IBM

Field Engineering Theory of Operation

Restricted Distribution

This manual is intended for internal use only and may not be used by other than IBM personnel without IBM's written permission.

Specifications contained herein are subject to change without notice. Revisions and/or supplements to this publication will be issued periodically.

2030 I/O Control

System/360 Model 30

2030 I/O Control
System/360 Model 30

Y24-3362-1

IBM®

Field Engineering

Theory of Operation

2030

I/O Control

System/360 Model 30

PREFACE

This manual contains information about the Input/Output controls of the System/360 Model 30. A knowledge of the IBM 2030 Processing Unit is assumed. Information pertaining to the 2030 Processing Unit is contained in a companion Field Engineering manual, 2030 Processing Unit, Form Y24-3360. Maintenance information is contained in the Field Engineering Maintenance Manual, 2030 Processing Unit, Form Y24-3390.

Fifth Edition (November 1966)

This edition, Y24-3362-1 contains minor changes to, but does not obsolete the previous edition, Y24-3362-0. Changed text is marked with a vertical line in the margin; changed figures are marked with a large bullet to the left of the figure title.

Copies of this and other IBM publications can be obtained through IBM Branch Offices. This manual has been prepared by the IBM Systems Development Division, Product Publications, Department 171, P.O. Box 6, Endicott, New York 13760. Address comments concerning the manual to this address.

SECTION 1. COMPREHENSIVE INTRODUCTION. 1-1

Input/Output Control. 1-1
Channels. 1-2
Input/Output Instruction Format 1-7
I/O Control Words 1-8
Functional Units. 1-14
Theory of Operation 1-15

SECTION 2. I/O INTERFACE 2-1

Interface Lines 2-2
Scan Controls 2-3
Outbound Tag Lines. 2-10
Inbound Tag Lines 2-14
Interlock Lines 2-16
Special Controls. 2-18
Initial Selection Sequence. 2-21

SECTION 3. MULTIPLEXOR CHANNEL 3-1

Comprehensive Introduction. 3-1
Data Flow 3-3
Functional Units. 3-3
MPX Storage (Subchannel Storage). . . . 3-5
MPX Channel Microprogram. 3-6
I/O Interface Control 3-7

THEORY OF OPERATION 3-15

Start I/O (I/O-Commands). 3-15
Multiplexor Share-Cycle 3-16
I/O Interrupt 3-27
PSW Store 3-28
CSW Store 3-29
New PSW Load. 3-33
CCW Flag Operations 3-33
Start I/O Transfer-in-Channel (TIC) Command 3-42
Initial Program Load (IPL). 3-42
Test I/O. 3-44
Halt I/O. 3-45
Test Channel. 3-46
Error Handling. 3-47

SECTION 4. SELECTOR CHANNELS 4-1

Comprehensive Introduction. 4-1

FUNCTIONAL-UNITS. 4-4

GA Register 4-4
GB Controls 4-5
GC and GD Registers 4-6
GE Register 4-6
GF Register 4-6
GG Register 4-7
GH Controls 4-7
GI Bus. 4-8
GJ Bus. 4-8
GJ Assembler. 4-9
GK Register 4-9
GO Register. 4-10
GR Register. 4-10
GR, GS, and GT Busses. 4-11
GU and GV Registers. 4-12
GW, GX Registers 4-12
Modifier 4-13
Clock (Selector Channel) 4-15

THEORY OF OPERATION. 4-18

Start I/O; Commands to Control Units . 4-18
I/O Interrupt Execution. 4-40
CCW Flag Operations. 4-41
Transfer in Channel (TIC). 4-50
Test I/O 4-51
Halt I/O 4-53
Test Channel 4-55
Initial Program Load (IPL) 4-56
Error Detection. 4-59

SPECIAL FEATURES 4-60

Memory Protect 4-60

SECTION 5. CONSOLE CHANNEL. 5-1

Comprehensive Introduction 5-1

FUNCTIONAL UNITS 5-18

IBM 1051 Attachment Clock. 5-18
Controls In-TT 5-24
Controls Out-TA. 5-25
Read-Write Register. 5-26
Share-Request Controls-Poll Control. . 5-27
IBM 1051 to 2030 Interface 5-32
Parity Checking. 5-43
Parity Bit Generators. 5-44

THEORY OF OPERATION. 5-45

Microprograms. 5-45
Write Operation. 5-53
Read Reader-2 Operation. 5-69
Read Inquiry Operation 5-73
Ready Share-Request Operation. 5-84
Sense Operation. 5-86
Chained Data Operation 5-86
Command Chaining Operation 5-87
Command Chaining with Command Immediate. 5-88

POWER SUPPLY AND INTERLOCKS. 5-90

SPECIAL FEATURES 5-91

Control Alarm Operation. 5-91

SHARED I/O PANEL 5-92

REVIEW QUESTIONS 5-96

Review Question Answers. 5-99

SECTION 6. DIRECT CONTROL FEATURE . . 6-1

Introduction 6-1
Write Direct 6-1
Read Direct. 6-1
Definitions of Interface Lines 6-3
Application. 6-4
Data Flow 6-5

SECTION 7. APPENDIX 7-1

INDEX. X-1

INPUT/OUTPUT CONTROL

- The IBM 2030 communicates with I/O devices through Channels.
- The Channel is a physical part of the CPU, but functionally separate.
- The Channels are connected to, and communicate with, the I/O control units via a standard interface.
- Each I/O device attached to a channel must have an associated control unit.

The transfer of information between the 2030 processing unit and an I/O device is by I/O control circuits and micro programs. These control circuits and micro programs, together, are called channels. Existing channel circuitry is located in the CPU, but in actual operation, the channels are completely separate units.

A channel is connected to an I/O control unit by a cable containing 34 signal lines. The 34 signal lines, and the sequence in which they become active and inactive, constitute the standard interface. By using a standard set of lines and signal sequences, the same I/O control units and devices can be used on most models of System/360.

When an I/O device is attached to the IBM 2030, a control unit must be used. The control unit can be separate from, or contained within the I/O device. Each device could 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 a difference in the rate of flow of data, or the time of occurrence of events, when transferring information between the 2030 and an I/O device. Figure 1-1 shows the concept of data flow between the CPU and I/O devices.

The channel itself can have many I/O devices physically connected to it but only one I/O device can be operational with the channel at any one time. To ensure that only one control unit and device is selected, each is given a unique address. This address is assigned by the customer and wired on an address card by the CE at the time of installation.

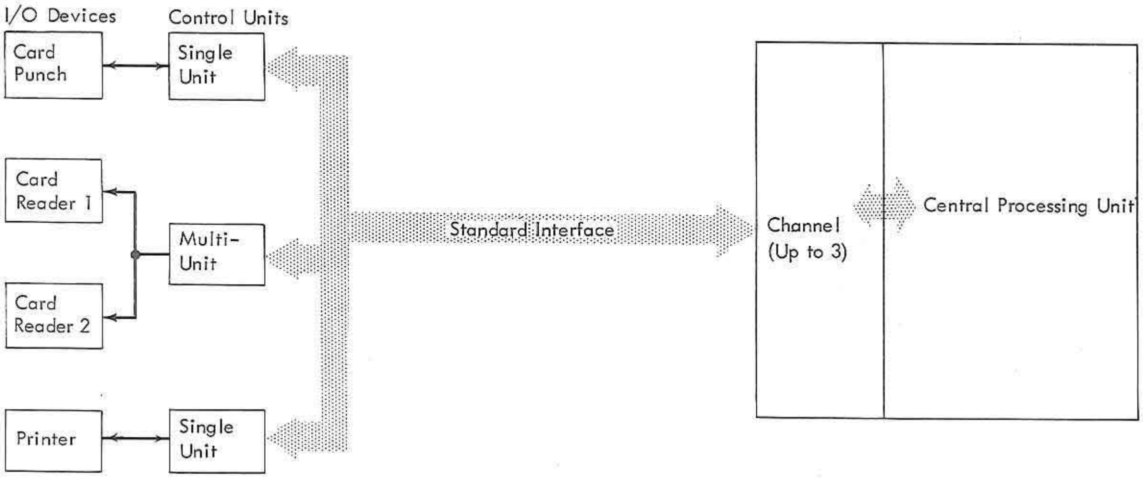


Figure 1-1. CPU to I/O Device Data Path

CHANNELS

- The IBM 2030 has two basic types of channels:

    1. Multiplexor

    2. Selector (one or two)

- A console subchannel is also available for the 1050 attachment.

The two channel types used by the 2030 perform the same function. They differ however, in the method by which they obtain their objectives:

1. The multiplexor channel is basically a microprogram routine. It can sustain operations with several I/O devices at a time on a time shared basis.

2. The selector channel is a high-speed channel. It operates with one I/O device until a complete data record has been transferred. It does not use microprogram routines to accomplish data transfers.

The console subchannel operates only with the 1050 attachment. It uses the microprogram routines of the multiplexor channel to accomplish many of its functions. The circuits for the console are also located in the 2030 frame.

CHANNEL INSTRUCTIONS

- The CPU uses four instructions to communicate with a channel:

    Start I/O
    Test I/O

    Halt I/O

    Test Channel

The start I/O instruction is used by the CPU to initiate all I/O operations. Test I/O is the CPU's means of obtaining information regarding the status of a particular I/O control unit and device. The halt I/O instruction is issued by the CPU to disconnect a particular I/O control unit and device from the channel. The test channel instruction involves only a channel and the CPU. It is used to determine the status of the channel.

These four instructions specify only the address of the channel, I/O control unit, and device. For the test I/O, halt I/O, and test channel instructions, this is sufficient. The channel must, however, obtain operating information to execute a start I/O instruction. It gets this information from a channel address word (CAW) at location 72 (hex 48) and a channel command word (CCW), located in CPU main storage. The CAW consists of:

1. A storage protection key for the I/O operation.

2. A command address, designating the location in main storage where the associated first CCW is located.

The CCW consists of:

1. The specific command to be executed by the I/O unit.

2. A data address that specifies the area of core storage to be used for the data transfers.

3. A count that specifies the number of data transfers, in the form of bytes, that are to be made.

4. Five flag bits that may extend or modify the basic command.

A separate CCW is necessary for each command the channel is to perform. For example, a read, write, rewind instruction to a tape drive is actually three separate commands and therefore would require three separate CCW's to execute the commands.

I/O INSTRUCTIONS

- The start I/O instruction selects the channel, control unit, and device.
- The channel, control unit, and device are connected electrically by the standard interface.
- After executing the instruction, the control unit presents ending status and causes an interrupt.

When the CPU issues a start I/O instruction to a channel, the channel selects the I/O device and control unit by executing an initial selection sequence.

The channel issues the I/O command to the control unit during the initial selection sequence, and electrically connects the control unit to the interface. The control unit decodes the command and raises control lines to the I/O device.

A control unit that is electrically connected to the channel can operate in either burst mode, or data interleave (byte) mode. When operating in burst mode, the control unit remains electrically connected to the channel after initial selection, until a complete data record has been transmitted. When operating in byte mode, the control unit disconnects from the channel after initial selection and between each data byte. When a control unit is connected to a selector channel, it always operates in burst mode. This is because a selector channel is a high speed channel that operates with one I/O device at a time.

When a control unit is connected to a multiplexor channel, it operates in either burst or byte mode, depending on the design of the control unit. Normal data transfer from the I/O unit to the multiplexor channel is in byte mode. The control unit may force burst mode operation. Normally, the low speed control units operate in byte mode and high speed units in burst mode. Intermediate speed units have a switch that can be set to either byte or burst mode. The selector channel overrides the switch setting and forces burst mode. When the channel is operating in byte mode, it may service several I/O units on a time-shared basis. It must disconnect from one I/O device before it can operate with another device. This means that the control unit must initiate a selection sequence each time it requires service from the channel.

When a channel and control unit have transferred the number of bytes required by the I/O command, the control unit sends its channel-end and device-end status to the channel. These may be sent together or separately, depending on the control unit.

The control unit always sends channel-end status to the channel at the end of the data transfer portion of a command. Channel-end may be alone or accompanied by other status bits. Device-end is sent when the control unit is in a position to accept another command.

When a channel receives ending status, it may either accept the status or it may command the control unit to hold (stack) the status. Once the channel responds to channel-end status with either an acceptance response or a stack response, the control unit disconnects from the channel. Figure 1-2 illustrates a basic selector channel data transfer.

When a channel accepts the control units device end status, it will either obtain new operating information (a new CCW), or initiate on I/O interrupt of the current program.

If the channel must obtain a new CCW because of the presence of the command chaining bit in the CCW, it must wait until it receives device-end status from the I/O device control unit.

An I/O interrupt causes the status of the interrupting channel and I/O unit to be placed in the channel status word (CSW) and the execution of the I/O interrupt program to begin.

Note: The CPU can begin the execution of an I/O interrupt only after it has completed the execution of the instruction it is currently executing when it receives the interrupt.

Refer to Figure 1-3 for a flowchart of the four channel and I/O instructions.

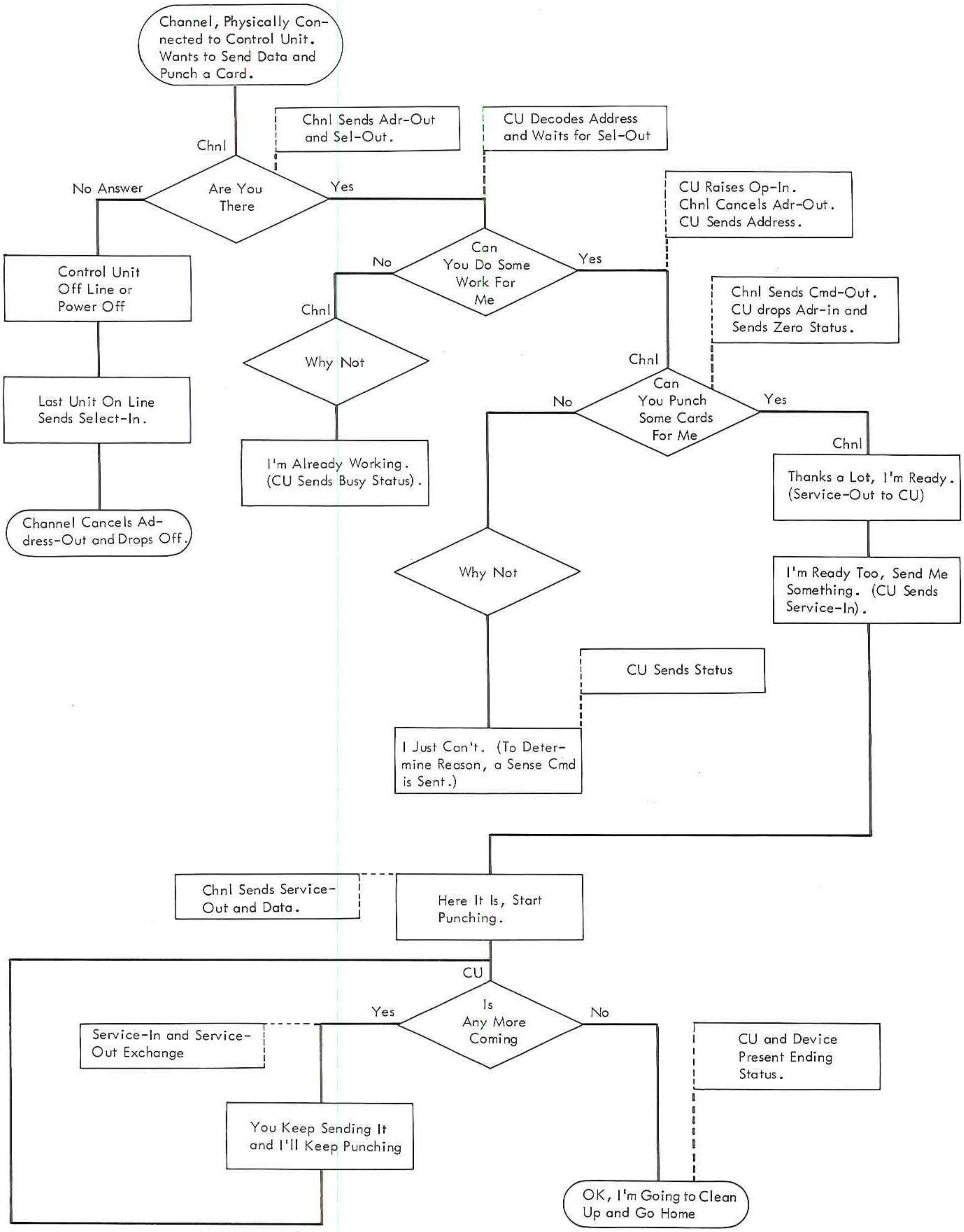

Channel, Physically Connected to Control Unit. Wants to Send Data and Punch a Card.
Chnl
Chnl Sends Adr-Out and Sel-Out.
CU Decodes Address and Waits for Sel-Out
No Answer
Are You There
Yes
Control Unit Off Line or Power Off
Last Unit On Line Sends Select-In.
Channel Cancels Address-Out and Drops Off.
CU Raises Op-In. Chnl Cancels Adr-Out. CU Sends Address.
No
Can You Do Some Work For Me
Yes
Chnl
Why Not
I'm Already Working. (CU Sends Busy Status).
Chnl Sends Cmd-Out. CU drops Adr-in and Sends Zero Status.
Chnl
No
Can You Punch Some Cards For Me
Yes
Chnl
Thanks a Lot, I'm Ready. (Service-Out to CU)
I'm Ready Too, Send Me Something. (CU Sends Service-In).
Why Not
CU Sends Status
I Just Can't. (To Determine Reason, a Sense Cmd is Sent.)
Chnl Sends Service-Out and Data.
Here It Is, Start Punching.
CU
Yes
Is Any More Coming
No
Service-In and Service-Out Exchange
You Keep Sending It and I'll Keep Punching
CU and Device Present Ending Status.
OK, I'm Going to Clean Up and Go Home

