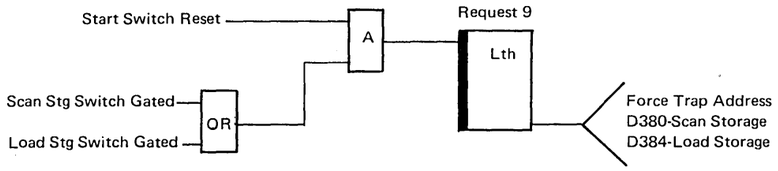
Branch to the common routine.

In the common routine, the suppress all traps line is lifted and the forced alter/display trap address (D840) is checked out.

Bit 4: Start Switch Reset

This diagnostic bit is used to help generate a trap request 9. In normal operation, a request 9 is turned on as a result of the Scan Storage or Load Storage switch positions of the Diagnostic/File Control switch. When the System/370 microprogram is in control storage and the start push button is pressed, a trap occurs to the microroutine that scans or loads storage.

In order to test the trap and priority circuits for this function, the start push button must be simulated.



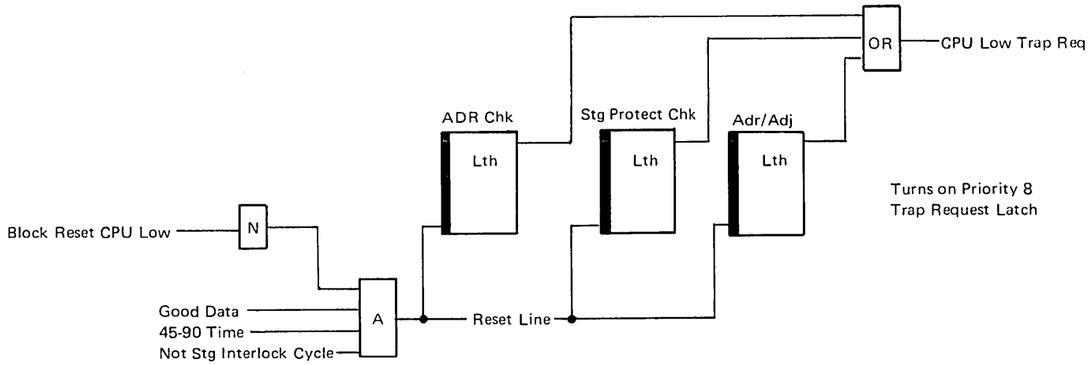
Microprogram Example:

LS170,A-,K01	Set Up Force Load Storage Trap
LS171,A-,K08	Set Up Force Start Switch Reset
DIAG=LS17,S0	Load ABRTY
DIAG0=0,OE,K08	Turn On Group Gate 1

A trap to D384 should occur during the next cycle.

Bit 5: Block Reset CPU Low

An address check, a storage-protect violation, or an address-adjustment trap request brings up the CPU low request, which turns on the request 8 priority latch. In normal System/370 operation, these latches are active for only one cycle and rely on request 8 priority latch to remember the request. In diagnostic mode, it is necessary to hold the CPU trap low request on so that the priority latches can be checked properly.

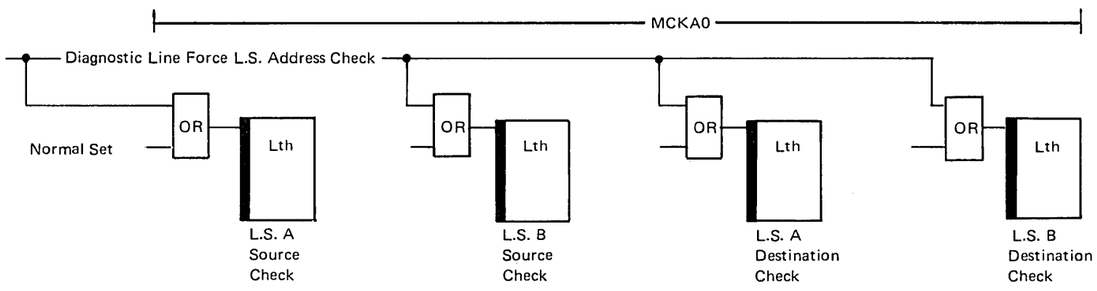


The Block Reset CPU low line prevents the normal reset to the CPU low request latches. Now, when a diagnostic test resets the priority request latches after a trap-1 cycle, the request 8 latch turns on again.

Bit 6: Spare

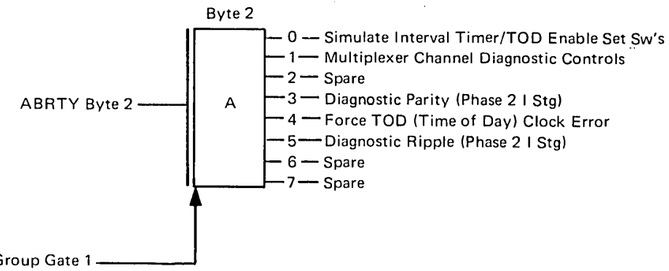
Bit 7: Force local-storage address check

This diagnostic line is used to force LS A&B Source and Destination Checks. It is used by the Machine-Check Register Diagnostic section (ECXX) to verify that the machine-check register can be set and read properly.



(Refer to Maintenance Diagram 1-31)

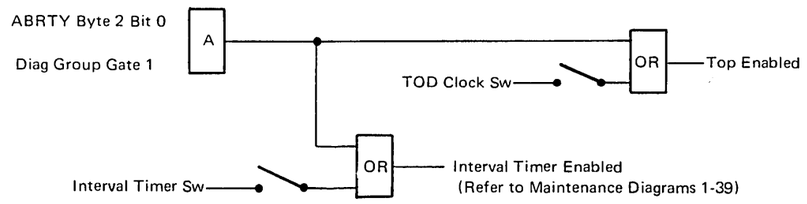
ABRTY Group Gate 1, Byte 2



Bit 0: Simulate interval timer and TOD enable set switches

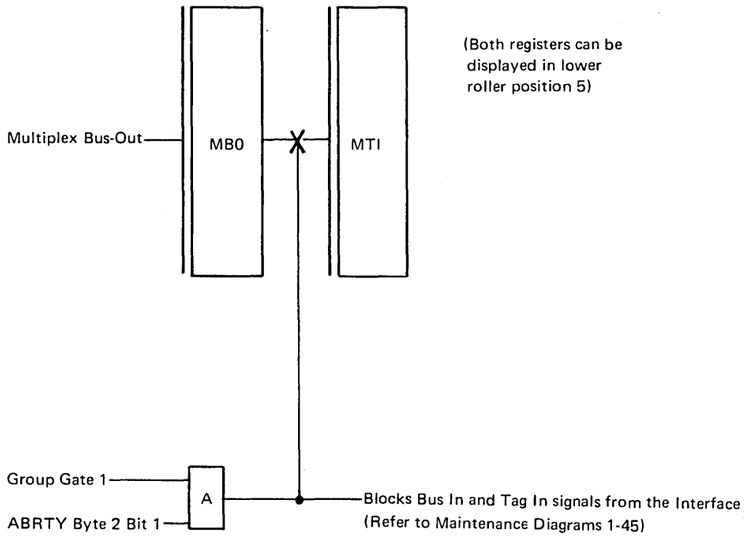
This bit simulates the interval timer switch and the Time of Day Clock Enable switch.

This diagnostic bit enables the Interval Timer and TOD Clock routine (EYXX) to run without any manual intervention. The console switches for the interval timer and the TOD clock do not have to be on to run the EY tests.



Bit 1: MPX (Multiplex) Diagnostic Control

This line allows checking of the MPX bus-out and tags-in signals. The multiplex channel tests (EFXX) check the multiplex channel controls and the bus and tag lines up to the interface drivers.



This diagnostic line gates the bus-out lines back through the tags-in lines. The normal tags-in and bus-in lines are degated by this control, allowing a positive control on the signals coming to the channel from a control unit. If the control-unit problem is forcing a bit on, it does not affect the MPX channel microdiagnostics.

Bit 2: Spare

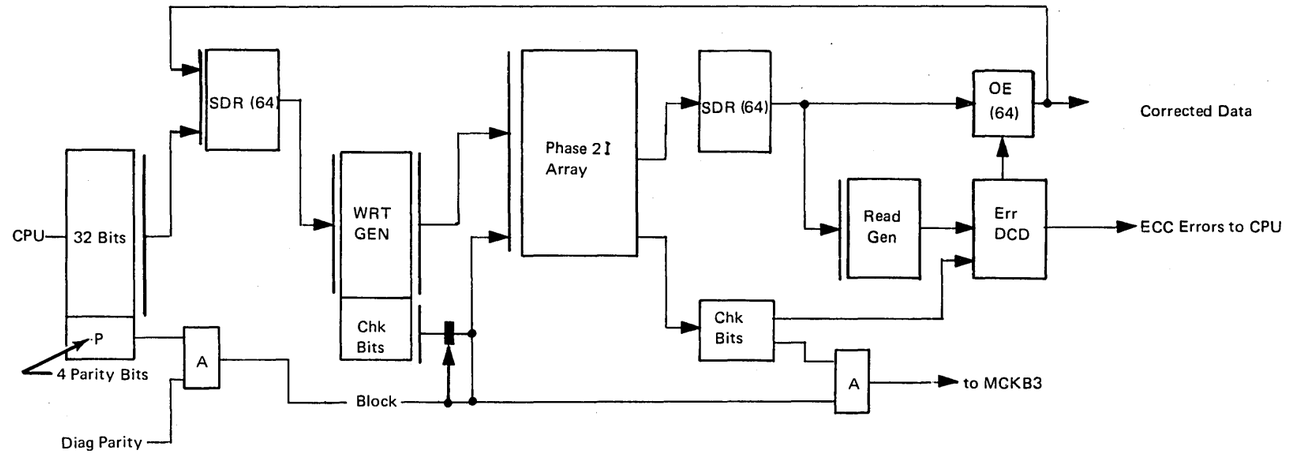
Bit 3: Force Time-of-Day (TOD) Clock Error

This bit forces the TOD Clock error latch to come on and set machine-check Reg A byte 3 bit 6. After the machine-check Reg bit is set, it resets the TOD Clock error latch. This diagnostic control is used in the machine Reg tests (ECXX) and the timer test (EYXX). (Reference Maintenance Diagram 1-39.)

Bit 4: Diagnostic Parity (Phase 2I Diagnostic Control)

This bit is used to help analyze the ECC board hardware and the Phase 2I storage. It provides two diagnostic functions.

On a normal write operation, 32 bits are sent to storage from the CPU and combined with 32 bits from the addressed word and sent to the write generator. The write generator stores the 64 data bits and generates eight ECC check bits.



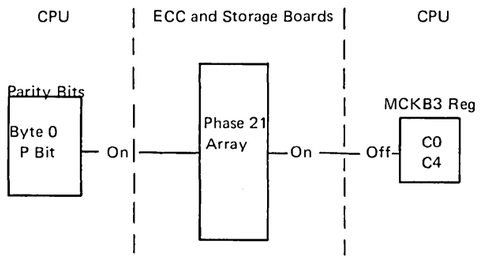
On a normal read operation, data is read from storage and new check bits are generated in the read generator. These new check bits are compared with the check bits read from storage. If a single error is detected, the bit in error is inverted before it is sent to the CPU.

Diagnostic parity mode was designed so that check bits could be written and read from the Phase 2I storage using a minimum of circuitry. In diagnostic parity mode, the four parity bits sent to storage bypass the write generator and are written directly into the Phase 2I array. Because there are eight check bits, each parity bit must be used to write two check bits.

Parity for byte 0—Writes check bits C0, C4
Parity for byte 1—Writes check bits C2, CT
Parity for byte 2—Writes check bits C8, C16
Parity for byte 3—Writes check bits C1, C32

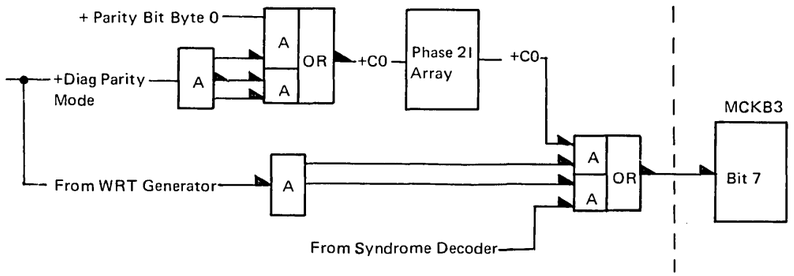
The other function of diagnostic parity mode is to send the check bits read from storage directly to the machine-check register B (MCKB) byte 3. This allows reading the check bits with a minimum of circuitry.

In diagnostic parity mode, an inversion of the parity bits takes place.



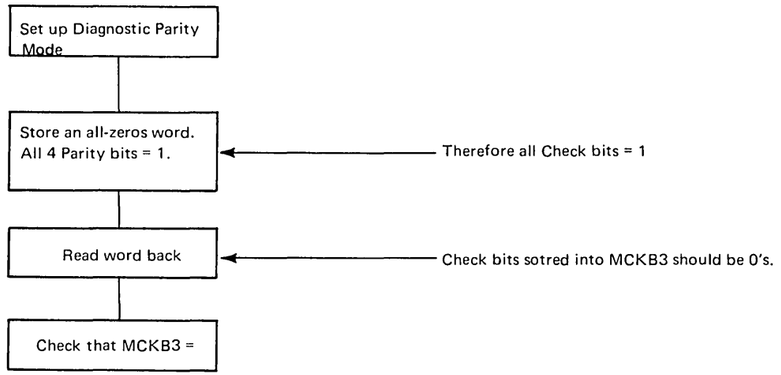
If the parity bit is on going to storage, it will be off when it is read back into the MCKB3 register.

The circuits look something like this:



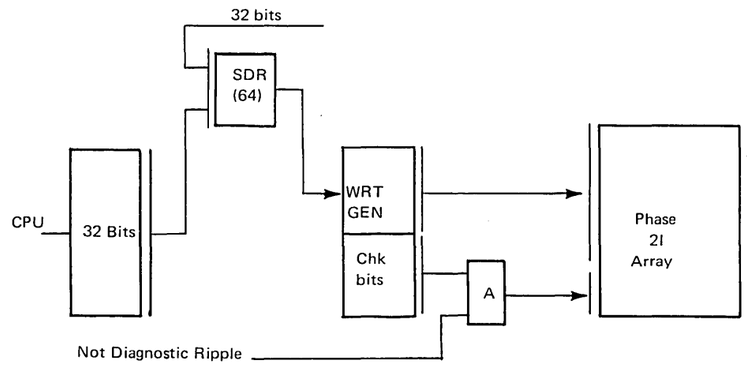
If the parity bit going to the ECC board is a 1 (+), it will be written into the Phase 2I array as a 1 (+), read out as a 1 (+). But the output And OR will be deconditioned and the MCKB byte 3 bit 7 latch will not get set.

This bit is used by the basic tests (BGAX). In test BGA3, the microprogram stores various patterns to verify that the eight check bits can be written and read properly. The first pattern is an all-zero word.



Bit 5: Diagnostic Ripple (Phase 2I Diagnostic Control)

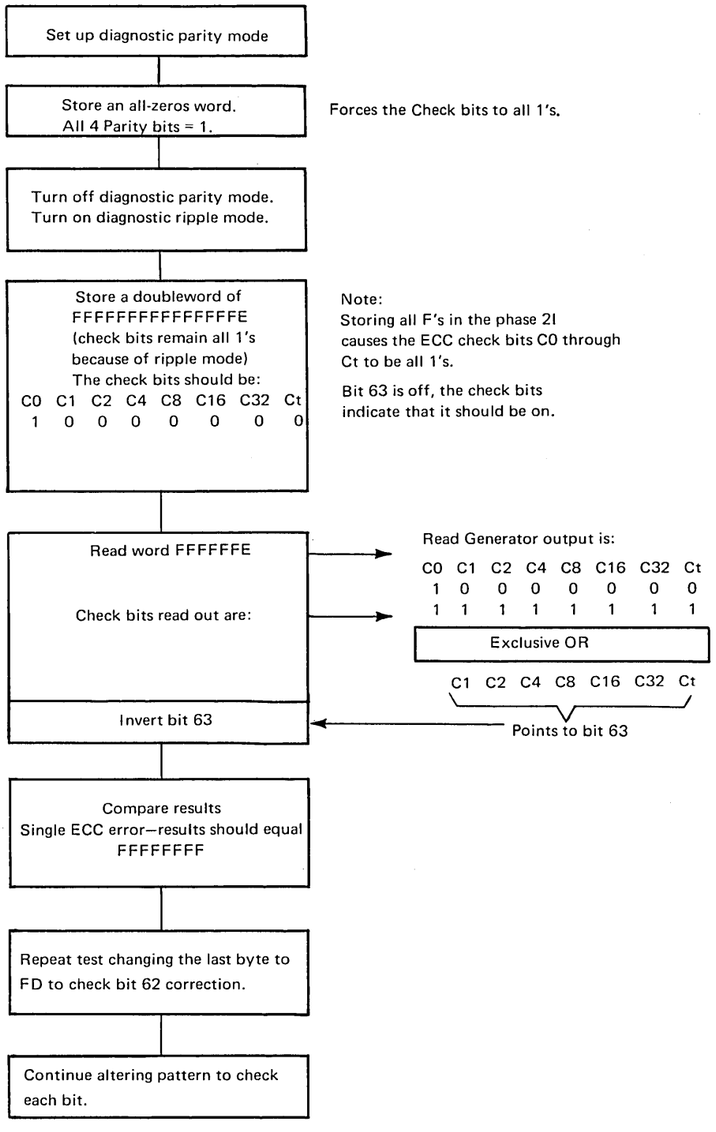
This bit is used by the storage diagnostics (BGAX) to check the error-correction circuitry of the ECC board.



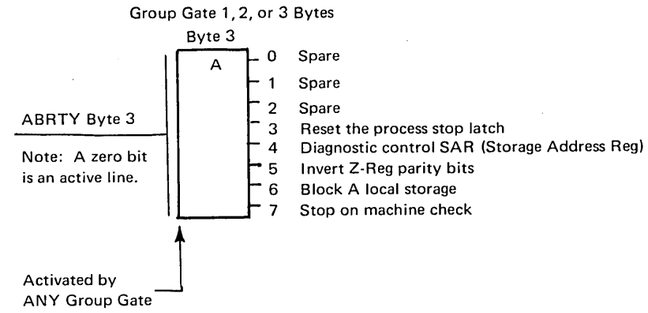
In Diagnostic Ripple Mode, the check bits from the write generator are blocked from entering the Phase 2I array. Thus, the check bits in the array are held frozen while the data bits change. This arrangement is used to force ECC correction to take place.

Microprogram Example:

Microprogram Example



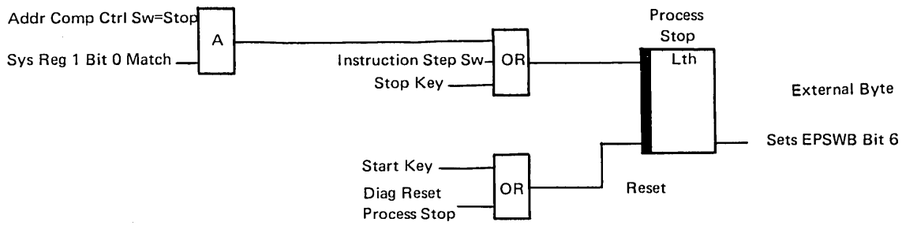
ABRTY Group Gate 1, 2, or 3, Byte 3



This byte is active if any group gate bit in DIAG0 is on.

Bits 0, 1, 2: Spares

Bit 3: Reset the Process Stop Latch



The process stop latch is interrogated by the micromonitor. It can be set by the instruction step switch or the stop key. Both of these switches can be used to request the micromonitor option to Cycle Each Test. A third way of setting the process stop latch is by setting the address-compare control switch to the STOP position. When a match on address contents occurs, the process stop latch will be set. The diagnostic function resets the process stop latch so that the manual intervention of pressing the start key can be eliminated.

This function is also used extensively by the console manual tests (MAXX).

REMEMBER

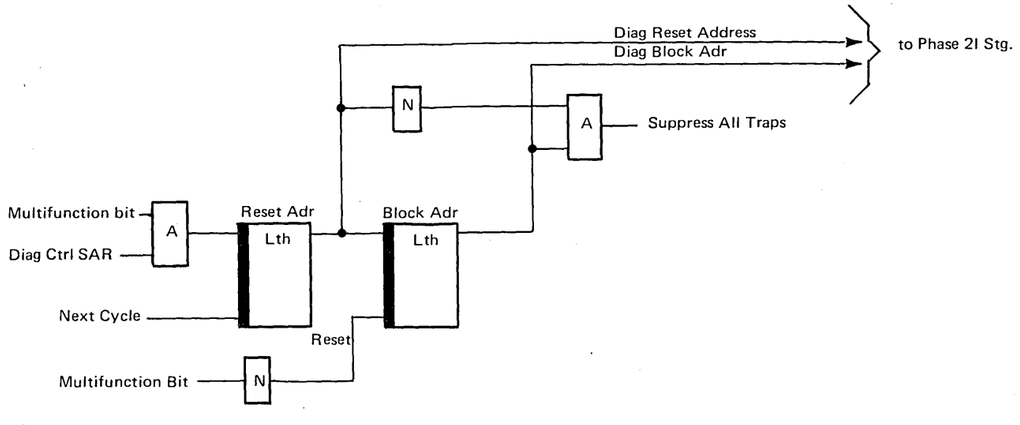
There is a Reader's Comment Form at the back of this publication.

Bit 4: Diagnostic Control SAR

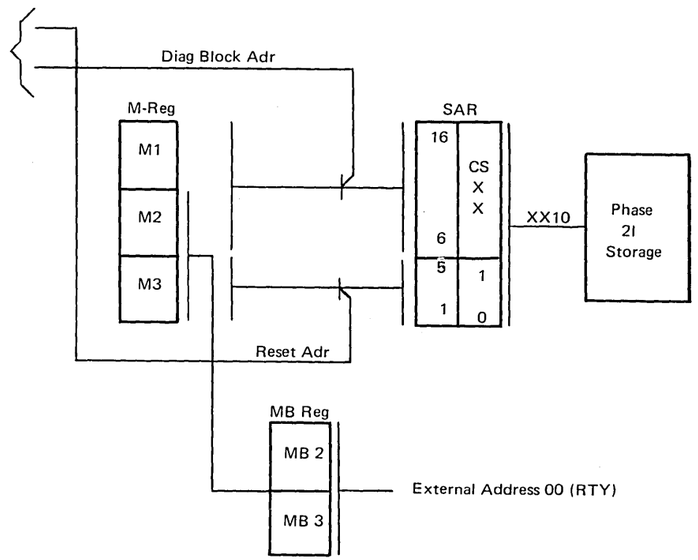
This bit is used primarily by the I-cycles tests (EDXX) and the priorities tests (EGXX).

If the wrong address is gated to SAR (Storage Address Register), the microprogram could branch to an invalid area and lose control. This diagnostic function allows the microprogram to retain control of the machine during checking for failures in the address-generation circuits (trap addresses and I-cycles generated addresses).

Two latches are activated by the 'diagnostic control SAR' line. Note that the multifunction bit (DIAG0, bit 1) must be on.

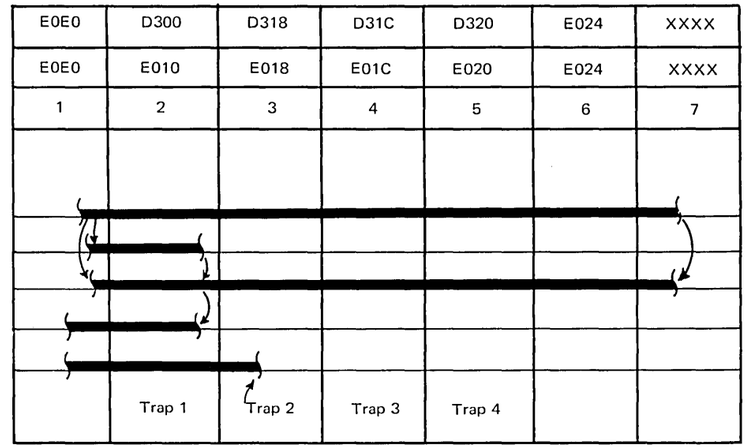


The diagnostic reset address is active for one cycle only, and the diagnostic block address is active until the microprogram resets the multifunction bit in DIAG0. The Suppress All Traps function is active as long as Diag. Block Address is on and Reset Address is off.



The diagnostic block address blocks the set/reset pulse to SAR bits 6-16. They will remain as they were in the last cycle for as long as the block address latch is on. The diagnostic reset address resets SAR bits 1 through 5 to hex 10.

Microprogram Example:



MICROPROGRAM: Diag Control SAR is on, Set IC trap is pending (D300) cycle

1.	E034	DIAG0=0,OE,K48	Allow trapping, turn on group Gate 1 set multifunction bit
2.	E0E4	(dummy word)	This should be the trap-1 cycle
3.	E010	LS1D=RTY,SC	Save MB2, MB3 from the trap-1 cycle
4.	E018	LS1E=RTY, S8	Save MB2 from the trap-2 cycle
5.	E01C	LS1F=RTY,S8	Save MB2 from the trap-3 cycle
6.	E020	(branch word)	Module switch to E0XX so that the M-Reg and N-buffer reg are realigned
7.	E024	DIAG0=0,OE,K10	Set suppress traps and reset the multifunction bit

Explanation of the Timing Chart and the Microprogram:

Cycle 1: Suppress All Traps is lifted to allow a trap to occur. The multifunction bit is turned on; this activates the Diag control SAR latches. Group gate 1 allows the Set IC Trap.

Cycle 2: This is the Trap-1 cycle. The trap address is gated into the M-Reg. However, the SAR latches are held to E010 because of the diagnostic controls.

Cycle 3: This is the Trap-2 cycle. The contents of E010 are read out and placed into the C-Reg (LS1D=RTY,SC). By now, the MB-Reg should be set to D300. The trap address of D300 will be stored away in LS1D. Also, the gate from M3 to SAR is opened (because Diag Reset Adr is off) and the next address becomes E018. (Note: The trap-2 latch resets group gate 1 to deactivate the diagnostic Set IC Trap request.)

Cycle 4 & 5: These are the trap-3 and -4 cycles. Again, save the forced module address of D3 by storing the MB2-Reg into LS.

Cycle 6: This cycle sets the M-Reg back in sync with SAR by executing a branch word with a module of E0.

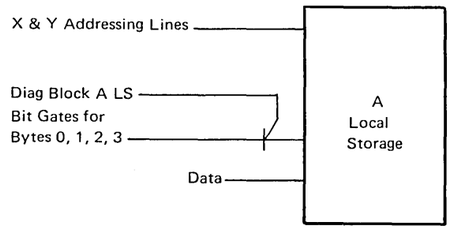
Cycle 7: Reset the multifunction bit and turn off the Diag block address latch. Branch to compare results.

Bit 5: Invert Z-Register Parity Bits

This bit causes the Z-Reg parity bits to be inverted. Data with bad parity is gated to the destination register. The next time the register with bad parity is gated out, either an A-Reg or a B-Reg check should occur.

Bit 6: Block A Local Storage

This bit is used to block writing into LS A. It does this by blocking the bit gates for each byte.



When a write to LS occurs, LS B is updated to the new value while LS A retains the old data. Flush through Check (FTC) and LS AB compare check circuits are tested by using this diagnostic control.

The micromonitor uses this function to compare expected and actual results from a test. This function allows a compare on all bits including the parity bit.

Bit 7: Stop on Machine Check

This bit, along with the 'any machine check OR' signal:
A. Stops the CPU Clock
B. Turns on the Diagnostic Stop light
C. Turns off console-file power.

This diagnostic function is used primarily by the Console Manual Tests (MAXX).

ABRTY GROUP 2

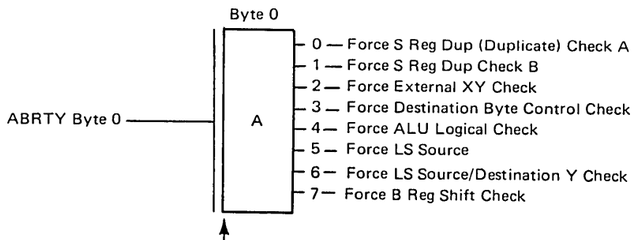
The bits in this group are active when:

1. The diagnostic key is on.
2. DIAG0 bit 5, Diagnostic Group Gate 2 is on.
3. ABRTY is loaded and frozen when the microstatement, DIAG = LSXX S0.

A group bit is active when it is a zero.

All the bits in group 2 check error-detection circuitry where it is not possible to generate an error by data manipulations.

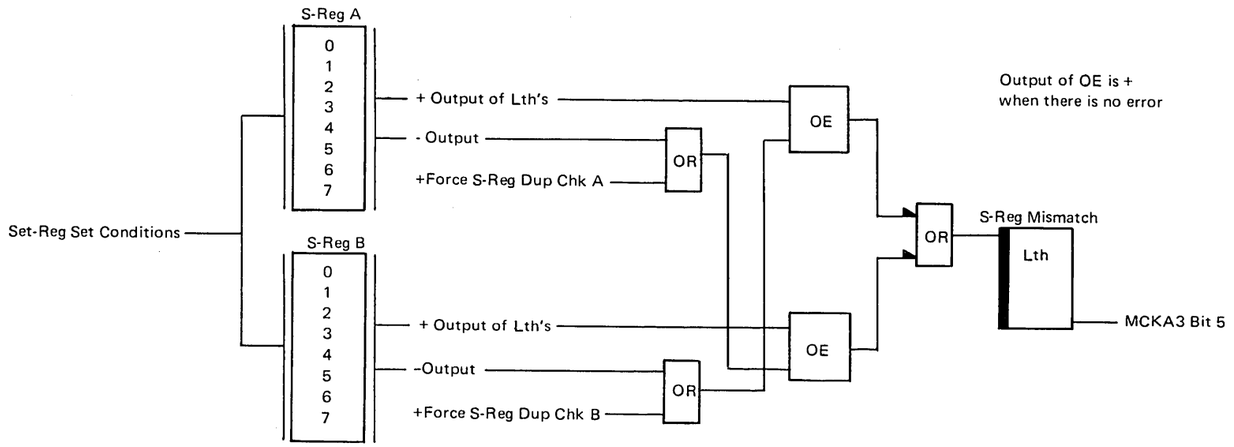
ABRTY Group Gate 2, Byte 0



Note: A zero bit is an active line

Bits 0 and 1: Force S-Register Duplicate Checks A and B

The S-register is duplicated in the machine. S-Register A is compared against S-Register B in every machine cycle; and if a no-compare exists, an error occurs.