Figure 1-2. Basic Selector Channel Operation

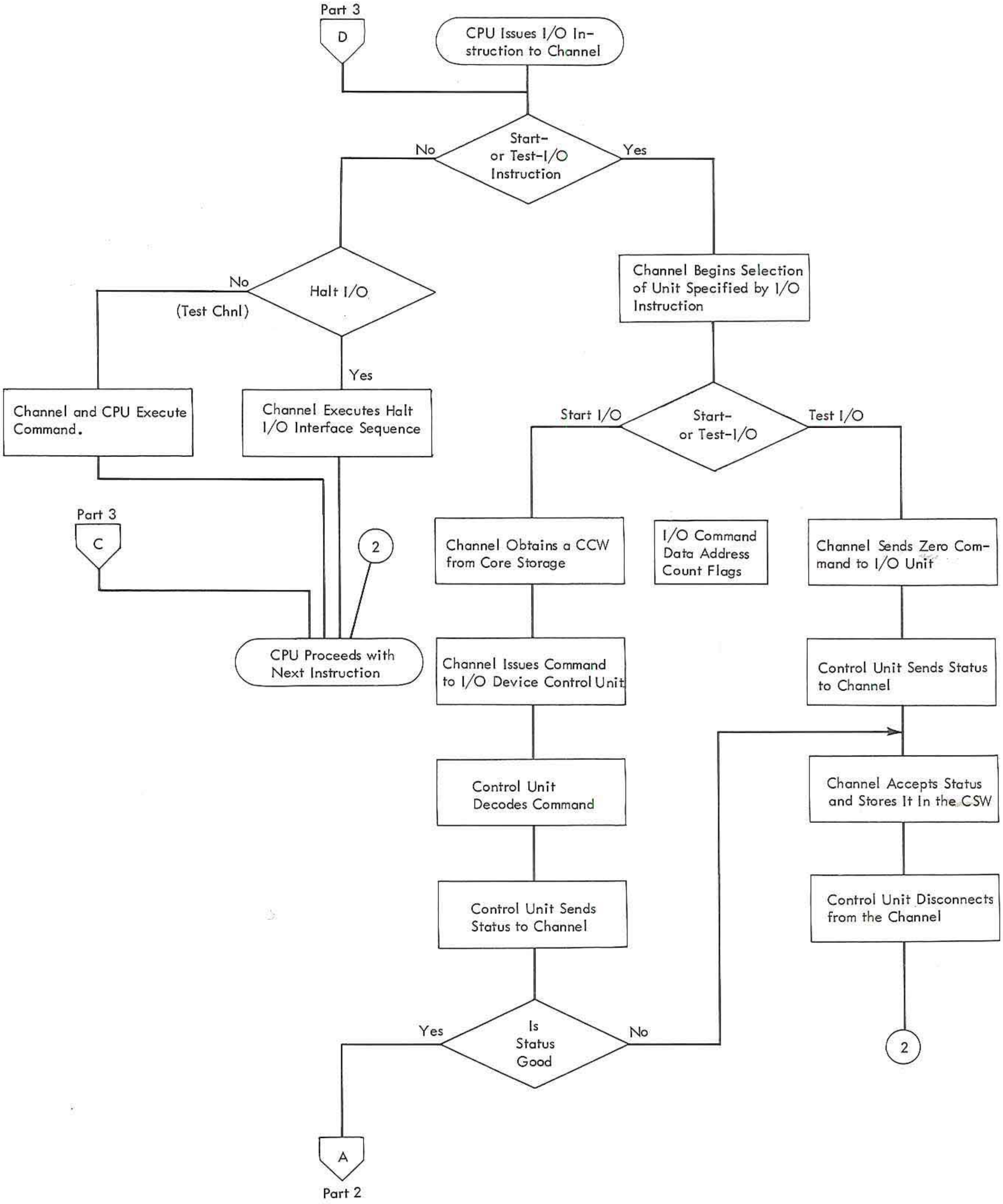

Part 3
D
CPU Issues I/O Instruction to Channel
Start- or Test-I/O Instruction
No
Yes
Halt I/O
No
(Test Chnl)
Yes
Channel Begins Selection of Unit Specified by I/O Instruction
Channel and CPU Execute Command.
Channel Executes Halt I/O Interface Sequence
Start I/O
Start- or Test-I/O
Test I/O
Part 3
C
2
Channel Obtains a CCW from Core Storage
I/O Command
Data Address
Count Flags
Channel Sends Zero Command to I/O Unit
CPU Proceeds with Next Instruction
Channel Issues Command to I/O Device Control Unit
Control Unit Sends Status to Channel
Control Unit Decodes Command
Channel Accepts Status and Stores It In the CSW
Control Unit Sends Status to Channel
Control Unit Disconnects from the Channel
Yes
Is Status Good
No
2
A
Part 2

Figure 1-3. I/O Instructions (Part 1 of 3)

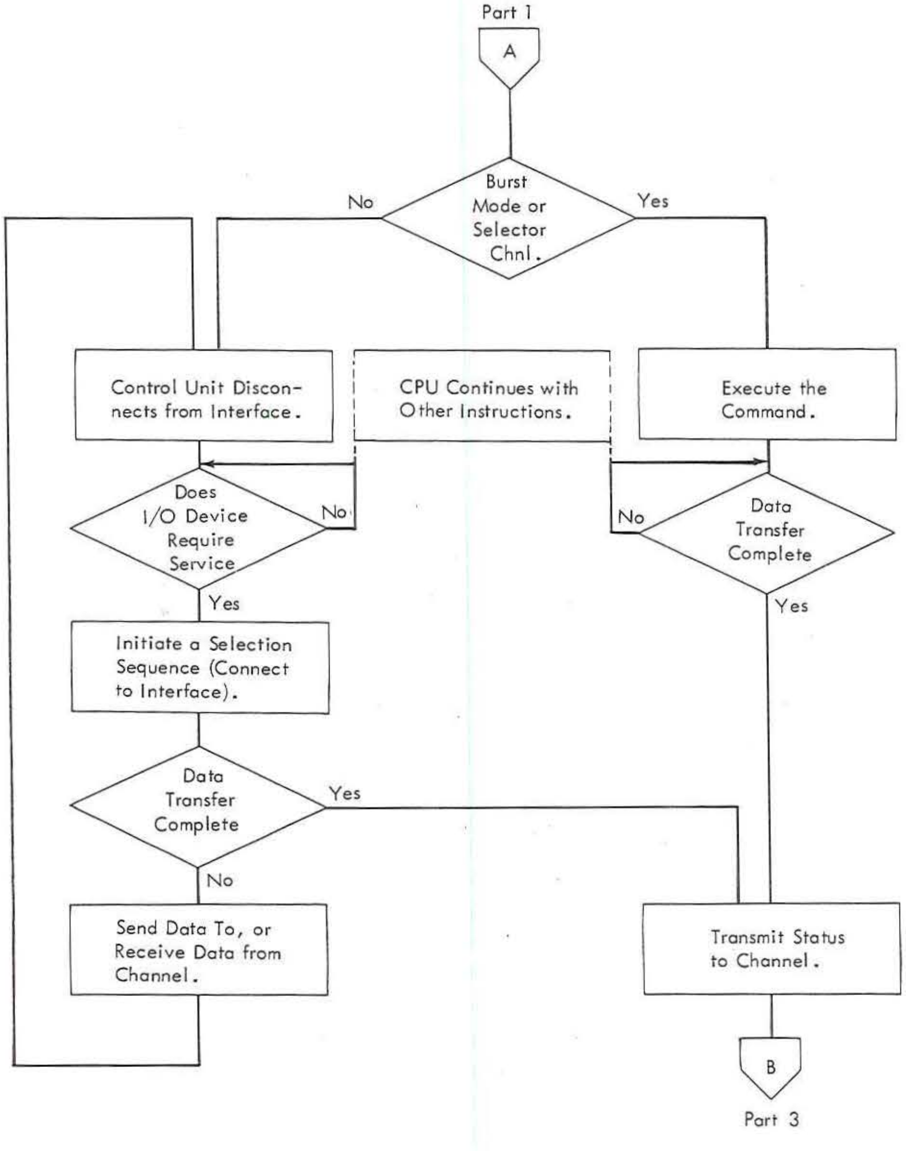

Part 1
A
Burst Mode or Selector Chnl.
No
Yes
Control Unit Disconnects from Interface.
CPU Continues with Other Instructions.
Execute the Command.
Does I/O Device Require Service
No
Yes
No
Data Transfer Complete
Yes
Initiate a Selection Sequence (Connect to Interface).
Data Transfer Complete
Yes
No
Send Data To, or Receive Data from Channel.
Transmit Status to Channel.
B
Part 3

Figure 1-3. I/O Instructions (Part 2 of 3)

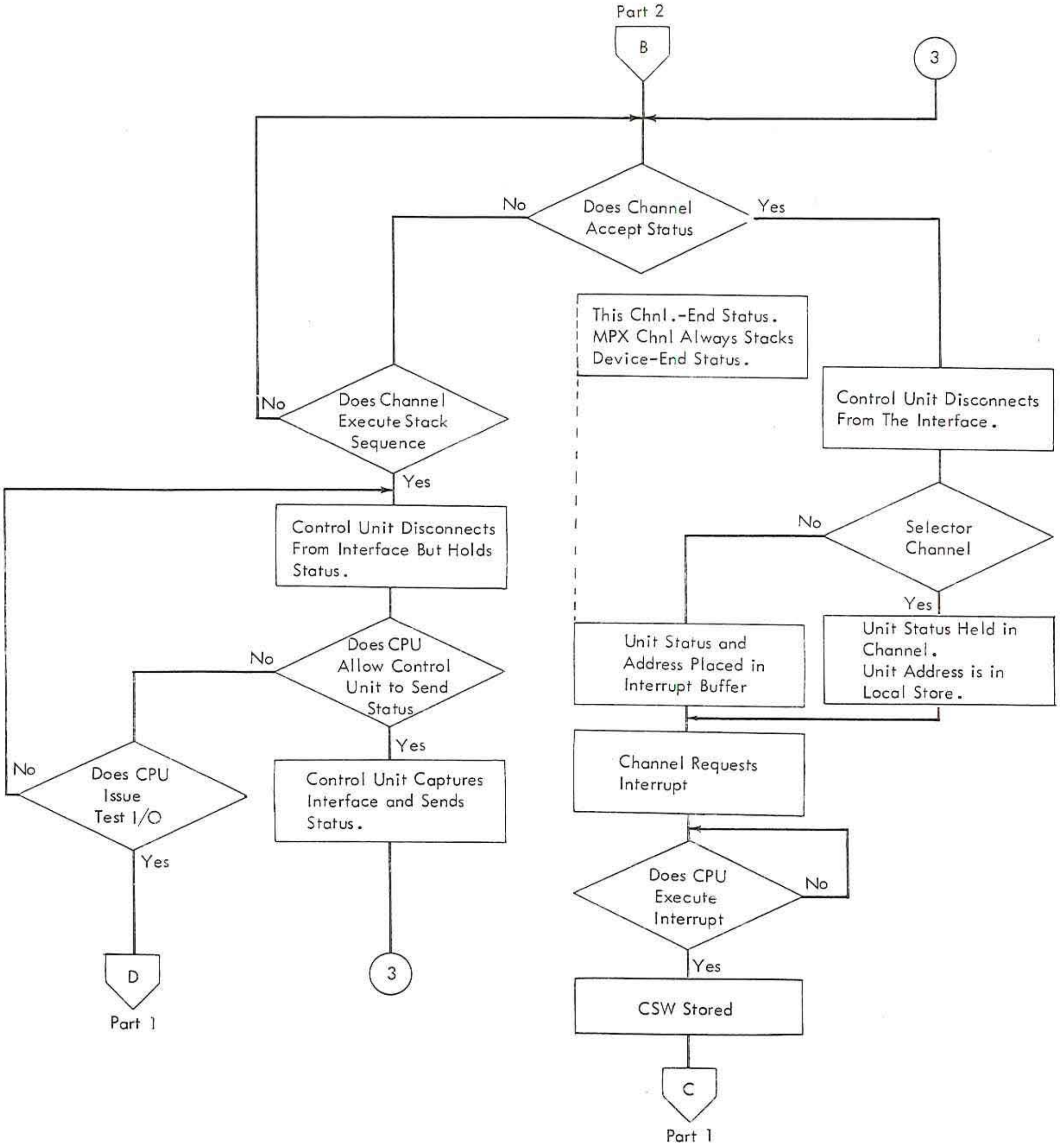

Part 2
B
3
No
Does Channel Accept Status
Yes
This Chnl.-End Status. MPX Chnl Always Stacks Device-End Status.
No
Does Channel Execute Stack Sequence
Control Unit Disconnects From The Interface.
Yes
No
Selector Channel
Control Unit Disconnects From Interface But Holds Status.
Yes
Unit Status and Address Placed in Interrupt Buffer
Unit Status Held in Channel. Unit Address is in Local Store.
No
Does CPU Allow Control Unit to Send Status
No
Does CPU Issue Test I/O
Yes
Control Unit Captures Interface and Sends Status.
Channel Requests Interrupt
Does CPU Execute Interrupt
No
Yes
Yes
D
Part 1
3
CSW Stored
C
Part 1

Figure 1-3. I/O Instructions (Part 3 of 3)

INPUT/OUTPUT INSTRUCTION FORMAT

- The four I/O instructions have the SI format in a 32 bit word.

The I/O instruction format is as follows:

0 7	8 15	16 19	20 31
Op. Code	Ignored	B1	D1

B1+D1=I/O Channel and Device Address

BITS 0-7: An eight-bit field giving the operation code of the instruction as follows:

10011100	(9C)	Start I/O
10011101	(9D)	Test I/O
10011110	(9E)	Halt I/O
10011111	(9F)	Test channel

BITS 8 - 15: These eight bits are not used and are ignored.

BITS 16 - 19: A four-bit field which designates a 32-bit general purpose register in local storage. A zero in the B1 field is used to indicate the absence of a base register component, and a value of zero is used in forming the address regardless of the contents of general register zero.

BITS 20 - 31: A 12-bit field which holds a literal value known as the displacement.

The contents of the general purpose register designated by B1 is added to the value of D1. This forms the channel and unit address. In the resulting 32-bit summary:

Bits 0-20: Ignored

BITS 21 - 23: These three bits for channel address are assigned as follows:

000	- Multiplexor Channel
001 - 010	Selector Channels 1 & 2 (011-110, Selector Channels 3-6, are invalid on the 2030)
111	- Unused

BITS 24 - 31: These eight bits define the unit number to a maximum of 256 units, each unit having a unique address.

I/O CONTROL WORDS

- Four control words are common to I/O operations:

1. Channel Address Word (CAW) }
2. Channel Command Word (CCW) } Start I/O Only
3. Channel Status Word (CSW)
4. I/O New and Old Program Status Words (PSW)

COMMAND ADDRESS WORD (CAW)

- The CAW is a 32-bit word located in main storage 72 - 75 (48-4B hex).
- It provides the address of the first Channel-Command Word (CCW) to be used in the I/O operation.
- It provides the I/O protection key to be used in the I/O operation.
- It is automatically read out when the instruction start I/O is given.

The CAW is a 32-bit word stored in the fixed location 72-75 in main memory. The CAW is loaded into the channel by the instruction start I/O and specifies the address of the first control word. The CAW has the following format:

0 3	4 7	8 31
I/O Protection Key	Zeros	CCW Address

BITS 0 - 3: A four-bit I/O protection key which is compared with the four-bit storage key obtained from the storage-protect stack when a main storage location is addressed.

If the protection key and the storage key do not compare, the protection key is not zero, and the operation is INPUT, the address is violating a protected area. This causes the channel to generate an I/O interrupt, with the protection-check indicated in the channel-status of the associated CSW.

BITS 4 - 7: This four-bit field must be zeros.

BITS 8 - 31: A 24-bit field which defines the address of the first control word used in the I/O operation. The CCW specified in this address must not be a Transfer-in-Channel command.

CHANNEL COMMAND WORD (CCW)

- The CCW is a 64-bit word located in a main-storage area designated by the program.

- It contains control information specifying:

  1. Type of operation (command code).
  2. Main storage area to be used (data address and byte count).
  3. Action to be taken upon completion of operation (Flags).

- Accessed only during the start I/O microprogram when initiating the I/O operation or chaining from one CCW to another.

- Used as a basis to form the unit-control word (for multiplexor channel only).

The address contained in the CAW is the address in main storage of the first CCW to be used in an I/O operation. This CCW is a 64-bit word and specifies the operation to be performed. Because the CCW must be located on a double word boundary, the three low-order bits of its address must be zeros.

The information contained in the CCW defines:

1. The operation to be performed.
2. The main-storage area to be used.
3. The action to be taken upon completion of the operation.

The format of a CCW is:

0 7	8 31	32 36
Command Code	Data Address	Flags

37 39	40 47	48 63
Zeros	Ignored	Count

BITS 0 - 7: Command Code. These eight bits specify one of the following six channel commands:

1. Read Backward
2. Write
3. Read
4. Control
5. Sense
6. Transfer-in-Channel

Each of the six commands, with the exception of Transfer-in- Channel (TIC), initiates the I/O operation. The TIC command CCW contains the address of another CCW in bits 8 - 31. The remaining bits (32 - 63) are ignored.

Figure 1-4 shows the CCW command code format. Bits marked x and M have the significance:
x = ignored
m = modifier

	BIT
Command	0 1 2 3 4 5 6 7
Invalid	x x x x 0 0 0 0
Sense	m m m m 0 1 0 0
Transfer in channel	x x x x 1 0 0 0
Read Backward	m m m m 1 1 0 0
Write	m m m m m m 0 1
Read	m m m m m m 1 0
Control	m m m m m m 1 1

Figure 1-4 Command Code.

The modifier bits of the command code specify to the I/O unit the detail conditions under which the operation is to be executed. It covers such conditions as recording-density, parity, and byte size. The meaning of the modifier bits depends on the type of I/O device.

BITS 8 - 31: Data Address. These 24 bits specify the location of an eight-bit byte in main storage. It is the first location referred to in the area defined by the CCW. The location is the starting address from which data is to be fetched or the starting address of the locations where data is to be stored.

BITS 32 - 36: Flags. Five flag bits modify the basic CCW command :

1. Bit 32, Chain Data CD. This bit specifies chain data addresses. It allows a channel to obtain the data address and count from the next CCW after the CCW in which it appears has been fully executed.

2. Bit 33, Chain Command (CC). This bit specifies chain command. It allows the channel to obtain and execute the next CCW if the CCW in which it appears is successfully executed.

3. Bit 34, Suppress Length Indication (SLI). This bit allows the channel to ignore incorrect length indications when an I/O device signals the end of an operation before the number of bytes specified by the CCW count have been transferred, or vice versa.

4. Bit 35, Skip. This bit allows the channel to bypass storing input information in main storage.

5. Bit 36, Program Controlled Interrupt (PCI). This forces the channel to attempt an interrupt of the major program. It is used by programmers to force an interrupt to the major program during the execution of a chain of CCW's. It allows the CPU to find out how the operation is going, and to initiate other action if the conditions warrant it.

<u>BITS 37 - 39:</u> These three bits must be zero. If they do not contain zeros, the operation is terminated with a program error indication.

<u>BITS 40 - 47:</u> These eight bits are not used and are ignored.

<u>BITS 48 - 63:</u> Count. This 16-bit field defines the number of byte storage locations in the area defined by the CCW. The count in conjunction with the data address specifies the complete storage area used by the current CCW.

<u>CHANNEL STATUS WORD (CSW)</u>

- The CSW is a 64-bit word located at main-storage word location 64 (40).
- It describes the status of the I/O operation just completed.
- It identifies the I/O protection key, the last CCW address used plus eight, the residual count, and channel and unit status at end time.
- It is stored in main-storage location 64 at the completion of an I/O operation when the I/O interruption (for that operation) is taken.
- During the initiation of an I/O instruction, the status portion of the CSW may be stored in main-storage storage 68 and 69 (44 and 45 hex) due to the I/O instruction being rejected, or if an immediate operation has been performed.

When the I/O operation sequence is completed at the unit an interruption to the CPU program is generated. The interruption informs the program that the operation initiated by the start I/O instruction has been completed. The CPU, in accepting the interruption, also stores, in a fixed memory location, a channel-status word. This 64-bit channel status word refers to the unit and operation that has just terminated and describes the results obtained with the status conditions prevailing. The CSW is formed in the process of an interruption and it is stored upon interruption at the fixed location 64 in main storage. It is available at this location (for interpretation by the program) until a new CSW is stored.

```
┌──────────────┬───────┬───────────────────┐
|0            3|4     7|8                31|
| Protection   | 0000  | Command Address   |
| Key          |       |                   |
└──────────────┴───────┴───────────────────┘

┌──────────────────────┬───────────────────┐
|32                  47|48               63|
|        Status        |       Count       |
└──────────────────────┴───────────────────┘
```

1. <u>Bits 0 - 3:</u> Protection Key. This four-bit key is the same as the protection key described previously under <u>Command Address Word (CAW).</u>

2. Bits 4-7: These four bits must be zero.

3. Bits 8-31: Command Address. Identifies the last CCW address used plus eight.

4. Bits 32-47: Status. This area defines the conditions in the I/O device and channel that caused the storing of the CSW.

   Bits 32-39 indicate the conditions detected by the I/O unit or its control unit and are sent via the interface in a status byte.
   Bits 40-47 indicate the conditions detected by the channel.

5. Bits 48-63: Count. This area contains the residual count for the last CCW used.

PROGRAM STATUS WORD (PSW)

- Four PSW's are associated with input/output operations:

  1. Current PSW
  2. Initial Program Load PSW
  3. I/O old PSW
  4. I/O new PSW.

The bit positions in a PSW which are utilized in an I/O operation are listed with their function:

Bits 0-7: System mask

Bit 0 Multiplexor Channel Mask

Bit 1 Selector Channel 1 Mask

Bit 2 Selector Channel 2 Mask

Bits 3-7 are not used.

Bits 16-31: Interruption Code (I/O Old PSW)

Bits 34-35: Condition Code (CC)

The system mask bits 0, 1, and 2 provide the facility to prevent all I/O interruptions occurring when the appropriate bit is zero. This gives the possibility of preventing I/O interrupts from one channel while allowing I/O interrupts on another channel.

The 16 bits which form the interruption code (PSW 16-31) are stored in the I/O old PSW in main storage 56 when an I/O interruption is accepted by the CPU. The interruption code in this case contains the channel and unit address which caused the interrupt. The channel and unit address occupy PSW bit positions 21 - 31. PSW bits 16 - 20 are made zero. Because only the channel and unit causing the interrupt are identified in the interruption code, the program must now refer to the status field of the CSW to determine the cause of the interrupt.

The condition code is set after the initiation of an I/O instruction and informs the program if the I/O instruction was successfully started or if it was rejected. The I/O instruction could be rejected because the channel or unit were unavailable or busy. The condition-code settings for each of the four I/O instructions are:

	0	1	2	3
Halt I/O	not working	halted	stopped	not oper.
Start I/O	available	CSW stored	busy	not oper.
Test Channel	not working	CSW ready	working	not oper.
Test I/O	available	CSW stored	working	not oper.

The meanings of the condition codes are:

1. Available	Unit and channel available
2. Busy	Unit or channel busy
3. CSW ready	Channel status word ready for test or interruption
4. CSW stored	Channel status word stored
5. Halted	Data transmission stopped. Unit in halt-reset mode
6. Not oper	Unit or channel not operational
7. Not working	Unit or channel not working
8. Stopped	Data transmission stopped
9. Working	Unit or channel working

The condition code also may be changed by LOAD PSW, SET SYSTEM MASK, and by an interrupt.

Current PSW

- The fields of the current PSW are scattered throughout local storage and CPU-registers

The current PSW fields and locations are:

1. System Mask: Located in local storage K24. It is also in CPU latches.
2. Storage Protect Key: Located in four high-order bits of local storage location K25 and Q-register. This is the storage protect key for CPU operation. It is not the same as the I/O storage protect key in the CAW.
3. AMWP: Located in the four low-order bits of local storage location K25.
4. Instruction Length Code (ILC): Located in the four low-order bits of local storage K4.
5. Condition Code (CC): Located in four high-order bits of local storage K27.
6. PSW bit: Located at bit-1 of local storage location K4.
7. Instruction Counter (IC): Located in local storage locations K17 and K18 when the PSW bit is on. It is in the IJ-registers when the PSW bit is off.

Initial Program Load (IPL) PSW

- The IPL PSW is loaded from the first input record on IPL operations.
- After the first three records have entered the CPU, the IPL PSW enters local storage as the current PSW.

I/O NEW PSW

- Located at main storage location 120 (hex 78).
- The I/O new PSW enters local storage on I/O interrupts as the current PSW.

I/O Old PSW

- Located at main storage location 56 (hex 38).
- The current PSW is loaded into the I/O Old PSW location during I/O interrupts.

UNIT CONTROL WORD (UCW)

- The UCW is actually a subchannel of the multiplexor channel.
- Refer to subchannels under the multiplexor channel functional units for a description of the UCW.

FUNCTIONAL UNITS

LOCAL STORAGE

- The contents of the CPU data flow registers are stored in local storage when a channel microprogram routine is being executed.
- Two local storage positions serve as the multiplexor channel interrupt buffer.
- Local storage also contains the K-addressable byte positions.

A channel breaks into the CPU microprogram at various times. For this reason the channel's microprogram must store the contents of the CPU data-flow registers in local storage. This allows the channel microprogram to use the CPU data-flow registers. When the channel is finished with the the CPU data-flow registers, they are restored to their original state from the information that was stored in local storage.

Local storage also serves as storage for other channel operating information. This information is stored in the K-addressable byte locations. The K-addressable byte locations also contain some of the current PSW.

The K-addressable byte utilization is as follows:

1. K0 (hex 88) is the Interrupt Buffer (IB) for the address of a MPX channel I/O unit that is requesting an interrupt.
2. K1 (hex 89) stores the T-register during MPX channel operations.
3. K2 (hex 8A) stores the R-register during MPX channel operations.
4. K3 (hex 8B) is the interrupt buffer for the status of a MPX channel I/O unit that is requesting an interrupt.
5. K4 (hex 8C) contains the current PSW instruction length counter (ILC) in the four low-order bits. Bit 1 serves as the PSW bit. The instruction counter (IC) is in local storage at K17 and K18 when the PSW bit is on.
6. K5 (hex 8D) holds the unit address of the I/O unit operating on selector channel 1.
7. K6 (hex 8E) holds the high-order address of selector channel 1 next-CCW address.
8. K7 (hex 8F) holds the low-order of selector channel 1 next-CCW address.
9. K8 (hex 98) serves as temporary storage for the unit address of a unit on the MPX channel.
10. K9 (hex 99) is used by the console subchannel to store unit status.
11. K 10 (hex 9A) through K15 (hex 9E) is CPU working storage.
12. K16 (hex A8) is instruction counter unavailable.
13. K17 (hex A9) holds the high-order byte of the instruction counter when it must be stored in local storage.
14. K18 (hex AA) holds the low-order byte of the Instruction Counter (IC) when it must be stored in local storage.
15. K19 (hex AB) is not assigned.
16. K20 (hex AC) stores the R-register during selector channel microprogram routines.
17. K21 (hex AD) contains the addresses of an I/O unit operating on selector channel 2.
18. K22 (hex AE) holds selector channel 2 next-CCW address high-order byte.
19. K23 (hex AF) holds selector channel 2 next-CCW address low-order byte.
20. K24 (hex B8) contains the current PSW system mask.
21. K25 (hex B9) contains the current PSW storage protect key and AMWP.
22. K26 (hex BA) is not assigned
23. K27 (hex BB) contains the current PSW condition register and program mask.
24. K28 (hex BC) is not assigned.
25. K29 (hex BD) stores the S-register during selector channel microprogram operations.
26. K30 (hex BE) stores the U-register during selector channel microprogram operations.
27. K31 (hex BF) stores the V-register during selector channel microprogram operations.

THEORY OF OPERATION

- The I/O instruction I-cycles are named SI format I-cycles.
- After reading out the instruction the CPU reads out the channel address word (CAW).
- The channel microprogram reads out the CCW and begins its execution.
- When the execution of the CCW is complete the channel either obtains new operating information or attempts to interrupt the CPU program.
- An I/O interrupt loads the CSW with the required information.

Before issuing an I/O instruction, the CPU program loads the CAW with the address of the CCW to be executed, and loads the CCW with the required information. Also the CPU must be in the supervisory state before issuing any I/O instructions. These are privileged instructions that may be issued in the supervisory state only, or a program error results.

During I/O I-cycles the instruction reads out normally. The instruction code enters the G-register. The B-field and D-field specify the channel and device address. The channel address enters the U-register, the device address enters the V-register. When this is complete, the CPU reads out the CAW. The storage protect key enters the high order of the U-register and the CCW address enters the IJ-registers. During the read out of the CAW, the PSW is interrogated to determine if the CPU is in the supervisory state, and the CCW address is checked to ensure that it is on a double word boundary.

When the CAW has been read out, the channel microprogram program reads out the CCW, the channel selects the desired I/O device and issues a command. The I/O device responds with status. After this is accomplished, the channel microprogram analyzes the status byte received from the I/O device during initial selection, and sets the PSW condition code. The channel microprogram then branches to CPU I-cycles.

In the case of a start I/O instruction that has been successfully started, the channel continues to operate with the I/O device either on a time-share basis with other I/O devices and the CPU or constantly with the I/O device until the entire record specified by the CCW count is transferred. This depends on the channel and I/O device whose address the start I/O instruction specifies. In most case the CPU begins the execution of the next instruction. The channel and CPU operate concurrently on a time-share basis. It is possible for the CPU to be executing a program while all three channels are operating concurrently, all on a time-share basis.

When an I/O device has completed the execution of a CCW, it sends an end signal to the channel. The channel in turn either obtains new operating information or attempts to generate an interrupt to the CPU program. The CCW flag bits determine if the channel is to obtain new operating information. The current PSW system mask determines whether or not a particular channel can generate an interrupt.

When the CPU executes an interrupt, it does so at the end of the execution of its current instruction. An I-O interrupt consists of:

1. Storing the current PSW at the I/O old PSW.
2. Loading the CSW with the required information.
3. Loading the I/O new PSW into the current PSW locations in local storage and corresponding registers. The CPU now executes the I/O program defined by the I/O new PSW.

- The IBM System/360 I/O interface provides a uniform method of attaching I/O equipment to the System/360.
- This interface consists of 34 data and control lines.
- Up to eight input/output control units may be attached to one System/360 channel.
- Most control units can accommodate one input/output device although some can accommodate more than one device.

The input/output interface provides a uniform method of attaching input/output (I/O) control units to IBM System/360 channels. The interface can accommodate up to eight control units with addressing capabilities for up to 256 I/O devices. This means that some control units can control a single I/O device and some can control a number of I/O devices.

The interface, consisting of 34 lines, establishes requirements for signal transfers between control units and the servicing channel. Therefore, interface lines provide a common information format and signal sequence for all input/output devices. Figure 2-1 categorizes interface lines according to their general functions.

Except for signals that establish priority among control units, all interface signals are sent over a common bus. Any interface signals that the channel generates is available to all control units, although only one control unit may be serviced by the channel at any given time.

After a control unit is selected, it remains logically connected to the channel until it transmits or receives the required information or until the channel signals the control unit to disconnect.

The rise and fall of signals transmitted over the interface are interlocked with the corresponding responses. This interlocking makes the interface applicable to a wide variety of circuits and data rates, and permits the connection of control units of different circuit speeds and data rates.

Each control unit contains an address card that designates its interface address; no two control units on the same interface can have identical addresses. To begin an I/O operation, the channel must transmit the address of the desired unit. During the selection sequence, the selected control unit must identify itself by transmitting its address to the channel.

A similar identification procedure is necessary if the control unit initiates the signal sequence: the control unit must identify itself by transmitting its address before it can indicate the purpose of the sequence.

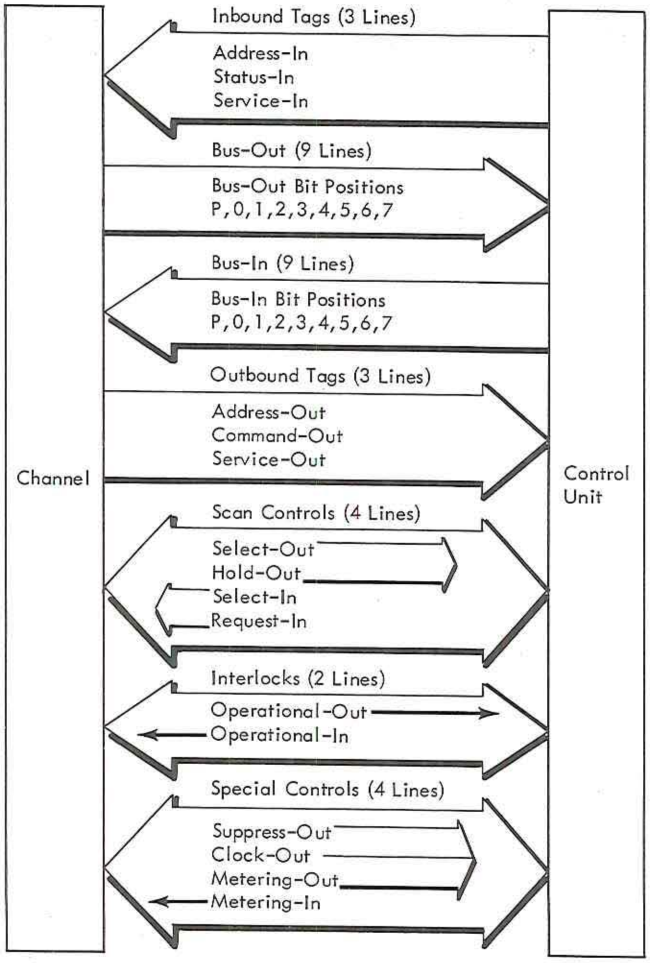


Figure 2-1. Interface Lines

Interface adapter circuits located in each control unit must:

1. Convert interface-line sequences and coded commands to control lines necessary to operate the control unit. In turn, the control unit controls the desired I/O device.
2. Establish communication between the control circuits in the unit and the interface.

INTERFACE LINES

- Interface lines are divided into five types:

  Bus lines
  Scan controls
  Tag lines
  Interlock lines
  Special controls

Figure 2-2 shows channel and control unit connections to the interface. Observe that the select lines (select-out and select-in) connect serially through each control unit for the purpose of establishing priority, other lines connect in parallel.

Priority is assigned to the different control units on the channel during installation. The customer establishes priority of control units by evaluating the type of channel involved, the data rates of the control units to be attached, and the type of operations to be performed. The highest priority control unit can receive the most service from the channel, whereas the lowest priority control unit receives the least service from the channel.