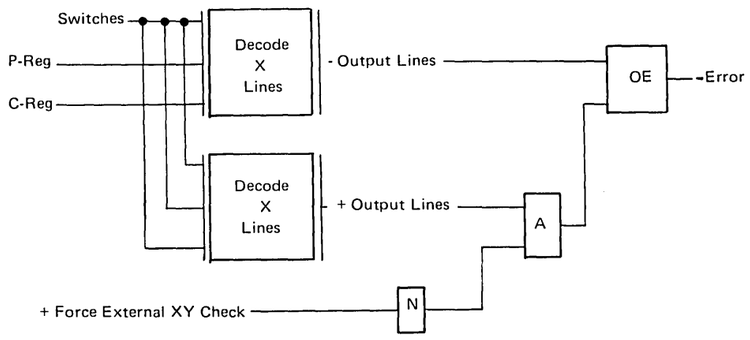


The inputs to the OE circuits should always be exclusive. When there is not an exclusive condition on any of the eight bits, machine check A, byte 3, bit 5 (S-Reg mismatch) is set. By forcing either S-Reg Dup check A or B, it is possible to determine S-register bit position failures and also the error-detection circuit failures.

Bit 2: Force External XY Check

This line forces the following errors:

1. External destination X-compare—MCKA3 Bit 0
2. External destination Y-compare—MCKA3 Bit 1
3. External source Y-assembler—MCKA3 bit 2



These decode lines are duplicated; an error exists if a mismatch occurs. The force external XY check blocks the output to the compare circuits and forces the error(s) to occur.

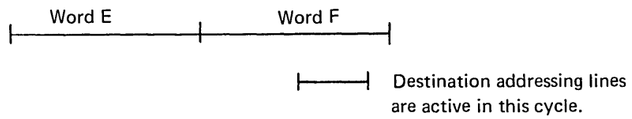
Microprogram Example: (MCKA3 should = 80)

A	LS160,A-,K20	Set DIAG mask to force EXT XY check
B	DIAG=LS16,S0	DIAG mask to ABRTY
C	DIAG=0,OE,K55	Set allow multimachine check Suppress all traps Group gate 2 and delay function
D	MCKA=NOREG,SF	Reset machine-check Regs
E	CPU=LS17,D0	Set up EXT DEST X Compare
F	BRANCH TO G	Dummy word, error occurs in this cycle Set MCKA3 Bit 0
G	LS180=MCKA3, OE,0	Save MCKA3 for results

Microwords A, B, and C set up the diagnostic controls to force an External XY check. Word D makes sure the machine-check register is reset. Word E (CPU=LS17,D0) is really a do-nothing word because no data will be moved into external address 08 (CPU), although the addressing lines are still decode. Examining the external address 08,

$$0\ 0\ \overbrace{0\ 0\ 1}^{X}\ \overbrace{0\ 0\ 0}^{Y}$$

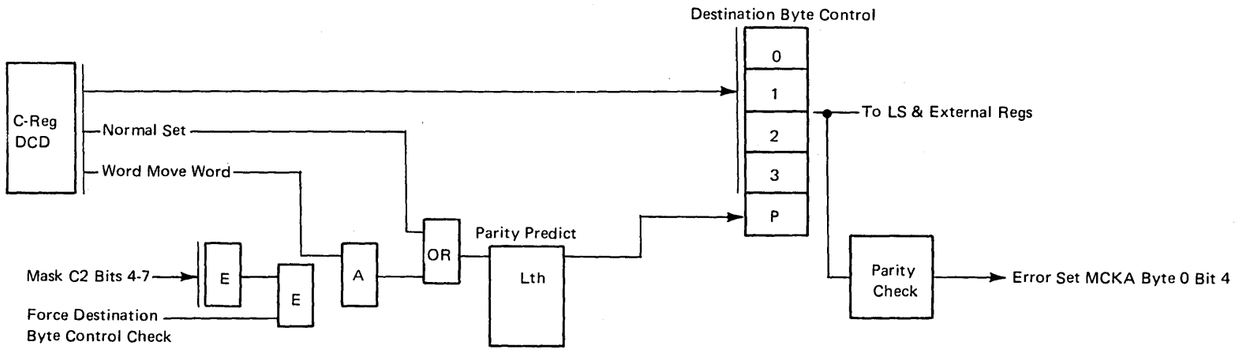
X destination decode is '1' and Y destination decode is a '0'. Word F is a branch word to next microstatement. It is during microword F that the destination X,Y lines are active.



The delayed function is also active in cycle F, blocking the decode lines from comparing. Because only 1 X line is active, the Y lines = 0; an external destination X compare occurs. The result stored in LS18 byte should be 80.

Bit 3: Force Destination Byte Control Check

The destination byte control lines control which bytes are stored into LS and the external registers.

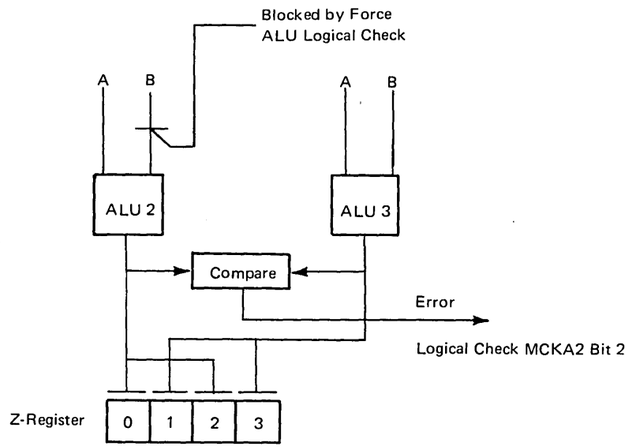


The C-Reg decoding determines which byte(s) are to be stored away into LS or the external registers. A bit is set in the destination byte control latches for each byte that is stored. For example, a word-move word LS10=LS11,SA Byte 0 and byte 2 are moved from LS11 to LS10; therefore, destination byte control latches 0 and 2 will be turned on. A parity predict latch will also be turned on to generate odd parity. If odd parity does not exist at check time, an error is generated.

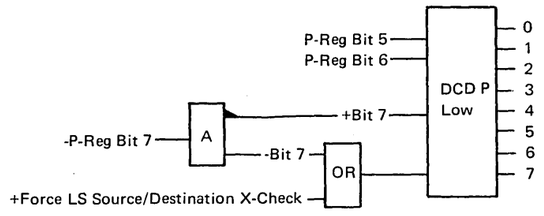
The force destination byte control diagnostic line inverts the input to the parity predict latch, forcing a parity check to occur. This diagnostic line is effective only on a word type 3 (word-move word). Reference second level diagram 1-6, E-3.

Bit 4: Force ALU Logical Check

ALU logical checks are forced by blocking the B input to ALU 2.



ALU logical operations are one byte operations. The same calculation is performed in both ALUs, and the outputs should compare equal. ALU2 is gated to Z-Reg bytes 0 and 2, and ALU3 is gated to bytes 1 and 3. If the output of the ALUs does not compare, a logical check results. A Z- and D-register check also occur because ALU2 gates to Z-Reg bytes 0 and 2.



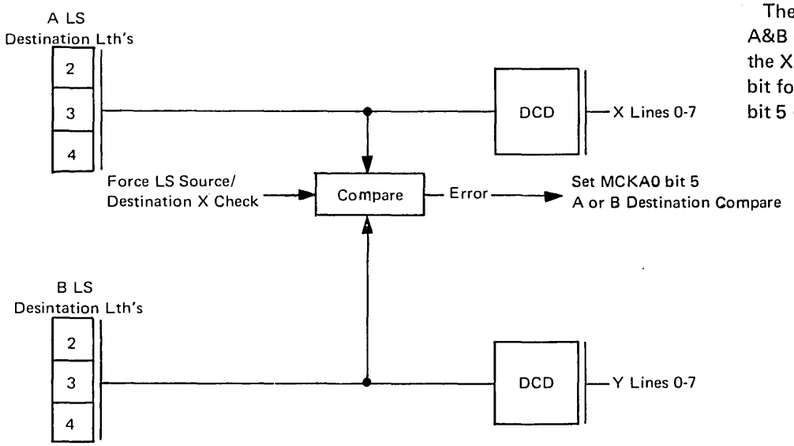
(Ref logic LA011)

Bit 5: Force LS Source/Destination X Check

This bit serves a dual function. First it forces an LS source check by conditioning 2 X drive lines.

By inverting one input of P-register bit 7 to the X-decode circuits, two X-drive lines are forced on. The X-drive lines to LS A and B are rechecked to see whether one and only one line is active for each LS access. An error will set the LS A and B source check latches, MCKA0 bits 0 and 1.

The other function of this bit forces a miscompare between the destination X lines for A and B LS.



The A and B destination latches should always be equal. The diagnostic line 'Force LS Destination X Check' causes a miscompare and set machine-check A byte 0 bit 5 error latch.

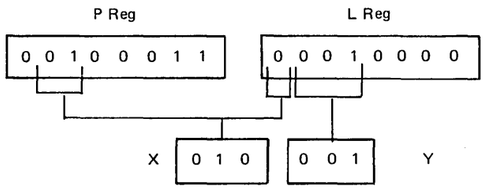
Bit 6: Force LS Source/Destination Y Check

This bit serves a dual function. It forces a LS source check by conditioning 2 Y-drive lines. This is accomplished by forcing the L-register high decode of 0 to always be active. The L-register is used to decode LS drive lines on indirect addressing arithmetic words.

Microprogram example:

A	LS160,A-,K02	Set up to force LS S/D Y checks
B	DIAG=LS16,S0	Freeze ABRTY
C	DIAGO=O,OE,K14	Turn on Suppress All Traps and group gate 2
D	P=O,OE,K23	Set P-register to indirect address LS10 to LS1F
E	L=O,OE,K10	Set L-register for indirect addressing
F	Z=LH),OR,LH)	Read out LS10 and LS11. A&B LS source checks should come on.

Microwords A, B, and C set up the diagnostic controls. Microwords D and E set up P and L to use indirect addressing and read out LS address 11.

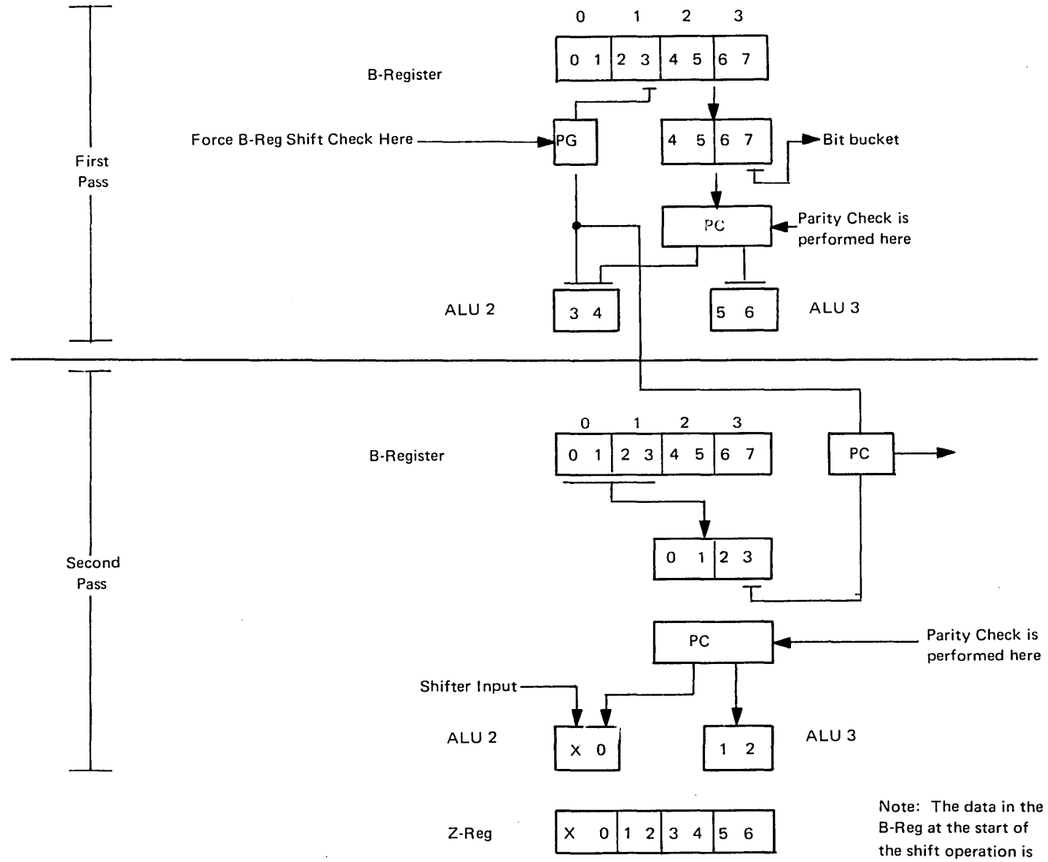


With an X-decode of 2 and a Y-decode of 1, LS11 is addressed. The diagnostic bit Force Y-Source Check also gates a Y-decode of 000, which addresses LS10.

The other function of this bit is similar to the previous bit, bit 5. A&B destination Y-lines bits 5, 6, 7 are compared the same as the X-lines. A miscompare results in a check. This diagnostic bit forces a miscompare, causing machine-check register byte 0 bit 5 (A or B destination compare) to be set.

Bit 7: Force R-Register Shift Check

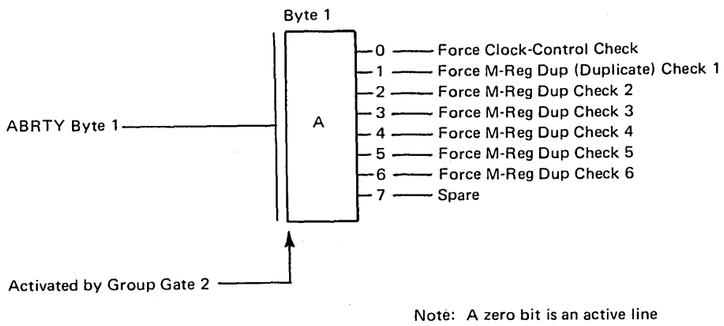
A B-register shift check can occur only during an arithmetic right-shift operation. It is a parity check of the output of the B-byte assembler and a special half-byte parity check of byte 1 bits 4-7. (Refer to the control-word section for a complete description of an arithmetic-shift operation.)



Note: The data in the B-Reg at the start of the shift operation is 01234567. After a right shift of 4, the data is X0123456 in the Z-Reg.

Byte 1 low is gated directly to ALU2 in the first pass. A special parity generator is used to form half-byte parity. This half parity generated in the first pass is compared against the parity generated on the second pass through the B-assembler. The diagnostic line 'force B-reg shift check' generates bad parity on the half byte and should cause the B-reg shift check to set (MCKA2 Bit 3).

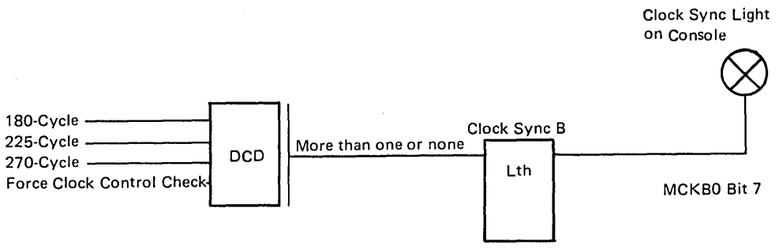
ABRTY Group Gate 2, Byte 1



Note: A zero bit is an active line

Bit 0: Force Clock-Control Check

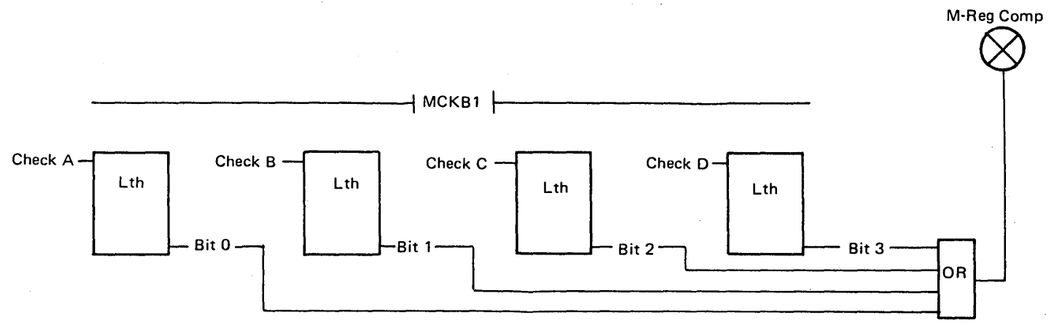
The clock-checking circuit checks that one of the three different clock cycles is active at one time. This diagnostic line forces the checking circuit to have two active at once.



Note: There are three basic clock cycles in the machine. These cycle times are extended by 22.5-nanosecond delays. However, all logic references still relate to the basic times of 180, 225, and 270.

Bits 1-6: Force M-Register Duplicate Checks

Most control and gating lines for the M- and N-registers are generated twice for checking purposes. The checking is done by an OE circuit. Both control lines must be on or off. (Refer to maintenance diagram 1-29).



The M-register compare checking circuits are composed of four check latches. Any of these check latches turn on the M-reg compare light on the console. There are many ways to set each of the four latches.

The six diagnostic control lines are used to check the control and gating lines. Errors are forced by blocking one input to the OE from the duplicate circuits. Each diagnostic bit covers several different control lines.

Bit 1 Force M-Reg Dup Check 1: This diagnostic gate checks the following control lines.

M2	Set Normal	Check A
M3	Set Normal	Check B
M2	Reset	Check C
M3	Reset	Check D

Bit 2 Force M-Reg Dup Check 2: This diagnostic gate checks the following control lines.

M2	Set traps	Check D
M3	Set traps	Check A, B
	Gate N3 to M3	Check C
	Gate trap to N2	Check D

Bit 3: Force M-Reg Dup Check 3: This diagnostic gate checks the following control lines.

Gate N2 to M2	check A
Gate N2 to M2	check A
Gate ADR/ADJ to M2	check A
Gate B2 to M2	check A
Gate C2 to M2	check A
Gate C2 to M3	check A
Gate C3 to M3	check B
Gate B3 to M3	check B
Gate ADR/ADJ to M3	check B
Gate N3 to M3	check C
Gate B1 to M1	check B
Gate ACB1 to M1	check C
N3 Bfr Bit-P Set-Reset	check D

Bit 4: Force M-Reg Dup Check 4: This diagnostic checks the following control lines.

M1 Set-Reset	check A
M Bfr set-reset	check B
Gate N2 normal	check C
S-reg branch cond. met	check D

Bit 5: Force M-Reg Dup Check 5:

Gate N2 set-reset	check D
Gate N2 Bfr set-reset	check B
Gate N2 set-reset	check A

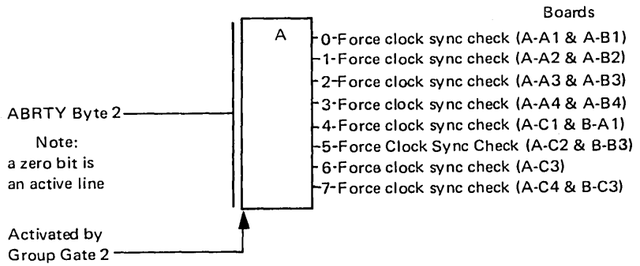
Bit 6: Force M-Reg Dup Check 6: This diagnostic checks the following control lines.

MB Reg reset	check A
M Reg byte 3 bit 4	check B
M3 bit 5 to SDBO SEL	check C

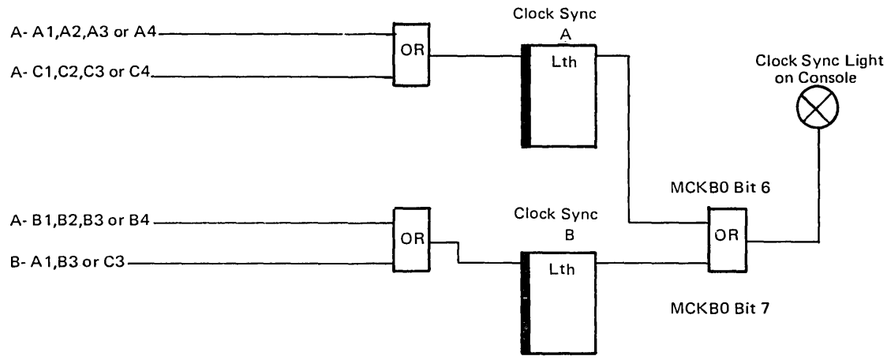
Note: When working in the logics, both names appear—M-Reg Dup check and M-Reg compare check. Both are referring to these errors.

Bit 7: Spare

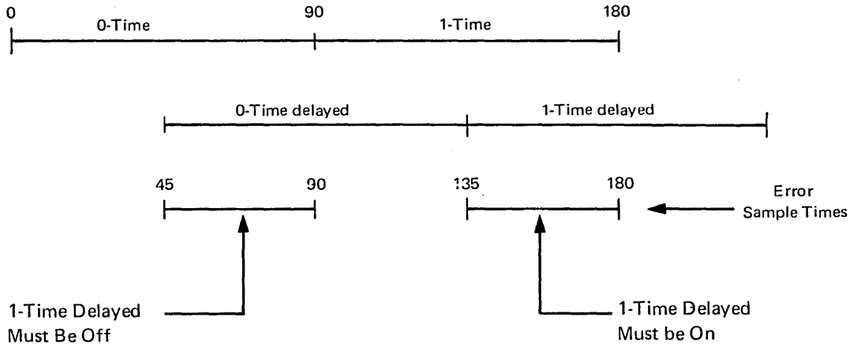
ABRTY Group Gate 2 Byte 2



Each board has its own clock card. The delays on these cards must be synchronized so that 0 time starts at exactly the same time on each board. The clock sync check assures that all boards are in step.



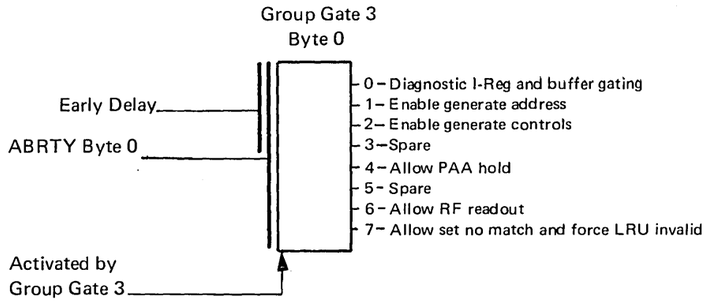
The clock sync check is performed by sampling the 1-time delayed pulse in each board and checking this pulse against timing pulses on the B2 board. (The error latches are on the B2 board.)



The diagnostic lines degate the 1-time delayed clock pulses generating an error at sample time.

ABRTY GROUP 3

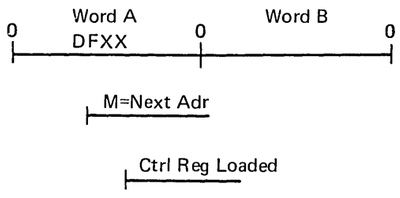
ABRTY Group 3, Byte 0



This byte is used by the I-cycles tests (EDXX) and the address-adjustment tests (EEXX). It is activated by Group Gate 3. Bits 0, 1, and 2 are used by the I-cycles controls hardware, and bits 4, 6, 7 control the diagnostic functions of the address-adjustment hardware.

Early Delay Function

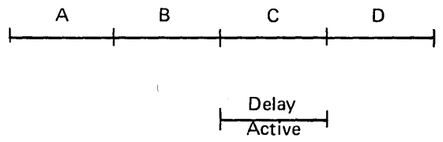
When the diagnostic delay function (DIAG0 bit 7) is used by I-cycles, it must be activated one cycle earlier than normal. This is because the I-cycles controls do not operate on a 0-time to 0-time basis. The I-cycle control register is loaded at 135 time, of the previous cycle.



I-cycles decoding for word B begins at 135 time of the previous cycle.

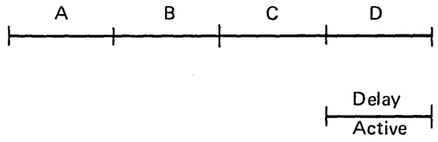
Example of the early delay function

Assume that Word A sets the diagnostic delay for the I-cycles controls.



Example of the normal delay function

Assume that word A sets the diagnostic delay for the other Group Gate bits.



Note that in both cases Word D is the microword affected by the diagnostic delay function.

Use of the Mulrifunction Bit (DIAG0 Bit 1) by I-Cycles Diagnostics

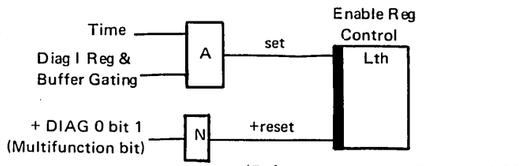
The multifunction bit is used by the I-cycles tests to perform the following functions.

1. Blocks DF module recognition.
2. Blocks the generate address line.
3. Blocks the generate controls line.
4. Blocks the normal set to the I-cycles control register.
5. Allows the use of group gate 3 byte 0 bits
6. Allows a special set to the I-cycles control register with a RDW X DC, YY. The address, YY, goes to the M3-reg. At storage 1 cycle time, M3 is gated to the I-cycle control register. The data is held in the I-cycle control reg until another RDW is executed. Example: RDW W DC, 0C. The I-cycle control register is set to 0C. Subsequent steps of the microprogram can test the I-cycles controls to permit checking the hardware function of the word DF0C.

Notes:

1. Refer to maintenance diagrams 1-70 B2,3 for I-cycles diagnostic controls.
2. Do not confuse the control registers in the I-cycles area with the C-register.

Bit 0 Diagnostic I-Reg and Buffer Gating



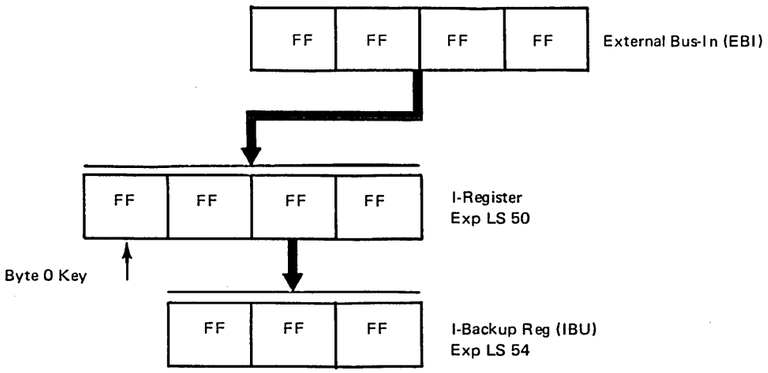
(Refer to Maintenance Diagrams 1-70)

The enable reg control latch is set with group gate 3 byte 0 bit 0 and reset when the multifunction bit (DIAG0 Bit 1) is reset.

The enable reg control latch has the following functions:

1. It allows the I-backup register (IBU) to be set with the same data that sets the I-reg when the I-reg is used as the destination.

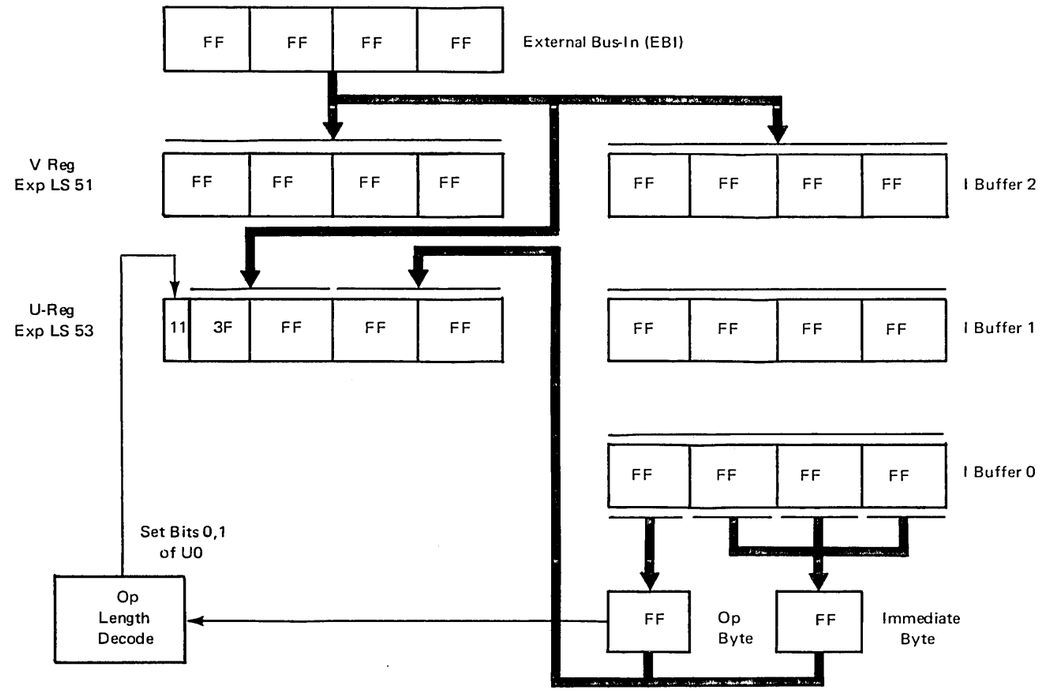
   Example: VC=0+LS14 (LS14=FFFFFFFF)



The IBU reg gets loaded with the same data as the I-reg.

2. It allows the I-buffers, the U-reg, the Op Reg and the Immediate Byte Reg to be set with the same data that sets the V-Reg when the V-Reg is the destination.

   Example: VC=0+LS 14 (LS 14=FFFFFFFF)

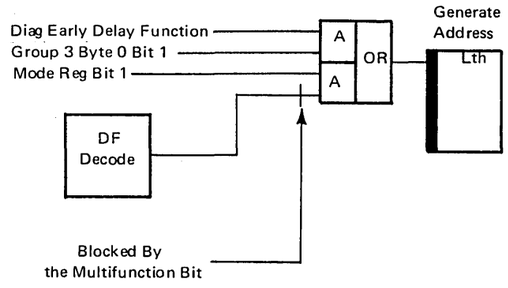


If Op is:	Then Op Reg Bits 0,1 are:	Generate or Length Decode of:
RR	00	01
RX	01	10
RS,SI	10	10
	11	11

The diagnostic function allows the data to be flushed from the External Bus-In (EBI) through I-Buffer 2 to I-Buffer 1, to I-Buffer 0 and into the Op and Immediate Byte regs. The Op and Immediate Byte regs are gated to the U-reg bytes 2 and 3. U reg, byte 1, and byte 0 (bits 2-7) are loaded from EBI. Bits 0,1 of the U-reg will be set from the instruction length decode circuits.

In this example (loading all F's) U-reg, byte 0, bits 0,1 are set to 11 because the instruction-length decode from the Op reg will indicate an SS Op.

Bit 1: Enable Generate Address

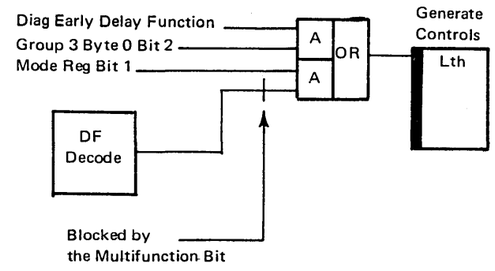


This bit, along with the diagnostic delay function, enables the generate address latch to be active for one cycle.

Normal turn-on to generate address is blocked by the multifunction bit (DIAG0, bit 1).

The purpose of the generate address latch is to enable the I-cycles control hardware to generate the next address to the M-reg. For example, if the present address was DF24, the generate address latch would cause the next address to be DF28. (Refer to Chapter 2, the I-cycles section, to see "Generate Address Gating and Controls.")