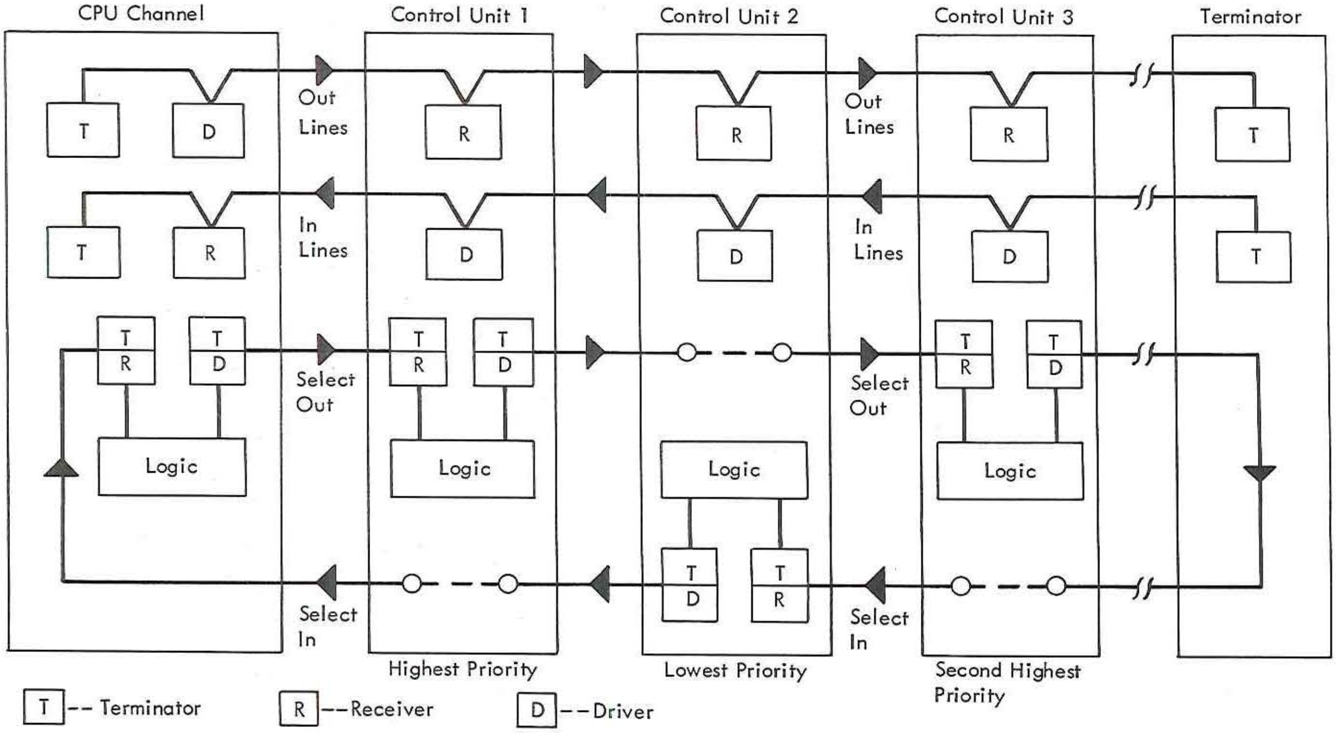
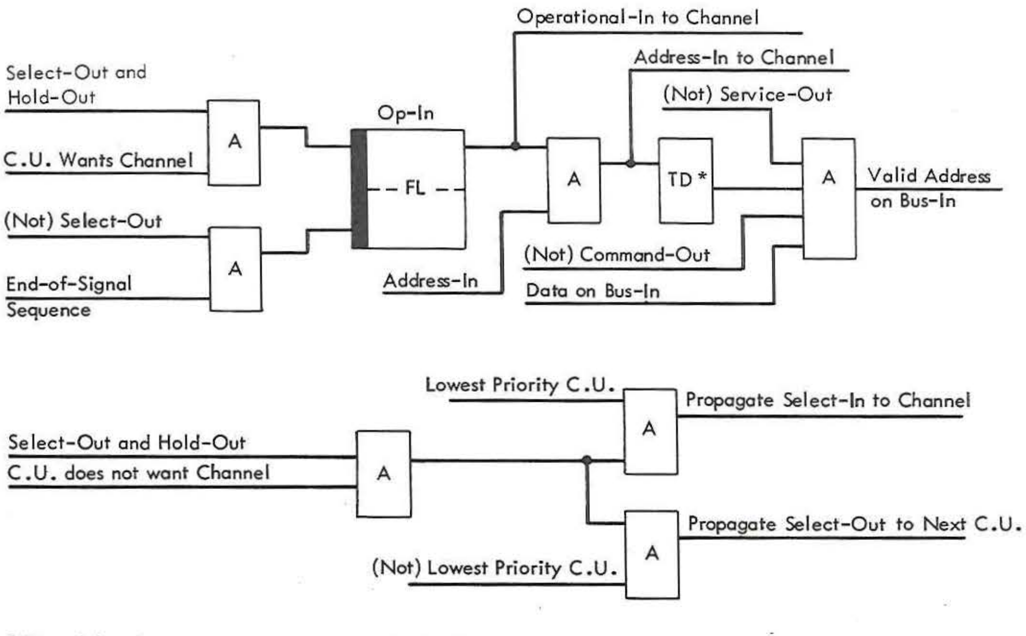


Figure 2-2. Interface Connections

SCAN CONTROLS (FIGURE 2-3).

- Select-out, hold-out, and select-in are controlled by the channel.
- Request-in is controlled by the control unit.
- The scan controls are independent of processing unit non-I/O operations.
- The purpose of the scan controls is to establish initial communication between the channel and attached control units.
- The scan controls establish contact between the channel and the control units on a priority basis.

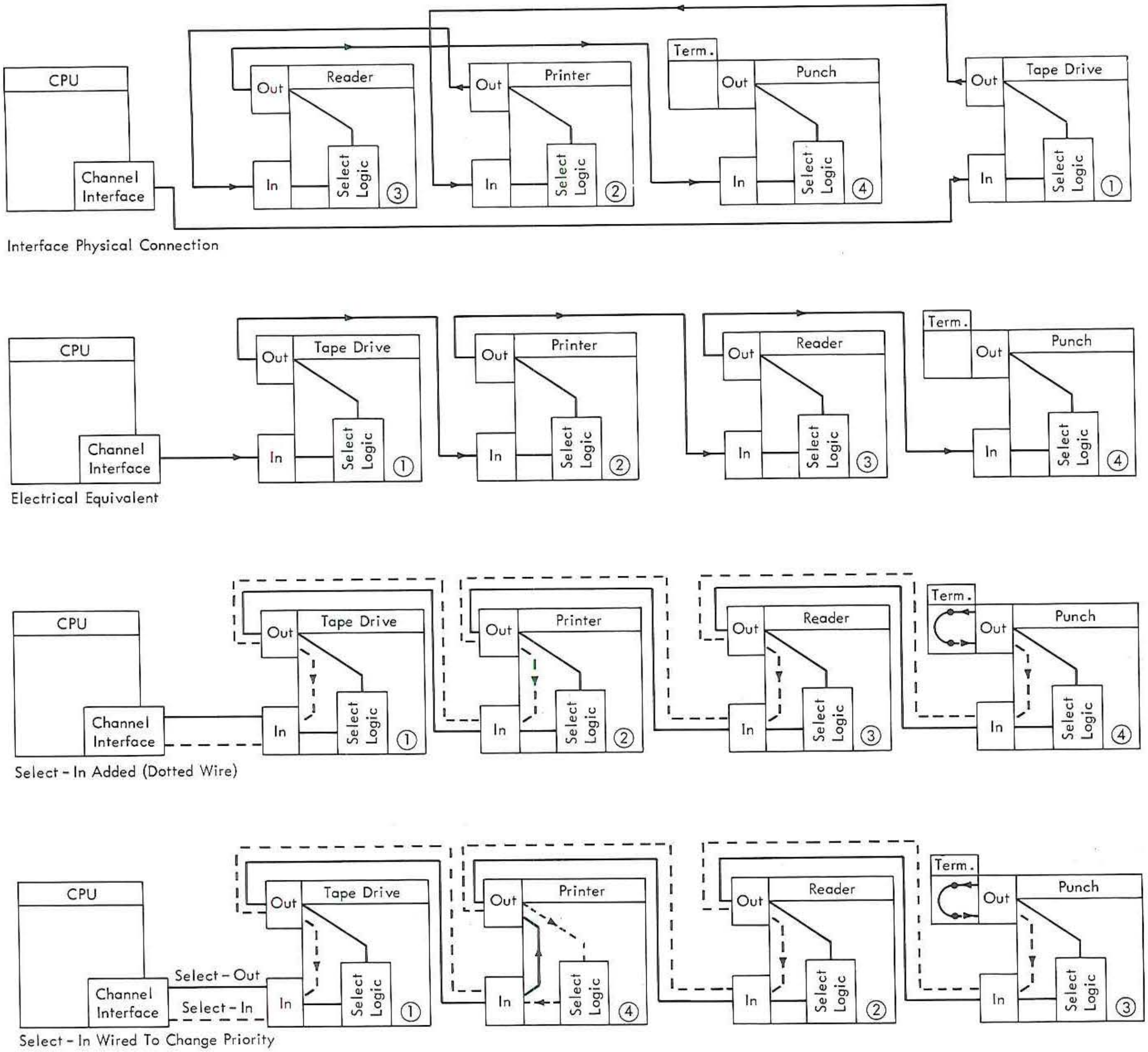


* Time delay duration dependent upon control unit.

●Figure 2-3. Scan Controls

The primary objective of scanning is to establish a connection with a control unit. During initial selection any control unit that receives select-out and hold-out at the same time compares its address to the one on bus-out. An equal compare is the first part of any selection sequence. Scanning, then, is the process of presenting select-out and hold-out in an established sequence until a control unit indicates that its address compares equal.

The sequence in which select-out and hold-out is presented to the control units is called priority. The first control unit to compare-equal blocks the select-out line to the next control unit in line. To further define priority, the first control unit that can receive select-out has the highest priority, the last, the lowest. In reality, there are two ways to establish priority. The method presented here is the simplest and is used primarily for clarity.



Interface Physical Connection

Electrical Equivalent

Select - In Added (Dotted Wire)

Select - In Wired To Change Priority

Figure 2-4. Select-In, Select-Out Priority Control

For this example, consider four control units positioned by the customer in this sequence:

1. Reader (closest to the processing unit)

2. Printer

3. Punch

4. Tape Drive (farthest from the processing unit)

For the system to operate efficiently, assume that the desired order of priority is:

1. Tape Drive (highest priority)

2. Printer

3. Reader

4. Punch (lowest priority)

To establish this priority, the CE must connect the tape drive to the system interface first, then connect the tape drive to the printer, the printer to the reader, and the reader to the punch. The output of the punch is connected to a terminator.

Disregarding, then, the physical location of the units, they are electrically coupled in the desired priority sequence. That is, the first control unit to receive select-out is the tape drive. If this unit compares addresses and finds them equal, it connects to the interface. Select-out is prevented from reaching the other control units by the selection logic. This prevents two or more units from connecting to the interface at the same time. If the unit compares addresses and finds them unequal, it propagates the select-out signal to the next control unit in line. This indicates that the highest priority unit does not require service and others are free to use the interface.

To complete the process of establishing priority, each unit must be jumper wired internally for "high-priority." This term means that each control unit, as it receives select-out, attempts to compare addresses. This wiring also causes select-out to be passed along to the next control unit if the addresses compare unequal.

Using the same cabling sequence, this priority can be altered by jumper wiring a control unit internally for a condition called low priority. Low-priority jumper wiring essentially shorts out the select-out signal directly to the next control unit and picks up the selection pulse on its return path to the processing unit. This return line is called select-in for physical identification only. Electrically, it is the select-out line on its return path to the processing unit. Thus, if any <u>one</u> of the four units in the example is wired for "low priority" its priority would change to the last in the series. Figure 2-4 shows the printer altered from the number 2 position to the number 4 position simply by changing the jumper wiring from high-priority to low-priority.

Considering only one unit wired for low priority, these are the possibilities:

Unit 1 (Tape Drive)	highest or lowest
Unit 2 (Printer)	2nd highest or lowest
Unit 3 (Reader)	3rd highest or lowest
Unit 4 (Punch)	no change

Except for the first unit, altering two or more control units offers unlimited priority arrangements. The first unit can only be first or last, not 2 or 3.

SELECT-OUT

- The select-out line connects each control unit in series.
- Control unit priority is determined by the order that the select-out line goes through the control unit logic.

The select-out line provides the scan loop that allows the channel to interrogate each control unit in priority sequence.

When an I/O control unit receives the select-out signal, it must either raise its operational-in line in response to it (request service) or immediately propagate the select-out signal to the next control unit in the series. Once a control unit has propagated select-out, it cannot raise its operational-in line until the next incoming select-out line to the control unit rises.

The channel must hold select-out active until it receives a signal on either the select-in or the address-in line. When select-out is transferred to the control unit with the lowest priority, the control unit either:

1. raises its operational-in line, and later, its address-in line, initiating a signal sequence with the channel, or
2. propagates the incoming select-out to the channel (select-in).

If a control unit conditions operational-in when the incoming select-out is active, it does not transfer select-out to the next control unit (or select-in to the channel). By conditioning operational-in, the control unit interrupts the channel's scan loop. Then, the control unit transmits an address byte on bus-in lines and conditions address-in.

The control unit must hold operational-in active until communication with the channel is complete. The channel can drop select-out after receiving address-in, or can hold select-out active through the complete I/O operation. In no case can the control unit cancel operational-in before the channel drops select-out.

SELECT-IN

- Select-in is the select-out line's return path to the channel.
- The select-in signal notifies the channel that no control unit has decoded the address sent out.

The control unit provides an option of connecting its selection logic in series on either the select-out or select-in line (see Figure 2-2). Descending-order priority from the channel can be established on the select-out line, and the remaining control units can maintain the descending-order priority to the channel on the select-in logic. For clarity, assume that the selection logic is connected to the select-out line.

Select-in is a line from the lowest priority control unit to the channel. It is the outgoing select-out line of that control unit and provides a return path to the channel for the select-out signal. The definition of the select-in line is the same as that of a select-out line coming from any control unit.

REQUEST-IN

- Request-in allows a control unit to indicate to the channel that it needs service.
- At the channel, request-in initiates the polling sequence.

A control unit conditions the request-in line to indicate that it will initiate a signal sequence when select-out polls that unit again. The channel need not condition select-out to scan the attached control units until the request-in line indicates that a control unit requires servicing. This operation allows the control unit with the highest priority to receive attention in a minimum amount of time after the request-in line is conditioned. To illustrate the function of request-in, consider the following example.

Assume that the highest interface priority is assigned to the I/O device with highest data rate and that the device with the slowest data rate has the lowest priority. The unit with the lowest priority can wait longest for service; so, each time that the request-in line is conditioned, that unit is the last to be polled. Because the I/O device with the highest priority can wait the least period of time for service, each request-in indication causes that unit to be polled first.

HOLD-OUT

- The hold-out signal controls the effect of select-out.
- This signal allows additional channel control over the polling operation.

The channel conditions the hold-out line to all control units in parallel to allow select-out to perform its function in a control unit. The hold-out line allows the channel to cancel the effects of select-out at each control unit at the same time. If the channel is holding select-out active and cancels hold-out, no control unit can make use of select-out. A unit can only propagate the incoming select-out to the next unit. Therefore, the channel can interrupt the polling sequence and cause select-out to fall in all units in the shortest possible time.

BUS LINES

- The bus lines carry information between the channel and all attached control units.
- There are nine bus-in lines and nine bus-out lines.
- The basic unit of information in the System/360, the 8-bit byte, may be transmitted over the bus lines.
- A parity bit is added to the 8-bit byte when necessary to maintain odd parity.

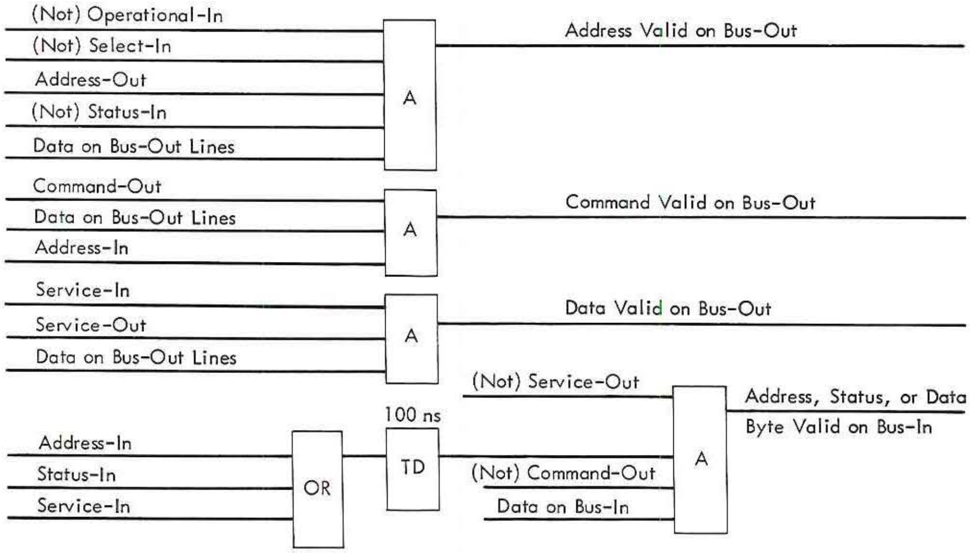


Figure 2-5. Bus Lines

Information is transmitted over the interface from the channel to a control unit on bus-out, and from a control unit to the channel on bus-in. Figure 2-5 shows the different conditions necessary to have information on bus-in and bus-out at the correct time. Bus-in and bus-out each contain eight information lines and one line for odd parity. Thus the basic unit of information in the System/360, the 8-bit byte, can be transmitted over the interface.

Information on the in and the out bus is arranged so that bit position 7 on a bus always carries the lowest order bit within an 8-bit byte. The highest-order bit is in position 0, and intervening bits are in descending order from position 1 to position 6.

When a byte transmitted over the interface consists of fewer than eight bits, the bits must be placed in the highest-numbered bit positions of the bus (low area). Any unused lines of the bus must include the low-numbered positions of the bus. Thus, if a paper tape reader reads 5-channel tape, bus-in positions 0, 1, and 2 are always zero. The parity bit of any byte must appear in the parity position (P). The byte formed by the nine bits must always contain odd parity.

BUS-OUT

- Bus-out transfers information from the channel to a control unit.
- The information can be either data, control, or address information.
- Tag lines identify the bus-out data.
- Bus-out consists of eight data lines and one parity line.

The nine bus-out lines transfer information from the channel to control units. The channel conditions an out- bound tag line to identify the type of data transmitted on bus-out lines. For example, when the address-out tag and bus-out lines are active concurrently, the information on the bus-out lines designates an address.

Tag lines control the period during which bus-out lines contain valid information. When the address of an I/O device is transferred, the information on bus-out lines is valid from the rise of address-out to the rise of one of the following: operational-in, select-in, or status-in. When information other than an address is transferred, signals on bus-out lines are valid from the rise of the identifying outbound tag to the fall of the responding inbound tag.

BUS-IN

- Bus-in transfers information from a control unit to the channel.
- Bus-in information can be either data, addresses, or controls.
- Inbound tag lines identify the information on bus-in.
- Bus-in consists of eight data lines and one parity line.

The nine bus-in lines transfer information from the selected control unit to the channel. The control unit conditions an inbound tag line to identify the type of information transmitted on bus-in lines. For example, when the status-in tag and bus-in lines are active concurrently, the bus-in lines contain a status byte.