Bit 2: Enable Generate Controls



This bit, along with the diagnostic delay function, enables the generate controls latch to be active for one cycle. The normal turn-on is blocked by the multifunction bit (DIAG0, bit 1).

This latch will be used by the I-cycles diagnostic to activate the control reg decodes.

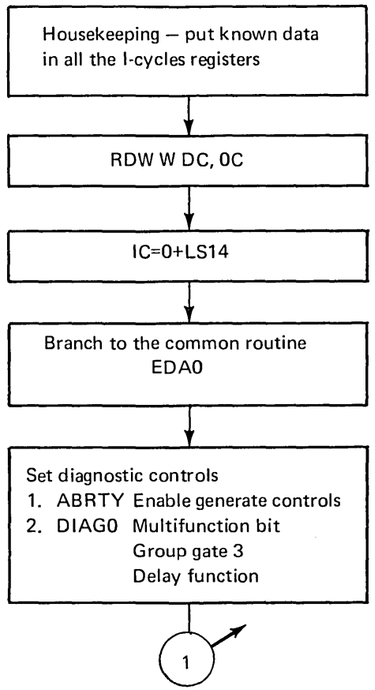
An I-cycles microprogram example is on the next page.

Microprogram Example: (Test ED A9)

This test generates the I-cycle controls necessary to load the I buffer registers and the Op and Immediate Byte registers. The starting I Address is 0000, and the Op at that location is D25A05550AAA.

The I-cycle addresses and their functions executed are:

ADDRESS	FUNCTION
DF0C–	Branch load, use I-Reg to address memory and load data to the I buffers 2, 1, 0.
DF00–	Load Op and Immediate registers with the data from I buffer 0 bytes 0 and 1.



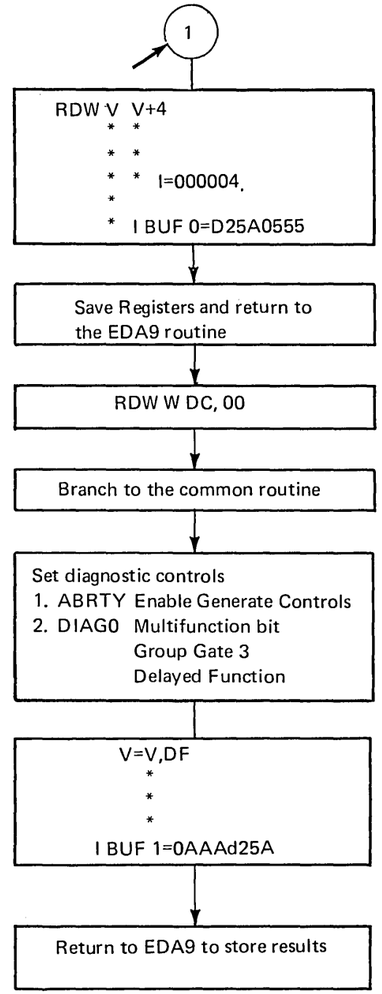
Use Diag Function—Enable Reg Control

Multifunction bit is onput 0C in the I-cycle control reg

Set the instruction counter to 000000. The first instruction is fetched from main stg location 0.

Note: In routine EDA0, the contents of main stg used by the I-cycles tests are shown; that is, PGM 0000–D25A05550 AAD25A

The common routine executes the I-cycles word DF0C



Delay function is active. This word simulates the I-cycles word DF0C
—fetch from main stg using the I Reg
—load the data into the I buffers

Note: I-cycles hardware is enabled for I-cycle only because of the diagnostic controls.

Put 00 in the I-cycle control reg

The common routine executes the DF00 word

Delayed function is active. This word simulates the I-cycles word DF00, a delay word to complete loading the I-buffers.

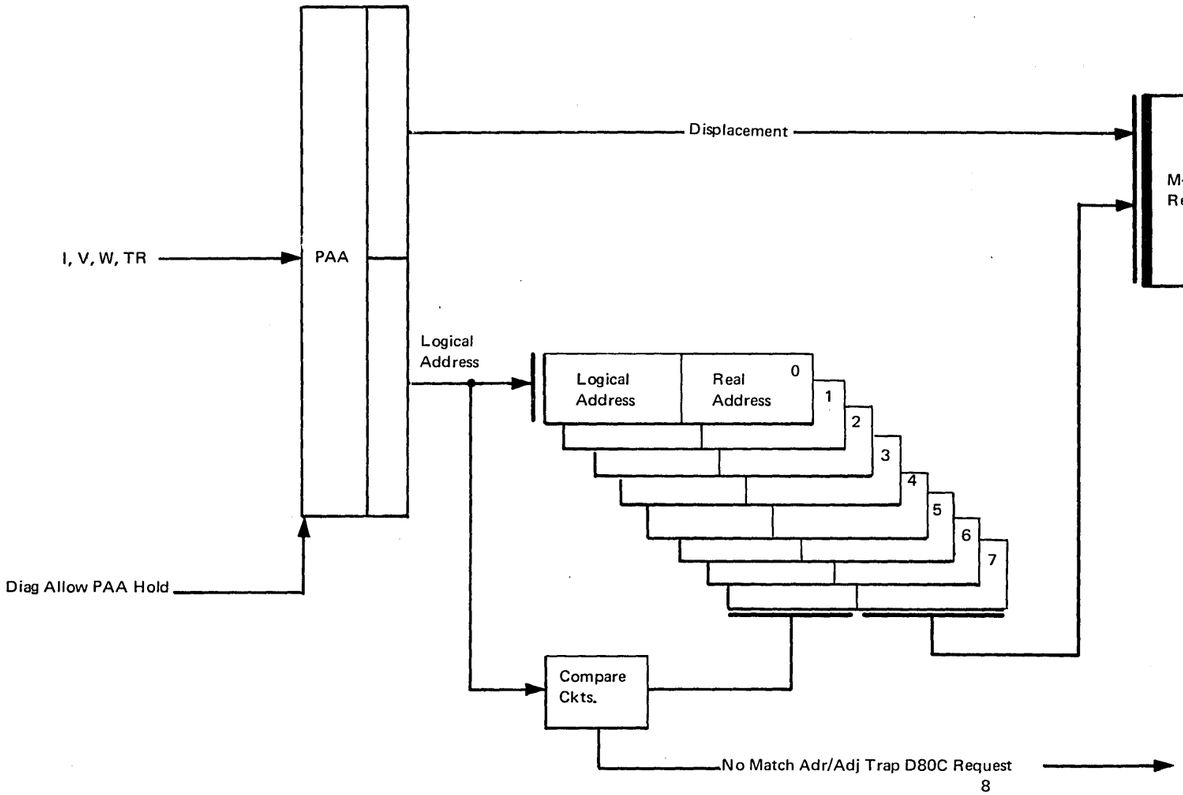
Summary: By using diagnostic hardware, two I-cycles words were simulated, DF0C and DF00. The next test, EDB0, builds on this test by completing the SS Op of D2. DF7C and DF70 are simulated next.

Bit 4: Allow Preaddress Assembler (PAA) Hold

During an address-adjustment trap, the address that caused the trap is held frozen in the PAA.

At the end of the trap routine, the M-Reg is loaded from the PAA (displacement bits) and the Real Address portion of the DOS table buffer registers.

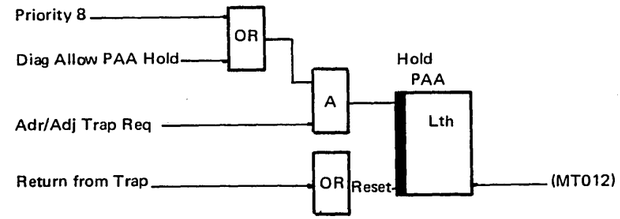
Because the trap and priorities tests have not been executed yet, the trapping hardware circuits cannot be used. The Suppress All Traps function is still activated by DIAG0, Bit 3.



The diagnostic line 'Allow PAA Hold' simulates the priority 8 signal from the trapping circuits and holds the PAA frozen.

The hold PAA diagnostic function becomes active during a Stg 2 cycle (see test EEB7) RDW LS15 ADJ,I. The contents of the I-Reg are held on PAA. Now the address-adjustment tables can be loaded and then verified with the next diagnostic function.

After the diagnostic test has been completed, the PAA is released by the microword RDH DK CS, WK. This is normally the last word of the address-adjustment trap routine.



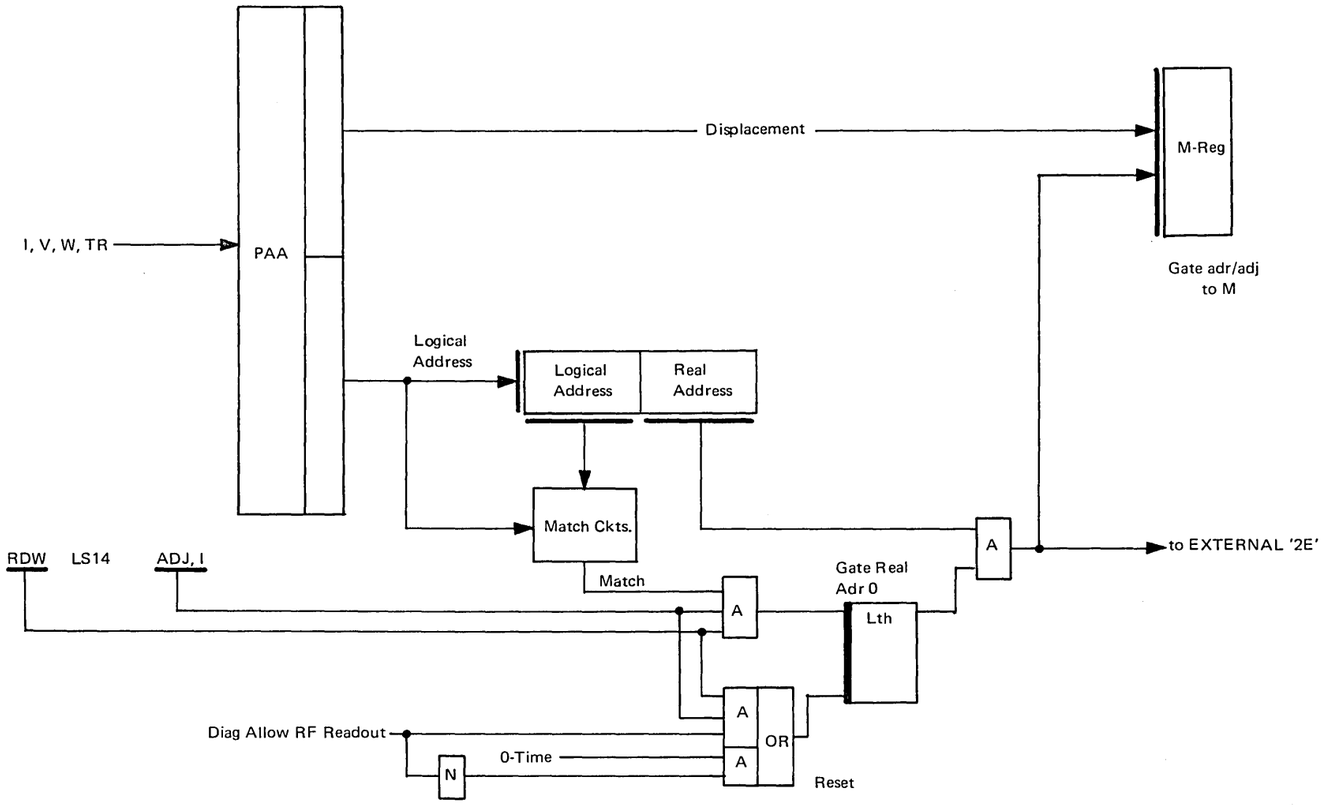
Bit 6: Allow RF Readout

Normally, when the machine is running, there is no way to examine the contents of the address-adjustment registers.

Only when the CPU clock is stopped is it possible to display the address-adjustment registers through External Address 2E, using switch H to determine which Reg to display.

This diagnostic function allows the address-adjustment Regs to be displayed dynamically through the external address 2E and read by the microcode.

When a read word with address adjust is specified, the Logical Address registers are scanned. If a match occurs, the corresponding real address is gated to the M-Reg along with the displacement bits from the PAA. Normal operation is to reset the Gate Real Address latches at 0 time of the next cycle.



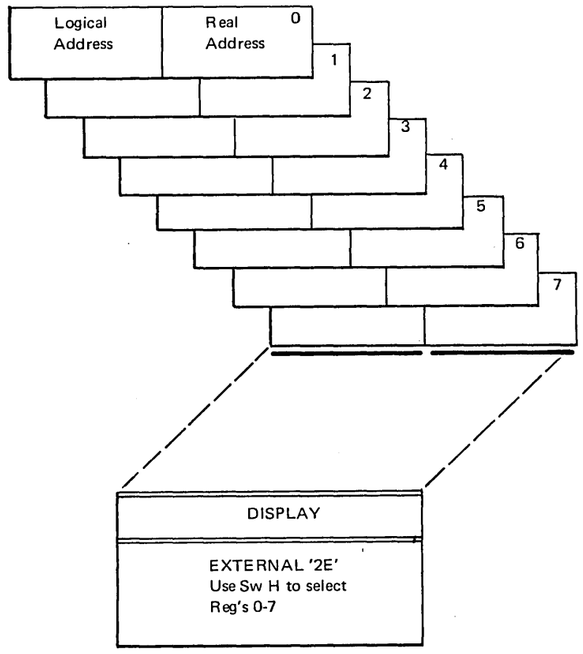
This diagnostic control blocks the reset until another storage word with address adjust is given. Therefore, the real address is held frozen and can be checked by reading External Reg 2E bytes 2 and 3; that is, LS153=ADJT3,OE,0.

Note: Both the Logical address and the Real Address can be displayed manually. With the diagnotic control 'allow RF readout' only the Real Addresses can be read dynamically in External Address 2E.

Bit 7: Allow Set No Match and Force LRU Invalid

This bit forces the following errors:

1. No Match (MCKB1 Bit 4)--Caused by the Set Eliminate signal on MT012.
2. LRU Invalid (MCKB1 Bit 6)--Caused by the Force LRU invalid signal on MT014.



SELECTOR CHANNEL GA DIAGNOSTIC FUNCTIONS

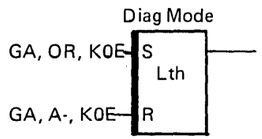
GA,OR,K0E	Set Diagnostic Mode, Reset Command Out	GA High/Low Decodes
GA,A-,K0E	Reset Diagnostic Mode	
GA,A-,K0D	Diagnostic Service Signal	
GA,OR,K80	Set Diagnostic Block Buffer Clock	
GA,A-,K80	Reset Diagnostic Block Buffer Clock	
GA,A-,K60	Set BF, BP Pulse, Shift Buffers	
GA,OR,K90	Set Diagnostic Status	
GA,A-,K90	Reset Diagnostic Status	
GA,OR,K11	Set Diagnostic Block Share Cycle Latches	GA Straight Decodes
GA,OR,K22	Reset Diagnostic Block Share Cycle Latches	
GA,OR,K33	Diagnostic Check Reset	
GA,OR,K44	Set Diag Function latch and Reset Expanded LS	
GA,OR,KFF	Reset Diagnostic Function latch	
GA,OR,K66	Reset Block MPX Scan latch (Set Diag Trap 1)	
GA,OR,K88	Set Block MPX Scan latch (Block Diag Trap request)	

SELECTOR-CHANNEL DIAGNOSTIC CONTROLS

The selector-channel tests use several GA decodes to perform diagnostic functions in the channel. The GA decodes are the channel-set/reset-functions performed in the word type 1.

The set/reset source (C-Reg, byte 2, bits 23=11) in word type 1 specifies the use of the GA decodes.

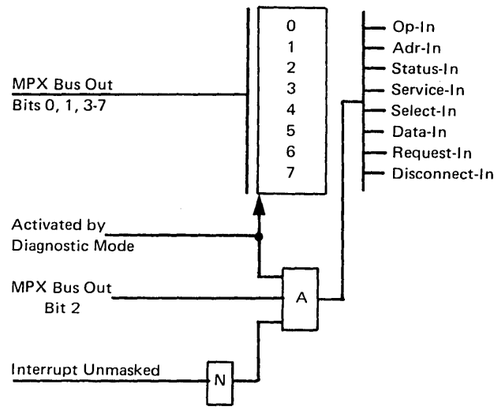
The decodes that follow are active *only* on the channel being tested. These decodes are called, Set/Reset GAH (high) and GAL (low).



GA, OR, K0E – Set Diagnostic Mode, Reset Command Out
GA, A–; – Reset Diagnsotic Mode

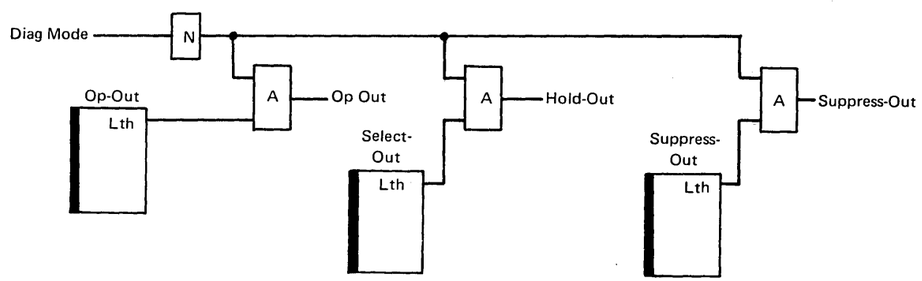
In order to test the selector-channel responses to incoming lines from the control units, it is necessary to simulate interface lines with diagnostic controls.

The diagnostic mode latch is set and reset with a GAL decode of 0E. The diagnostic mode latch degates Op Out, Hold-Out, and Suppress-Out from going to the interface drivers. Diagnostic mode and a GA decode of KOC (GA, OR, KOC) allows a set to the GR Full latch to control the data transfers of the GR Buffers. The diagnostic mode latch set also provides a reset to the command-out latch.



Note:
Use lower roller position 5 to display these indicators.

(Refer to Maintenance Diagrams 1-47 A1)



(Refer to Maintenance Diagrams 1-47 I 1-6)

GA, A-,K0D - Diagnostic Service Signal

In normal operation, the Service Signal line is made active by different conditions of the Tag-In lines. Service signal allows the set of Command-Out, Service-Out, or Data-Out.

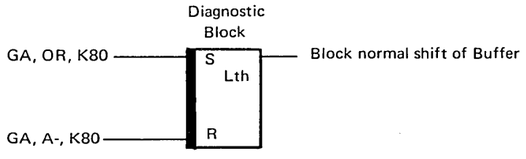
Because the normal tag-in signals cannot be used to generate the service signal while microdiagnostics are run, a diagnostic service signal must be generated.

Other functions of the service signal are:

1. Provides a conditioning level to set the 'GR full' latch.
2. Provides a reset to the 'buffer position 0 full' latch.
3. Provides a conditioning level to the channel 'data check' latch.

GA, OR, K80-Set Diagnostic Block Buffer Clock

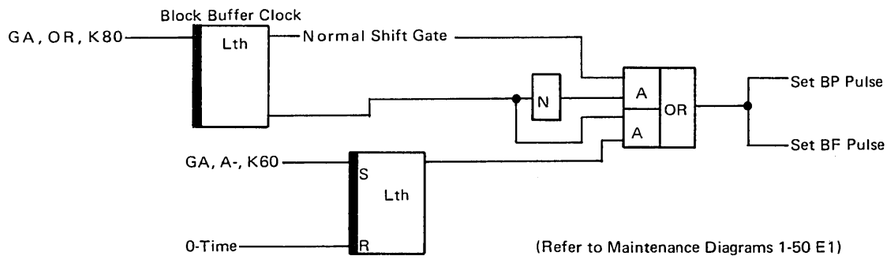
GA,A-,K80-Reset Diagnostic Block Buffer Clock



The 'Diagnostic Block Buffer Clock' latch blocks the automatic shifting of data from one buffer reg to another.

GA,A-,K60-Set BF, BP Pulse, Shift Buffers

Normal operation of the buffers allows data to be shifted through as many buffers as possible, depending on the buffer controls.



The 'set and BP pulse, shift buffers' is a diagnostic latch that turns on for one cycle and allows one buffer clock cycle. This control enables the microprogram to check every buffer position.

SELECTOR-CHANNEL MICROPROGRAM EXAMPLE

```
*** TEST EJD5  BUFFER CHANNEL ONLY ***
A              TEST THE ACTION OF BUFFER FULL BITS AND THE
               BFR BYTE COUNTER (GB).

          * TEST FORM - ROUTINE
          * RSLT FORM - NORMAL

               * DESCRIPTION *

B              DIAG MODE AND DIAG BLK BFR CLK LATCHES BEING
               ON ALLOW SINGLE CYCLE CONTROL OF THE BUFFER.
               A DATA PATTERN IS SET INTO GR AND GR FULL IS SET
               VIA INPUT AND SERV SIG DLY. GBF-BFR FULL BITS AND
               GBD -BYTE COUNTER ARE GATED TO MONITOR RSLT
               FIELD. GR FULL IS RESET AND THEN SET AGAIN TO
               ENABLE FULL BIT PROPAGATION VIA THE DIAGNOSTIC
               SHIFT - (GA,A-,K60) WHICH ALLOWS 1 SET BF AND
               1 SET BP EVERY MICRO INSTRUCTION.  THE TEST
               CONTINUES UNTIL ALL FULL BITS ARE ON - BF0-6 AND
               GR FULL. GB WILL INDICATE 8 BYTES IN THE BUFFER.

               * ALD PAGES INVOLVED - GC31X,GC41X,GB213,GC512

               * EXPECTED RESULTS *

               BYTE 1 SHOULD BE 00        GBF - BFR FULL BITS
               BYTE 2 SHOULD BE 01        GBD - BFR BYTE CNTR

*** NOTE                                                        *
C              THE FIRST PORTION OF THIS TEST WHICH CONSISTS OF  *
               THE FACILITIES FOR - LOOPING,RESET OF ATTACHMENT, *
               SETTING THE PROPER CHANNEL GATE,ETC, IS LOCATED   *
               IN TEST - EJA0 AND IS LABELED - ALL TESTS START   *
               HERE.                                             *
*****************************************************************

************** START TEST ***********************************

D              A     P=0,0E,K94          SEL CH L/S AREA.      E04
               B     RDW GD DM,08        SET INPUT (CCW1).     L20
               C     GR=0,0E,KFF         SET FF INTO GR REG.   E23
               D     GA,A-,K0D           ALLOW DIAG SERV SIG.
               E     GA,A-,K60           *ALLOW BF/BP TO SET
                                          GR FULL LATCH.
               F     P=0,0E,K93          LS18-1F AND EX20-3F.  E04
               G     LS180=GBF,0E,0,ABCK       R E S U L T  1. L18 E21
                                         GBF - ALL BITS OFF.
               H     LS181=GBD,0E,0,ABCK       R E S U L T  2. L18 E21
                                         GB3 ON,1 BYTE OF DATA
                                         IN BFR(GR FULL IS ON)
```

Explanation of Microprogram Example

A Test EJD5 will run only on a buffered selector channel. The test name EJXX indicates that channel one is being tested.

EKXX - indicates that channel 2 is being tested.
ELXX - indicates that channel 3 is being tested.
EMXX - indicates that channel 4 is being tested.

Tests ENXX are common tests to all selector channels.

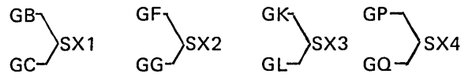
IMPORTANT: All of the selector channel tests in the micro-listings are labeled EJXX. Example: Assume that a failure occurred on selector channel 3; the console printout would be:

```
S    ELD5
     Actual Results     0000
     Error ID             **
     Expected Results   0001
```

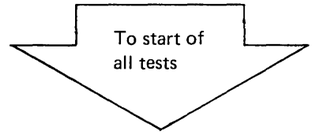
If fault-locating information fails to resolve the problem, the microlisting must be used to analyze the failure. The failing routine in the micro-listing is EJD5. There is no microroutine labeled ELD5.

The common routine EJA0 is used to set up the proper channel gates and cause the test ID to change when testing switches from one channel to another. This routine also contains descriptive text on all of the L.S. and External Channel words.

B A description of the test follows. The ALD pages reference channel 1 *only.*



C This note appears at the beginning of each selector-channel (SX) test. Routine EJA0 contains all the necessary setup procedure to run this test. To find out what happens in the common routine, go to EJA0 and see the full-page note.



(This note appears in EJA0, about sequence No. 750)

Test Start Here

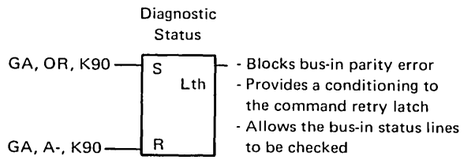
D Microstep A sets the P-Reg to address the channel LS area. Note the LS and EXT addresses to the right of the microprogram comments, L20, E23, and E21. The actual addresses differ from the listing depending on the channel being tested. For example: L20 specifies the LS address for the GD-Reg for selector channel 2. If selector channel 1 is being used in this routine, the real local-storage address is LS28. Use the lights in the upper roller, position 1, to determine which channel is being tested.

Microword B loads Local Storage (L20-SX2), (L24-SX3), (LS28-SX1) or (LS2C-SX4) and also sets the channel 'input forward' latch. Step C puts FF into the GR-Reg. Setting the diagnostic Service Signal in step D allows the GR Full Latch to be set with the next microword (E), the diagnostic latch that allows a BF/BP pulse. Steps G and H save the first two results. No data has been shifted into the buffer yet, so GFB (Buffer Full bits) should equal 00. GBD (Buffer Byte Counter) should be equal to 01 because one byte has been transferred into the GR-Reg.

NOTE: Lower roller positions; 5-8 are helpful when trouble-shooting channel problems. Be sure to select the right channel on the Storage Select switch.

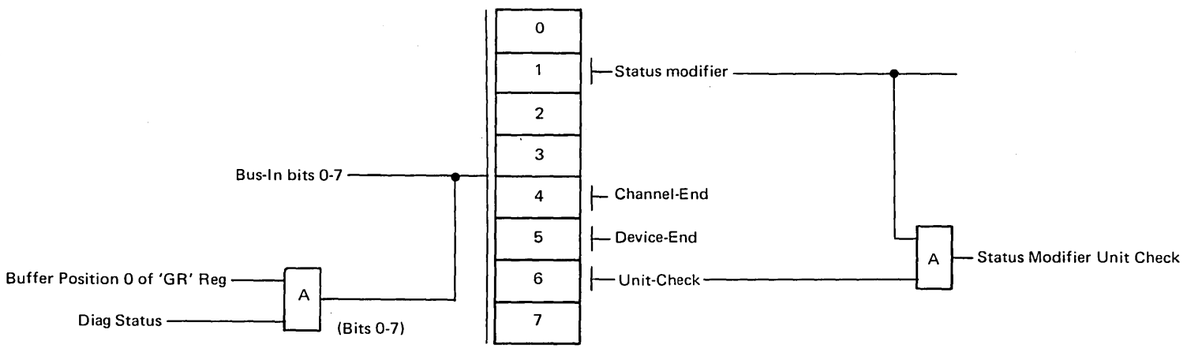
GA,OR,K90--Set Diagnostic Status

GA,A-,K90--Reset Diagnostic Status



Diagnostic Status blocks bus-in parity errors from generating an interrupt condition in the channel. It also simulates the Mark-In input to the command retry latch.

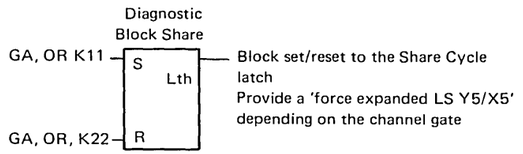
The primary reason for the diagnostic status latch is to simulate bus-in status signals so that these signals can be verified without using the interface. Buffer position 0 (buffered machine) or GR-Register (nonbuffer) is loaded with the status bit(s). The diagnostic status latch provides the conditioning level to activate the status conditions. The bus-in status signals that are generated are: Channel-End, Device-End and Status Modifier, Unit Check. Status Modifier, Unit Check is the signal that conditions a command retry operation.



The decodes that follow are active for all channels. These decodes deal primarily with the generation of share cycles and trap requests. (The decodes are called 'set GA, straight decodes'.)

GA,OR,K11--Set Diagnostic Block Share Cycle Latches

GA,OR,K22--Reset Diagnostic Block Share Cycle Latches



(refer to Maintenance Diagrams 1-46 B6)

The diagnostic block share cycle latch blocks the set/reset to the share cycle latches for all channels.

The selector-channel microtests use this line to block the share cycle latches; then the test sets up the conditions to force a share cycle. The GA,OR,K22 resets the blocking function and allows one share cycle to occur. This diagnostic control permits the microdiagnostic to check the share-cycle controls without losing control of the microprogram.

GA,OR,K33--Diagnostic Check Reset

This decode simulates the check reset key on the console and resets all of the channel check control latches. The check latches are reset with this GA decode to prevent the manual intervention of pressing the check reset key.

GA,OR,K44--Set Diagnostic Function Latch and Diagnostic Reset Expanded L.S.

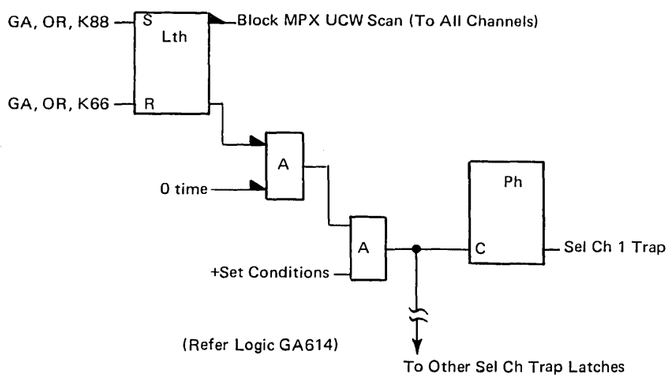
GA,OR,KFF--Reset the Diagnostic Function Latch

These two decodes are used to set and reset controls and latches in the Expanded L.S. area used by the selector channels.

GA,OR,K66--Reset the Block Multiplex UCW Scan Latch Allow Trap Latch SET/RST In Common Channel Board.

GA,OR,K88--Set the Block Multiplex UCW Scan Latch Diagnostic Block Trap Latch SET/RST In Common Channel Board

The block-multiplexer UCW Scan Latch is used in diagnostic mode to control the set/reset to the selector-channel trap latches.

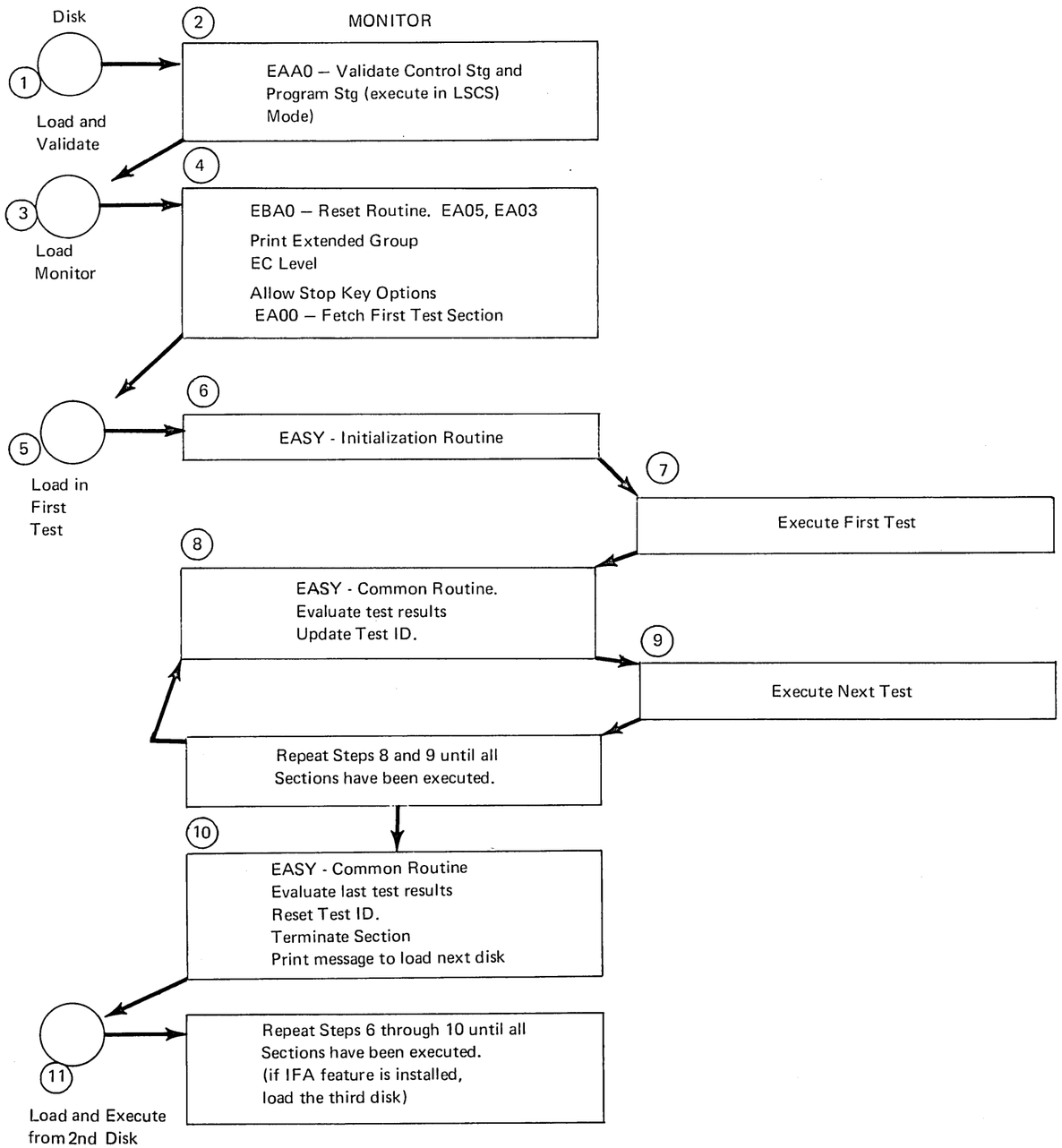


By resetting (GA,OR,K66), the block MPX UCW scan latch, the selector-channel trap latches are conditioned to turn on. When the set block MPX UCW Scan (GA,OR,K88) is issued, the selector-channel trap latches are blocked again. This control enables the microdiagnostic to check out the trap and priority circuits in the channel (Section ENXX).

NOTE: The block-multiplexer scan latch is used only during IMPL when the microcode is being loaded. This GA decode allows the block MPX plug card in each channel to be scanned and check the type of devices on the channel. The plug cards are scanned so that the UCW tables in control storage can be loaded properly. Because this is the only function of this GA decode, the diagnostic can make use of an unused leg of the block multiplex UCW scan latch.

MICROMONITOR INFORMATION

Flowchart of Micromonitor Control



Diagnostic Local Storage Assignment

	0	1	2	3
00		Common Work		
01		Area		
02				
03		Undefined		
04				
05		Work		
06				
07		Area		
08				
09				
0A				
0B				
0C				
0D				
0E				
0F		Sys Reset Link		
10		Section		
11				
12		Work		
13				
14		Area		
15	5 5	5 5	5 5	5 5
16	F F	F F	F F	F F
17	F F	F F	F F	F F
18	Actual Results			
19	Field For			
1A	Exfended			
1B	CPU FLT			
1C	Common Work Area			
1D	Used By Both			
1E	Monitor and			
1F	Tests			

	Selector			
	Channels			
30	FLT Table Pointer			
31	Backup Pointer for Text Loop			
32	Setup Area Pointer			
33	Test Number Link Register			
34	Monitor Work Register			
35	Monitor Work Register			
36	Monitor Work Register			
37	Monitor Work Register			
38	Print Message Link			
39	Flags	Search	Test	Ident
3A	Monitor		Link	
3B	Flags Section		Alpha Dec Flags	
3C	Current Test Link			
3D	Error Flags			
3E	Flags	Flags Err Cor		Feat 3
3F	Section Link		Reg	

Sense Switches in LS3B0

0 Print Exp-Act Rslts
1 Loop Test
2 Loop Section
3 Test ID
4
5 Print Bypass
6 Print Errors
7 Print Instructions

Flags in LS3E1

0
1 *EX1 was run
2 *EX2 was run
3 *EX3 was run

Flags in LS390

0 Cannot Single-Cycle
1 Test Search
2 Last Test
3 Overlay
4 No Loop SSW
5 Feature Test
6 1 = Read 0 = Write
7 Main Storage Print

Flags in LS3B3L

4 Already In Test
5
6 Cycle Each Test
7 Retry Flag

Flags in LS3E0

0 No New Line
1 Retain Overlay
2 Log Failing Test
3 Log Present
4 Extint Bit 1
5 Print Section Message
6 Printer 2 Reset
7 Restart Manuals

APPENDIX A. MAINTENANCE AIDS

Control-Word Bit Charts (Part 1)

Bit Definition of the Branch and Module-Switch Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch and Module Switch				**Branch High**				**Branch Source**						**Branch Source Dest**		**Module Address**								**Next Address**				**Branch Low**			
								Word				**Byte**																			
0	0	0	0	0000=0 0001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6 0111=BH 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7						00=0 01=1 10=2 11=3		00=--- 01=(S=Branch Source) 10=(T=Branch Source) 11=(L=Branch Source)												0000=- 0001=1 0010=Z0 0011=NZ 0100=S3 0101=S5 0110=S7 0111=BL 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7			

Bit Definition of the Branch and Link or Return Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch and Link or Return			**Link Rtn**	**Branch High**				**Link Address**								**K/Module**								**Next Address**					**Branch Low**		
								LS or EXT	**Y**			**X**		**Spare**																			
0	0	1	0	0=link 1=rtn				0=LS 1=EXT								BAL— If C3 bit 4=0, this field is set into the P-register. If C3 bit 4=1, this field contains the Module address for M3. RTN—Any bit in this field resets the corresponding bit in the H-register											BAL 0=P 1=MS RTN 0=--- 1=use NA				0000=0 0001=1 0010=Z0 0011=IO 0100=S3 0101=S5 0110=S7 0111=I1			

Control-Word Bit Charts (Part 2)

Bit Definition of the Branch Word

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Branch				Branch High				Branch Source						K Hi-Lo			S/R	S/R Source		K				Next Address				Branch Low			
								Word				Byte																			
0	0	0	1	0000=0 0001=1 0010=S1 0011=S0 0100=S2 0101=S4 0110=S6 0111=BH 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7								00=0 01=1 10=2 11=3				00=MS 01=L 10=H 11=St		0=--- 1=DK	0=,OR, 1=,A,	00=BS 01=S 10=P 11=GA								0000=0 0001=1 0010=Z0 0011=NZ 0100=S3 0101=S5 0110=S7 0111=BL 1000=B0 1001=B1 1010=B2 1011=B3 1100=B4 1101=B5 1110=B6 1111=B7			
																◄ - - - - - - - - Module Address - - - - - - - - ►															

Bit Definition of the Word-Move (Version 0)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word Move				Version	Branch High			Destination								Source Note 1				Mask				Next Address					Branch Low		
								LS or EXT			Y			X	LS ELS																	
0	0	1	1	0	000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6			0=LS 1=EXT							0=LS 1=ELS											0=--- 1=Stop			000=0 011=1 010=Z0 011=--- 100=S3 101=S5 110=S7		

Note 1. The source cannot specify an external register. SPTL, however, can be specified by C2, B0-3.

Bit Definition of the Word-Move Word (Version 1)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Word Move				Version	Branch High			Source								Destination				Mask				Next Address					Branch Low		
								LS or EXT			Y			X	Spare																	
0	0	1	1	1	000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6			0=LS 1=EXT																		0=--- 1=Stop			000=0 001=1 010=Z0 011=--- 100=S3 101=S5 110=S7		

Control-Word Bit Charts (Part 3)

Bit definition of the Storage Word (Non-K-Addressable)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Storage Word		**Subform**			**Branch High**			**Data Register**				**K or Inc/Dec**		**Stat Set**		**Address Source**				**Modes**		**Special Stat Set**		**Next Address**					**Branch Low**		
0	1	000=Read Word 001=Store Word 010=Read Half Wd 011=Store Half Wd 100=Read Byte 101=Store Byte 110=RDWRL/RDMP 100=Read Key 101=Store Key Subform for (special-stat set			000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6 111=M6							00=K-Addr 01=No Addr update 10=+ 11=-		00=— 01=S2 10=S45 11=Z6						00=CS 16 bit address 01=MS 10=ADR ADJ 11=CPU prot		00=— 01=TA 10=TB 11=special 00=— 01=TH 10=— 11= special		Read Store			0=— 1=Dec Cnt			000=0 001=1 010=Z0 011=SDC/VAL(0) 100=S3 101=S5 110=S7 111=M7	

Bit Definition of the Storage Word (K-Addressable)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Storage Word		**Subform**			**Branch High**			**Data Register**				**K Addr**		**K Modes**		**Address Source**				**K**				**Next Address**					**Branch Low**		
0	1	000=Read Word 001=Store Word 010=Read Half Wd 011=Store Half Wd 100=Read Byte 101=Store Byte			000=0 001=1 010=S1 011=S0 100=S2 101=S4 110=S6 111=M6							00 00=MS (0 00 KK) 01=CS (CM KK) 10=CS (FF KK) 11=CS (FK 8 bit Addr)														0=—					000=0 001=1 010=Z0 011=— 100=S3 101=S5 110=S7 111=M7	

Bit Definition of the Arithmetic Word (Type 10 Byte Version)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		**Form**		**Operation**				**A Source**						**Stat Set**		**B Source**						**A Hi-Lo**		**Next Address**						**Branch**	
								Word				**Byte**				**Word**				**Byte**											
1	0	00=A=A/K 01=Z=A/B 10=A=A/B 11=B=A/B 00=Z=A/K		0011=,OE, 0100=+ 0101=+ (+1) 0110=C + (Rst S0) 0111=,A, 1000=,OR, 1001=C + - (+C) 1010=C ,D+-, (+C) 1011=ABCK 1100=- 1101=-(+1) 1110=C-(+1) (Set S0) 1111=,A-,							00=0 01=1 10=2 11=3		00=— 01=S12 10=S45 11=Z6						00=0 01=1 10=2 11=3		00=block 01=L 10=H 11=St						00=— 01=S2,S3 10=S4,S5 11=S6,S7	

Control-Word Bit Charts (Part 4)

Bit Definition of the Arithmetic Word (Type 10 Fullword Version)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith OP Form		Form		Operation				A Source						Stat Set		B Source						Shift		Next Address						Branch	
								Word				A Input				Word				B Input											
1	0	00=A=A/K 01=Z=A/B 10=A=A/B 11=B=A/B		0000=C + (Rst S0) 0001=C - (+1) (Set S0) 0010=C+-(+C)								00=block 01=16 bits 10=24 bits 11=32 bits		00=--- 01=S12 10=124 11=Z24						00=4 bits 01=8 bits 10=12 bits 11=32 bits		00=--- 01=(SR4,0) 10=(SR4,S0) 11=(SR4,TH)						00=--- 01=S2,S3 10=S4,S5 11=S6,S7	

Bit Definition of the Arithmetic Word (Type 11)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith Op Form		Form		Op	A Gating			A Source						Stat Set		B Source						B Hi-Lo		Next Address						Branch	
								Word				Byte				Word				Byte											
1	1	00=A=A/K 01=L=A/B 10=A=A/B 11=B=A/B		0=,OE, 1=C + (Rst S0)	000=block 001=L 010=H 011=St 100=X block 101=XL 110=XH 111=X							00=0 01=1 10=2 11=3		00=--- 01=S12 10=S45 11=Z6						00=0 01=1 10=2 11=3		00=block 01=L 10=H 11=St						00=-- 01=S2,S3 10=S4,S5 11=S6,S7	

Bit Definition of the Arithmetic Word (Indirect Byte Type 10 or 11)

C0								C1								C2								C3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Arith Op Form		Form		Operation or Op/AXHL				A Source						Stat Set		B Source						A/B Hi-Lo		Next Address						Branch	
								Word								Word															
10 or 11		Refer to bit definition for 10 or 11 word		Refer to bit definition for 10 or 11 word				Indirect byte addressing is specified by the value 1010 1011 1110 1111				00=--- 01=--- 10=+TA 11=-TA		00=--- 01=S12 10=S45 11=Z6		Indirect byte addressing is specified by the value 1010 1011 1110 1111				00=--- 01=--- 10=+TB 11=-TB		00=block 01=L 10=H 11=St						00=-- 01=S2,S3 10=S4,S5 11=S6,S7	

LOCAL-STORAGE MAP (370 MICROPROGRAM IN CONTROL STORAGE)

Word Name	LS Location	Byte 0	Byte 1	Byte 2	Byte 3	X and Y Line
	00		General Register 0			X0 Y0
	01		General Register 1			X0 Y1
	02		General Register 2			X0 Y2
	03		General Register 3			X0 Y3
	04		General Register 4			X0 Y4
	05		General Register 5			X0 Y5
	06		General Register 6			X0 Y6
	07		General Register 7			X0 Y7
	08		General Register 8			X1 Y0
	09		General Register 9			X1 Y1
	0A		General Register A			X1 Y2
	0B		General Register B			X1 Y3
	0C		General Register C			X1 Y4
	0D		General Register D			X1 Y5
	0E		General Register E			X1 Y6
	0F		General Register F			X1 Y7
AX	10		SRTN Temp Link			X2 Y0
DI	11		Alter/Display Log Link			X2 Y1
RTX	12		Retry Link			X2 Y2
DTX	13		Translate Link			X2 Y3
X	14		Working			X2 Y4
R	15		Working			X2 Y5
Y	16		Working			X2 Y6
Q	17		Working			X2 Y7
MA	18					X3 Y0
MBS	19					X3 Y1
MX	1A					X3 Y2
MC	1B					X3 Y3
MD	1C					X3 Y4
MF	1D					X3 Y5
MW	1E					X3 Y6
CX	1F	CPU	Link	Register		X3 Y7

	Word Name	IFA Name	LS Location	Byte 0	Byte 1	Byte 2	Byte 3	X and Y Line
SX 2	GD		20					X4 Y0
	GC		21				Count	X4 Y1
	GM		22		Protect CCW Address			X4 Y2
	GW		23					X4 Y3
SX 3	GD		24					X4 Y4
	GC		25				Count	X4 Y5
	GM		26		Protect CCW Address			X4 Y6
	GW		27					X4 Y7
SX 1	GD	FD	28					X5 Y0
	GC	FC	29				Count	X5 Y1
	GM	FM	2A		Protect CCW Address			X5 Y2
	GW	FW	2B					X5 Y3
SX 4	GD	FA	2C					X5 Y4
	GC	FB	2D				Count	X5 Y5
	GM	FS	2E		Protect CCW Address			X5 Y6
	GW	FL	2F					X5 Y7
			30		Floating-Point Register 0			X6 Y0
			31		Floating-Point Register 0			X6 Y1
			32		Floating-Point Register 2			X6 Y2
			33		Floating-Point Register 2			X6 Y3
			34		Floating-Point Register 4			X6 Y4
			35		Floating-Point Register 4			X6 Y5
			36		Floating-Point Register 6			X6 Y6
			37		Floating-Point Register 6			X6 Y7
	SO		38					X7 Y0
	PM		39		P.E. Control P.E. Code Group Alter Mask			X7 Y1
	DM		3A		Adjustment Factor			X7 Y2
	RW		3B		Address Adjustment Working			X7 Y3
	DP		3C		IFA Low-Priority Link			X7 Y4
	LNK		3D		I-Cycle Link			X7 Y5
	P4X		3E		SX-4 Link Register			X7 Y6
	P3X		3F		SX-1, 2, 3, Link Register			X7 Y7

Note: Words 28 through 2F are shown with Selector Channel designations.

Expanded Local Storage

EXPLS	Word Name	Byte 0	Byte 1	Byte 2	Byte 3	X and Y Line
50	I	Key		I-Register		X2 Y0
51	V			V-Register		X2 Y1
52	W			W-Register		X2 Y2
53	U			U-Register		X2 Y3
54	IBU			IBU-Register		X2 Y4
55	TR			TR-Register		X2 Y5
56	ICS	I-Cycle Control Display				X2 Y6
		57 through 5F unassigned				
60	G2DRL		DATA ADDR (SX 2)			X4 Y0
61	G2DBRL		BACKUP DATA ADDR			X4 Y1
62						X4 Y2
63						X4 Y3
64	G3DRL		DATA ADDR (SX 3)			X4 Y4
65	G3DBRL		BACKUP DATA ADDR			X4 Y5
66						X4 Y6
67						X4 Y7
68	G1DRL		DATA ADDR (SX 1)			X5 Y0
69	G1DBRL		BACKUP DATA ADDR			X5 Y1
6A						X5 Y2
6B						X5 Y3
6C	G4DRL		DATA ADDR (SX 4)			X5 Y4
6D	G4DBRL		BACKUP DATA ADDR			X5 Y5
6E						X5 Y6
6F						X5 Y7
		70 through 77 unassigned				
78	SN					X7 Y0
79	PN					X7 Y1
7A	WK		Working Register			X7 Y2
7B	NP	PAA byte 1,2	Latched Control		Control	X7 Y3
7C	DK			Real Addr Reg		X7 Y4
7D	SS					X7 Y5
7E						X7 Y6
7F						X7 Y7

EXTERNAL ASSIGNMENT CHART

Word Address	Word Name	Byte 0	Byte 1	Byte 2	Byte 3	X Y Line
00	RTY	MB 2	MB 3	ECNT	RCNT	0 0
01	NO REG	NOREGO	NOREG1	NOREG2	NOREG3	0 1
02	DIAG	DIAG0	DIAG1	FEAT2	FEAT3	0 2
03	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX	0 3
04	SPTL *	S-REG	P-REG	T-REG	L-REG	0 4
05	SYS *	SYS0	SYS1	SYS2	H-REG	0 5
06	MCKB *	MCKB0	MCKB1	MCKB2	MCKB3	0 6
07	MCKA	MCKA0	MCKA1	MCKA2	MCKA3	0 7
08	CPU	MODE	CFDAR	LRUM	MATCH	1 0
09	CFDR	CFDR	CFDR	CFDR	CFDR	1 1
0A	ACB	ACB0	ACB1	XXXXXXXX	XXXXXXXX	1 2
0B	SW	SW0	SW1	SW2	SW3	1 3
0C	SPTL *	S-REG	P-REG	T-REG	L-REG	1 4
0D	SYS *	SYS0	SYS1	SYS2	H-REG	1 5
0E	MPX	MTO	MT1	MB1	MBO	1 6
0F	DOC	TI	TA	TT	TE	1 7
10	PSWCTL			MSKA	MSKB	2 0
11	CTCAX	CTCAX0	CTCAX1	CTCAX2	CTCAX3	2 1
12	MISC	EXTINT				2 2
13	CTCAY	CTCAY0	CTCAY1	CTCAY2	CTCAY3	2 3
14	SPTL *	S-REG	P-REG	T-REG	L-REG	2 4
15	SYS *	SYS0	SYS1	SYS2	H-REG	2 5
16	IN	INTA	INTB	SER2	SER3	2 6
17	DC	DCBO	DCHI	TSBO	DCHI	2 7
18	ABRTY	ABRTY0	ABRTY1	ABRTY2	ABRTY3	3 0
19	SPTLB	SRTY	PRTY	TRTY	LRTY	3 1
1A	HMRTY		HRTY	MRTY2	MRTY3	3 2
1B	CPURTY	BYDST	RTYFLG	LSDST	EXTDST	3 3
1C	SPTL *	S-REG	P-REG	T-REG	L-REG	3 4
1D	SYS *	SYS0	SYS1	SYS2	H-REG	3 5
1E	PIR	PIR0	PIR1	PIR2	PIR3	3 6
1F	PIRM	PIRM0	PIRM1	PIRM2	PIRM3	3 7

Word Address	Word Name	Byte 0	Byte 1	Byte 2	Byte 3	X Y Line	
20	GBUF FBAK	GBO FWB	GB1 FCH	GB2 FCL	GB3 FOP	4 0	
21	GBS FCND	GSP FDS	GBF FHC	GCT FED	GBD FMOD	4 1	
22	GSTAT FSTAT	GF FFL	GE FCS	GS FST	GL FGL	4 2	SX1/IFA
23	GTAG FTAG	GTO FTO	GTI FTI	GO FBO	GR FDR	4 3	
24	SPTL *	S-REG	P-REG	T-REG	L-REG	4 4	
25	SYS *	SYS0	SYS1	SYS2	H-REG	4 5	
26	FERR	FSB	FGT	FTS	FAT	4 6	IFA
27	CKCPT	CKCPT 0	CKCPT 1	CKCPT 2	CKCPT 3	4 7	
28	GBUF	GBO	GB1	GB2	GB3	5 0	
29	GBS	GSP	GBF	GCT	GBD	5 1	SX4/IFA
2A	GSTAT	GF	GE	GS	GL	5 2	
2B	GTAG	GTO	GTI	GO	GR	5 3	
2C	SPTL *	S-REG	P-REG	T-REG	L-REG	5 4	
2D	SYS *	SYS0	SYS1	SYS2	H-REG	5 5	
2E						5 6	
2F						5 7	
30	GBUF	GB0	GB1	GB2	GB3	6 0	
31	GBS	GSP	GBF	GCT	GBD	6 1	SX2
32	GSTAT	GF	GE	GS	GL	6 2	
33	GTAG	GTO	GTI	GO	GR	6 3	
34	SPTL *	S-REG	P-REG	T-REG	L-REG	6 4	
35	SYS *	SYS0	SYS1	SYS2	H-REG	6 5	
36	TODH	TODH0	TODH1	TODH2	TODH3	6 6	
37						6 7	
38	GBUF	GB0	GB1	GB2	GB3	7 0	
39	GBS	GSP	GBF	GCT	GBP	7 1	SX3
3A	GSTAT	GF	GE	GS	GL	7 2	
3B	GTAG	GTO	GTI	GO	GR	7 3	
3C	SPTL *	S-REG	P-REG	T-REG	L-REG	7 4	
3D	SYS *	SYS0	SYS1	SYS2	H-REG	7 5	
3E	TODL	TODL0	TODL1	TODL2	TODL3	7 6	
3F						7 7	

may not be used as a destination

*Not flush-through-checked

Both MCKA and MCKB are set to zero when MCKA is used as a destination in a word-move word, with the NOREG as the source.

3145 FEATURE CODE LISTING

ACC	DESCRIPTION
CHC	CHANNEL TO CHANNEL
CHMP	CHANNEL TO CHANNEL OR 3215 MOD 1
DBNM	SEL CHNL 4 BUF OR DIRECT CONTROL AND NO 3215
DCMP	DIRECT CONTROL AND MPK
DCN4	DIRECT CONTROL AND NO SEL CHNL 4
DCT	DIRECT CONTROL
DC4B	DIRECT CONTROL OR SEL CHNL 4 BUFFER
DPK	3210 MODEL 2
DSNM	DCT OR SEL CHNL COMMON, NO MPK
MDPK	3215 MOD 1 AND 3210 MOD 2
MNBD	3215 AND NO DIRECT CTRL AND NO SEL CHNL 4 BUFF
MNDC	3215 AND NO DIRECT CONTROL
MNSD	3215 NO SEL CHNL 4, NO DIRECT CONTROL
MPK	3215 MODEL 1
NBPF	SEL CHNL 1 NO BUFFER AND NO IFA
NCHC	NO CHANNEL TO CHANNEL
NDCM	NO DIRECT CONTROL AND NO MPK
NDCT	NO DIRECT CONTROL
NMPK	NO 3215 MOD 1
NSB2	SELECTOR CHANNEL 1 BUFFER NO CHANNEL 2
NSB3	SELECTOR CHANNEL 2 BUFFER NO CHANNEL 3
NSB4	SELECTOR CHANNEL 3 BUFFER NO CHANNEL 4
NS43	SEL CHNL 3 NO CHNL 4
NSC2	SEL CHNL 1 NO CHNL 2
NSC3	SEL CHNL 2 NO CHNL 3
NSDM	NO SX4 AND NO DCT AND NO MPK AND NO CHNL
NSX2	NO SELECTOR CHANNEL 2
NSX3	NO SELECTOR CHANNEL 3
NSX4	NO SELECTOR CHANNEL 4
NS09	N/128K MAIN-STORAGE FRAME
N4DC	NO SX4 AND NO DIRECT CONTROL
SBC	WORD BUFFER COMMON
SBN2	SEL CHNL 1 BUF W/O SEL CHNL 2
SBPF	SEL CHNL 1 BUFFER OR IFA
SB2F	SELECTOR CHANNEL 2 AND INTEGRATED FILE ADAPTER
SB12	SEL CHNL BUFFERED 1 AND 2
SB14	SEL CHNL 1 AND 2 AND 3 AND 4 BUF
SB1	SELECTOR CHANNEL 1 BUFFER
SB2	SELECTOR CHANNEL 2 BUFFER
SB3	SELECTOR CHANNEL 3 BUFFER
SB4	SELECTOR CHANNEL 4 BUFFER
SC1	SELECTOR CHANNEL 1
SC2	SELECTOR CHANNEL 2
SC3	SELECTOR CHANNEL 3
SC4	SELECTOR CHANNEL 4
SC12	SEL CHNL 1 AND 2
SCPF	SELECTOR CHANNEL 1 OR INTEGRATED FILE ADAPTER
SC2F	INTEGRATED FILE ADAPTER SELECTOR CHANNEL 2 NO BUFFER
SDPK	3210 MOD 1 OR MOD 2

ACC	DESCRIPTION
SFIU	SEL CHANL 1 BUF AND NO SEL 2, OR SEL CHANL 1 UNBUF, OR IFA WITH SEL CHNL 2 UNBUF
SF2B	SEL CHNL 1 BUF OR IFA WITH SEL CHNL 2 BUF W/O SEL CHNL 3
SF3B	SEL CHNL 1 BUF OR IFA WITH SEL CHNL 2 AND 3 BUF W/O SEL CHNL 4
SF2U	SEL CHNL 1 UNB OR IFA WITH SEL CHNL 2 UNB W/O SEL CHNL 3
SF3U	SEL CHNL 1 UNB OR IFA WITH SEL CHNL 2 AND 3 UNB W/O SEL CHNL 4
SPK	3210 MODEL 1
SPF	INTEGRATED FILE ADAPTER
SUN2	SEL CHNL 1 UNB W/O SEL CHNL 2
SU14	SEL CHNL 1 AND 2 AND 3 AND 4 UNB
S1NB	SELECTOR CHANNEL 1 NO BUFFER
S2NB	SELECTOR CHANNEL 2 NO BUFFER
S3NB	SELECTOR CHANNEL 3 NO BUFFER
S4NB	SELECTOR CHANNEL 4 NO BUFFER
S4DC	SELECTOR CHANNEL 4 OR DIRECT CONTROL
S4DM	SX4 OR DIRECT CONTROL AND MPK
S04	MAIN STORAGE 112K
S06	MAIN STORAGE 160K
S75	MAIN STORAGE 208K
S08	MAIN STORAGE 256K
S09	MAIN STORAGE 384K
S10	MAIN STORAGE 512K
IBN2	SELECTOR CHANNEL 1 BUFFER NO CHANNEL 2
2BN3	SELECTOR CHANNEL 2 BUFFER NO CHANNEL 3
3BN4	SELECTOR CHANNEL 3 BUFFER NO CHANNEL 4
ICN2	SEL CHNL 1 NO CHNL 2
2CN3	SEL CHNL 2 NO CHNL 3
3CN4	SEL CHNL 3 NO CHNL 4
4BDM	3215 AND DIRECT CTRL OR SEL CHNL 4 BUFF

3145 ALD VERSION CODES	
CODE	FEATURE
000	BASIC
001	MATRIX PRINTER (3215)
003	IFA
004	SELECTOR CHANNEL (WORD BUFFER)
005	SELECTOR CHANNEL (NO WORD BUFFER)