Tag lines control the period during which bus-in lines contain valid information. Signals on bus-in lines are considered valid from 100 nanoseconds after the rise of the identifying inbound tag to the rise of the responding outbound tag.

- Select-out, hold-out, and select-in are controlled by the channel.
- Request-in is controlled by the control unit.
- The scan controls are independent of processing unit non-I/O operations.
- The purpose of the scan controls is to establish initial communication between the channel and attached control units.
- The scan controls establish contact between the channel and the control units on a priority basis.

The primary objective of scanning is to establish a connection with a control unit. Any control unit that receives select-out and hold-out at the same time compares its address to the one on bus-out. An equal compare is the first part of any selection sequence. Scanning, then, is the process of presenting select-out and hold-out in an established sequence until a control unit indicates that its address compares equal.

OUTBOUND TAG LINES

- There are three outbound tag lines:

    1. Address-Out
    2. Command-Out
    3. Service-Out

- These lines control the basic communication between the channel and a control unit.

- No two of these lines can be active at the same time.

Each of the outbound tag lines has a basic purpose:

Address-Out — The data on the bus-out lines is an address.

Command-Out — The data on the bus-out lines is a command.

Service-Out — Responds to data received from the control unit on a read operation, and indicates data being sent to the control unit on a write operation.

These signals have a positive sequence depending on the operation involved. Example: Initial selection share-cycle, ending procedure. It is as much the sequence that governs the operation as the exact name or purpose of the line. These sequences must be thoroughly understood to become completely familiar with the operation of the interface.

ADDRESS-OUT

- The basic function of address-out is to indicate to a control unit that the information on bus-out is an address.

- Address-out can also be activated at the same time as select-out to disconnect a selected control unit from the interface.

The channel transmits a signal over the address-out line (Figure 2-6) to indicate either of two conditions:

1. Address-out initiates selection of an I/O device causing all attached control units to attempt to decode the address on the bus-out lines. Because each control unit address is different, only one unit can decode the address. If the control unit that recognizes the address is not busy, it must respond by conditioning operational-in when select-out is conditioned to the control unit.

    Address-out precedes the rise of select-out by 400 nanoseconds. The channel must hold address-out active until it receives operational-in, select-in, or status-in. Select-in indicates that no control unit decoded the address. This occurs when the specified control unit is off-line. Status-in indicates that the designated control unit is busy and cannot be interrupted to execute another operation. The channel responds to the status-in reply by cancelling select-out. It then waits for status-in to fall, and cancels address-out.

2. Address-out causes a control unit to disconnect from the interface. The channel:

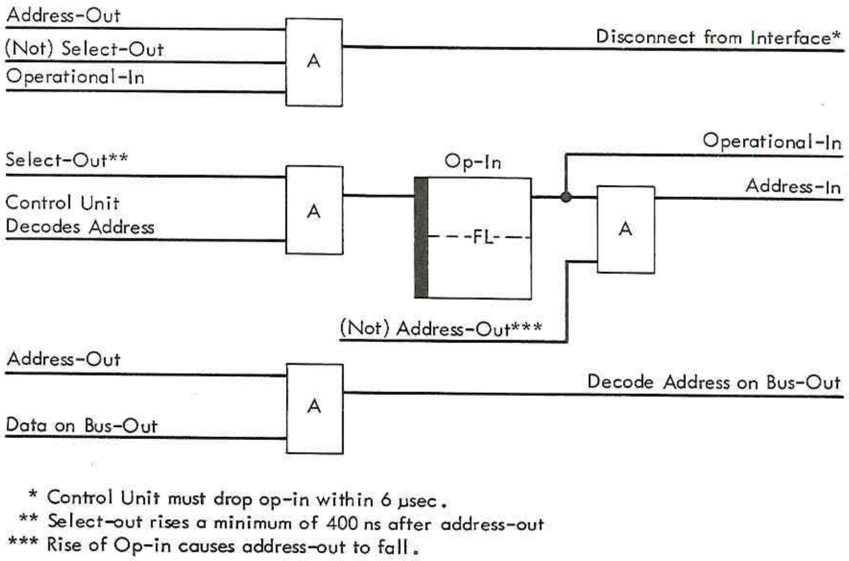
    a. Conditions the select-out line,

b. Receives operational-in from a control unit,

c. Conditions the address-out line (or allows address-out to remain active), then after 250 ns,

d. Cancels select-out and holds address-out active until the control unit allows operational- in to fall.

The control unit must cancel operational-in within 6 microseconds after receiving the interface disconnect indication. The I/O operation proceeds to the normal end, but no data is transferred across the interface.



* Control Unit must drop op-in within 6 µsec.
** Select-out rises a minimum of 400 ns after address-out
*** Rise of Op-in causes address-out to fall.

Figure 2-6. Address-Out

COMMAND-OUT

- The basic function of command-out is to indicate to a control unit that the information on bus-out is a command.
- Command-out also is used to indicate "proceed" at the end of initial selection.
- In response to status-in command-out signals a control unit to "stack" its status.
- In response to service-in, command-out indicates "stop".

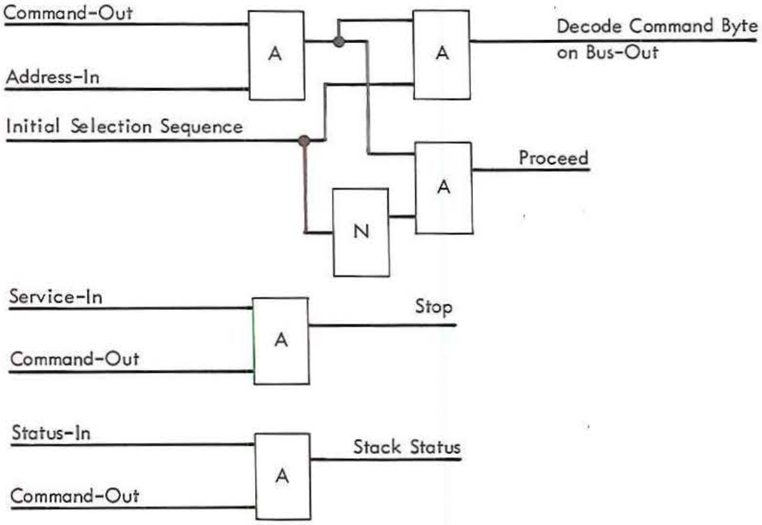


Figure 2-7. Command-Out

The channel conditions the command-out line (Figure 2-7) to respond to a signal on an inbound tag line. During the initial selection sequence, the channel activates command-out to reply to address-in, indicating that a command byte is on the bus-out lines. This command byte specifies the I/O operation to be performed. Only at this point in the initial selection sequence does command-out cause the selected control unit to decode the byte on bus-out lines. After the initial selection sequence, the command-out response to address-in means proceed.

A command-out response to service-in means "stop", and causes the control unit to terminate the operation and reset. Whether during the initial selection sequence or at the end of the operation, the command-out reply to status-in causes the selected control unit to "stack" (hold) the status data. The control unit can present the stacked status only if suppress-out and address-out are down when select-out rises at the control unit.

SERVICE-OUT

- The basic function of service-out is to indicate to a control unit that data has been received or is being sent.
- Service-out responds to service-in on read or write operations and to status-in during a status sequence.

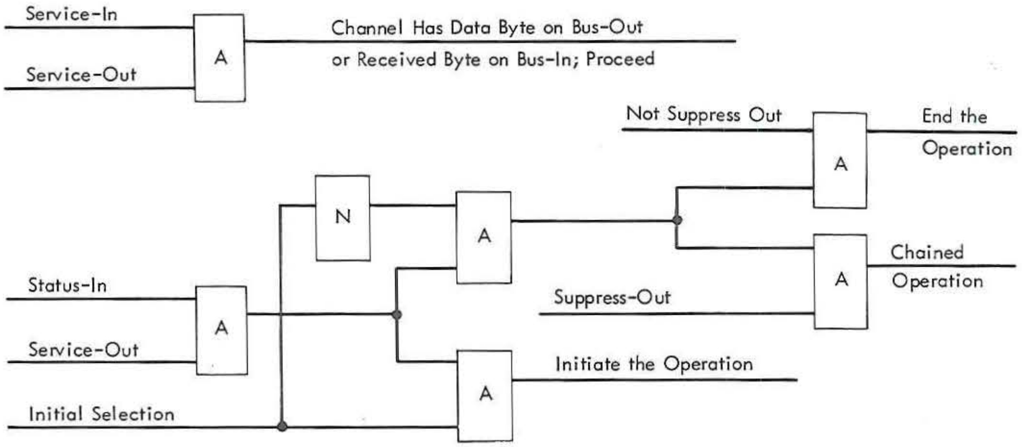


Figure 2-8. Service-Out

Service-out is a line from the channel to all attached control units and is used to signal the selected device in recognition of a signal on the service-in or status-in line (Figure 2-8). A signal on the service-out line indicates to the selected device that the channel has accepted the information on bus-in or has provided on bus-out the data requested by service-in.

When service-out is sent in response to service-in during read, read-backwards, or sense operations, or to status-in, the service-out signal must rise after the channel accepts the information on bus-in. In these cases, the rise of service-out indicates that the information on bus-in has been received.

When service-out is sent in response to service-in during a write or control operation, the rise of service-out indicates that the channel has provided the requested information on bus-out. In this case, the signal must rise after the information is placed on the bus. Service-out must stay up until the fall of the associated service-in or status-in signal. Service-out cannot be up concurrently with any other out-tag.

A service-out response to status-in while suppress-out is up indicates to the control unit that the operation is being chained. See Suppress-Out for further details. The status is accepted by the channel.

INBOUND TAG LINES

- There are three inbound tag lines:

  Address-In
  Status-In
  Service-In

- These lines control the basic communication between a control unit and the channel

- No two of these lines can be active at the same time.

Each of these lines has a single basic purpose:

Address-In — The data on the bus-in lines is an address.

Status-In — The data on the bus-in lines is status information.

Service-In — The information the bus-in lines is data on a read operation. On a write operation, this line indicates that the control unit is ready to receive data.

The control unit uses the inbound tag lines to respond to signal sequence developed by the channel. For example, during an initial selection sequence the channel activates address-out. The control unit that is selected responds with address-in. The channel then sends command-out. When the control unit receives the command, it sends status-in. The channel responds with service-out. When the control unit is ready to send or receive data, it activates service-in.

ADDRESS-IN

- Address-in identifies the information on the bus-in lines as a control unit address.

- When the channel identifies the control unit, it responds with command-out.

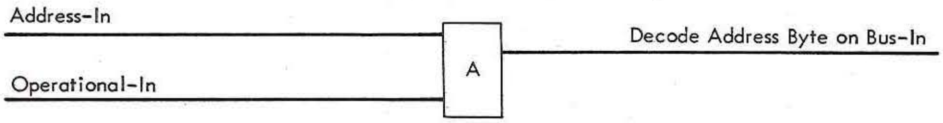


Figure 2-9. Address-In

Address-in is a line from all attached control units to the channel (Figure 2-9). It is used to signal the channel that the address of an I/O unit is on bus-in. The channel responds to address-in with command-out.

During the initial selection sequence, the channel checks the address received against the address initially sent out. If they are equal, the operation can proceed. If not, the channel branches to an error routine.

During data transfer cycles, the channel identifies the control unit when address-in is received. The channel can then select the correct information to proceed with data transfer.

Address-in remains active until the rise of command-out. It must then fall in order that command-out can fall.

STATUS-IN

- Status-in identifies the information on *he bus-in lines as a status byte.
- The line must remain active until the channel responds.

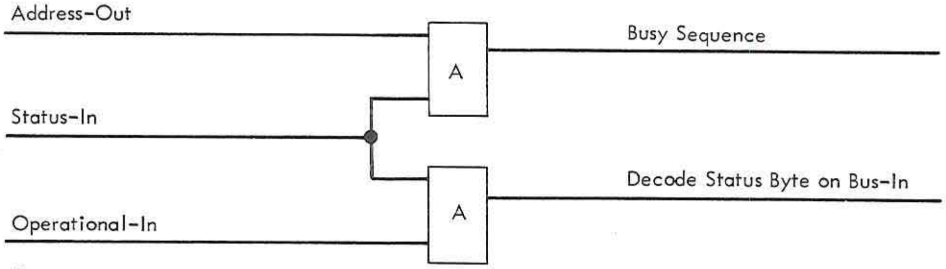


Figure 2-10. Status-In

A control unit activates the status-in line to indicate to the channel that a status byte is on the bus-in lines (Figure A-10). The status byte has a fixed format and contains bits describing the current status at the control unit.

Status-in must remain up until the channel responds with:

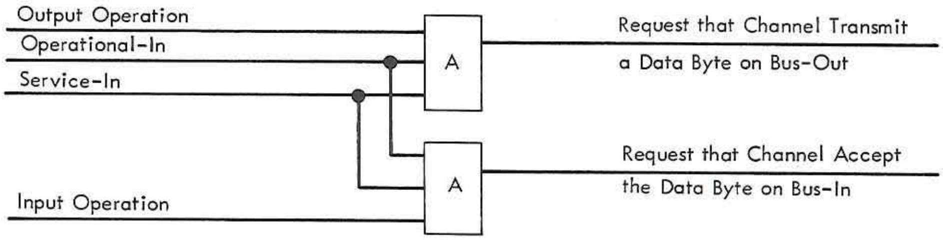
1. Service-out, indicating that the channel accepted the status.

2. Command-out, indicating that the channel has stacked the status.

If status-in is the control unit's reply to address-out during initial selection, it must remain up until select-out falls.

SERVICE-IN

- On a read operation, service-in indicates that data is ready on the bus-in lines.
- On a write operation, service-in indicates that the control unit is ready to receive data.
- This line remains active until the rise of either command-out or service-out.



●Figure 2-11. Service-In

A control unit activates the service-in line (Figure 2-11) to:

1. Signal the channel that a data byte is on the bus-in lines (read, read backward, and sense operations).

2. Request that the channel transmit a

data byte on the bus-out lines (write or control operations).

The channel responds to service-in with:

1. Service-out, when data is accepted or transmitted.
2. Command-out, to stop data transfers and end the operation.

INTERLOCK LINES

- There are two interlock lines:

  Operational-in
  Operational-out

- These lines permit only one control unit at a time to communicate with the channel.

The interlock lines provide additional control of the interface. They are used to gate other signals, reset all control units and signal the channel that a unit is selected.

OPERATIONAL-OUT

- All outbound lines are controlled by operational-out.
- To reset all the control units, the channel drops operational-out and suppress-out.

Operational-out originates at the channel when the processing unit is power-on-reset. It stays up as long as the channel is operable. The operational-out line enables all control units on the interface to communicate with the channel (providing the correct signal sequences are maintained). If operational-out is down, all interface signals become ineffective.

The fall of operational-out resets control units on the interface either selectively or concurrently, depending on the status of suppress-out, a special control line (Figure 2-12). If the channel drops operational-out while a control unit is operating on the interface, the control unit is reset. If the channel wishes to reset all control units on the interface, the suppress-out and operational-out lines are both dropped (for a period of at least 6 microseconds to ensure a complete reset).

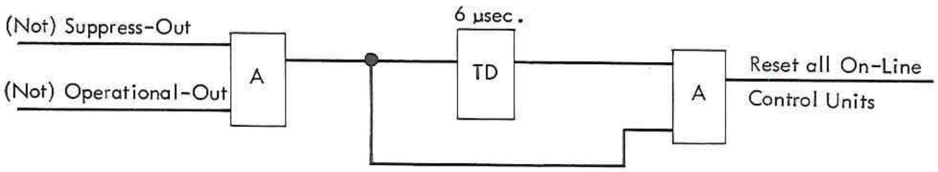


Figure 2-12. Operational-Out

OPERATIONAL-IN

- Operational-in signals the channel that a device is selected.
- When select-out falls and the signal sequence is completed, operational-in is dropped.

To initiate an interface signal sequence, a control unit conditions operational-in when the incoming select-out line is up. At the same time, the control unit blocks select-out from reaching the next control unit. No other control unit can connect to the interface while operational-in is up.

When operational-in is raised for a particular signal sequence, it stays up until all required information is transmitted between the channel and the device. Operational-in drops at the same time as the transfer of the last byte of information, if select-out is down.

Signals on bus-in and on the inbound tag lines are significant only when operational-in is up, except in the case of the control-unit-busy selection sequence. When operational-in is down, the channel disregards any signal on these lines. However, most control units are interlocked so that no signals are placed on any inbound lines when operational-in is down. This is necessary to prevent erroneous indications to the channel when another control unit first raises its operational-in line.

SPECIAL CONTROLS

CLOCK-OUT

- Clock-out designates when a control unit is allowed to change to CE mode.
- Clock-out indicates that the processing unit is not in a halt or wait condition.

A control unit can switch to CE mode only when the System-360 processing unit is halted or waiting. The clock-out line indicates that the processing unit is not halted or not waiting, and could address any control unit. Therefore, a control unit is prevented from changing in or out of CE mode unless the clock-out line is down. The clock-out line prevents the setting of the CE mode latch by the CE key switch. Clock-out is raised by the channel when the conditions are as shown in Figure 2-13.

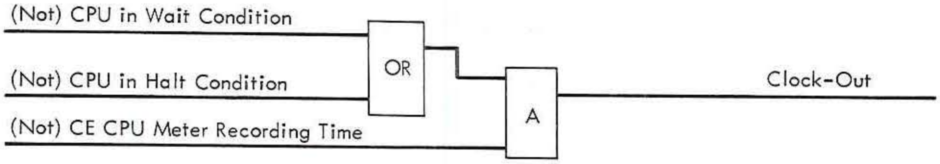

(Not) CPU in Wait Condition
(Not) CPU in Halt Condition
(Not) CE CPU Meter Recording Time
OR
A
Clock-Out

Figure 2-13. Clock-Out

METERING-OUT

- Metering-out enables the control unit usage-meters to record time.

The channel raises the metering-out line to all control units whenever the processing unit customer usage-meter is recording time. Metering-out causes customer usage-meters to register time in each control unit that is not in the disabled state (CE mode).

METERING-IN

- Metering-in indicates to the channel that the control unit is recording time.