LIST OF DEVELOPMENT TERMS THAT MAY APPEAR IN THE LOGICS

Previous		Present	
SPF	SYSTEM PRIME FILE	IFA	INTEGRATED FILE ADAPTER
LDF	LOAD DIAG FILE	CF	CONSOLE FILE

LOGIC REFERENCE

A BYTE ASM CTRLS	BA011
A BYTE ASM B ENTRY	BA015
A BYTE ASM BYTE 0 BITS P-7	BA021-BA022
A BYTE ASM BYTE 1 BITS P-7	BA023-BA024
A BYTE ASM BYTE 2 BITS P-7	BA025-BA026
A BYTE ASM BYTE 3 BITS P-7	BA027-BA028
A BYTE CTRLS S/R	BA013
ACB M REG COMP	MC013
ACB REG BYTE 0	MC015-MC017
ACB REG BYTE 1	MC016
ADR. ADJ. CONTROLS	MT011-MT015
ADR ADJ. TO L.S.	MT036-MT037
ADR ADJ MATCH	MT311-MT345
ADR. ADJ REGS 0-8	MT111-MT134
ALU FUNCTION CTRLS	AC011
ALU GATE BA2 & BA3 HI & LO TO ALU 2 & 3	BK011
ALU BUS BYTE 3 BITS 0-7	AL113-AL123
ALU BUS BYTE 2 BIT 0-7	AL133-AL143
ALU HI INPUT ASM CTRLS	BB111
ALU LO INPUT ASM CTRLS	BB121
ALU A SW CTRLS & ALU S/R	AC014
ALU A SW GATE CTRLS	AC012
ALU A INPUT PARITY GATES	AM011-AM012
ALU 2 A INPUT BITS 4-7	AL131-AL132
ALU 2 A INPUT BITS 0-3	AL141-AL142
ALU 3 A INPUT BITS 4-7	AL111-AL112
ALU 3 A INPUT BITS 0-3	AL121-AL122
ALU CARRY-IN LATCHES	AM014
ALU CARRY & COMPL CTRLS	AC013
ALU 16 BIT CARRY LOOKAHEAD	AH015
ALU PARITY PREDICT & 4 BIT HS CHECK	AL117-AL127
ALU PARITY PREDICT & 4 BIT HS CHECK	AL137-AL147
ALU LOGICAL CHECK	AD011
ALU HS ERROR LATCHES	AM013
ALU DECIMAL CTRLS	AD012
ALU 2 4 BIT HS TRANSMITS & CARRIES	AL135-AL145
ALU 3 4 BIT HS TRANSMITS & CARRIES	AL115-AL125
ALU3 4 BIT GROUP CARRY COLLECTION FS	AL116-AL126
ALU 2 4 BIT GROUP CARRY COLLECTION FS	AL136-AL146
A-REG BYTE 0 BITS P-4	RA111-RA112
A-REG BYTE 0 BITS 5-7	RA121
A-REG BYTE 1 BITS P-4	RA122-RA131
A-REG BYTE 1 BITS 5-7	RA132
A-REG BYTE 2 BITS P-4	RA141-RA162
A-REG BYTE 2 BITS 5-7	RA151
A-REG BYTE 3 BITS P-4	RA152-RA161
A-REG BYTE 3 BITS 5-7	RA162
B BYTE ASM CTRLS	BA012
B BYTE CTRLS S/R	BA013
B BYTE ASM 2B INPUT ASM BITS P-7	BB112-BB123
B BYTE ASM 3B INPUT ASM BITS P-7	BB113-BB122
BACK UP ASM BYTE 1 BITS P-7	RR126-RR128
BACK UP ASM BYTE 2 BITS P-7	RR136-RR138
BACK UP ASM BYTE 3 BITS P-7	RR146-RR148
BACK UP REG BYTE 0 BITS P-4	RR116-RR117
BACK UP REG BYTE 1 BITS P-7	RR121-RR123
BACK UP REG BYTE 2 BITS P-7	RR131-RR133
BACK UP REG BYTE 3 BITS P-7	RR141-RR143
BASIC ASM BYTE 0 BITS P-7	GB611-GB612
BASIC ASM BYTE 1 BITS P-7	GB621-GB622
BASIC ASM BYTE 2 BITS P-7	GB631-GB632
BASIC ASM BYTE 3 BITS P-7	GB641-GB642
B-REG BYTE 0 BITS P-4	RA113-RA114
B-REG BYTE 0 BITS 5-7	RA123
B-REG BYTE 1 BITS P-4	RA124-RA133
B-REG BYTE 1 BITS 5-7	RA134
B-REG BYTE 2 BITS P-4	RA143-RA144
B-REG BYTE 2 BITS 5-7	RA153
B-REG BYTE 3 BITS P-4	RA154-RA163
B-REG BYTE 3 BITS 6-7	RA164
B SOURCE BRANCH HI DECODE A	RM211-RM213
B SOURCE BRANCH LO DECODE A	RM212-RM213
B SOURCE BRANCH HI DECODE B	RM221
B SOURCE BRANCH LO DECODE B	RM222
BL1 & BL2 DEC	MT034-MT035
BLOCK 1 COMP TO BL SIZE	MT033
BLOCK 1 & BLOCK 2 WORK REG	MT021-MT027
BRANCH HI-LO GATING 8	RM223
BRANCHING CTRLS	RM042
CF CLOCK CTRLS LIGHTS IND	KF042-KF054
CF COMMAND REG CTRLS & DECODE	KF022-KF026
CF COMMAND REG	KF034
CF DATA REG BYTE 0	KF014
CF DATA REG BYTE 1	KF015
CF DATA REG BYTE 2	DK016
CF DATA REG BYTE 3	KF017
CF DA COMPARE	KF032
CF DATA CHECK	KF011
CF DISPLAY CHECKS	KF035-KF041
CF DISK ADDR REG	KF031
CF INTERFACE TO 23FD	SS011
CF READY & HEAD CTRLS	KF021
CF TRACK INC DEC CTRLS	KF033
CF SHIFT REG	KF012-KF013
CLOCK START CTRLS OSCILLATOR & DRIVE	KC021-KC253
CLOCK SYNC CHECK	RE045
CLOCK SYNC GATING	RD023
CONSOLE ADDR COMPARE	PA341-PA351
CONSOLE CPU LAMPS & DRIVERS	PL141
CONSOLE CPU SYS CHECK	PL142
CONSOLE LDF CHECKS LAMPS & DRIVERS	PL161-PL162
CONSOLE MATCH CIRCUITS	PM011
CONSOLE PUSH BUTTON & ROTARY	PA251-PA331
CONSOLE ROLLER SW A REG DISPLAY	PL101-PL132
CONSOLE SW DATA ENTRY A & B	PA011-PA211
CONSOLE SW DATA ENTRY C & D	PA021-PA221
CONSOLE SW DATA ENTRY E & F	PA031-PA231
CONSOLE SW DATA ENTRY G & H	PA041-PA241
CONSOLE SWITCHES MISC	PA051-PA101
C-REG ASSEMBLY	RC112-RC122
C-REG ASSEMBLER	RC162-RC172
C-REG ASSEMBLER	RC182-RC192
C-REG ASSEMBLER	RC142-RC152
C-REG ASSEMBLER	RC132-RC134
C-REG CTRLS & SDBO	RC091-RC093
C-REG BYTES 0-3 BIT P	RC111
C-REG BYTES 0-3 BIT O	RC121
C-REG BYTES 0-3 BIT 1	RC131
C-REG BYTES 0-3 BIT 2	RC141
C-REG BYTES 0-3 BIT 3	RC151
C-REG BYTES 0-3 BIT 4	RC161
C-REG BYTES 0-3 BIT 5	RC171
C-REG BYTES 0-3 BIT 6	RC181
C-REG BYTES 0-3 BIT 7	RC191
C-REG BYTE 0 DECODE 1-2	DC011-DC012
C-REG BYTE 0 DECODE 3-4	DC013-DC014
C-REG BYTE 1 DECODE 0-7	DC021-DC022
C-REG BYTE 2 DECODE P-7	DC023
C-REG BYTE 2 BITS 4-7 BFR	BK012
DESTINATION BYTE CTRLS	KD011-KD015
DISPLAY ASM BYTE 0	PB011-PB015
DISPLAY ASM BYTE 1	PB021-PB025
DISPLAY ASM BYTE 2	PB031-PB035
DISPLAY ASM BYTE 3	PB041-PB045
D-REG BYTE 0 BITS P-4	RA115-RA116
D-REG BYTE 0 BITS 5-6	RA125
D-REG BYTE 1 BITS 0-5	RA126-RA135
D-REG BYTE 1 BITS 6-7	RA136
D-REG BYTE 2 BITS P-4	RA145-RA146
D-REG BYTE 2 BITS 5-7	RA155
D-REG BYTE 3 BITS 0-5	RA156-RA165
D-REG BYTE 3 BITS 6-7	RA166
DIRECT CTRL INTERFACE	WR071
DIRECT CTRL INTERFACE BUS TAG	JA021-JA131
DIAG REG BYTE 0,1	RD021,RD024
DIAG CTRLS	RD022
DIAG MANUAL CTRL	KD111-KD112
DOC CTRL INTERFACE	WP011-WP012
DOC CONSOLE RD WR CYCLE CTRL	PD012-PD013
DOC CONSOLE WD DATA REG	PD015
DOC CONSOLE KEYBOARD INTERFACE	PD021-PD022
DOC CONSOLE MAGNET DRIVER	PD031-PD121
DOC CONS PRT INTERFACE	WP011-WP022
DOC CONSOLE PRINTER SW & CTRLS	PA361-PA381
ELIMINATE REG.	MT014
EPSW A B REG	RM812
ERROR COUNTER	RE361-RE362
EXT ASM GATES	BE111-BE121
EXT ASM GATES	BE151-BE161
EXT ASM BYTE 0 BITS P-2	BE112-BE113
EXT ASM BYTE 0 BITS 3-7	BE114-BE122
EXT ASM BYTE 0 BIT 7	BE123
EXT ASM BYTE 1 BITS P & 0	BE132
EXT ASM BYTE 1 BITS 3 & 4	BE124
EXT ASM BYTE 1 BITS 1 & 5	BE133
EXT ASM BYTE 1 BITS 6 & 7	BE134
EXT ASM BYTE 2 BITS P-2	BE142-BE143
EXT ASM BYTE 2 BITS 3-6	BE146-BE152
EXT ASM BYTE 2 BIT 7	BE153
EXT ASM BYTE 3 BITS 2-4	BE153-BE154
EXT ASM BYTE 3 BITS P & 0	BE162
EXT ASM BYTE 3 BITS 1 & 5	BE163
EXT ASM BYTE 3 BITS 6 & 7	BE164
EXT BUS IN DRIVERS BYTE 0	RC211-RC212
EXT BUS IN DRIVERS BYTE 1	RC221-RC222
EXT BUS IN DRIVERS BYTE 2	RC231-RC232
EXT BUS IN DRIVERS BYTE 3	RC241-RC242
EXT CTRL ASM MISC	DE001-DE002
EXT CTRL ASM MISC	DE003-DE004
EXT CTRL ASM MISC	DE005-DE006
EXT CTRL ASM SOURCE DECODE	DE011-DE012
EXT CTRL ASM SOURCE DECODE	DE013-DE014
EXT CTRL ASM SOURCE DECODE	DE015-DE016
EXT INTERRUPT-RFG P-0	RM813
EXT DECODE BFR	RR119-RR129
EXT DECODE BFR	RR139-RR149
EXT DESTINATION DECODE	DE021-DE023
EXT DESTINATION ADDRESS LT	DE022-DE024
EXT DEST X DECODE	DE026
EXT DEST Y DECODE	DE025-DE027
EXT FEATURE SOURCE DECODE	DF011-DF016
EXT INTERRUPT REG	JA011-JA012
EXP EXT ASM BYTE 0 BITS P-2	BF112-BF113
EXP EXT ASM BYTE 0 BITS 3-6	BF114-BF122
EXP EXT ASM BYTE 0 BIT 7	BF123
EXP EXT ASM BYTE 1 BITS P & 0	BF132
EXP EXT ASM BYTE 1 BITS 2-4	BF123-BF124
EXP EXT ASM BYTE 1 BITS 1 & 5	BF133
EXP EXT ASM BYTE 1 BITS 6 & 7	BF134
EXP EXT ASM BYTE 2 BITS P-2	BF142-BF143
EXP EXT ASM BYTE 2 BITS 3-6	BF144-BF152
EXP EXT ASM BYTE 2 BIT 7	BF153
EXP EXT ASM BYTE 3 BITS P & 0	BF162
EXP EXT ASM BYTE 3 BITS 2-4	BF153-BF154
EXP EXT ASM BYTE 3 BITS 1 & 5	BF163
EXP EXT ASM BYTE 3 BITS 6 & 7	BF164
FEATURE BYTES 1 & 2	RD051-RD072
FTC ERROR BITS P-1	RA125-RA156
FTC ERROR BITS 6-7	RA166
FLUSH THRU CHECK ERROR	RE025
FTC ERROR 1 S/R	RA116
FTC ERROR 2	RA126-RA127
FTC ERROR 3	RA136-RA137
FTC ERROR 4	RA146-RA147
FTC ERROR 5	RA157
FTC ERROR 6	RA167
H-REG BITS 0-7	RH022-RH023
H-REG BACK-UP REG	RR125
H-REG PARITY CHECK	RH021
I BACK-UP	RV111-RV315
I BUFFERS	RU111-RU128
I BUFF CNTLS	RU041-RU045
I CYC. CONTROL GEN.	RU031-RU036
PRE-ADDR ASSY	RU051-RU052
I-CYCLE CONTROLS	RU053
I-CYCLE ERROR LATCHES	RU054
I REG	RV111-RV315
K ASM BITS 0-7 LATCHES	BK015
K ASM FORCE BITS P,5-7	BK014
KEY REG	RV111-RV315
L-REG BITS P-7	RL011
L-REG BACK-UP REG	RR144
LOGICAL ADR, DISP. ASSY	MT042-MT093
LRU	MT211-MT216
LS CONTROLS	MB111-RC092
LS GATES BYTES 0-3	LC011
LS A DEST ADDR LT X	LA212
LS A DEST ADDR LT Y	LA222
LA A B FAST X ADDRESS	LA011-LA018
LA A FAST Y ADDRESS	LA111-LA117
LS A SLOW X ADDR ASM	LA211
LS A SLOW Y ADDR ASM	LA221
LS B DEST ADDR LT X	LA232
LS B DEST ADDR LT Y	LA242
LS B FAST Y ADDRESS	LA121-LA127
LS B SLOW X ADDR ASM	LA231
LS B SLOW Y ADDR ASM	LA241
LS MISC X ADDRESS CTRLS	LA021-LA024

LS MISC Y ADDRESS CTRLS	LA031-LA034
LS 64X18 MONO BUFFER	LA311-LA347
LS SEL CHAN ADDR FORCE	BB012
LA A B COMPARE ERROR 1 & 2	RA117-RA127
LS A B COMPARE ERROR 3 & 4	RA137-RA147
LS A B COMPARE ERROR 5 & 6	RA157-RA167
MANUAL STORE DISPLAY	KM011-KM031
MASK A-REG 4 INTRO REG	RJ011
MASK B-REG 4 INTRA REG	RJ012
MCK-REG CTRLS	RE011-RE015
MCK-REG A BYTE 0 BITS P-7	RE021-RE022
MCK-REG A BYTE 1 BITS P-7	RE022-RE024
MCK-REG A BYTE 2 BITS P-7	RE031-RE033
MCK-REG A BYTE 3 BITS P-7	RE034-RE036
MCK COUNTER	RE053
MCK-REG B BYTE 0 BITS P-7	RE041,RE045
MCK-REG B BYTE 1 BITS P-7	RE045,RE046
MCK-REG B BYTE 2 BITS P-7	RE051,RE053
MCK-REG B BYTE 3	RE052
MEMORY CTRL	MS011-MS015
M-REG DATA GATES A & B	RM111-RM114
M-REG DATA GATES A & B	RM114-RM124
M-REG SET-RESET CTRLS	RM112-RM113
M-REG SLT-RESET CTRLS	RM122
M-REG LATE SET-RESET CTRLS	RM123
M-REG DUP CHECK ASM	RM061-RM064
M-REG DUP CHECK	RD024
MB-REG BYTE 2 BITS P-7	RM035
MB-REG BYTE 3 BITS P-3	RM073
M1-REG	MC011-MC022
M2 REG BYTE 1 CTRL	MC014
M2-REG BACK-UP	RR135
M3-REG PRE ASM P-7	RM021
M2-REG BYTE 2 BITS P-3	RM031-RM033
M3-REG BYTE 2 BITS 4-7	RM032
M3-REG BITS P,0,2	RM041-RM043
M3-REG BITS 1,3	RM044
M3-REG BITS 6-7	RM052
MR-BITS 4-5 ASM A	RM214-RM215
M3-BITS 4-5 ASM B	RM224-RM225
M3-REG PARITY GEN & CHECK	RM065
M3-REG BACK-UP	RR145
MPX INTERFACE	WA011
MPX CHANNEL TAGS & REG & DECODE	FA011-FA151
NO REG BLOCK TO A REG ONLY	
N2-REG BYTE 2 BITS P-2	RM033
N2-REG BYTE 2 BITS 3-7	RM034
N3-REG BITS P-7	RM045-RM052
NSP REG	MT031-MT032
P REG	RP011-RP012
P-REG BACK-UP REG	RP124
P-REG PARITY PREDICTION	RP015
PLUG ON TERM	ZA012-ZE234
PRIORITY DECODE	RH013
RETRY BFR	RR021
RETRY COUNTER	RH031
RETRY FLAGS	RR012
RETRY HM BYTE 0	RR115
RETRY REG CTRL	RR011-RR012
RETRY CODE CTRLS	GB711-GB713
RETRY-DIAG-ACB ASM BYTE 0 BITS P-3	BE211-BE212
RETRY-DIAG-ACB ASM BYTE 1 BITS P-7	BE221-BE222
RETRY-DIAG-ACB ASM BYTE 2 BITS P-7	BE231-BE232
RETRY-DIAG-ACB ASM BYTE 3 BITS P-7	BE241-BE242
RETRY BACK-UP REG BYTE 0 BITS P-7	RR111-RR113
S-BRANCH HI-LO ASM A	RM216
S-BRANCH HI-LO ASM B	RM226
S-REG CHECK	RS126
S REG A	RS115,RS214
S REG B	RS125,RS221
S-REG BITS 0-3	RS115-RS125
S-REG BITS 2,4,5 STAT SET COND	RS211-RS221
S-REG BITS 4-7	RS241-RS224
S-REG SET-RESET CTRL & CHECK	RS112-RS114
S-REG SET-RESET CTRL	RS122-RS124
S-REG SET-RESET CTRL	RS212-RS222
S-REG OPERATION DECODE	RS111-RS121
S-REG BACK-UP REG	RR114
SDBI DRIVER BYTE 0 EXTERNAL	MB151
SDBI DRIVER BYTE 1 EXTERNAL	MB161
SDBI DRIVER BYTE 2 EXTERNAL	MB171
SDBI DRIVER BYTE 3 EXTERNAL	MB181
SDBI DRIVER BYTE 0 INTERNAL	MB111
SDBI DRIVER BYTE 1 INTERNAL	MB121
SDBI DRIVER BYTE 2 INTERNAL	MB131
SDBI DRIVER BYTE 3 INTERNAL	MB141
SDBO PRE ASSEMBLER	RC113-RC123
SDBO PRE ASSEMBLER	RC133-RC143
SDBO PRE ASSEMBLER	RC153-RC163
SDBO PRE ASSEMBLER	RC173-RC183
SDBO PRE ASSEMBLER	RC193
SDBO ASSEMBLER	RC114-RC124
SDBO ASSEMBLER	RC134-RC144
SDBO ASSEMBLER	RC154-RC164
SDBO ASSEMBLER	RC174-RC184
SDBO ASSEMBLER	RC183-RC184
SDBO INTERCEPTOR	RV011-RV015
SP PARITY & PAA CNTL	RV021-RV024
STORAGE WORD DECODE	DC031-DC032
STP STACK	MS115-MS621
STP X, Y DRIVE & TIMING	MS111-MS114
SYS REG BYTE 0	RS011
SYS REG BYTE 1	RS012
SYS REG BYTE 2	RS013
TA REG DOC CONSOLE	PD011
TT REG DOC CONSOLE	PD014
TIMER	CH113
TIMER	CH111-CH112
TIMER OSC DRIVE	CT111
TIMER DECODE	PM014
TIMER CARD 2 CTRLS	CH021-CH022
TIME OF DAY ADVANCE CTRLS SET CTRLS	CT111-CT112
TIME OF DAY CLOCK OSCILLATOR & DELAY	CT011-CT115
TIME OF DAY CLOCK DETECT & BITS ASM	CT113-CT114
TIME OF DAY HI TO LO BYTE 0 BITS 0-5	CT211-CT212
TIME OF DAY HI TO LO BYTE 0 BITS 6-7	CT213
TIME OF DAY HI TO LO BYTE 1 BITS 0-1	CT213-CT214
TOD SPF ASM GATING BYTE 0 BITS 0-7	CT214
TOD PARITY PREDICT BYTE 0	CT214
TOD PARITY PREDICT BYTE 1	CT225
TOS HI LO BYTE 1 BITS 1-5	CT221-CT222
TOD HI LO BYTE 2 BITS 0-1	CT221-CT223
TOD SPF ASM GATING BYTE 1 BITS 1-7	CT224
TOD SPF ASM GATING BYTE 2 BITS 1-2	CT224
TOD CTRS BYTE 2 BITS 4-7	CT311-CT312
TOD CTRS BYTE 3 BITS 1-4	CT312-CT313
TOD CTRS BYTE 3 BITS 5-7	CT314
TOD ASM BYTE 2,3 PARITY GEN & GATES	CT315-CT316
TOD BYTE 3 BITS 3-7	CT317
T-REG INPUT	RT011
T-REG BITS 0-3	TR012
T-REG BITS 4-7	RT013
T-REG ASM & 2 REG ENTRY	RT014
T-REG PARITY PREDICTION	RP013-RP015
T-REG BACK-UP REG	RR134
TRAP ADDRESS LT	RH014
TRAP ADDRESS BYTE 3 BITS 3-4	RH017
TRAP ADDRESS BYTE 3 BITS 5-P	RH018
TRAP CTRL SIGNAL DRIVERS	RH016
TRAP CYCLE LT	RH021
TRAP REQ LT	RH012-RH015
U2 REG, OP & OPDEC.	RU011-RU016
U3 REG, IMM. BYTE	RU021-RU026
V REG	RV111-RV315
W REG	RV111-RV315
Z REG BYTE 0 BITS 0-7	AL134-AL144
Z REG BYTE 0 & 2 BIT P	AL137-AL147
Z REG BYTE 1 BITS 0-7	AL114-AL124
Z REG BYTE 1 & 3 BIT P	AL117-AL127
Z REG BYTE 2 BITS 0-7	AL134-AL144
Z REG BYTE 3 BITS 0-7	AL114-AL124
Z REG GATE CTRLS	AC015
Z-REG TO A REG CTRLS	RA012
Z-REG RESET	RA013
Z-REG COMBINATIONS EQUAL 0	RS213-RS221

STORAGE (SQ001-SQ020)

ADR. & INST. TO STOR	SQ600-SQ601
DATA TO STOR	SQ700-SQ712
DELAY LINE	SQ311-SQ315
ERROR DEC	SQ290-SQ292
INST. EXIT TO SYS.	SQ800-SQ805
PARITY OUT GEN	SQ301-SQ304
READ GEN	SQ261-SQ262
SAR	SQ401-SQ404
SDBI SDBO INT-EXT	SQ100-SQ805
SDBO INT. & EXT.	SQ900-SQ915
SDR	SQ202-SQ248
SDBO BYTES 0-7 EXT-INT	SQ900-SQ915
STOR. CLOCK	SQ407-SQ410
SYNDRONE GEN	SQ271-SQ287
WRITE GEN	SQ251-SQ252

SELECTOR CHANNEL

SEL CHNL INTERFACE	GF414-GZ555
SEL CHNL INTERFACE RECEIVERS & DRIVERS	GB911-GB951
SEL CHANNEL GATES, SHARE CTRLS, TRAPS	GA111-GA511
SEL CHANNEL DECODES, GATES, TAGS, CTRLS	GB051-GB514
SEL CHNL SOURCE DECODE	GD011-GD017
SEL CHNL WORD BUFFER & CTRLS	GC311-GC513
SEL CHNL 1 INTERFACE	WA021
SEL CHNL 2 INTERFACE	WA031
SEL CHNL 3 INTERFACE	WA041
SEL CHNL 4 INTERFACE	WA051
SEL CHNL 1 ASSEMBLER	GC611-GC645

POWER

SEQ PANEL LITE DRV.	YD311-YD351
1ST-SEQ LOGIC	YD411-YD431
2ND-SEQ LOGIC	YD441
RELAY DEV.	YD511
REG. INPUT RESISTOR CARDS	YD111-YD121
REED RELAYS	YD811
SEQ-PANEL INDICATORS	YD611-YD612
THERMALS, SEQ & DELAY	YD711-YD721
UNDER VOLT. DET.	YD211-YD251
VOLTAGE NETS BOARDS	ZZ011-ZZ121
VOLTAGE NETS	ZZ011-ZZ310

IFA

ADDR. MARK	JL413-JL414
ADR ADJ. CONTROL	JV011
BCA REG.	JL213
BIT RING	JL312
BYTE COUNTER & DECODES	JL022-JL116
CC HWD ERR. LAT.	JL611
CC REG. FORCE	JL214
CC DEC & CNTL	JL216
CE TEST SW	JL021-JL022
CE CONTROLS	JL514
CLOCK PHASE GEN.	JL311
COMPARE GATE & DATA	JL215
CONTROL TAG	JL512
CYC CODE REG	JL211-JL212
DATA & CMD OVERRUN	JL513
DATA CK	JL811
DIAG. ASSEMBLER	JL711-JL715
FCH REG	JK211
FCL & FBO REG	JK311
FCS REG	JK212
FDR & FOP REG	JK411, JK412
FED REG	JK312
FHC REG	JK216-JK217
FLAG REG-FFL	JK111
FMOD REG	JK611
FSB & FGT PAR. GEN.	JL901
FST REG	JK313
GATED ATTENTION	JL811-JL814
HD COND.	JL411
IFA ASSY WD 20-23 BYTE 0	JK113-JK115
IFA ASSY WD 20-23 BYTE 1	JK213-JK215
IFA ASSY WD 20-23 BYTE 2	JK314-JK315
IFA ASSY WD 20-23 BYTE 3	JK414-JK416
IFA INTERFACE	WF101-WF111
INCORRECT LENGTH LAT	JK513
INDEX	JL411
MISSING AM LAT.	JL014
MOD SEL BITS	JL511
MOP REG	JL314
ORIENTATION LAT.	JL014
RD, WR ERASE GATES	JL412
RESETS	JL613
IFA RETRY CODE REG	JK612
SERDES READ BUFFER	JL011-JL015
SHARE CYC. CNTL	JK511-JK512
SHARE CYC ERROR LAT.	JK513
SLT & MST CONVERTS	JL415-JL416
	JL515-JL518
	JL813
TAG REG - FTO	JK112
TRAP CONTROLS & REQUESTS	JL611-JL612
WR. DATA & MOP ERROR	JL313
WRITE CONTROL	JL117
WRONG LENGTH RECORD	JL512

2319 MIXER BD

CE PANEL	FE110-FE120
DRIVE SELECT	FE070
INTERFACE ENTRY FROM DRIVE	FE010
INTERFACE EXIT TO DRIVE	FE020
LEVEL CONVERT	FE070-FE100
RD-WR COAX SW.	FE030
VFO	FE040-FE060
CHAN TO CHAN ADAPTER FEATURE	XX001-XX931
DIRECT CONTROL FEATURE	JA011-JA151

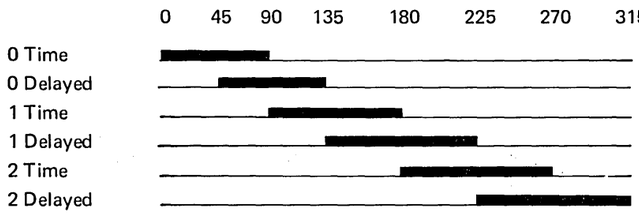
* NOTE: SQ Logics in Area & FTSC Only

MST LOGIC INFORMATION

CLOCK CARD LOCATIONS	
A GATE	**B GATE**
A1 - K2	A1 - C4
A2 - C4	
A3 - C4	B3 - V3
A4 - Q4	
	C3 - J2
B1 - C4	
B2 - M2	
B3 - H4	
B4 - K2	
C1 - G2	
C2 - J2	
C3 - G4	
C4 - E2	

PIN LOCATIONS OF TIMING PULSES ON EACH CLOCK CARD	
+0 Time—	G05
-0 Time—	J07
+0 Delayed—	D11
-0 Delayed—	G07
+1 Time—	B13
-1 Time—	J09
+1 Delayed—	J02
-1 Delayed—	J13
+2 Time—	J12
-2 Time—	G13
+2 Delayed—	D04
-2 Delayed—	G12
OSC Sync Point	G10

CLOCK TIMING CHART



CYCLE TIME FOR CONTROL WORDS

202.5 ns
247.5 ns
292.5 ns
315.0 ns

Note: Cycle times in the logic pages appear as 180, 225, 270 cycles— A delay of 22.5 nanoseconds is added to each of these cycles to obtain the cycle times of 202.5 to 292.5. Two delays are added to obtain the 315-cycle time. Refer to the control-word section for details on cycle times for each word type.

SPECIAL NOTE

- Signal lines cannot be tied down to -3V or ground
- Signal lines should not be tied up to +1.2V directly
- Signal lines can be tied up to +1.2V through a resistor network located on each board at the following locations.

A GATE

A1	-	K2 B07,B08	B1	-	C4 B07,B08	C1	-	G2 B07,B08
A2	-	C4 B07,B08	B2	-	M2 B07,B08	C2	-	J2 B07,B08
A3	-	C4 B07,B08	B3	-	H4 B07,B08	C3	-	G4 B07,B08
A4	-	Q4 B07,B08	B4	-	K2 B07,B08	C4	-	E2 B07,B08

B GATE

A1	-	C4 B07,B08
B3	-	V3 B07,B08
C3	-	J2 B07,B08

PULLING CARDS WITH POWER ON

Most cards in the A & B gates can be pulled out and extended or swapped without dropping power. Exceptions to this are:

1. Local-storage cards 01A - B4 — M2,P2;01A—C4—B2,C2
2. Storage-protect cards — 01A—A1—H4,J4,K4,L4
3. Phase 2I Memory Cards
4. ECC Board cards 01B-A3
5. Memory select card 01A-C1F2

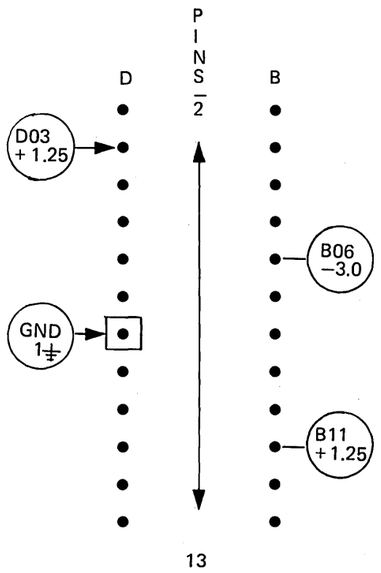
Note:

1. It may be necessary to IMPL after a card is extended or swapped.
2. Board covers should not be left open for an extended period of time.
   (Possible false errors or thermal checks)

4-Wide Card Socket

D	B
J	G
P	M
U	S

Wiring Side

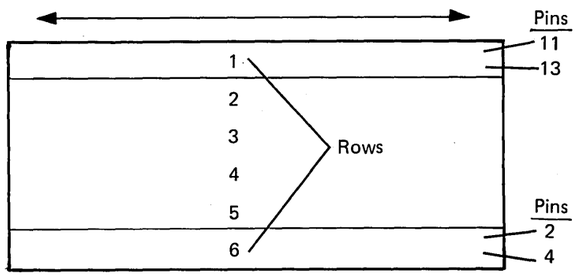


Voltage Locations On Phase 2 I STG Array Boards

Voltages are applied to EACH card. Each card occupies two connector positions

+7	G09	-3	B06
+2	B04	GND	D08/B13
+1.25	D03		

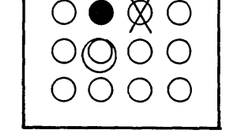
Upper and Lower Row Tri-Lead Locations



MST Board (Wiring Side)

Upper row (1) are pins 11 and 13
Lower row (6) are pins 2 and 4

MST — 2 Module -- Pin Side View

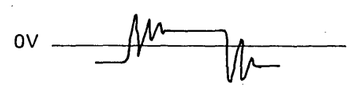


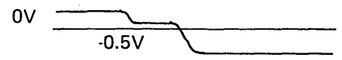
-3 Volts
GROUND
+1.2 Volts

VOLTAGE LEVELS—SCOPING INFORMATION

1. MST voltage swing is approximately +0.4V to -0.4V. Depending on the load this signal will vary slightly..
2. Scope probes must always be grounded.
3. Lamp driver +2.0 is the up level—+0.3V is the down level.
4. Interface lines +3.0 is the up level—0.0V is the down level.
5. All lines longer than 12 inches must be terminated (a 90 resistor is used). These resistors may be on an MST card or may be terminated with a POT (plug on terminator). In the FEALD's a terminator on the MST card is shown by an asterisk on an input line to a logic page and a note at the bottom of the page.