A control unit transmits metering-in to the channel when the customer meter on the control unit is recording time. Metering-in causes the customer meter that records use of the processing unit to accumulate time even though the processing unit may be in the halt or wait condition.

SUPPRESS-OUT

- The suppress-out line has these functions:

  Suppress status
  Suppress data transfer
  Chain command control
  Selective reset.

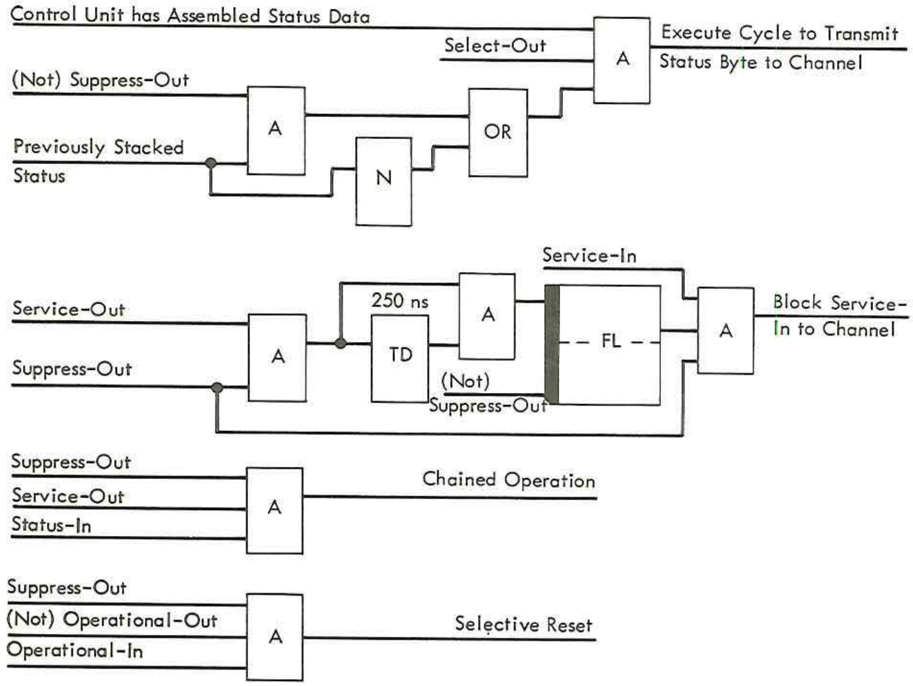


Figure 2-14. Suppress-Out

The suppress-out line (Figure 2-14) connects the channel to each control unit. Suppress-out is used alone or with an outbound tag line to provide the following special functions.

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. The status byte indicates whether errors were encountered in performing the operation and signals the channel that the operation is complete. A channel may respond to status-in with command-out, causing the control unit to stack the status data.

When select-out rises at a control unit holding stacked status data, that control unit will not capture the interface to present the status byte if suppress-out is active. The channel must condition suppress-out at least 250 nanoseconds before the control unit receives select-out to ensure that the stacked status data is not transmitted. The rise of suppress-out after a control unit begins a status cycle does not interfere with the transmission of the status byte. If a control unit conditions request-in to offer status data, and suppress-out rises before the control unit receives select-out, suppress-out drops request-in.

SUPPRESS DATA TRANSFER: For noncyclic I/O devices (buffered I/O devices, not applicable to tape) that can wait for data transfers without indicating an overrun condition, suppress-out blocks service-in. The channel must condition suppress-out at least 250 nanoseconds before the previous service-out tag drops to ensure that the subsequent request for data or offer of data will be suppressed.

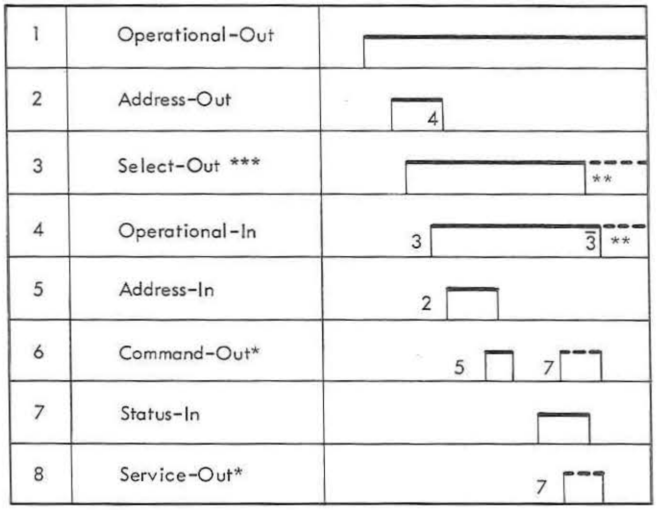
CHAIN COMMAND CONTROL: To indicate a chained command, the channel conditions suppress-out after the selected control unit begins the cycle to transfer the status byte and before the channel responds to status-in with service-out. The active state of the suppress-out line and the service-out response to status-in combine to hold selection of the control unit and the I/O device. The next command from the channel must be directed to that control unit and I/O device.

SELECTIVE RESET: If the channel conditions suppress-out at least 250 microseconds before allowing operational-out to fall and holds suppress-out active until 250 nanoseconds after operational-out rises again, only the I/O device presently operating on the interface is reset.

INITIAL SELECTION SEQUENCE

- During the initial selection sequence, the channel selects a control unit and specifies the operation to be performed.
- This sequence is 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 (Figure 2-15). Regardless of the unit selected or the operation designated, the signal sequence in the initial selection is standard.



* A multiplex channel will respond to status-in with either command-out or service-out. Normally, a selector channel will respond 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 up together.

Figure 2-15. Initial Selection Sequence

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 is active 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. However, the operational-in response to address-out does not commit the control unit or the channel to perform an operation.

If no control unit decodes the address byte (specified control unit is off-line; 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 cancel address-out and terminate the selection sequence.

When operational-in causes the channel to cancel address-out, the selected control unit 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 answered.

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 no-op), 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, the status byte is zero. If the channel accepts this status byte, it responds with service-out. This signal completes the initial selection sequence.

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 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. The remainder of the sequence is the same as a channel-initiated initial selection sequence.

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 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 receipt of the status byte 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 second status byte to the channel. The meaning and format of this status byte are identical to the purpose and format of the status byte transmitted during the initial selection sequence.

To acknowledge receipt of the status byte, 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 is active, the control unit sends its address, and re-transmits the status byte to the channel.

If the channel does not condition suppress-out before select-out rises at the control unit holding stacked status data, the control unit initiates another cycle to transmit the status byte again.

COMPREHENSIVE INTRODUCTION

- The IBM 2030 multiplexor channel shares data-flow registers with the CPU.
- The multiplexor channel consists mainly of microprogram routines.
- The multiplexor channel can sustain operations with several I/O devices at once on a data interleaving basis.
- The basic IBM 2030 multiplexor channel has 32 subchannels.

The IBM 2030 multiplexor channel is basically a microprogram that uses the CPU data-flow registers to perform the multiplexor channel functions (Figure 3-1). The I/O instructions issued to the multiplexor channel (Start I/O, Test I/O, Halt I/O, and Test Channel) are all executed by the channel microprogram.

The multiplexor channel may operate in either one of two modes, BURST or BYTE mode. In burst mode, the channel selects and operates with one I/O device, only, until the entire CCW or chain of CCW's is executed. In byte or data interleave mode, the channel selects and operates with a device until one, two or four 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 IBM 2030 multiplexor channel operates under control of 32 subchannels. The subchannels maintain the operating information for each CCW being executed 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 insure that it specifies supervisory state.

If the PSW does not specify supervisory state, a program interrupt occurs. If the PSW is in supervisory state, the channel begins the execution of the instruction.

When executing either a test channel, a halt I/O, or test I/O instruction, the channel executes the entire instruction before returning the CPU to I-cycles. When executing a start I/O instruction the channel allows the CPU to return to I-cycles before the execution is complete, but only after the selected I/O control unit and device have disconnected from the channel as in the case of byte-mode devices.

Whenever the CPU returns to executing other instructions, the operating information for a selected device is stored in the subchannel. When the device requires service, the channel breaks into the current microprogram routine, stores the CPU data flow register information in local storage, obtains the operating information from the subchannel, 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-PSW instruction.

When the entire CCW sequence specified by the start I/O 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 will normally prevent other I/O interrupts by masking them off with the system mask. It will normally interrogate the CSW and execute some type of error routine if the CSW indicates the instruction was not executed correctly. Otherwise, the I/O program will normally allow 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 interrupt.

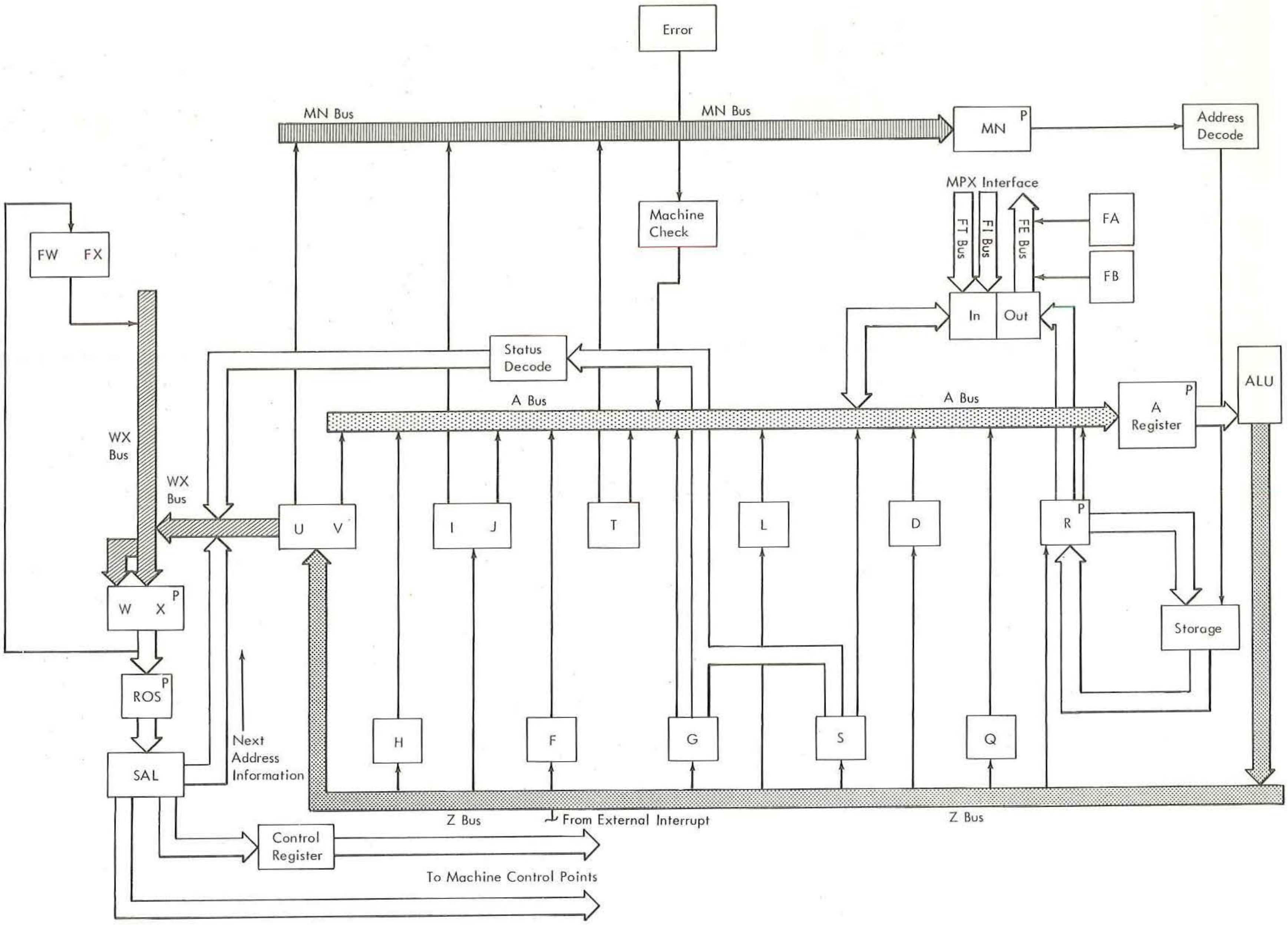


Figure 3-1. Multiplexor Data Flow

DATA FLOW

- All data between the MPX channel and CPU main storage must pass through the R-register.
- The UV-registers maintain the data address
- The LD-registers maintain the count.

Because the MPX channel uses the CPU registers to accomplish its objectives, the MPX data flow is quite similar to normal CPU data flow.

On input operations the channel receives data on bus-in. The MPX channel microprogram puts the bus-in information into the R-register by specifying the R-register as the destination and bus-in as the A-register entry. The data goes through the A-register and Arithmetic Logic Unit (ALU). Any parity errors are corrected in the ALU, and a channel control check is set if running in process mode. If running in stop mode, an A-register error is set. The data enters the R-register from the ALU. It is then stored in main storage by a store cycle.

On output operations data is read out of main storage into the R-register. From the R-register the data (control data or record data) enters the bus-out register which conditions the bus out lines directly.

During both input and output operations, the UV-registers maintain the data address while the LD-register maintains the count.

FUNCTIONAL UNITS

SUB CHANNELS

- Operating information necessary to sustain a single I/O operation is contained in a subchannel.
- The MPX channel may have up to 224 subchannels.
- A MPX channel subchannel is a double word called a Unit Controlled Word (UCW).
- UCW's are stored in MPX storage.
- A UCW is initially generated by the microprogram from the CAW and CCW information.

The high speed of the IBM 2030 CPU and the slow speed of many I/O devices makes it desirable to have a channel that can operate several I/O devices at a time. The multiplexor channel fulfills this requirement. By having many subchannels, each of which contains a current record of one I/O operation, the multiplexor channel is able to read or write from several low-speed I/O devices at a time. When an I/O device requests service, the multiplexor channel selects the subchannel associated with that I/O device and services the I/O device under control of the subchannel. The IBM 2030 has 32 subchannels as a standard feature and may have up to 224 subchannels as optional features.

The first 8 subchannels can be either single or shared subchannels. The remaining subchannels are all single. A shared subchannel can be used by more than one I/O device. Shared subchannels are normally used by multiunit control units, such as tape or disk drives which control several I/O devices. When several I/O devices share a subchannel, only one of the I/O devices can operate at any given time.

A subchannel contains the necessary information that is needed to sustain channel operation with a particular I/O device. Unit Control Words (UCW) serve as the MPX channel subchannels.

A basic IBM 2030 has 32 UCW's stored in a portion of auxiliary storage called MPX storage. The UCW's remain in MPX storage when not being used for channel operation. They are fetched from the MPX storage when

UNIT CONTROL WORD
UCW

000	001	010	011	100	101	110	111
Channel Status	Op. and Flags	Count High	Low	Data Address High	Low	Next CCW Address High	Low

Channel Status Byte

0 - Count Zero (1050 Only)
1 - Channel Control Check
2 - Interface Control Check
3 - First Status Received
4 - (No Name)
Coded explanation below.
5 - Incorrect Length
6 - Program Check
7 - Protection Check

Op. and Flags Byte

0 - Chain Data Address (CDA)
1 - Command Chaining (CC)
2 - Suppress Length Indication (SLI)
3 - Skip
4 - Program Controlled Interrupt (PCI)
5 - Active
6 - Output (Write)
7 - Decrement Data Address

3	4	
0	0	Handling Data; Expecting Data.
0	1	Device Instructed to Stop; Expecting Status.
1	0	Status Stacked at Control Unit.
1	1	Status Is in Interrupt Buffer (IB).

● Figure 3-2. Unit Control Format

needed, and placed in the CPU registers. After they have been used and updated, they are restored into the MPX storage.

Figure 3-2 shows the UCW format. The initial UCW is generated in the CPU registers during E-phase of a start I/O instruction. The address of the first CCW is fetched from the CAW. Adding 8 to this address generates the next CCW address, which becomes part of the UCW.

The data address and count are fetched from the first CCW and placed in the UCW. Because the 2030 storage address is a maximum of 16 bits long, only 2 bytes are needed in the UCW for the data address. The same holds true for the next CCW address.

The op-and-flag byte in the UCW is now generated from the CCW command byte and flag byte. The 5 flags are transferred intact. The command is compressed to two necessary bits: bit-6 indicating input when 0, and output when 1, and bit-7 indicating increment or decrement data address. Bit-5 is the active bit for the UCW, and is set during a start I/O and reset by the store CSW portion of an I/O interrupt or test I/O instruction.

The status byte contains information necessary to generate the CSW. It is kept current during an I/O operation, and with some interchange of bits, becomes the channel status byte in the CSW. Bits-3 and-4 are not used in the CSW. One is replaced by the PCI flag, and the other is unused in the CSW. Bits 3 and 4 have meaning only when the active bit (5) is on.

Bit-3 only indicates the device status byte has been queued in the device.

Bit-4 alone indicates that status-in, not service- in, should be sent by the I/O device next. If not, it is an error.

When bit-3 and bit-4 are on together, a Channel End (CHE) type of unit status (UCW is active) has been stored in the IB and the multiplexor-interrupt latch turned on.

Both bits 3 and 4 off (0), indicate the channel is expecting data and has not yet come to the end of the specified count.

All other positions in the status and op-and-flags bytes have the same meaning as they do in the CCW flags byte and the CSW channel-status byte.

MPX STORAGE (SUBCHANNEL STORAGE)

- 32 double words used to store unit control words.
- Each double word, or UCW, is associated with a control unit or device attached to the multiplexor - channel interface.
- The UCW address is developed from the I/O device address by microprogramming.
- The XH and XL latches in the FB register are used to distinguish between the local storage and one of the MPX storage locations.
- UCW 31 is assigned to the console subchannel.
- Figure 3-3 shows UCW address development.

MPX storage is a 256-byte portion of auxiliary core storage. It holds 32 double words called Unit Control Words (UCW).

UCW's 0 through 7 are used by multidevice control units. These eight UCW's are called shared subchannels, and multi unit control units, which use these UCW's, must have a device address which contains a bit in position 0. Each of these shared subchannel UCW's can be shared by up to 16 I/O devices. The limitation is due to addressing.