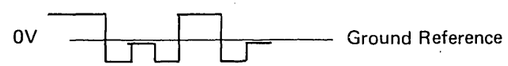
Example: -1 time buffered — RT011 BB6* —
A POT is another terminator that plugs into the tri-lead cable. POTS are shown in the ZA and ZB logic pages. Missing POT can be detected by an excessive amount of oscillation on a signal level.





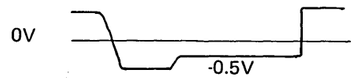
Note: Missing POTS cause the machine to be very sensitive to noise.

6. Bad levels may appear to come from the external assemblers (BE & BF logic) because of spare inputs. Spare inputs are left floating. This causes the output to appear like a bad level.

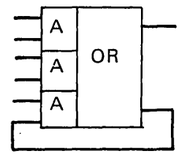


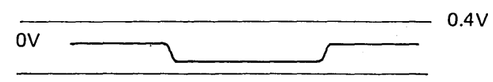
The output signal should not be gated anywhere during the time the signal does not pass through ground. Also, there should be a note on the logic pages involved.

7. Examples of outputs from correctly operating nets that may look as if they are double-terminated.

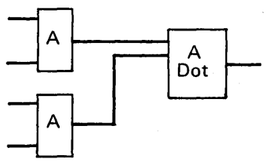


The above wave forms are typical outputs of an A0 type circuit. The double hump on the negative swing indicates that two input AND circuits are active.

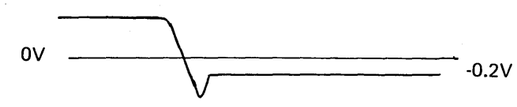




The above wave form is typical of A dot AND Ckt. With two nets tied together, the positive signal will pull up a negative signal.



8. The following is an example of a double-terminated net output.



GATE LAYOUT

	A	B	C
1	Backup Regs Stg protect TOD Clock Sel Ch (common)	Selector Channel 1 or Integrated File Adapter	M, N, MB Regs Traps and Priorities Branch Controls Interval Timer
2	Selector Channel 2 *SX4	External Assemblers Diagnostic Reg	A, B, Z, D, Regs ALU FTC Latches A, B Assemblers
3	Selector Channel 3 *SX4	External Assemblers	ALU Controls SPTL C Reg Decode Display Assembler
4	Console File MPX Channel System Reg Printer/KBD Manual Controls	C Reg SDBO Secondary Ctrl Asm LS 'A' Array Cards	LS Controls LS 'B' Array External Addressing Ctrls

'A' GATE (CARD SIDE)

	A	B	C
1	Selector Channel 4 Direct Control	3215 Printer/Keyboard Channel to Channel	
2	Phase 2I STG (control Stg. and high main Stg.)	Phase 2I STG (112 or 160K)	Phase 2I STG (208 or 256K)
3	ECC	Address-Adjust Circuits I.V.W.U. I.BU, TR Regs. Logical Regs.	Channel Ctrls LRU Reg CPU ADDR ADJ CTRLS I-Cycle Ctrl. Op Code and I Buffers
4	Phase 2I STG (Control stg. and high main stg.)	Phase 2I STG (112 or 160K)	Phase 2I STG (208 or 256K)

'B' GATE (CARD SIDE)

* Several cards for Selector Channel 4 are in A2 and A3 boards

Console Lamp Drivers
Console-File Interface Drivers
Printer/Keyboard Magnet Drivers & Lamp Drivers

01F – A1
(Behind Console)

Under–Over Voltage Detect
Power Sequence Relays
Thermal/CB Detect
Indicator Driver for Power Panel

02 – A2
(Below CE Power Panel)

Channel Cable Plug Locations

LOC	A	B	C	D	E	F	G	H	J
2	DC In	DC Out							
3	X Bus In	X Bus Out	Y Bus In	Y Bus Out	MPX Bus	SX1 Bus	SX2 Bus	SX3 Bus	SX4 Bus
4	X Tag In	X Tag In	Y Tag In	Y Tag Out	MPX Tag	SX1 Tag	SX2 Tag	SX3 Tag	SX4 Tag

B	2----13	2----13	G
D	2----13	2----13	J

DC = Direct Control

Channel-to-channel feature uses positions A3, A4, B3, B4, C3, C4, D3, and D4.

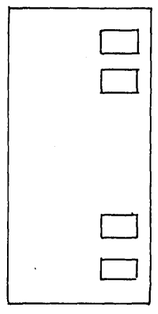
Note: Plug the terminator cards shipped in the I/O connector rack sockets into the appropriate positions in the last control unit on each channel. Refer to the installation manual for each control unit to determine the terminator location.

VOLTAGE DISTRIBUTION

Voltage Location Chart

Regulator	Logic Page	Rating	Gate	Boards
101	YE310	+1.25V @69A -3V	A	A3, B3, C3
102	YE310	+1.25V @69A -3V	A	A1, B1, C1
104	YE300	+1.25V @69A -3V	A	A2, B2, C2
107	YE301	+1.25V @69A -3V	A	A4, B4, C4
108	YE300	+6V @12A	See Special Voltage Distribution	
109	YE303	+1.25V @69A -3V	B	A1, B1, C1
103	YE300	+1.25V @69A -3V	B	A2
106	YE300	+7V @69A	B	A2
105	YE300	+1.25V @89A	B	A3
110	YE302	+2V @250A	B	A2, A4
111	YE302	+2V @250A	B	B2, B4
112	YE302	+2V @250A	B	C2, C4

Duo Reg Adj Card



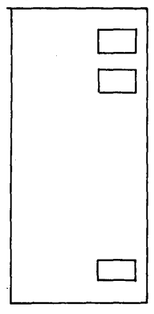
Overcurrent Potentiometer (sealed)

Overvoltage Potentiomter (sealed)

Voltage Adj Potentiometer 1.25V

Voltage Adj Potentiometer 3.0V

Single Reg Adj Card



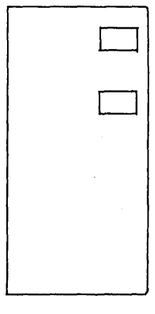
Overvoltage Adj Potentiometer

Overcurrent Adj Potentiometer

Voltage Adj Potentiometer

Usually Sealed

Phase 21 Control Regulator Card



Sealed Potentiometer

Voltage Adj. Potentiometer

Special Voltage Distribution Chart

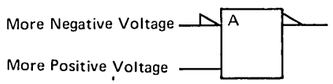
	A-Gate		B-Gate	
01A	A1L1D12	GND	01B	A1C6D04	+6
	A1L1D11	+6		A1C6E03	GND
	A2C1D11	+6		A1C6E04	+6
	A2C1E12	GND		A1T6B04	+6
	A2C1E11	+6		A1T6A03	GND
	A3C6E04	+6		A1T6A04	+6
	A3C6E03	GND		B1N1D11	+6
	A3C6D04	+6		B1N1E12	GND
	A4L1D12	GND		B1N1E11	+6
	A4L1D11	+6		B1N6D04	+6
	B1C1E11	+6		B1N6E03	GND
	B1C1E12	GND		B1N6E04	+6
	B1C1D11	+6		B1C6E03	GND
	B4C1E11	+6		B1C6E04	+6
	B4C1E12	GND			

01AF1 (Mixer Board) Special Voltage Distribution Chart

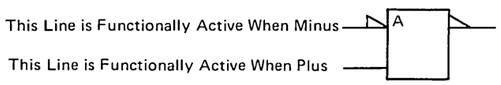
3210-1		3210-11		3215		Console File		Audible Alarm	
Pin	Voltage	Pin	Voltage	Pin	Voltage	Pin	Voltage	Pin	Voltage
C2B08	+6V	D2B08	+6V	B7D08	GND	L2B12	+24V	K2B08	+6V
C3B08	+6V	D3B08	+6V	B7D13	+24V	L2D13	+24V	K3B08	+6V
C4B08	+6V	D4B08	+6V	B7B06	-3	M2B12	24V Ret	F6B08	+6V
J2B08	+6V	C5B08	+6V	B7B08	+6V				
J3B08	+6V	D5B08	+6	M7B06	-3				
J6B07	24V Ret	G6B07	24V Ret						
K6B07	24V Ret	H6B07	24V Ret						
L6D02	+24V	M5D02	+24V						
L6D13	+24V	M5D13	+24V						
L6D10	24V Ret	M5D10	24V Ret						
L6B08	+12V	M5B08	+12V						

BASIC LOGIC SYMBOLOGY

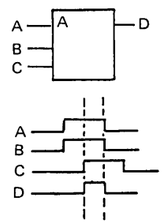
POLARITY--Is indicated by a wedge (◺) or no-wedge.



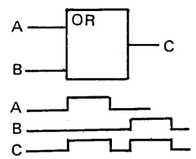
ACTIVE LEVEL--Is the line level that conforms to the edge of block character for that line.



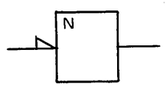
AND--The output of the AND block is active only when all of its inputs are active. The letters in the block are the symbol of the function. In this case, 'A' is the symbol for the AND function. The input may be mixed to any block.



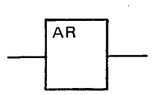
OR--The output of the OR block is active only when one or more of its inputs are active.



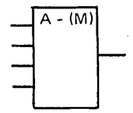
INVERTER--The output of the inverter is of opposite potential to the input.



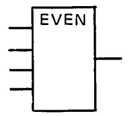
AMPLIFIER--The amplifier provides adequate driving energy and an appropriate impedance match to other blocks. The amplitifer output is active only when the input is active. An amplifier having input or output of other than standard logic signal voltage has distinctive labeling at the block.



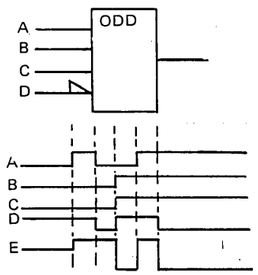
THRESHOLD--The output of the threshold is active only when the number of active inputs reaches or exceeds the number specified in the function symbol. (N) - minimum number of active inputs required for an active output.



EVEN COUNT--The output of even count (Even) is active only when an even number (such as 0, 2, 4, and 6) of inputs are active.

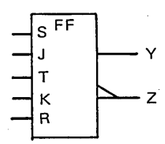


ODD COUNT--The output of odd count (odd) is active only when an odd number (such as 1, 2, 5, and 7) of inputs are active.

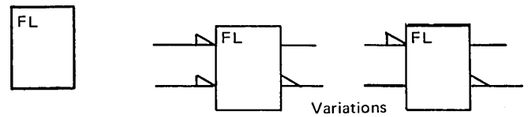


FLIP-FLOP--The flip-flop has two stable states. One of these is the 'I' state or set state: the other is the '0' state or clear state. The flip-flop block normally has two outputs. A 'I' output and a '0' output. In the ALD's a line from the upper part of the block represents the 'I' output and a line from the lower part of the block represents the '0' output.

A flip-flop can have five types (S, R, J, K, and T) of inputs in different combinations. Inputs J and K, respectively act like inputs S and R in the flip latch except that simultaneous application of a J set and K reset will complement the output. The T input complements each output. In the FF example a simultaneous S-R (set-reset) input causes output Y to follow the set (+) and output Z to follow the reset (+). If any other inputs are active during simultaneous S-R input, the outputs are undefined.

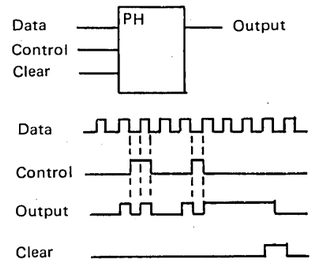


FLIP FLOP LATCH OR FLIP LATCH--The definition of this device is the same as that given for flip flop except that simultaneous application of active signals at the 'I' input and the '0' input will cause the 'I' output and '0' output to both go to the negative polarity or both go to the positive polarity (depending upon the characteristics of the particular circuit type) for the duration of such simultaneous input application. Complement input is not applicable to this block.

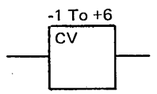


POLARITY HOLD--The output of this block will follow the data (CD) line as long as the control line is active. When the control input goes inactive the output remains at whatever polarity it possesses at that moment. The PH block may have a clear input. If so, when the clear input is active the output is inactive.

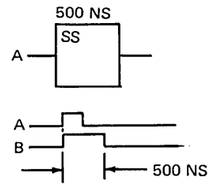
The output line is toward the top of the block. The data line is the input line toward the top of the block. The control line is centered on the input side of the block. The clear line is toward the bottom of the block.



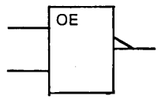
CONVERTER--The converter block provides the necessary conversion between two types of logic. Voltage mode to current mode, voltage to voltage, etc. An indication of input and output voltage levels, or line types, may be shown in the title area of the block.



SINGLESHOT--The output of the singleshot becomes active when the input is active. The output remains in this state for a time characteristic of the particular block. Regardless of the length of the input signal, the singleshot always has the time duration shown in the title area of the block. If a singleshot has more than one output not of the same duration, the block is labeled or a reference note on the page relates pin numbers to time durations.

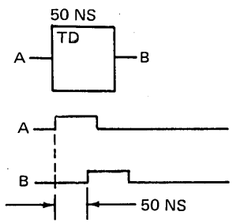


EXCLUSIVE OR--The output of an exclusive OR block is active when only one of its inputs is active.



TIME DELAY--The time delay block delays a signal without distoration of the signal. The time delay symbol must always be accompanied by the time delay.

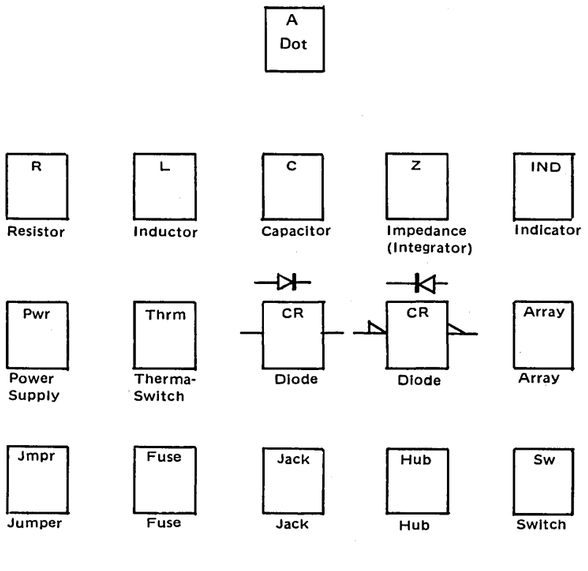
Time delays having a delay time for the leading edge of the output that is different from the time delay for trailing edge are identified by the placement of an 'L' for leading and A '1' for trailing immediately prior to the separate delay times in the block area. The input polarity at the block must be that associated with the leading edge of the output.



DOT BLOCK--A dot block represents an external connection of two or more nets. If one of the nets becomes active, it will force all nets to the active level. Blocks which are connected in this fashion will have a wedge ( ) or a plus (+) symbol under their output lines indicating the level of the active dot.

Note: Dot And blocks operate as And's--output is active only when all input nets are active.

COMPONENT BLOCKS



EDGE OF BLOCK CHARACTER--An edge of block character, alongside an FE ALD (MST) block serves the following functions

E An extender. In combination with a 'K' input, shows that additional blocks act as inputs to the first block.

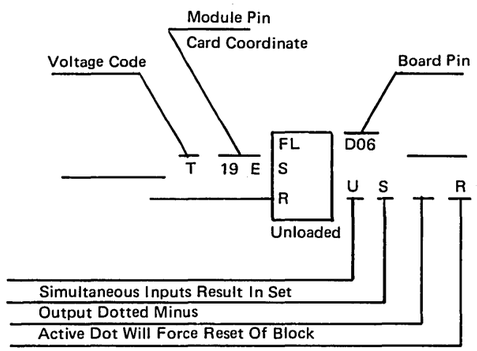
K At the output of a block, 'K' indicates that the line connects to another output. At the output of a block And connected to an 'E' output of another block, 'K' indicates the nonlogical function of an extender.

X A nonlogic input or output. The driving circuit to this input is usually a fixed voltage or bias. An X line does not influence the state of a circuit.

P A positive-going shift or pulse activates the block.

N A negative-going shift or pulse activates the block.

T A test point. Do not confuse this with 'T' as an input voltage character.



Voltage Codes

Code	Minimum Up Level (Volts)	Minimum Down Level (Volts)
B	2.5 to 2.1	1.9 to 1.5
C	2.5 to 2.1	1.4 to 1.0
D	1.9 to 1.6	1.4 to 1.0
E	2.0 to 1.6	0.9 to 0.6
F	4.0 to 3.5	0.5 to 0.3
G	2.5 to 2.1	0.5 to 0.3
H	2.0 to 1.6	0.5 to 0.3
J	1.5 to 1.1	0.5 to 0.3
L	0.7 to 0.5	0.4 to 0.2
T	0.3	-0.3
Z	-1.0	-1.5

Block Characters

C	Control line of PH
CD	Controlled data line of PH
J	Set line. See flip flop
K	Reset line. See flip flop
R	Reset line
S	Set line
T	Complement line. See flip flop
U	Unloaded output
X	Nonlogical line (Exm bias)
*	Indicated off board connection or labeled load resistor

DOCUMENTATION SUMMARY

3145 Related Manuals & SRL's (Not Complete)

FETOM MST Packaging, Tools & Wiring Change Procedure	SY22-6739
FETOM Components Circuits	SY22-2798
FETOM Power Supplies	Y22-2799
3145 Parts Catalog	S124-0120
3345 Parts Catalog	
3215 Parts Catalog	S124-0107
3210 Parts Catalog	S124-0114
Selectric I/O-2 Parts Catalog	S131-0024
23FD Integrated Theory/Maintenance	SY26-4154
3215 FETMM	SY24-3560
3215 Maintenance Analysis Procedures (Order by Part Number Per Sheet)	
3210 Theory/Maintenance	SY24-3559
Selectric I/O-2 Theory/Maintenance	SY27-0078
3345 Theory/Maintenance	SY24-3581
3145 MDM Second Levels	SY24-3580
A Guide to System 370/145	GC20-1734
IBM System/370 Model 145 Functional Characteristics	GA24-3557
IBM System/370 Model 145 Operating Procedures	GC38-0015
IBM System/370 Model 145 Installation Information Physical Planni.	GA22-6976
IBM System/370 System Summary	GA22-7001
IBM System/370 I/O Configurator	GA22-7002
IBM System/370 Principles of Operation	GA22-7000
IBM 3210 Console Printer-Keyboard Model 2 Component Description	GA24-3552
IBM 3215 Console Printer-Keyboard Model 1 Component Description	GA24-3550
Emulating the IBM 1401, 1440, and 1460 on IBM System/370 Models 155 and 145 Using OS/360	GC27-6945
Emulating the IBM 1410 and 7010 on IBM System/370 Models 155 and 145 Using OS/360	GC27-6946
Emulating the IBM 1410 and 7010 on IBM System/370 Models 155 and 145 Using DOS/360	GC33-2005
Emulating the IBM 1401, 1440, and 1460 on IBM System/370 Models 155 and 145 Using DOS/360	GC33-2004
DOS OLTEP	GC24-5086

System/370 General Group (Microfiche)

These manuals will be shipped to each System/370, on microfiche, form MID. (Initial shipment approximately two weeks before machine ship.)

Document	Microfiche Form No.	Hard Copy Form No.
Meters - Parts Catalog	S1B4-0056	S124-0056
Tools/Test Equipment Catalog Vol. 1	S1B3-0330	S123-0330
Tools/Test Equipment Catalog Vol. 2	S1B3-0438	S123-0438
SMS Instruction/Reference Manual	S2B3-6900	S223-6900
Metering-FETOM	S2B3-2728	S223-2728
FETOM Components Circuits, SLT, SLD, ASLT, MST	SYB2-2798	SY22-2798
FETOM SLT Packaging, Tools, Wiring Change Procedures	SYB2-2800	SY22-2800
FETOM-Power Supplies SLT, SLD, ASLT, MST	SYB2-2799	SY22-2799
SMS Power Supply 60-Cycle	S2B5-6478	S225-6478
Transistor-Theory-Application	S2B3-6783	S223-6783
Transistor-Component Circuits	S2B3-6889	S223-6889
FETOM MST Packaging Tools and Wiring Change Procedure	SYB2-6739	SY22-6739

New 370 Op-Codes

OPERATION	MNEMONIC	OP CODE	FORMAT	OPERANDS	DESCRIPTION	INTERRUPTION	COND. CODE
Compare logical characters under mask	CLM	BD	RS	R1, M3 D2 (B2)	Second operand is compared with first operand under mask, (bits 12-15 = bytes 0-3), and result indicated in condition code.	Oper. Prot. Adr.	0 = Selected byte or mask = 0 1, Selected R1 Lo 2, Selected R1 Hi 3, Not used
Compare logical long	CLCL	OF	RR	R1, R2	1st Op compared with 2nd Op. and results indicated in cond. code. Note: R1 & R2 designate a pair of Regs and must start with an even Reg R1, R2 (8-31) = Start of comp. field R1 + 1, R2 + 1 (8-31) = length of comp. field R2 + 1 (0-7) = Padding char. R1, R2, R1 + 1 (0-7) = ignored	Oper. Prot. Adr. Spec.	0, = or both fields 0 length 1, 1st op lo 2, 1st op hi 3, not used
Insert Char. Under Mask	ICM	BF	RS	R1, M3, D2 (B2)	Bytes in contiguous order from 2nd Op. inserted into 1st Op. under mask, (bits 12-15 = bytes 0-3) in	Oper. Prot. Adr.	All 0, inserted B1 = 0, or mask is 0 1, 1st bit of field = 1 2, 1st bit of field = 0 0, not used
Load Control	LCTL	B7	RS	R1, R3, D2 (B2)	Control Regs. specified by R1 (start) and R3 (ending) are loaded from location designated by D2 (B2).	Oper. Privileged Op. Prot. Adr. Spec.	Unchanged
Move Long	MVCL	DE	RR	R1 R2	2nd Op. is placed in 1st. Op. location Note: Specs. of R1 & R2 are the same as CLCL inst.	Oper. Prot. Adr. Spec.	0, 1st & 2nd Op. count are = 1, 1st Op. Cnt. low 2, 1st Op. Cnt. hi 3, no movement, destructive overlap
Set Clock	SCK	B204	SI	D1 (B1)	8-byte field spec. by D1 (B1) is placed in the TOD clock (0-51 bits used)	Oper. Privileged Op. Prot. Adr. Spec.	0, Clk. valid 1, Clk. valid 2, not used 3, Clk not operational
Shift and Round	SRP	F0	SS	D1 (L1, B1), D2 (B2), I3	1st Op. is shifted in the direction and no. of digit positions specified by 2nd Op 1st Op is rounded by factor I3 on right shifts. Note: D2 + B2 bit 27-31 = no. of shifts Bit 26 = 0 = left shift Bit 26 = 1 = right shift	Oper. Prot. Adr. Data Dec. Over-Flow	0, result is 0 1, result < 0 2, result > 0 3, result overflows

OPERATION	MNEMONIC	OP CODE	FORMAT	OPERANDS	DESCRIPTION	INTERRUPTION	COND. CODE
Start I/O Fast Release	SIOF	9C + Bit 15 = 1	SI	D1 (B1)	Non-block multiplex mode: Same as SIO. Block-multiplex mode: SIO is started if sub-channel, channel are available or in interrupt pending state & no exceptional condition has been detected	Privileged Op.	0, I/O operation initiated & channel proceeding with execution 1, CSW stored 2, Channel or sub-channel busy 3, not op.
Store Channel ID	STIDC	B203	SI	D1 (B1)	Information identifying the designated channel is stored in 4 byte field at MS 168 0-3 Type 4-15 Channel Model No. 16-31 Max. IOEL Length	Oper. Privileged Op.	0, ID stored 1, CSW stored 2, Chan. busy, no store 3, not oper.
Store Char. Under Mask	STCM	BE	RS	R1, M3, D2 (B2)	1st Op. bytes, selected by mask (bit 12-15 = bytes 0-3) are placed in 2nd Op. field in contiguous order.	Oper. Prot. Adr.	Unchanged
Store Clock	STCK	B205	SI	D1 (B1)	Value of T.O.D. clock is stored in 8 byte field designated by D1 (B1)	Oper. Prot. Adr.	0, Clock in set state 1, not set state 2, error state 3, (not valid for 3145)

OPERATION	MNEMONIC	OP CODE	FORMAT	OPERANDS	DESCRIPTION	INTERRUPTION	COND. CODE
Stored CPU ID	STIDP	B202	SI	D1 (B1)	CPU data stored in 8-byte field designated by D1 (B1) 0-7 Reserved 8-31 CPU Serial 32-47 CPU Model 48-63 Max. MCEL Length	Oper. Privileged Op. Prot. Adr. Spec.	Unchanged
Store Control	STCTL	B6	RS	R1, R3 D2 (B2)	Starting Control reg. Spec. by R1 & ending with control reg. Spec. by R3 are stored at Loc. addressed by D2 (B2).	Oper. Privileged Prot. Adr. Spec.	Unchanged
Halt Device	HDV	9E	SI	D1 (B1)	Current I/O Op. of addressed device is terminated. Chaining is broken. (More detail is Sys/370 Prin. of Op.)	Privileged Op.	0, Sub-chan busy with another device or interrupt pending 1, CSW Stored 2, Chan. Working 3, Not Operational

DIAGNOSE INSTRUCTION FUNCTIONS FOR THE MODEL 145

The specific diagnose functions are designated by the 24-bit address formed by adding the contents of B^1+D^1. The two low-order bits of this address are ignored so that the address always appears to be on a word boundary. The following specific Diagnose Instruction functions have been defined for the Model 145.

Load Patch Words (B+D=000000)

This diagnose function loads patch control words into control storage from main storage. This op code is enabled when the service representative sets a bit in an external register (as yet undefined) by preparing a programmable card. This Op code is disabled after the control words have been successfully patched (condition code 0). Thus, there can be only one successful patch per IMPL. If this Op code is issued again, no control words are changed (condition code 3). This diagnose function has the following format:

83	R–	BD	DD

The address in general register R specifies the main-storage location of the patch control words and their addresses. This main-storage address is forced to a word boundary by circuitry. Data for each patch word occupies two words in main storage. Bytes 2 and 3 of the first word contain the 16-bit control-storage address where the patch word is to be inserted. The second word contains the actual control word. If byte 0 of the first word is nonzero, the operation is ended: no control word is moved to control storage.

Module BC of control storage is reserved for adding microprogram changes (adding control words). Any added control words should be assigned to this module.

This diagnose function sets the following condition codes.

0 Patch control words properly loaded.
1 Patch address is outside of control storage.
2 FE pin not plugged (the service representative has not enabled this Op code).
3 Second patch attempt: Op code is now disabled.

83	I^2	B^1	D^1

format:

PSW Restart (B + D = 000004)

This diagnose function causes the same action as pressing the PSW restart key on the console.

Main-Storage Full-Recording Mode (B + D = 000008)

This diagnose function sets the mode register to full-recording mode for single-bit corrections in main storage. There is no change to the condition code.

Control-Storage Full-Recording Mode (B + D = 000010)

This diagnose function sets the mode register to full-recording mode for single-bit corrections in control storage. The condition code remains unchanged.

Main-Storage Quiet Mode (B + D = 00000C)

This diagnose function sets the mode register to quiet mode for single-bit corrections in main storage. The condition code remains unchanged.

Control-Storage Threshold Mode (B + D = 000014)

This diagnose function is used to set threshold mode for single-error corrections in control storage.

Control-Storage Quiet Mode (B + D = 000018)

This diagnose function sets the mode register to quiet mode for single-bit corrections in control storage. The condition code remains unchanged.

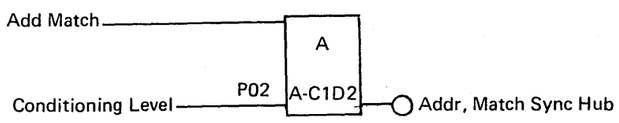
TRAP LOCATIONS AND ROUTINES

OPERATION	H-REGISTER BIT	TRAP ADDRESS	ROUTINE
Selector share cycles	None	None	
Machine check without I/O	H0	– –	
a. Normal		D000	GHEC
b. H0 is already on		D004	GHEC
c. One or more machine checks have already occurred (SYSO, Bit 2=1)		D008	GHEC
d. H0 and SYSO Bit 2 are already on		D00C	GHEC
Machine check with I/O	H0	– –	
a. Normal		D010	GHEC
b. H0 is already on		D014	GHEC
c. One or more machine checks have already occurred (SYSO, Bit 2=1)		D018	GHEC
d. H0 and SYSO Bit 2 are already on		D01C	GHEC
Retry	H1	– –	
a. Normal		D200	GHRT
b. H1 is already on		D204	GHRT
c. A retry trap operation is in progress (SYSO, Bit 1=1)		D208	GHRS
d. H1 and SYSO Bit 1 are already on		D20C	GHRS
CPU High	H2	– –	
a. Set IC		D300	GKCC
b. CA Trap		D304	GKCC
c. Address Contents		D308	GKCC
d. System Reset		D30C	GRST
IFA	H3	– –	
a. Mini-Op End		D128	GPBH
b. Error End		D12C	GPCG
c. Index		D124	GPCE
d. Gated Attn		D120	GPBK
Selector Channels 1, 2, 3 No IFA	H3	--	
a. Exceptional Status Trap		D120	GSES
b. Chaining (CC or CD)		D124	GSTR
c. UCW Handling		D128	GSTR
d. Unused		D12C	GSTR
Selector Channels 2, 3 with IFA	H4	– –	
a. Exceptional Status Trap		D100	GSES
b. Chaining (CC or CD)		D104	GSTR
c. UCW Handling		D108	GSTR
d. Data (unchained)		D10C	GSTR
Multiplexer	H5	D400	GMSR
IFA	H6	– –	
a. Return Low		D480	GPBK
b. Unused		D484	
c. Unused		D488	
d. Diagnostic		D48C	GPDO
Store/Display	H7	– –	
a. Store/Display		D840	GKAD
CPU Low without I/O	None	– –	
a. Spare		– –	
b. Storage Protect		D804	GICM
c. Address Check		D808	GICM
d. Address Adjust Exception		D80C	GGST
CPU Low with I/O	None	– –	
a. Spare		– –	
b. Storage Protect		D814	GMDR
c. Address Check		D818	GMDR
d. Spare		– –	
Scan/Clear	None	– –	
a. Scan Storage		D380	CSTS
b. Clear Storage		D384	CSTS
Single-Cycle Allow I/O and Soft CA Match	None	DC00	GKCC

H-Reg Bit	Blocks Trap Request for H-Reg Bit
H0	None
H1	H2, 3, 4, 5, 6, 7
H2	H2
H3	H3, 4, 5, 6
H4	H4, 5, 6
H5	H5, 6
H6	H6
H7	H7

CPU MAINTENANCE AIDS

1. To prevent I-cycles prefetch, tie B-B3 G2 D04 to B-B3 V3 B08. (ALD reference page RV023.)
2. To provide a conditioning level to the address-compare match circuits, tie the conditioning level (it must be a minus signal) to A C1 D2 P02.