Single subchannel UCW's are called for by I/O devices which do not have a bit in position 0 of their address. There are 24 single subchannel UCW's in the basic MPX storage with up to 88 single subchannel UCW's possible. Any of the first 8 subchannels, not used by multi-unit control units, may be used by a single unit control unit.

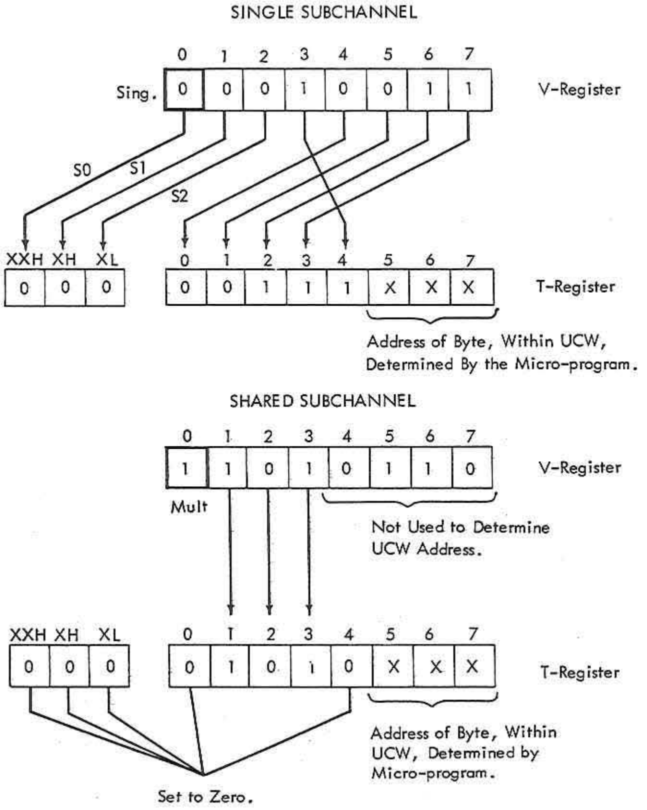


Figure 3-3. Conversion of I/O Device Address to MPX Storage Address

UCW address 31 is reserved for use by the console subchannel. When this address is detected during a start I/O, a branch is taken to the console subchannel channel microprogram.

The address of each UCW is developed from the I/O device address. The development is done by microprogramming.

The I/O device address is stored in the V-register during I-cycles. The bits in the V-register are then transferred to the T-register, and the XXH,XH, and XL latches. This is done in one of two ways, as determined by position 0 of the V-register.

If position 0 is a zero, a single subchannel channel conversion is made. On a machine equipped with 224 UCW's, a multicontrol unit must use each device seperately. The XXH latch is set by position zero. Position one is transferred to the XH latch; position two is transferred to the XL latch. These latches determine which of the three MPX storage areas will be used. If XH and XL are both off, one of the first 32 UCW's will be addressed. The two optional MPX storage areas are addressed by either XH only or XL only being on. This also means that a device address with 011 in bit positions 0, 1, and 2 would be invalid because this would cause local storage to be addressed. Validity checking is done by the microprogram.

Next, the 3-bit of the V-register is transferred to the 4-bit position of the T-register. Then, the 4-, 5-, 6-, and 7-bits are transferred to the 0-, 1-, 2-, and 3-bit positions of the T-register. Now one out of a possible 244 UCW's has been selected. Bits-5, -6, and -7 of the T-register are now set to a constant by the microprogram, thus the T- register contains the address of one byte of the proper UCW.

A shared subchannel UCW is indicated when position 0 of the V-register is 1. In this case positions 1, 2, and 3 in the V-register are transferred to positions 1, 2, and 3 in the T-register. The XXH, XH, and XL latches are turned off (set to 0) and positions 0 and 4 of the T-register are set to 0. Positions 4, 5, 6, and 7 of the V-register are not used.

Again, bits-5, -6, and-7 of the T-register are set to address one of the 8 bytes in the UCW selected.

Device addresses 00 through 1F and 80 through FF are valid on a basic Model 30. By adding two more MPX storage areas, device addresses 20 through 5F are also valid.

MPX CHANNEL MICROPROGRAM

- The MPX channel microprogram performs all the MPX channel functions.
- The MPX microprogram may be entered from four different sources:
    1. The I/O instructions.
    2. The store CSW routine.
    3. MPX share requests.
    4. IPL.
- The MPX microprogram has exits for:
    1. Selector channel I/O instructions.
    2. Console subchannel instructions.
    3. Return to the CPU microprogram.
    4. Load PSW.

All multiplexor-channel operations are performed under control of the channels microprogram. The CPU may enter the MPX channel microprogram by means of a normal microprogram branch, as on start I/O instructions, or the CPU may be forced into the MPX channel microprogram by the MPX stacking latch as on MPX share-cycles.

The four entries to the microprogram point out the four basic uses of the microprogram. The four I/O instructions: Start I/O, Test I/O, Halt I/O and Test channel enter the channel program from I-cycles. Start I/O uses the majority of micro steps, while the test-channel instruction uses very few.

A forced branch to the second entry point, is caused by a multiplexor share-from some I/O device. This entry consists of the store CPU register routine, plus several other routines, which perform the share operation.

A forced entry into the MPX channel microprogram is accomplished when a share-request sets the WX-registers to the address of a channel microprogram word. To be able to return the CPU to its present operating state, the address in the WX-registers is stored in the MPX channel read only back up address register (ROBAR). The branching conditions of the last addressed CPU micro program word are also stored in the MPX channel. These are stored in the buffer restore latches.

The third entry is used when an I/O interrupt causes a store old PSW, store CSW, load new PSW, operation. The store CSW portion of the I/O interrupt is performed by the multiplexor channel microprogram, if the multiplexor channel caused the interrupt.

The fourth entry is another forced branch caused by the initial program load (IPL) key in the console. The routine entered by this branch sets up a dummy CCW and start I/O to cause IPL.

The MPX channel exits are:

1. When the program is entered from I-cycles, the channel address is tested. A selector channel address causes a branch to the selector channel microprogram.
2. If the device address on a basic machine calls for UCW 31, a branch to the console subchannel microprogram is taken.
3. At the end of either a store CSW or an IPL operation, the program branches to the load PSW routine.
4. An I/O instruction for the multiplexor channel will return the CPU to I-cycles on completion.
5. A microprogram trap, caused by a share-request, returns the CPU to the micro word that was interrupted by the trap if no other share requests are pending.

Several new microprogram statements are introduced by the multiplexor channel. These instructions control the FA and FB registers, gate data through the channel, and allow the microprogram to branch on multiplexor channel signals.

Data passes from bus-in to the FI-bus, and from the MPX bus-out register to bus out. The FT-bus is a group of signals from the interface and channel, put together to form a byte that can read onto the A-bus for testing.

The status bus is made up of signals used to control microprogram branching.

I/O INTERFACE CONTROL

- Out-bound lines are controlled by latches in the FA and FB registers.
- In-bound lines are sensed by the microprogram or used to reset latches in the FA register.

Some of the latches in the FA and FB registers are used to control the outbound interface lines. Microprogramming sets these latches, and either the inbound interface lines or microprogramming can reset the latches.

The inbound lines are either sensed directly by the microprogram, or mixed together first, and then sensed.

FA-REGISTER

- The FA-register is comprised of five latches (Figure 3-4).
- It controls the tag-out and bus-out interface lines.
- It controls polling.

The FA-register latches are turned on by microwords that specify FA in the CS field. The value of the CK- field selects a specific FA-register latch or latches.

The five FA-register latches are:

1. Command-start latch
2. Service-out latch
3. Command-out latch
4. Address-out latch
5. Bus-out-control latch

COMMAND-START: This latch turns on when a microword CS-field specifies FA. The command-start latch is controlled by the parity-bit position of the CK-field. When it is 1, the latch turns on. When it is 0, the latch turns off.

When the command start latch is on it prevents the MPX-channel from polling (selecting) any unit that is requesting service from the MPX channel . This includes the console subchannel. It does this by holding the <u>FT3-bit MPX Share Req</u> line at a negative level. It also prevents the MPX stacking latch from turning on by holding the <u>MPX Share Req</u> line at a positive level.

SERVICE-OUT: This latch activates the service-out tag line. It is turned on when a microword specifies FA in the CS-field and CK = 1 or 9 (CK SAL bit-3). It is turned on alone when CK = 1. It is turned on with the bus-out control latch when CK = 9.

The MPX channel microprogram turns on the service-out latch in response to either service-in or status-in. Once the service-out latch turns on, it remains on until the inbound tag line (either service-in or status-in) falls.

COMMAND-OUT: This latch activates the command-out tag line when it is on. It is turned on by a microword that specifies FA in the CS-field and CK = 2, or 10. (CK SAL 2 bit). When CK = 2, it is turned on alone. When CK= =10, it is turned on with the bus-out control latch. The command-out latch is turned on by the MPX channel microprogram in response to either address-in, service-in, or status-in. It turns off when the initiating inbound tag-line falls.

ADDRESS-OUT: This latch controls the address- out tag line. It is turned on to activate address-out.

The address-out latch is turned on by a microword that specifies FA in the CS field and CK = 4 or 12 (CK SAL bit 1). It may be turned on alone (CK = 4) or with the bus-out control latch (CK = 12). It is reset by the microword that specifies FA in the CS field when the CK SAL 1 bit is off (0).

BUS OUT CONTROL: This latch turns on when the microword CS-field specifies FA and the value of CK is 8, 9, 10, or 12 (CK SAL bit 0). The bus- out control latch allows the contents of the R-register to enter the bus-out register . The bus-out register, in turn, conditions the standard interface bus-out lines.

The bus-out control latch is always turned on with a tag-out control latch (CK = 9, 10 or 12). It remains on until the associated tag-out latch turns off.

The bus-out register latches remain on until the next time the bus-out control latch is turned on at which time any bus-out register latches that are not conditioned to turn on by the R-register contents turn off at T1 time.

Once the R-register contents enter the bus-out register, the R-register is available for other functions.

FA Name	CK Field Decode (FA → K)	Reason/ Use

Command Start Latch	CK-P
	P = 1, on
	P = 0, Off
		Prevents MPX Chan. From Polling (Selecting) any unit Requesting Service From MPX Channel (Including Console)

Service - Out Latch	(CK SAL Bit 3) CK = 1 or 9 Turned on Alone When CK = 1. Turns on with Bus - Out CTRL Latch When CK = 9.	Activates Service Out Tag Line, in Response to Either Service - in or Status - in.
Command Out Latch	CK SAL 2 Bit When CK = 2, turns On Alone. When CK = 10 (A), Turns on With Bus-Out CTRL Latch.	Activates Command Out Tag Line in Response to Either Address-in, service-in, or Status - in. Turns off when initiating Tag - Line Falls.
Address Out Latch	CK SAL Bit 1 When CK = 4, turns on alone. When CK = 12, Turns on With Bus - Out CTRL Latch. Resets With (FA → K) Field When CK SAL 1 Bit is off.	Activates Address Out Tag Line.
Bus - Out Control	CK SAL Bit 0 Turns on with Tag-Out CTR: Latch (CK = 8, 9, 10, 12)	Allows Contents of R. Reg. to enter Bus-Out Reg.

● Figure 3-4. FA Register

FB-REGISTER

- The FB-register is comprised of 11 polarity hold latches (Figure 3-5).
- The FB-register latches are turned on or off by a microword that specifies FB as the CS-field destination.
- The CK-field selects a specific FB-register latch or group of latches.

The FB-register controls the operational-out and suppress-out interface lines. It also contains the I/O and external mask latches associated with the current PSW and the XH and XL latches associated with auxiliary storage.

The FB-register is a group of polarity hold latches. These latches turn on or off when a microword specifies FB as the CS field destination. The eleven FB-register latches are:

1. Multiplexor channel interrupt latch.
2. Operational-out control latch
3. Suppress-control latch
4. Multiplexor operation latch
5. Multiplexor channel mask
6. Selector channel 1 mask
7. Selector channel 2 mask
9. XXH-latch
10. XH-latch
11. XL-latch

MPX CHANNEL INTERRUPT: This latch is turned by the MPX channel to indicate that the MPX interrupt buffer (IB) is full. This means that an I/O interrupt is pending.

The MPX channel interrupt latch turns on when a microword CS field specifies FB and CK = 12 and parity bit on (CK SAL bits 0, 1 and parity). It turns off when CS is FB, CK = 12, and CK parity bit is off.

OPERATIONAL-OUT CONTROL LATCH: This latch (FA051) controls the operational-out interface line in conjunction with a 6 usec single-shot It is turned on by a microword that specifies FB in the CS field when CK = 5, 1 (CK SAL bits 1, 3 and P). It is turned off by a microword that specifies FB in the CS-field when CK=5, (parity bit off).

The 6 usec single-shot is activated by the microprogram to cause operational-out to fall during a system reset, recycle reset or power-on reset.

SUPPRESS CONTROL LATCH: This latch is under microprogram control as one of the controls for the suppress-out line. It is turned on by the MPX channel to indicate the channel is command chaining or to cause a selective reset.

The suppress-control latch is turned on by a microword that specifies FB in the CS-field when CK = 10, 1(CK SAL bits 0, 1 and P). It is turned off by a microword that specifies FB in the CS-field when CK = 10, 0 (parity bit off).

MULTIPLEXOR OPERATION LATCH: This latch defines that portion of a MPX share-request cycle when the CPU registers are being used for channel functions. It is controlled by microprogram. It is turned on after the CPU has been stored and turned off before the CPU is restored.

The multiplexor-operation latch is used to tell when to specify channel errors and when to specify CPU errors. All errors that occur when the latch is on are charged to the multiplexor channel.

The multiplexor-operation latch is turned on by a microword that specifies FB as the CS-field destination with CK = 6, 1 (CK SAL bits 1, 2 and P). It is turned off when CS destination is FB and CK=6,0.

I/O AND EXTERNAL-MASK LATCHES: The multiplexor-channel mask, selector-channel 1 mask, selector-channel 2 and external-mask latches are conditioned to set to the contents of the current PSW system mask.

The PSW system mask byte is transferred into four FB-register latches during PSW load. Here they provide active control of I/O and external interrupts. Bits 0, 1, 2, and 7 of the R-register are used to set the mask latches, after the system mask has been read into the R-register. The microword that sets the PSW system mask from the R-register into the FB-register specifies FB as the CS field destination and CK = 3, 1 (CK SAL bits 2, 3 and P). A microword to reset these latches specifies FB in the CS-field with CK = 3, 0 (parity bit off).

XH, XL, AND XXH AUXILIARY STORAGE ADDRESS LATCHES: These latches are used by microprogram routines to address one of the 7 MPX storage areas.

The XXH, XH and XL latches are conditioned by the S- register contents, as well as a microword that specifies FB as the

CS-field destination when CK = 9, 1 (CK SAL bits 0, 3 and P). XXH turns on when the S-register-0 latch is on. XH turns on when the S-register-1 latch is on. XL turns on when the S-register-2 latch is on. These latches are reset by a microword with FB in the CS-field, and CK = 9, 0 is selected.

When XXH, XH, and XL are all off, the first 32 UCW's of MPX storage are selected.

The use of the XXH latch is optional and is used for auxiliary storage addressing on systems that require more than 96 UCW's.

FB Name	CK Field Decode (CS = FB)	Reason Use
MPX Channel Interrupt	CK SALS 0, 1, P (on) 0, 1, P̄ (Off)	MPX Interrupt Buffer (IB is Full.) (I/O Int. Pending)
Operational Out Control Latch	CK SALS 1,3,P (Off) 1,3,P̄ (on)	Controls OP-Out Interface Line With 6 usec S.S.
Suppress Control Latch	CK SALS 0, 2, P (on) 0, 2, P̄ (Off)	Controls On State of Suppress - Out. Indicates Channel is chaining commands or to cause selective Reset.
Multiplexor Operation Latch	CK SALS 1,2,P (on) 1,2,P̄ (Off)	Defines when to Specify Channel Errors and when to specify CPU Errors. (On = Channel Errors, Off = CPU Errors)
(I/O and External Mask Latches) MPX SX1 SX2 External	CK SALS 2,3,P (on) 2,3,P̄ (on)	Provides Active Control of I/O and External Interrupts. Bits 0, 1, 2, and 7 of R-Register are used to Set the Mask Latches.
XXH Latch	CK SALS 0,3,P (on) 0,3,P̄ (Off)	S Register 0 on (optional) used when addressing more than 96 UCW's
XH Latch	CK SALS 0,3,P (on) 0,3,P̄ (Off)	S Register 1 on Used to Address One of 7 MPX Stor Areas
XL Latch	CK SALS 0,3,P (on) 0,3,P̄ (Off'	S Register 2 on. Used to Address One of 7 MPX Stor. Areas
	When XXH, XH, & XL are all off, the first 32 UCW's are addressed.	

●Figure 3-5. FB Register

POLL CONTROL

- The initial-selection sequence uses the command-start and start-select-out latches to control select-out.
- Automatic response to Request In uses the Select Out Request and Select Out Interlock latches.

The multiplexor channel automatically polls the multiplexor channel interface in response to request-in, if both the command-start and select-out interlock latches are off. Select-out is sent over the interface to the unit that sent request-in. When this unit receives select-out, it sends back address-in and operational-in.

ADR-I and OP-I turn on the select-out-interlock latch, which then resets the select-request latch, and select-out is dropped. Another request-in will not be detected until select-out-interlock is reset. Select-out interlock is reset when the control unit that has captured the interface drops operational-in.

Request-in can now be responded to, as long as command-start is off.

The initial selection sequence brings up Select-Out as follows: When the microprogram has arrived at the point in an I/O instruction where it needs the interface, it captures the polling circuits by turning on the command- start latch.

Turning on the command-start latch prevents a request-in signal from turning on the select-request latch, and also prevents the multiplexor share-request signal from being generated.

The start-select-out latch generates the Sel-0 signal during initial selection. This latch is turned on when both Adr-0 and Bus-0 are generated. When the address-out and bus-control latches turn on, their outputs are ANDed to turn on the start-select-out latch.