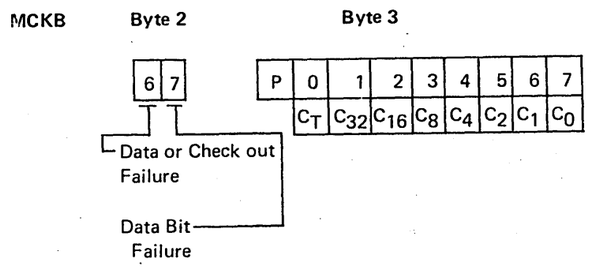


ADL PM 013

3. Pressing the store push button runs the CPU clock for three cycles. Pressing the display push button runs the CPU clock for two cycles. It is possible to single-cycle through these operations by adding a jumper from A—A4 R2 B08 to A—A4 Q4 B07 (Ref. Logic KM012).
4. To hardstop the machine while running microdiagnostics (that is, error occurs in a timing loop),
   a. Put the check control switch in hardstop.
   b. Tie-up a B3 M2 B09 to A—B3 H4 B08 (Ref. Logic RE012—Suppress all traps)
5. Automatic recycle of a failing program loop (PSW Restart)
   a. Use the control-word address-trap position on the address-compare switch. (Refer to the console section of this manual.)
   b. Alternate method: Use this method if the switches are needed to generate an address-compare sync.
      1. Find a word in the microlisting where the restart operation should begin.
      2. Build a branch and module switch word to branch to the GGAD routine, statement LNK 04. This performs a branch to the PSW restart function. The address should be in the IPL PSW at location 0.
6. Another method of performing a restart operation is by use of the diagnose instruction. The PSW restart function of this instruction (83000004) can be stored in the program stream. A write-up on the diagnose instruction is included in this section.
7. To stop on single ECC errors, tie B2N2D07 to B2M2B07 (RE061). To determine which list is being corrected, perform the following.
   a. Turn on the full-recording bits in the mode register (external address 08, byte 0). Bit 3 is for main-storage full recording. Bit 4 is for control-storage full recording. Turn on these bits before running the failing job.
   b. Put on the jumper to stop on single ECC errors.
   c. The machine will hardstop when the single ECC error occurs. The SNG ECC light will be on.
   d. Display MCKB (machine check B), external address 06, byte 3, to determine the failing bit.
   e. Determine the failing address by using the information from the microlisting and the indicators.

Examples

To determine the failing bit, total the binary value of the C bits in MCKB3. Bits 6 and 7 of MCKB2 determine whether the failing bit is a data bit or a check bit.



1 1	0 0 0 0 0 1 1 0	Bit 3 is failing
1 1	0 1 0 1 0 0 1 0	Bit 41 is failing
1 0	0 0 0 0 0 0 0 1	Check Bit C_0 is failing
1 1	0 0 0 0 0 0 0 0	Bit 0 if failing
1 0	1 0 0 0 0 0 0 0	Check Bit C_T is failing

8. To loop the storage scan or storage load function of the diagnostic/console file control switch, refer to the comments in the CSRD routine of the System/370 microlisting (about sequence number 300).
9. When troubleshooting problems with the System/370 microprogram load, be sure to take advantage of the Alter/Display capabilities of the console printer (GKAD Routine).

Alter	Display	Address Range	Storage Area
AM	DM	0 Thru 7FFFF	Main Storage
AK	DK	0 Thru 7FFFF	Storage Keys
AS	DS	0 Thru FFFF	Control Storage
AL	DL	00 Thru 7F	Local Storage
AC	DC	0 Thru F	Control Registers
AG	DG	0 Thru F	General Regs
AF	DF	0, 2, 4, 6	Floating-Point Regs
AP	DP	----	Current PSW
T	T		Test

*Requires CE Key

NOTES:

1. Expanded local storage (40 through 7F) can be displayed but not altered.
2. The T mnemonic can be used to correct mincr mechanical typewriter problems between customer jobs without the necessity of loading the micro-diagnostic disk. Refer to GKTM routine.
3. When AM/DM is used, it is not necessary to type a six-digit address. Example: To alter main memory location 0-type AM0; then carriage-return.
4. If a mistake is made while typing in data, instead of trying to determine which byte to alter, type in AM to the nearest word & use the space bar to get the correct byte. Each pressing of the space bar causes the character in storage to be printed.

CHANNEL MAINTENANCE AIDS

1. To use the address-compare switch functions during an IPL operation (address compares uses switches F, G, H normally used for the IPL address):
   a. Method 1——Control address stop in the GMPL routine before the word that moves the switches to the LS Reg (about Sequence No. 33). Put the IPL address in the switches and single-cycle through until the switches are read. Put the new stop word in the switches and return the rate switch to process.
   b. Method 2——Change the microprogram in the GMPL routine. Remove the word that reads the switches (R=SW, S3). Insert in its place RDH R DM, FC (Bit structure is 5 x 50 FC XXX).

      Next Address

      Store the IPL address in main-storage location 00 FC. Now the switches C—H are free to use for address-compare functions. (That is 00FC = 0280)

2. The selector channels gate the following words into the C-Reg for share cycles. The share cycle light should be on.

Operation	Buffer	No Buffer
Output	40B80C08	60B80C08
In forward	48880C08	68B80C08
In backward	488C0C08	68BC0C08
Skip	48840C08	68B40E08

3. All selector channels share the same microroutines in the 370 microprogram listing. Therefore, the WA1 - WA2 local storage and external addresses that appear in the far-right column of the microprogram listings may not be correct. To determine the local-storage and external addresses for the selector channels, use the maps for LS and externals in this manual. There are four indicators in the upper roller, Position 1, to assist you in determining which channel is operating or which one was operated last. The selector-channel routines are GSXX.
4. Selector Channel Logics

Chan	1	GB & GC
	2	GF & GG
	3	GK & GL
	4	GP & GQ

GA pages are common to all channels.

CONSOLE-PRINTER MAINTENANCE AIDS

1. To change Address of Console Printer
   –Patch C. S. (FF60) DC area.

		32 UCW's	16 UCW's
If Doc Console Address =	09 C.S. FF60 =	00 E090	00 09 E090
If Doc Console Address =	1F C.S. FF60 =	00 E1F0	00 1F E0F0
2nd Doc Printer =	1E C.S. FF68 =	00 E1E0	00 1E E0E0

2. To single-cycle alter/display
   Refer to KM012 Line name (+ Store Display Single Cycle)
   Tie this line plus. (Use clock card tie-up point.)
3. To print single characters continuously, refer to Basic Micro Diag. routine BLA0.
4. To print single characters from keyboard without altering storage, use alter/display test mode. Refer to routine GKTM.
5. To scope keyboard entries, refer to: BLA0 (3210), BMA0 (3215)

PR-KB Troubleshooting Hints

This procedure can be used for both the 3210 and the 3215 except where noted otherwise.

This procedure is not intended to be a test procedure but assists to statically check out suspected trouble areas without a microprogram loaded.

This procedure checks out about 90% of the Selectric I/O adapter by sorting the specified data into the Selectric I/O registers. The Selectric I/O has four registers, (T1, TA, TT, and TE). Only two of these registers may be used as destinations (TA and TE), but the TI and TT registers can be checked out by storing specific data in the TA and TE registers.

The console PR-KB registers can be selected by setting the bottom roller switch to position 2. Rotary switch H selects which byte is to be stored. Switch H set to 2 stores in the TA-register whatever data is dialed into the A and B switches. To store data into the TE-register, set switch H to 3.

Dial the bottom roller switch to position 4. Set H switch to 3. Set the A and B switches to OE. Set the storage select switch to display the external registers and store. The H-register should contain OE. This prevents the adapter from trapping. Dial roller position 5, set byte 0 (MTO Reg) to 00. Return the bottom roller switch to position 2.

A. TA-register check
   1. Set switch H to 1 and the A and B switches to 00 and store. TA-register should be reset and the parity bit on.
   2. Store each bit individually into the TA-register to check for shorts, opens, and transpositions.
   3. Set TA-register to 00.

B. TE-register check
   1. Set switch H to 3, set A and B switches to 00 and store. All TE-register bits should be off, and the parity bit on.
   2. Store each bit individually into the TE-register to check for shorts, opens, and transpositions.

C. TT register reset check
   1. With the PR-KB not ready, set the H switch to 1, set the A and B switches to 18 and store. The TT register should be 00 or 08. Bit 4 is the ball side bit and may be in either state.
      Note: Ball side does not apply to 3215.
   2. Set the TA-register to 00, and the TT-register should not change.

D. TT-Register set check
   1. Set the rate switch to SINGLE-CYCLE IMMEDIATE STOP. Make the PR-KB ready, press START once. The TT-register bit 1 (operational), bit 5 (end), and bit 6 (console request) should be on. Intervention required light off.
      Set TA-register to 08.
      TT-register bits 5 and 6 should be off.
      Set TA register to 00.
   2. Press the request key.
      Press START.
      TT-register bit 0 (attention), bit 6 (console request), and the request pending indicator should be on.
   3. Set the TA-register to 08. TT-register 0 and 6 bit should turn off. Also request-pending indicator should be off.
      Set the TA-register to 80, (the proceed indicator should be on). TT-register should not change.
   4. Press the END key,
      Press START
      TT-register bit 5 (end) and bit 6 (console request) should be on.
      Set TA-register to 08, TT-register bits 5 and 6 should turn off. Proceed indicator is off.
      Set TA-register to 80.
   5. Press the cancel key and start button. TT bit 6 (console request) and TT bit 7 (cancel) should be on.
      Set the TA-register to 08. TT bit 6 and 7 should turn off.
      Set the TA-register to 80.
   6. Press the alter/display key, then the start key, once.
      The TT-register bit 2 (alter/display bit) should be on.
      Set H-reg to OF.
      Set the TA-register to 84. TT-register bit 2 should be off.
      Set the TA-register to 00.

7. Set the TA-register to 20, TT 6 (console request) should turn on.
   Dial bottom roller to position 5.
   Set byte 0 bit 6 (MPX suppress out) on.
   Dial roller position 2. TT bit 6 should be off.
   Reset MT0 bit 6 (MPX suppress out). TT-register bit 6 should be on.
   Set TA-register to 00. TT-register bit 6 should be off.
8. Set TA-register to 04. TT-register bit 6 should be on.
   Set TA-register to 00. TT-register bit 6 should be off.
9. Set TA-register to 40. TT-register bit 6 should be on.
   Set TA-register to 00. TT-register bit 6 should stay on.
   Set TA-register to 10. TT-register bit 6 should turn off.

E. Clock circuit and printing check (does not apply to 3215)
   1. Set the rate switch to PROCESS.
      Set TA-register to 50.
      Set TE-register to 08. Typewriter should have done a carriage return.
   2. Watch the TT-register while storing 01 repeatedly.
      TT bit 4 (ball side) should be turning on and off.
   3. Set TE-register to 02. Typewriter should print an A.
      (In the event that the ball is in a different case than the one desired, set the TE-register to 01 to shift to the opposite case.) Any character can be printed by storing the proper T/R code into the TE-register.
      (See pages 7-6 and 7-7.)

F. Keyboard and TI-register check
   1. Set TA-register to 90.
      Press desired key, and its appropriate code will appear in the TI-register. TI-register (modified keyboard) codes are on page 7-7.
      Note: EBCDIC characters appear in the TI-register for the 3215.

G. To operate the second (remote) 3210, turn on bit 3 of sys byte 0 before starting this procedure.

Character	TI reg.	Character	TI reg.
1	31	=	F1
2	32	<	F2
3	33	;	F3
4	34	:	F4
5	35	%	F5
6	36	'	F6
7	37	>	F7
8	38	*	F8
9	39	(	F9
0	30	)	F0
-	20	—	E0
&	10	+	D0
q	18	Q	D8
w	26	W	E6
e	05	E	C5
r	19	R	D9
t	23	T	E3
y	28	Y	E8
u	24	U	E4
i	09	I	C9
o	16	O	D6
p	17	p	D7
@	3C	¢	FC
a	01	A	C1
s	22	S	E2
d	04	D	C4
f	06	F	C6
g	07	G	C7
h	08	H	C8
j	11	J	D1
k	12	K	D2
l	13	L	D3
$	1B	|	DB
≠	3B	"	FB
z	29	Z	E9
x	27	X	E7
c	03	C	C3
v	25	V	E5
b	02	B	C2
n	15	N	D5
m	14	M	D4
,	2B	¦	EB
.	0B	¬	CB
/	21	?	E1
Space	00	Space	C0
C/R	1C	C/R	DC
U/C	CE	L/C	3E

Side 0	TE reg.	Side 1	Side 0	TE reg.	Side 1
a	02	A	)	82	/
(	06	'	—	86	
e	12	E	v	92	V
d	16	D	u	96	U
i	1A	I	z	9A	Z
h	1E	H	y	9E	Y
c	22	C	t	A2	T
b	26	B	s	A6	S
g	32	G	x	B2	X
f	36	F	w	B6	W
|	3E	.	?	BE	,
j	42	J	<	C2	1
@	46	*	%	C6	0
n	52	N	;	D2	5
m	56	M	¬	D6	4
r	5A	R	+	DA	9
q	5E	Q	>	DE	8
l	62	L	¢	E2	3
k	66	K	&	E6	2
p	72	P	!	F2	7
o	76	O	:	F6	6
≠	7E	$	"	FE	=

INTEGRATED FILE ADAPTER (IFA) MAINTENANCE AIDS

1. a. WARNING: Use STOP position of address-compare control switch if disk files are in operation. If Hard Stop is used and IFA is in a write mode, the data can be incorrectly written. (Customers should be made aware of this also if they use address-compare functions.)
   b. Online diagnostics should not be running when customer performs an IPL.
   c. Online diagnostics affect customer's throughput. Record test results and analyze offline. Use FD test box whenever possible.
2. To change the control-unit address of IFA (Normally 30):
   Refer to 370 microprogram, routine GPAA.
   Patch control storage (D. C. Area) byte at FFAB

   FFAB = 3X 3 = Address of 30
   X= Not used (Formerly No. of Physical attached drives)

3. Scoping levels seen in IFA and 2319
   MST + .5V to – .5V
   SLT + 3.0V to 1.0V or 0V
   MST to SLT converters
   1.5V to –2.5V
   CE Panel Ckts.
   Float Lev. (approx.ground) to -3.0V
4. 'S' Service plug part number 2218608
5. To connect CA SYNC (or other IFA line) to 2319 CE Panel Sync Hub: (Temporary procedure)
   Attach one end of tri-lead to desired sync point.
   Remove plastic housing from the other end of tri-lead and insert in tri-lead at 01A–B1 E1 A11 (WF 111 – Probe Line). Be careful not to short signal to ground.
   20" tri-lead = P/N 817687 50" = 817095
   Note: A negative signal sent to the 2319 also turns on the 2319 probe light.
6. To gate the address-compare switch-function (Example: CA match with certain IFA line), refer to console area of manual, address-compare switch (PM 013).

   Note: The output can now be used as a sync for 2319 troubles or to use the 2319 probe light as a circuit monitor. (See IFA No. 6.)

7. IFA Microroutines GPAA – GPCG
   Online 2319 Tests GPDO – GPD2
   IFA Extended Microdiag. SB Series
   (2319 VFO Adjustment procedure is in SBAO and Chapter 10)

8. IFA retry bits as posted in extended logout.
   FGL (Ext 22-3 Bits 5-7)
   000 Error occurred while executing a test I/O
   001 Set when a drive is selected and before Mini-Op routines are entered from I-cycle
   010 Set when IFA decodes a valid CCW.
   011 Set on every successful share request.
   100 Set when IFA CCW not valid.
   101 Sets when a drive is selected.
   110 Unused.
   111 No other code applies

9. IFA Share Cycle Words (Seen in C-Reg)

		Main Stor.Share	
Input	68 B8	OC	08
Output	60 B8	OC	08
Input-Skip	60 B8	OE	08
		Control Stor.Share	
Input	68 B8	40	08
Output	60 B8	40	08
Input-Skip	68 B4	42	08

10. File drive head adjustment. Improving scope picture if using online diag's. (Cannot be used if customer is using IFA.)
    GPD 1 Routine
    Change word that sets CE Wait Time SS (GA,A–,K80 seq. 23 Approx.) to branch to next word.
    Patch Word = OXOOXXXX
    X's = ADR of next word.

IFA Latches Logic References

IFA LATCHES

ADDR MARK 1 & 2	JL413
ADDR MARK–MOP REG	JL314
ALLOW BCA CK	JL811
ALLOW FULL OSC.	JL311
ALLOW GA DEC.	JK513
ALLOW HD COND	JL411
ALLOW INDEX	JL411
ALLOW RESTART	JL612
ALLOW TRAPS	JL612
BIT RING LATCHES	JL312
BIT COUNT TIME	JL216
BIT COUNT APP. LATCHES	JL213
BLOCK CLK	JL313
BYTE CTR LATCHES	JL111-4
CC DATA	JL216
CC HWD ERROR	JL015
CC REG. INPUT	JL216
CC REG. LATCHES	JL211-2
CCW 0	JL117
CE MODE	JL514
CE TRAP REQ.	JL514
CE–PANEL SW LATCHES	JL021-2
CHAIN CMD	JK111
CHAIN DATA	JK111
CHAN DATA CK	JK212
CHAN CTRL CK	JK212
CHAN BUSY	JK313
CLK GAP SENSE RESTART	JL413
CMD OVERRUN	JL513
COMP GATE	JL215
COMPARE HI	JL215
COMPARE LOW	JL215
COMP. RD DATA	JL215
CONT RD TO INDEX	JL513
CONTINGENT CONNECTION	JK513
CONTROL	JK112
CONT. UNIT ADR. BITS	JL216
COUNT O GATE	JL116
COUNT O SHARE	JK512
CS COUNT RDY	JK512
CS SHARE CYC	JK512
CU BUSY	JK313
CUA LOAD	JK112
CYC ACTIVE	JK512
DATA GATE	JL117
DATA REQ	JL117
DATA REQ.HONORED	JK511
DATA FIELD PEND	JL513
DATA CK	JL811
DATA OVERRUN	JL513
DCC MODE	JK313
DELTA INDEX	JL411
DIAG DATA SENSE	JL513
DIAG. ERR. REG–FED–	JK312
DIAG. MODE	JL514
DIAGNOSTIC	JL411
END DATA FIELD	JL117
ERASE GATE	JL412
ERROR RESTART	JL612
ERROR TIME OUT	JL612
ERROR TRAP	JL611
FBO REG	JK311
FCH REG	JK211
FCL REG	JK311
FDR REG	JK411-2
FDR FULL	JK511
FDR FULL BUFF	JK416
FFL PARITY	JK111
FILE HD–CYC LAT.	JK216-7
FMOD LATCHES	JK611
FOP REG	JK411-2
FORCE DEC O	JL116
FORMAT	JL314
GATE LAST REQ.	JL414
GATED ATTENTION LATCHES	JL811
GATE INDEX	JL411
GATE BR–O D–O	JL311
GATED SERIAL DATA	JL011
HD CONDITION	JL411
HI TRAP	JL611
IDA	JK111
IFA SHARE 2	JK511
INCORRECT LENGTH	JK513
INDEX	JL411
INDEX START	JL314
INDEX TRAP	JL611
INPUT	JK512
INTERRUPT COND. BUFF.	JK416
INTERRUPT	JK313
LATE SHARE	JK513
LOST ON LINE	JL611
MISSING AM	JL014
MOP PARITY ERR.	JL611
MS COUNT RDY	JK512
MS COUNT RDY BUFF	JK416
MULTI–MOD. SEL.	JL511
NTO OP	JL612
ON–LINE BUFF	JK118
ORIENTATION	JL014
OUTPUT	JK512
PACK CHG BUFF	JK118
PCI	JK111
PRE–FULL OSC.	JL311
PROG. CK	JK212
PROTECT CK	JK212
RAW DATA	JL514
RD BUFF PARITY–SERDES–	JL013
RD GATE	JL412
RD SYNC GATE	JL413
READ	JL314
RETRY LATCHES 5, 6, 7	JK612
SCAN	JL314
SCAN BYTE	JL013
SEARCH	JL314
SELECTED GATED ATTENT	JL215
SERDES CHECK	JL013
SERDES LATCHES	JL012
SERDES OUTPUT	JL011
SET HEAD	JK112
SET CYC	JK112
SET DIFF.	JK112
SHARE CYC ERR. BUFF	JK416
SHARE ERROR	JK513
SKIP	JL314
SKIP	JK111
SLI	JK111
SPARE A, B, C	JK112
STANDARD INDEX	JL411
STD RD DATA	JL215
STW CYC 1 ADR ADJ.	JV011
STW CYC 2 ADR ADJ.	JV011
STW INTLK ADR ADJ.	JV011
TRACK OVERRUN	JL613
WR CLK GATE	JL412
WR CURRENT ERR	JL611
WR. SYNC	JL117
WR. ZERO	JL117
WRITE	JL314
WRITE GATE	JL412
WRONG LENGTH REC	JL512
WRONG SYNC	JL612
ZERO PAD PARITY	JL013
7TH ZERO	JL414
8TH ZERO	JL414

IFA Sense and Status Information

Sense Byte 0-3

	Byte 0	Byte 1	Byte 2	Byte 3
Bit 0	Command Reject	Data Check in Count Field	Unsafe	Ready
Bit 1	Intervention Required	Track Overrun		On Line
Bit 2	Bus-Out Parity	End-of Cylinder	Serdes Check	Unsafe
Bit 3	Equipment Check	Invalid Sequence	Selected Status	Write Current Sense
Bit 4	Data Check	No Record Found	Cyclic-Code Check	Pack Change
Bit 5	Overrun	File Protected	Unselected File Status	End-of Cylinder
Bit 6	Track Cond Check	Missing Address Marker		Multi-Module Select
Bit 7	Seek Check	Overflow Incomplete		Seek Incomplete

Byte 4: Physical File Address (See Text)
Byte 5: Record Overflow Codes (See Text)

Sense Byte—4

The bits of byte-4 identify the physical disk drive assigned to the given address. Only the four low-order bits are assigned. The same four bits are written as the low-order of the BCA indicator byte at the end of each field.

Bits 0 1 2 3 4 5 6 7	Physical Drive
0000 0000	A
0000 0001	B
0000 0010	C
0000 0011	D
0000 0100	E
0000 0101	F
0000 0110	G
0000 0111	H
0000 1111	Module not defined.

Note: Module not defined occurs when the identifier plug for the given address has not been inserted.

Sense Byte—5

This byte is zero at all times except when an overflow-incomplete (Byte—1, Bit-7) occurs.

0123	4567	Hex	Interrupted Condition
0000	0110	06	A read command was in progress.
0000	0101	05	A write command was in progress.
0010	0101	25	A search-KD-equal was in progress, and the record is equal to this point.
0100	0101	45	A search-KD-high was in progress, and the record is to this point.
0110	0101	65	A search-KD-high or -equal was in progress, and the record is equal to this point.
0101	0101	55	A search-KD (any type) was in progress, and the record at this point is low. A search-KD-equal was in progress, and the record is unequal. The status-modifier must not be set for the ending.
0111	0101	75	A search-KD-high or search-KD-high/equal was in progress and the record to this point is high. The status-modifier must be set for the ending.

Unit Status (CSW 32-39)

Bit 0	Attention (not used by IFA)
Bit 1	Status modifier
Bit 2	Control-unit end
Bit 3	Busy
Bit 4	Channel-end
Bit 5	Device-end
Bit 6	Unit check
Bit 7	Unit exception

Channel Status (CSW 40-47)

Bit 0	Prog. Control interrupt
Bit 1	Incorrect length
Bit 2	Program check
Bit 3	Protection check
Bit 4	Channel Data Check——This bit is set to indicate a bus parity error during IFA share–cycles
* Bit 5	Channel Control Check--This bit indicates that the CPU detected an error other than storage-protect or storage-wrap during a file share-cycle. The condition is detected by the CPU circuits but is referred to the IFA for posting. The operation and chaining are suppressed.
Bit 6	Interface Control Check--This bit indicates that a command overrun condition existed during the operation.
Bit 7	Chaining Check--Not used by IFA

*Red light indications if in hard stop.

370 MICROPROGRAM LOAD (IMPL)

Start console file

NOTES
1. Be sure that all Sw's are in process and that the disk is inserted properly.
2. Console file eight turns from pink to white.

Start loading from disk address 00

Validate Stg.

DACC Routine (Overlay 1)
1. Stop words are loaded into every position of control storage.
2. Clear MS if load follows a power-on reset (POR).
3. Stg. errors (if any) are blocked during this routine.
4. The EC number, serial number, and feature descripter cards are checked and compared to data on the disk to ensure that the proper microprogram is being loaded. If not, a diagnostic stop occurs.

3. The console writer should restore and carriage-return.
4. Use the M-Reg andthe address listing (Overlay 1) to determine what caused the stop.

The plug cards are located in A-B2 P3, P4, P5. Display external addresses 12 & 16 to read the plug card.

DGN 0-6 Routines (Overlay 2)
1. These tests perform a checkout of the CPU.
2. If a failure occurs, a CPU error may be on or a stop word may be cycling.
3. If a failure occurs during these tests, load the microdiagnostics to isolate the problem.

5. Console writer prints Go/No-Go complete.

Validate Stg.

DACS Routine (Overlay 3)
1. Once again, stop words are loaded into control storage.

Load the 370 Microprogram (Overlay 4)

1

1

ZZZ Routine
1. Perform a CHKSum on Cntrl Stg by Exclusive Or of all control words.
2. Bad CHKSum is indicated by a diagnostic stop and M Reg. byte 2 = BC.
3. Troubleshoot by loading microdiagnostics if stop occurs.

System Reset
IPL
PSW Rst

Trap to D30C

System Reset

GRST Routine
1. Validate LS, channels, externals to proper values (see descriptive text in GRST for details.)

Diagnostics

CSRD Routine
1. Checks the basic data paths and verifies that the checking circuits are functional.
2. Clear Stg. protect keys if Power on Reset or Enable CLR is active.
4. Clear main Stg. if Enable CLR is active.
5. Errors are indicated by the Diag. stop lite on.

6. If failures occur, load microdiagnostics.

System Reset

GRST Routine
1. Enable I-cycles (DF module ADR forces hardware operation).
2. Set control Stg. to threshold mode, main Stg. to quiet mode, turn off IMPL indicator.
3. CHK if patch required (TODL Byte 3 Bit 3 = 1 Ext. Adr. 3E).

Load Card Rdr. with Patch IPL Card RDR.

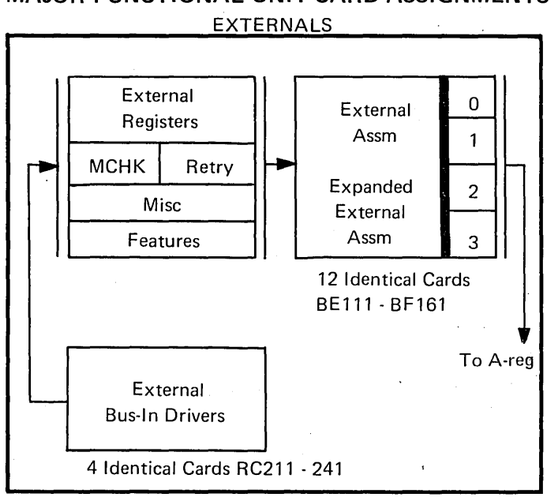
Patch Req'd	IPL	GIPW Routine	PSW Restart	GICM Routine	System Reset	Stop Word

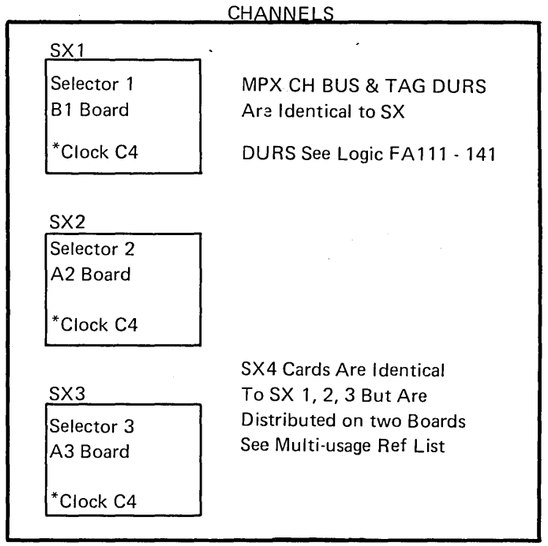
7. If a microprogram temporary fix (patch) is required, move the jumper on the card located in A-A1R2 from a zero to a one. Refer to the write-up in the microprogram section for a complete description of installing a microprogram patch.

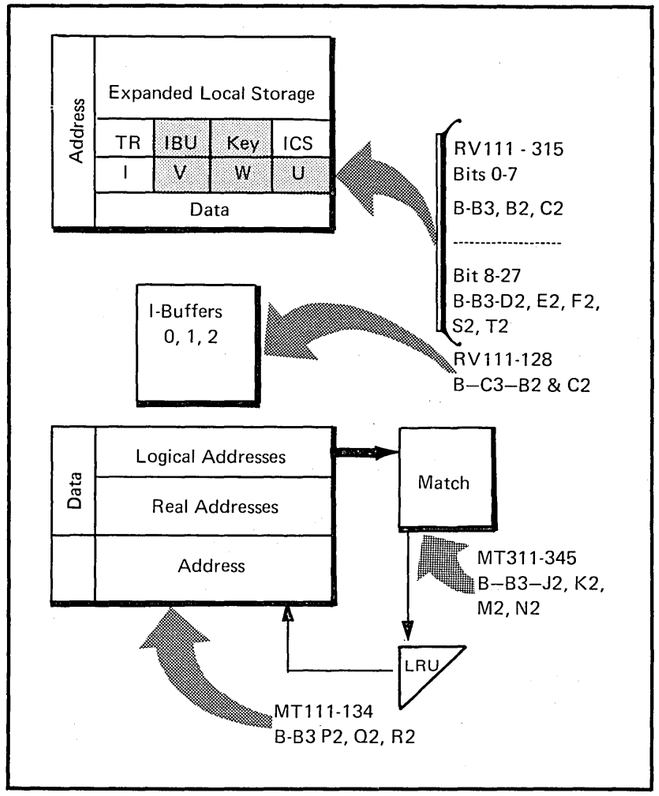
Block multiplex overlay routine will configure BLK MPX channel table for type of device on each channel.

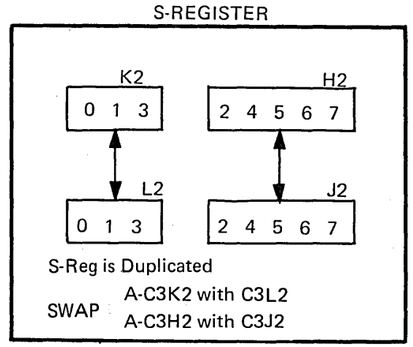
	Plug Card Locations	Logic Page
CH1	A-B1S2	GB63X
CH2	A-A2S2	GF63F
CH3	A-A3S2	GK63X
CH4	A-A2T2	GP61X

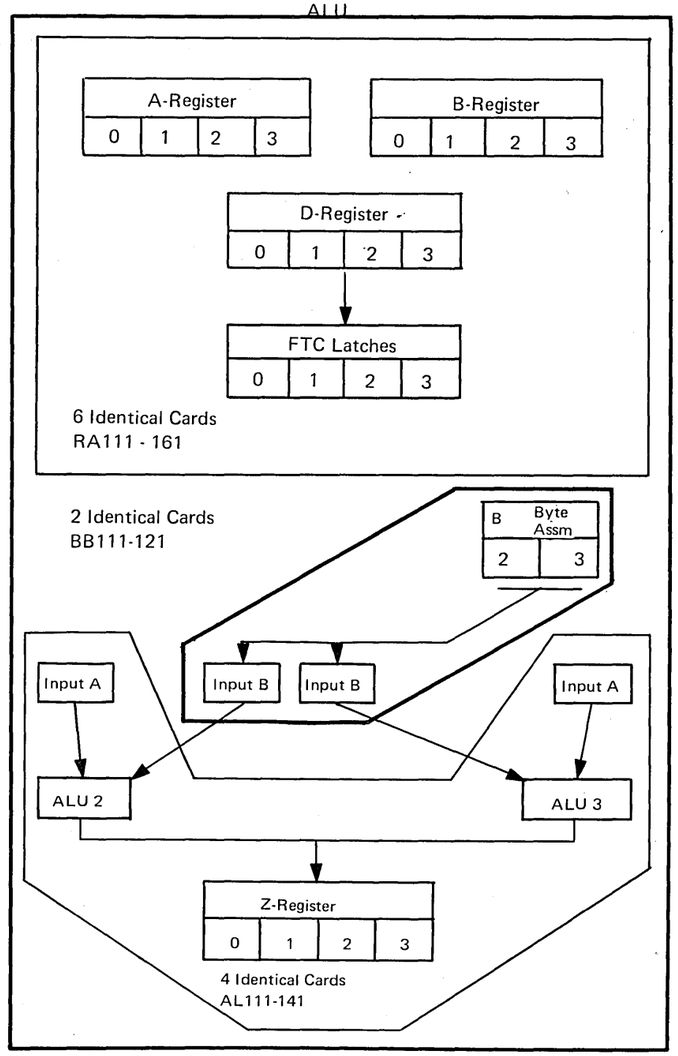
MAJOR FUNCTIONAL UNIT CARD ASSIGNMENTS

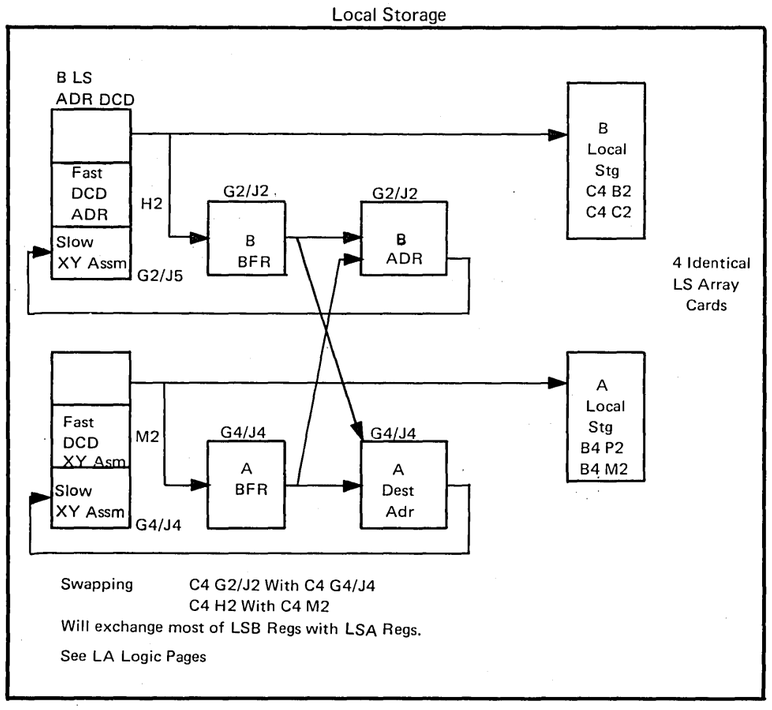












M-Reg

M-Reg Gating Lines
2 Identical Cards RM111, RM121

B Source Branch Decode
2 Identical Cards RM211, RM221

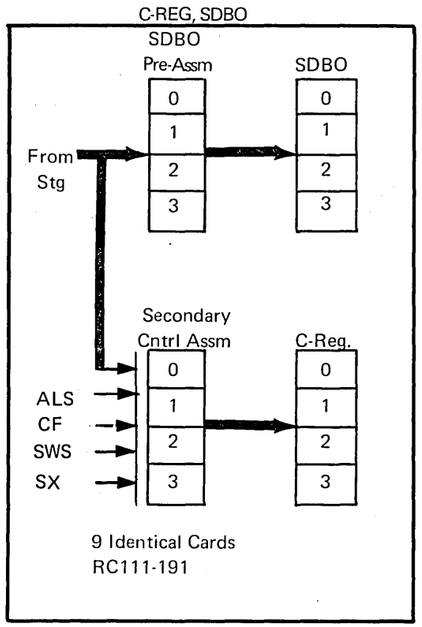
STG PROTECT

Stg Protect Stack
4 Identical Cards
MS 511-541

Clocks

All Clock Cards are identical. There is one on every A Gate Board. See KC 111-222

*If swapped, timing must be checked and changed if necessary.



MBA Storage Analyzation Test

Purpose:
To purge main and control storage of all single-bit cell failures.

1. Load MBA0 from Extended Diagnostics.
2. Check diagnostic printout, for MACH SER NUM,E C LEVEL, STORAGE xxxK.
3. Set system control panel rate switch to single-cycle.
4. Alter control-storage address FFF3 to 00, using system control-panel rotary switches. was 9A (154)
5. Press start key.

Note: When running MBA tests at or above EC 128658, refer to write-up in Test MBA7 for single-bit detection.

EXAMPLES:

A ERROR:

```
MBA   STORAGE ANALYZATION TEST.
SET SWITCH H TO A 1 TO LOOP AND DEPRESS START.

MACH SER NUM   012345      E.C.05       STORAGE 256K

               GATE     CHASSIS     CARD
REPLACE         01B      C4          S2
REPLACE         01B      B4          S2
EXCHANGE        01B      B4          R4

REPLACE OR EXCHANGE THE ABOVE CARDS AND RERUN THE TEST.
```

B NO ERROR:

```
MBA   STORAGE ANALYZATION TEST.
SET SWITCH H TO A 1 TO LOOP AND DEPRESS START.

MACH SER NUM   012345      E.C.05       STORAGE 256K

            GATE    CHASSIS     CARD

STORAGE IS IN SPEC

MONITOR MUST BE RELOADED TO CONTINUE AFTER RUNNING THESE STANDALONE TESTS.
```

Block Multiplexer-Feature Plug Cards

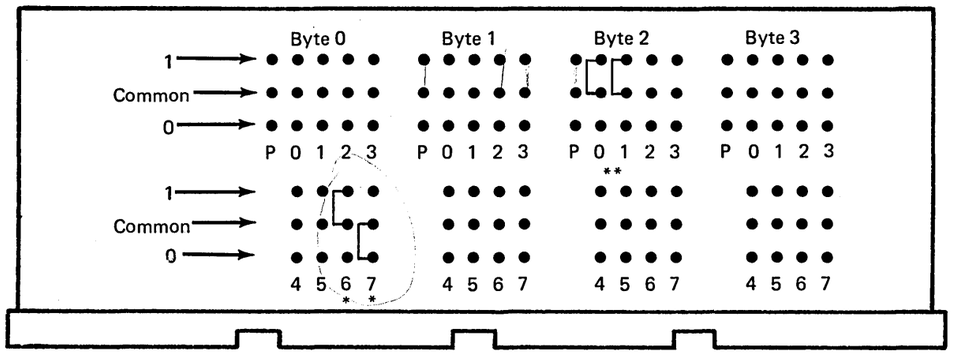
Each channel contains a plug card that is scanned at IMPL load time. Information pertaining to the type of devices on each channel is plugged on this card.

The block-multiplexer feature is installed on a 370/145 if the 370 microlisting contains routine GSLD.

IMPORTANT——This card may need to be plugged by the CE at installation time and also may need to be changed if new I/O devices are added to the channel.

The plug cards are located in the following positions. (All selector channels contain the plug even if the Block MPX feature is not installed.)

Channel 1—— A – B1S2 (no IFA installed)
Channel 2—— A – A2S2
Channel 3—— A – A3S2
Channel 4—— A – A2T2



**Asterisks indicate bits that are used in the example.

Plugging from common to the upper pin causes a logical one to be read. Plugging from common to the lower pin causes a logical zero to be read.

Addresses on the channel are blocked into groups of 16.

Two bits on the plug card are used with each group of 16 addresses.

Use the following table to determine which bits on the plug card are used with each address group.

Byte	Bits	Device Address
Byte 0	0, 1	00 - 0F
	2, 3	10 - 1F
	4, 5	20 - 2F
	6, 7	30 - 3F
Byte 1	0, 1	40 - 4F
	2, 3	50 - 5F
	4, 5	60 - 6F
	6, 7	70 - 7F
Byte 2	0, 1	80 - 8F
	2, 3	90 - 9F
	4, 5	A0 - AF
	6, 7	B0 - BF
Byte 3	0, 1	C0 - CF
	2, 3	D0 - DF
	4, 5	E0 - EF
	6, 7	F0 - FF

Example: Assume that selector channel 1 has the following:

TAPES—— (Addresses – 180 – 184) – 2803 with 4–2401's
DISK—— (Addresses – 130 – 137) – 3830 with 8–3330's

Using the table above and the plug card in A–B1S2: the addresses for tapes would be located in Byte 2 Bits 0, 1. The addresses for disks would be located in Byte 0, Bits 6, 7.

Next, determine how to plug these bits. Use the chart on the next page. Tapes are type 3 → byte 2. Bits 0 and 1 are plugged 1, 1. Disks are type 1 → byte 0. bits 6, 7 are plugged 1, 0. Be sure to plug odd parity for each byte.

The next page gives a brief description of the different types of devices that can be attached to the block-multiplexer channel.

The following description tells how to plug the two bits associated with each address group.

The block-multiplexer channel (even though it is a selector channel) operates much like a regular multiplexer channel. That is, many control units can be operating simultaneously. This necessitates the use of UCWs (unit control words). UCWs are located in control storage and contain all the information needed to keep a particular control unit operating (status, chaining information).

The microprogram needs to know just what kind of control unit is on the block multiplexer channel. Control units are of three types:

Type 1 Each device in the group requires an unshared UCW
Type 2 Each device in the group shares a UCW
Type 3 Each device in the group operates in selector-channel mode (block multiplexer not allowed)

The two bits assigned to each group of 16 addresses will be plugged as follows.

00 Continuance of shared UCW — Type 2
01 Shared UCW — Type 2
10 Unshared UCW — Type 1
11 Selector-channel Mode — Type 3

The plant plugs all address groups to the 1.1 state (type 3: selector-channel mode). At installation, it is necessary to change the plug cards if type 1 or type 2 devices are installed on your machine.

The following chart assists in determining what type of device is attached to the block-multiplexer channel.

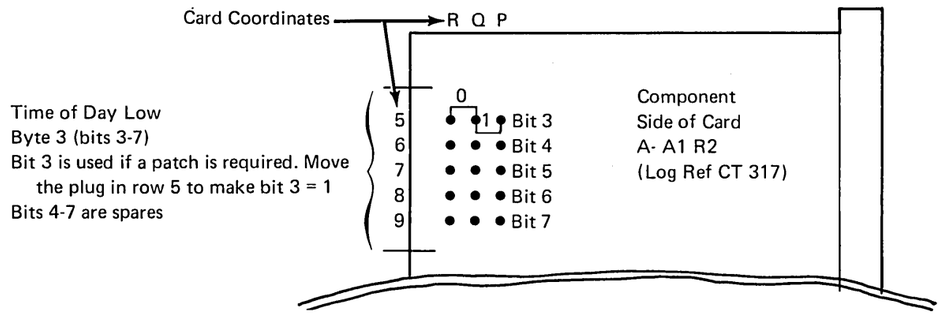
Type	Devices	Description
1	1442, 1443	Readers and Printers
1	2821, 2540, 1403	
1	2501, 2520	
1	3211	
3	2841-2311, 2303, 2321	Disk Files and Drum Storage
3	2314-2319, 2312, 2313, 2318	
1	2835-2305	
1	3830, 3300	
3	2415	Tapes
3	2803, 2401, 2402, 2420	
3	3420	
3	1419	Mag. Char. Reader
3	2701	Teleprocessing
3	2250 I	Displays
3	2840-2250 II	
3	2848-2260	
2	3270	

Note 1: The channel section of this manual gives a complete description of the block-multiplex channel operation It contains a description of the UCW tables in control. STG, the UCW Pool, etc.

Note 2: The 3270 Displays can have more than 16 devices on one control unit. All of these devices would share the same UCW. Assume 32 display units with addresses 00-1F
Byte 0 Bits 0, 1 would be plugged '01' - shared UCW
Byte 0 Bits 2, 3 would be plugged '00' - shared UCW
continued

Microprogram Temporary Fix Plug Card

A microprogram temporary fix (also called a patch) is a modification to the 370 microprogram. These patches are installed in the field on a as-needed basis. At IMPL time, the microprogram checks whether a patch is required. It does this by examining the TOD low byte 3 bit 3 (Ext Adr 3E) at card location 01A A1 R2. If this bit is plugged to a logical 1 (Bit on), a patch is required.



NOTE: Refer to the microprogram temporary fix procedure for a detailed description of installing a patch.

Memory Select Plug Card

This card is plugged in O1A-C1F2. The delay in the card is plugged so that the select to control storage occurs 57 ns after 0 time in the CPU. This card is set at the plant and should not have to be adjusted in the field unless the C1F2 is replaced. Refer to the adjustment procedure in the storage section of this manual (Logic Ref. MS011).

Selector-Channel Feature Plug Card

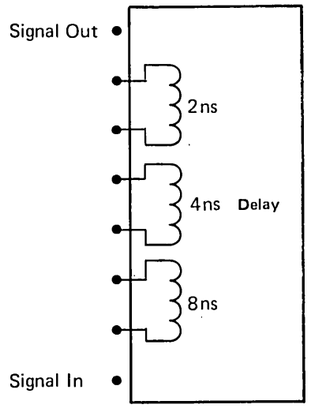
The selector-channel feature plug card has been modified to eight different part numbers. It is not necessary to plug this card.

P/N (EC 135342)	Feature Description
5858817	S x 1 unbuffered without S x 2
5858818	S x 1 buffered without S x 2
5859054	S x 1 unbuffered or IFA and S x 2 and not S x 3
5859055	S x 1 buffered or IFA and S x 2 and not S x 3
5859056	S x 1 unbuffered or IFA and S x 2, S x 3 and not S x 4
5859057	S x 1 buffered or IFA and S x 2, S x 3 and not S x 4
8216246	S x 1, S x 2, S x 3, S x 4 unbuffered
8216247	S x 1, S x 2, S x 3, S x 4 buffered

NOTE: Any subsequent engineering changes (EC) may have different part numbers.

Standard Clock Cards

A clock card is located in each board. The delay taps are adjusted so that the clock times match on each board.



The delay is plugged to obtain 0–14 ns of delay – (Logic KCXXX)

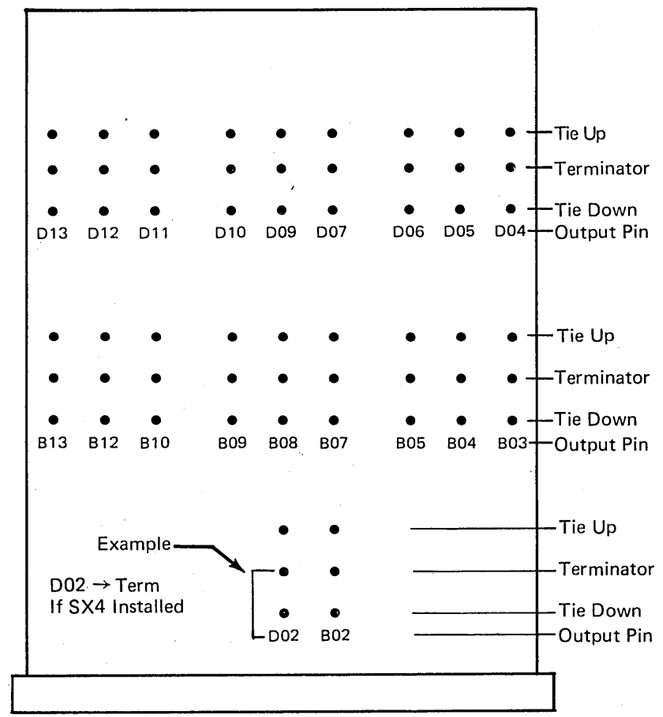
These clock cards are adjusted at the plant. If it is necessary to adjust clock time in the field, refer to the C-Reg and clock section in this manual.

Selector-Channel Feature Plug Card (Early Machine)

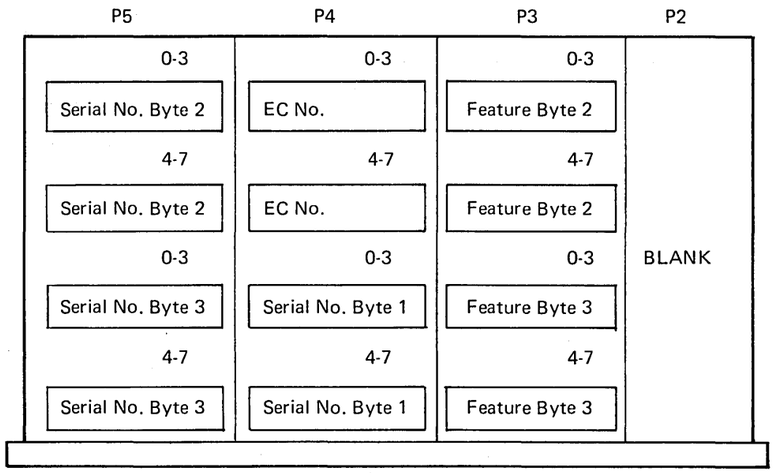
This card is plugged in O1A–A1U4. It is used to tie lines up or down depending on the particular channel being installed. It will be necessary to alter this card, only if a channel upgrade is made in the field (Logic Ref. GA511 and A0006).

Jumpers are installed as follows.

Feature	Installed	Not Installed
IFA/SX1	B03 → Term. B04 → Term. B13 → Term.	B03 → Tie Up B04 → Tie Down B13 → Tie Down
SX2	D11 → Term. D05 → Term B07 → Term.	D11 → Tie Up D05 → Tie Down B07 → Tie Down
SX3	B05 → Term. B10 → Term. D04 → Term	B05 → Tie Up B10 → Tie Down D04 → Tie Down
SX4	D02 → Term. B12 → Term. B02 → Term.	D02 → Tie Up B12 → Tie Down B02 → Tie Down
Buffered Channel	D06 → Tie Down	D06 → Tie Up



Serial Number, EC and Feature—Plug Card



Notes

1. Cards are located in A—B2P2 through P5. (ALD RD051 to RD072).
2. Feature card B2P3. This card is plugged to identify the features to the microdiagnostic program. It should not be necessary to alter this card in the field unless a feature is added or removed from the machine.
3. Card is plugged to the Hex value indicated below.

Byte 2, Bits 0 — 3 Main-Storage Sizes
1 = 112K 4 = 256K
2 = 160K 5 = 384K
3 = 208K 6 = 512K

Byte 2 Bits 4 — 7 Channel Information
Bit 4 — IFA
Bits 5 and 6 — Number of Channels (IFA counts as one channel)
00 = 1 10 = 3
01 = 2 11 = 4 Example: Machine with IFA & SX2 & word buffer (would be plugged B)
Bit 7 = Word Buffer

Byte 3 Bits 0 — 3 Misc. Features
Bit 0 Spare
Bit 1 3215 Matrix Printer
Bit 2 Second Selectric Printer
Bit 3 Spare

Byte 3 Bits 4 — 7 Misc. Features
Bits 4, 5, 6 Spares
Bit 7 Direct Control
EC Number and Serial Number Byte 1 (B2P4)

Notes:

1. EC number. The last two digits of the last EC installed is plugged in B2P4.
2. This card must be updated in the field after an EC is installed.
3. The 370 microprogram verifies that the disk and the machine are at the same level (otherwise the disk will not load).
4. The serial number byte 1 contains the first two digits of the serial number of the machine. The serial number is also verified when the disk is loaded.

Serial Number Bytes 2 and 3

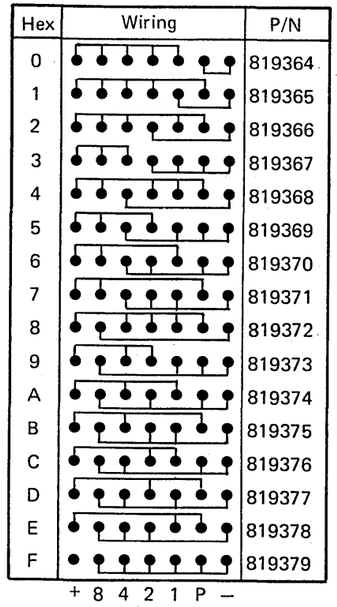
Notes:

1. The card in B2—P5 contains the last four digits of the machine serial number. The serial number is verified when the disk is loaded.

General Notes

1. Features are located in external Reg 02 bytes 2 and 3 and can be displayed on the console.
2. The EC number and serial number byte 1 are located in external Reg 12 bytes 2 and 3. After installing an EC, display this Ext Reg on the console.
3. Serial number bytes 2 and 3 are located in external Reg 16 and can be displayed on the console.

Wiring of the plug cards and the P/N of the pre-assembled hex configuration are shown below.



COMPLETE 370 MICROPROGRAM INDEX

(Routine Feature Sensitive)

* Helpful commentary of Routine, Instruction, or Operation.

OVERLAY 1	
AABA	Local Storage and External REG Maps
AACA	External Word Bit Designations
AADA	External Word Bit Designations (Cont.)
DACC	Validate Control Storage
OVERLAY 2	
DGNK	Go-No-Test
DGNR	Go-No-Test
DGN0	Go-No-Test
DGN1	Go-No-Test
DGN2	Go-No-Test
DGN3	Go-No-Test
DGN4	Go-No-Test
DGN5	Go-No-Test
DGN6	Go-No-Test
D999	Go-No-Test
OVERLAY 3	
DACS	Validate Control Storage
OVERLAY 4	
ABKA	K Addressable Index Addressable C.S. area (Cust's. Avail. Stor - last M.S. Addr.)
CSRD	System Reset Diagnostic (Looping Instruction for all of Stor. or Single Adr.)
CSTS	Storage SCAN and Clear
GAAI	Instruction Cycles*
GARR	Execute RR Instruction*
GARS	Execute RS/SI Instructions*
GARX	Execute RX Instructions*
GASS	Execute SS Instructions
GAXF	CVT. Binary & MPY Long I Cycles Completion
GBAB	RR Branches and Set Prg. Mask
GBAC	RX Branches and Execute
GBAD	Branch On Index
GCAE	RR Signed Loads
GCAG	RR Load and Fixed Point Arithmetic
GCBA	Half Store, LA, Store and Insert Char.
GCBE	RX Store
GCBH	ADD/SUB Logical, Start Multiply Divide
GCBJ	Load Multiple
GCBK	Store Multiple
GCCB	Convert to Binary
GCCD	Convert to Decimal
GCCR	Load and Store Mult. Control Registers (Control REG's MAP- F480 to F4BC)
GCDV	Fixed Point Divide*
GCMH	Fixed Point Half Word Multiply*
GCMP	Fixed Point Full Word Multiply*
GCSH	Shifts*
GDAA	Decimal I Cycle Completion
GDAO	Pack, Unpack, Move with Offset
GDAR	Decimal Arithmetic
GDMD	Decimal Multiply and Divide*
GDSR	Decimal Shift and Round
GEAA	RR and RX Logical Operations
GEAB	Immediate Logical Operations
GELG	SS Logical Operations
GELL	Long Move and Compare Logical*
GEMO	Insert Store and CMP. LOG. Under MASK
GEMV	SS Moves
GETE	SS Translates and Edits
GFAR	FLT PT Arithmetic*
GFDV	FLT PT Divide*
GFEP	Extended FLT PT Exponent Routine*
GFEX	Extended FLT PT Arithmetic Start*
GFHV	FLT PT Half*
GFLR	Floating Point Rounding Loads
GFMD	FLT PT Multiply and Divide Start*
GFME	Extended FLT PT Multiply Routines*
GFMF	FLT PT Multiply Start*
GFML	FLT PT Multiply Loops*
GFSL	FLT PT Signed Loads
GFST	FLT Point Store and Convert to Decimal
GGAA	TST Under MSK, Move Char, TST and Set
GGAB	Set System Mask and Load PSW
GGAD	Diagnose Instruction (C.S. Patch Information)
GGB2	'B2' Instruction*
GGCM	Monitor Call Instruction
GGCS	Set and Store Time of Day Clock
GGDC	Direct Control (Read/Write Direct)*
GGID	Store CPU/Channel ID's
GGPA	Set, Insert Keys, Supervisor Call
GGSL	
GGSM	
GGST	Translation TRAP
GGSX	Translation Exception
GHRM	Routine For Retry of Storage Words
GHRS	Routine For Retry of Second Error
GHRT	Retry Routine
GHEC	Machine Check Trap Routine*
GHEI	Machine Check Interruption Handling
GHLC	Log Routine (Log Buffer MAP)
GHYI	Emulator Interrupts
GHYP	DRI and RCCW Operations
GHLI	Machine Check Interrupt Code Routine*
GICM	Common Interrupt Routine
GIPW	Store and Load PSW Routines (BC & EC Mode PSW Data)
GITM	Interval Timer Routines
GKAD	Alter/Display Routine* (Instructions)
GKCC	Set IC and Console Traps*
GKDI	Doc. Console Start I/O*
GKDJ	Doc. Console Test I/O and Int.*
GKDT	Doc. Console Share Routine*
GKEL	Alter/Display Expanded Local Storage*
GKLA	Alter/Display Logical Address Translation
GKTM	Console Printer/Keyboard Test Mode (Instructions)
GLAA	I/O Operations*
GMCR	Multiplex Chain Routine*
GMCW	Multiplex Channel Unit Control Words
GMDR	Multiplex Channel Data Routine*
GMGR	Multiplex General Routine*
GMLO	Multiplex Channel Log-Out
GMMS	Main Status Handling Routine
GMPL	Initial Program Load*
GMSR	Share Trap and IPL
GMSW	Multiplex CSW Store Routine
GPAA	I-Cycles—IFA (UCW MAP - IFA Adr.)
GPBA	CCW Read Out—IFA
GPBB	OP Decode—IFA
GPBC	Search Ops—IFA
GPBD	Read Ops—IFA
GPBE	Write Ops—IFA
GPBF	Space Count Op—IFA
GPBH	Sub-routines—IFA
GPBK	Control Ops—IFA
GPCA	Error Sets—IFA
GPCC	Machine Check—IFA
GPCE	Index Trap—IFA
GPCG	Error Trap—IFA
GPD0	IFA CE In-Line Test Entry and Set Up (2319 In-line Test Desc.)
GPD1	IFA CE IN-line Error Posting + Resets
GPD2	IFA CE IN-line Trap Handling Routines
GQER	Program Event Rec.*
GQET	
GRST	System Reset*
GSER	SELR CHNL Error Routine*
GSES	SELR CHNL Exceptional Status
GSHT	SELR CHNL Halt Routine
GSIN	SELR CHNL Interrupts
GSLD	Block Multiplex UCW Routine*
GSTB	BLKMPX UCW Tables
GSOP	SELR Channel I/O Ops
GSSS	SELR CHNL Status Store
GSTR	SELR CHNL Chaining Traps
GTRC	Op Code Count Routine
GTRD	Trace 0 Diagnose Ops
GTRT	Trace Buffer Load Routine
GTRU	Trace 1 Diagnose Ops
HAUM	14XX Add and Subtract
HBVW	14XX High Speed Ops
HCOM	14XX Compare
HICY	14XX I Cycles
HIND	14XX Indexing
HIOA	14XX I/O Address
HMCP	14XX CPU Move Operations
HMIO	14XX Move I/O
HMOD	14XX Modify Address
HOPD	14XX Op Decode Routine
HOUT	14XX Exit Routine
HZAS	14XX Reset Add and Subtract
ZZZZ	Control Transfer (DISC to Control STG.) —(Check Sum of Coreload)

REMEMBER

The INDEX is in the front of the manual

READER'S COMMENT FORM

3145 Processing Unit
Theory-Maintenance

SY24-3581-1

- Are you able to find information in this publication easily?
- Are the illustrations adequate, clear, and concise?
- Which of the following are included in your occupational responsibilities?
  - System operating. . . . . .
  - Programming. . . . . .
  - Supervising. . . . . .
  - Instructing. . . . . .
  - Service representative . . . . . .
  - Other. . . . . .
- We would appreciate your comments; please give specific page and line references where appropriate.

Reply Requested?

Yes ☐

No ☐

NAME . . . . . .

STREET . . . . . .

CITY . . . . . .

STATE . . . . . ZIP CODE . . .

READER'S COMMENT FORM

3145 Processing Unit
Theory-Maintenance

SY24-3581-1

- Are you able to find information in this publication easily?
- Are the illustrations adequate, clear, and concise?
- Which of the following are included in your occupational responsibilities?
  - System operating. . . . . .
  - Programming. . . . . .
  - Supervising. . . . . .
  - Instructing. . . . . .
  - Service representative . . . . . .
  - Other. . . . . .
- We would appreciate your comments; please give specific page and line references where appropriate.

Reply Requested?

Yes ☐

No ☐

NAME . . . . . .

STREET . . . . . .

CITY . . . . . .

STATE . . . . . ZIP CODE . . .

SY24-3581-1

YOUR COMMENTS, PLEASE . . . .

Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM branch office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N.Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . . .

IBM Corporation
P.O. Box 6
Endicott, N.Y. 13760

Attention: Product Publications, Dept. K10

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

SY24-3581-1

YOUR COMMENTS, PLEASE . . . .

Your answers to the questions on the back of this form, together with your comments, will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM branch office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N.Y.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . . .

IBM Corporation
P.O. Box 6
Endicott, N.Y. 13760

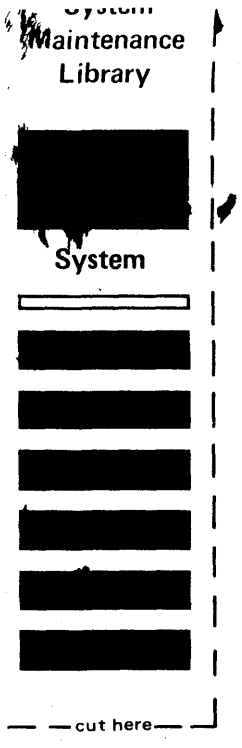
Attention: Product Publications, Dept. K10

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)



3145 Processing Unit Printed In U.S.A. SY24-3581-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)