This latch has three resets. Adr-I will reset the latch. Also, if Adr-O is down, the Sel-I or status-reset signal will reset it; status reset is a combination of STA-I, not CMD-O and not SRV-O, or Recycle reset.

Circuit timing prevents a conflict between request-in and the turn on of command-start. If request-in comes up during the micro step before the word that turns on command-start, a forced ROS branch prevents execution of the microword that turns on command-start until after the share-request operation.

If Req-I rises during the micro step that turns on the command-start latch, the command-start latch is allowed to turn on. The latch clamps off the Req-I signal before it can turn on the select-out request latch or set the MPX share-request stacking latch. Req-I will not be honored until command-start is turned off.

OPERATIONAL INTERLOCK

- Synchronizes Op-I signal to CPU timing.
- Reset every T1 time.
- Used in development of SVI and STI direct branch signals for the status bus.

The operational-interlock latch is turned on at T2 time if the Op-I interface line is up. It is reset every T1 time. Its output is used to force up SVI and STI direct-branch signals when Op-I is down. Refer to Direct Branch Signals for development of SVI and STI signals.

FT-BUS

- Eight selected signals which can be gated to the A-bus.
- There is no parity bit.
- The MPX channel uses 5 of the bus lines.

The FT-bus is made of eight selected signals put together to form a byte of information, with no parity, which can be gated onto the A-bus for testing by the MPX channel microprogram . Two micro steps are needed to test signals on the FT-bus. The first step masks the FT-bus for the signal to be tested. The second step branches on the results of the first step.

Five of the eight FT-bus signals come from the MPX channel circuits. The eight signals which make up the FT-bus are:

FT-0	Suppress-out (diagnostic use only)
FT-1	Hold-in latch (direct data channel)
FT-2	Multiplexor-operation latch
FT-3	Multiplexor-share-request signal (provided command start is off).
FT-4	Initial-program-load latch (CPU).
FT-5	Select-in interface signal
FT-6	Select-out (for diagnostic use)
FT-7	Multiplexor channel interrupt latch

The multiplexor share request signal is a combination of request-in and not command start.

DIRECT-BRANCH SIGNALS

- Direct branching is done on 5 MPX channel signals where speed is a factor.
- Two of the signals SVI and STI, are developed by the channel to provide the high-speed-data loop, with a high speed entry and exit.

Five signals from the MPX channel are sent to the CPU status bus for direct branch control of microprogramming. These signals are sent to the status bus rather than the FT-bus because direct branching is faster.

The signal names are:

OPI	(operational-in interface signal)
AI	(address-in interface signal)
SVI	(service-in signal)
STI	(status-in signal)
INT	(interrupt signal)

SVI and STI are signals developed by MPX channel circuits for use by the high-speed-data loop.

The status-in (STI), service-in (SVI) branch indicates these conditions:

STI	SVI	
1	1	Op-in is down.
0	1	Service-in and Op-in are up (No SRV-O or CMD-O)
1	0	Status-in and Op-in are up (No SRV-O or CMD-O)
0	0	None of the above, usually Op-in up and neither service-in or status-in are up.

The INT signal is brought up by the MPX channel if the multiplexor channel interrupt latch is on and the multiplexor-mask latch if off. This signals the CPU that a multiplexor channel I/O interrupt should be taken.

FI-BUS

- FI in a microword is the designation for bus-in.

FWX-REGISTER (ROBAR)

- Stores the contents of the WX-register register (ROAR) during MPX share cycles.
- FW is a six-position register, parity and 3 through 7.
- FX is a nine-position register, parity and 0 through 7.

The FWX-registers store the contents of the WX-registers when the CPU microprogram is interrupted by the multiplexor channel. This is when the channel is operating in data interleave mode and an I/O unit requires service. The channel is in data-mode when it is operating concurrently with the CPU program on a time-shared basis.

The contents of the WX-registers are gated into the FWX-registers by the <u>MPX Share Pulse</u>. This pulse occurs when an I/O unit requests service on the MPX channel and no higher priority for ROS control exists.

The last microword of the MPX channel routine must gate the FWX-register contents into the WX-registers so the CPU program can continue from the point it was discontinued.

LOAD LATCH

- Turned on by load key.
- Forces trap to IPL microprogram when a trap of higher priority does not exist.
- Turned off by C -> LOAD microstatement when I/O phase of IPL is completed without errors.
- Turns on load lamp on console to indicate IPL.
- Holds suppress-out on for the duration of its state.

The load latch turns on and causes a load request to be sent to the priority stacking latches when the console load key is released. Two latches, which eliminate switch noise, control the turn on of the load latch.

Because the load key also causes a system reset, the load request is not honored until after the system-reset request has been honored and a system reset executed.

Then the load request is honored, and a trap to the IPL microprogram is taken.

When the I/O portion of the IPL microprogram has been completed with no errors, the 2030 executes a micro step containing C -> LOAD in the CK field. This causes the load latch to turn off. The IPL is now complete except for loading the IPL-PSW from main storage address 0000 into the current PSW, which sets up the initial machine conditions and branches the macroprogram to the first instruction.

The IPL latch has an indicator on the console which indicates if the IPL I/O operation has been successfully completed or not.

START I/O(I/O-COMMANDS)

- The IBM 2030 issues all operating commands to its channels and I/O units with a start I/O instruction.
- The channel and I/O unit addresses are designated by the start I/O instruction.

The start I/O instruction is an SI-format instruction. The four- B-field of the start I/O instruction contains the high-order address of a general purpose register (LOCAL STORAGE). The D-field of the instruction contains a displacement. When the B-field of the start I/O instruction is zero, the D-field enters the U- and V-registers as the channel address and the unit address, respectively. (four high-order bits enter the low order of the U-register). When the B-field is not zero, the contents of the general purpose register, referenced by the B-field, adds to the contents of the D-field. The result appears in the D-field and enters the U-and V-registers as the channel and unit addresses, respectively.

When the CPU recognizes an instruction as an I/O instruction, it branches the microprogram to the I/O routine. (QC001) This routine reads the Channel Address word. (CAW) and checks the problem - state bit in the current PSW. It initiates a program interrupt if the bit is on ; I/O instructions are privileged instructions and may be issued in the supervisory state only.

When the start I/O instruction is not being executed as the result of an execute instruction, the microprogram stores the contents of the IJ-registers in two K-addressable byte locations in local storage. These are K17(A9) and K18(AA). The microprogram turns on the PSW bit, bit 1 of K4(8C), to indicate that the instruction counter is stored in local storage and not in the IJ-registers.

If start I/O is being executed as a result of an execute instruction, the IJ-registers are already stored in local storage and is therefore not stored by the microprogram.

Whether or not the I/O instruction is being executed as a result of an execute instruction, the microprogram reads out the CAW.

On start I/O instructions the entire CAW reads out:

1. The CCW address enters the IJ registers.
2. The microprogram checks the CCW address to insure that it is on the correct boundary (multiple of 8), and that the high order is zero.
3. The microprogram places the memory protect tag in the high order of the U-register if the memory- protect feature is present in the CPU.
4. The microprogram checks the first CAW byte to insure that it is zero if memory-protect is not installed in the CPU or that the four low-order bits are zero if memory-protect is installed in the CPU.

When this is accomplished, the microprogram initiates a program interrupt if any errors occurred.

Assume that no errors have occurred, the microprogram resets the multiplexor channel XXH, XH, and XL latches and branches to the start I/O routine associated with the channel addressed by the start I/O instruction, in this case, the multiplexor channel start I/O routine (QC001).

The multiplexor channel start I/O routine reads out the CCW under microprogram program control. It checks the command byte for a Transfer-In-Channel (TIC) command before reading the rest of the CCW. After determining that the data address of the CCW is valid, the count is not zero, and the three low-order bits of the flag byte are zero. The microprogram sets the PSW condition code to zero loads the UCW associated with the unit address with the required information and selects the I/O unit.

The UCW address is determined by the address of the I/O control unit and device which is in the V-register. If bit zero of the V-register is on, it indicates that the control unit to be selected controls multiple I/O devices. Multiple-unit control units operate with one device at a time and always use one of the first 8 UCW locations for a subchannel.

Because the basic IBM 2030 has 32 UCW's, the unit address is checked to determine that it does not exceed a value of 31. If the address is exactly 31, it indicates that the console is being requested. See section 5 for a description of 1050 operations.

Assume the unit address is valid and is not the address of the console. The micro program tests the interrupt buffer status. If the status is zero, the microprogram resets the interrupt latch.

The micro program now starts the initial selection of the I/O control unit and device. It sets the unit address in the R-register and turns on the address-out latch with the instruction FA= K12, 1. This instruction also turns on the bus-out control latch. Thus, the address of the desired unit is placed on the standard interface. The addressed control unit responds to address-out with either status-in or operational-in. Status-in, in this case, indicates that the control unit is busy. Assume the control unit is not busy and responds with operational-in.

The rise of operational-in allows the microprogram to reset the address-out latch.

After the channel drops address-out, the control unit places the address of the selected I/O device on bus-in and raises address-in. The channel compares the address sent to the address received in return and initiates an address mismatch error if they are not identical.

Assume the addresses match, The channel microprogram places the CCW command byte in the R-register, and turns on the command-out latch and bus-out control latch. This resets address-in at the control unit. When address-in falls, the channel microprogram reads out the CCW count and high-order data address, placing them in the D, L, and U registers for later use.

The control unit next presents its status to the channel. When the channel microprogram recognizes the rise of status-in it tests the status.

If the status is 0, as it is if the control and I/O device are in a position to execute a CCW, the microprogram loads the UCW with the CCW command and next-CCW address. The microprogram now responds to status-in with service-out (FA = K1, 1). The microprogram now sets the UCW channel status to 0 and places the memory-protect key in the Q-register. It also turns on the H-register-6 latch, which prevents multiplexor share-request traps. (The command-start latch also performs this same function by clamping off any requests from other control units).

When Op-in falls at the channel, it signals the CPU by activating both the service-in and status-in direct branching conditions. When the microprogram recognizes this condition, it loads the UCW with the count and data address. The microprogram also places the multiplexor protect key in the Q-register, resets the multiplexor operational latch, resets the H-register-6 latch, and returns the program to I-cycles.

The CPU now proceeds with program execution, while the multiplexor channel, the selected I/O control unit, and device execute the CCW command.

MULTIPLEXOR SHARE-CYCLE

- A multiplexor share-cycle:

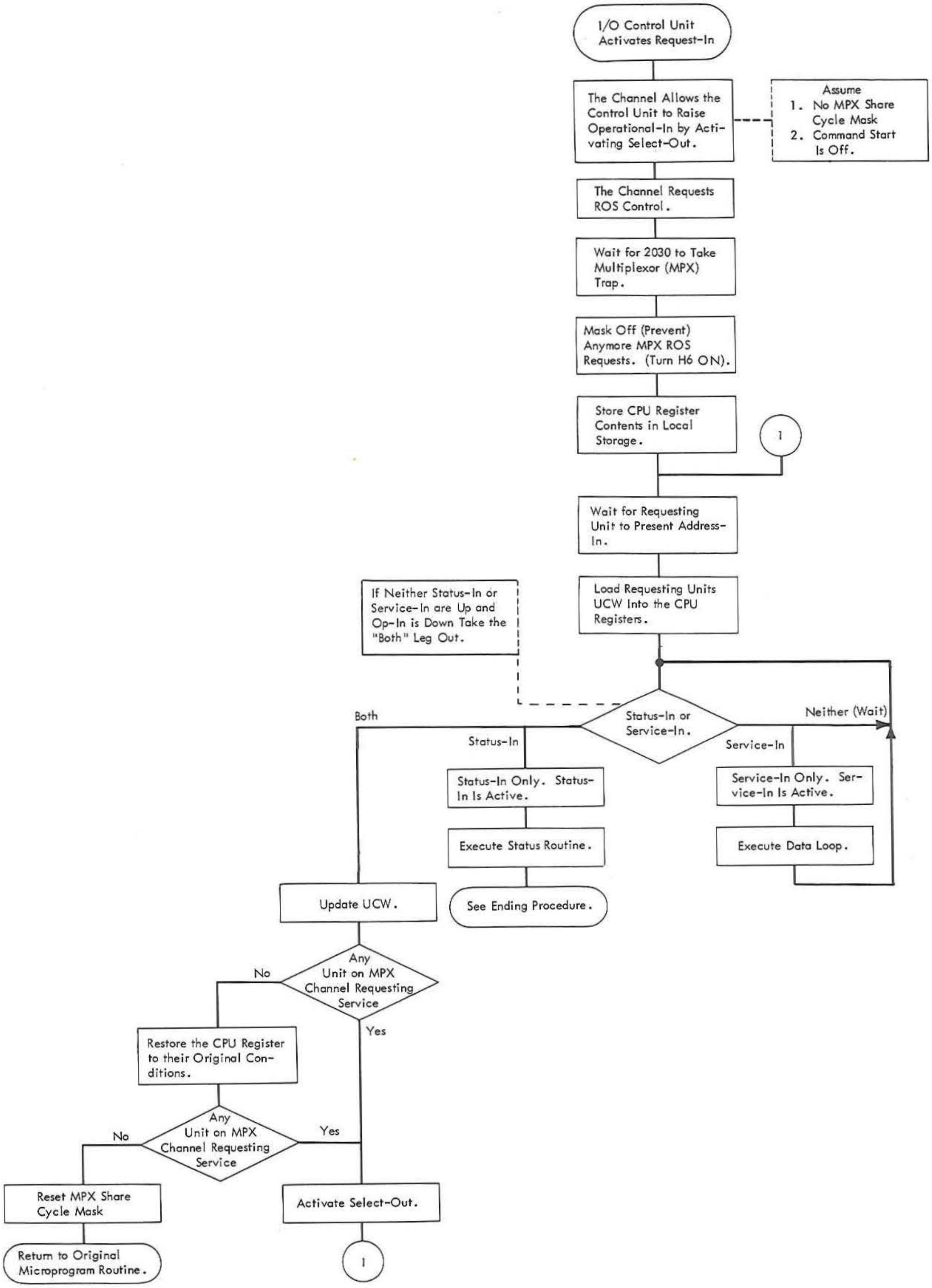
    1. Stores the contents of the CPU registers in local storage.

    2. Connects the requesting I/O unit to the interface.

    3. Loads the unit's UCW into the CPU data-flow registers.

    4. Controls the transfer of information between an I/O unit and the channel.

- Figure 3-6 shows the MPX share-cycle concept.


I/O Control Unit Activates Request-In
The Channel Allows the Control Unit to Raise Operational-In by Activating Select-Out.
Assume
1. No MPX Share Cycle Mask
2. Command Start Is Off.
The Channel Requests ROS Control.
Wait for 2030 to Take Multiplexor (MPX) Trap.
Mask Off (Prevent) Anymore MPX ROS Requests. (Turn H6 ON).
Store CPU Register Contents in Local Storage.
1
Wait for Requesting Unit to Present Address-In.
Load Requesting Units UCW Into the CPU Registers.
If Neither Status-In or Service-In are Up and Op-In is Down Take the "Both" Leg Out.
Status-In or Service-In.
Both
Status-In
Service-In
Neither (Wait)
Status-In Only. Status-In Is Active.
Service-In Only. Service-In Is Active.
Execute Status Routine.
Execute Data Loop.
See Ending Procedure.
Update UCW.
Any Unit on MPX Channel Requesting Service
No
Yes
Restore the CPU Register to their Original Conditions.
Any Unit on MPX Channel Requesting Service
No
Yes
Reset MPX Share Cycle Mask
Activate Select-Out.
Return to Original Microprogram Routine.
1

●Figure 3-6. MPX Share Cycle; General Logic Flow

To send information to the channel or receive information from the channel, an I/O control unit initiates multiplexor share-request cycles. A multiplexor share cycle consists of breaking into the present microprogram routine, executing the multiplexor share-request microprogram routine, and returning to the original microprogram routine.

An I/O unit activates its request-in line to initiate share-request cycles. Request-in activates the FT bus-3 line, multiplexor share-request. This turns on the select-out-request latch, activating select-out. It also turns on the multiplexor channel stacking latch in the CPU. This forces the microprogram to branch to the multiplexor channel microprogram (trap) at the end of the next CPU cycle that has no ROS write instructions pending (allow-write latch off). Also, to branch to the multiplexor channel trap, no higher priority stacking latches may be On and the H-register-6 latch must be Off.

Assume that these requirements are met. The IBM 2030 starts executing the multiplexor trap. The trap stores the contents of the CPU data-flow registers in local storage, loads the I/O units UCW into the CPU registers, and connects the I/O unit to the standard interface (turns on operational-in). The multiplexor trap also contains a data loop which controls the transmission of information between an I/O device and multiplexor channel.

Once an I/O unit captures the interface, it may hold it for the transmission of one byte of information, several bytes, or for a complete record. As long as an I/O unit keeps its operational-in line active, it has control of the interface and is thus connected to the channel. The multiplexor channel stays in its data loop as long as an I/O unit keeps its operational-in active.

When the I/O unit deactivates operational-in, the IBM 2030 resumes execution of the microprogram routine that was discontinued by the multiplexor share-request.

MULTIPLEXOR CHANNEL ROS REQUEST

- The multiplexor (MPX) channel requests ROS control when it receives request-in.
- The CPU allows the MPX channel to take ROS control following a cycle which has no write cycles pending and when there are no higher-priority requests for ROS control.
- The MPX channel stores the ROAR in ROBAR (FWX-register) before it assumes ROS control.
- The MPX channel stores the X6 and X7 branch conditions in its X6 and X7 buffer latches.

When the channel receives request-in from a control unit, it generates a request for ROS control. This request turns on the MPX channel stacking latch at T3 time if the allow-write latch is off and there are no higher-priority requests for ROS control.

When the MPX channel stacking latch turns on, it:

1. Stores the Read-Only Address Register (ROAR) contents in the MPX channel ROBAR (FWX-register).
2. Conditions the X-register to set to the address of the first ROS word of the MPX channel microprogram routine (address 010 in hexadecimal).
3. Stores the X6 and X7 branch conditions in their respective MPX channel buffer latches on the next CPU cycle.
4. Prevents the set of the control register on the next cycle.

Normally, a ROS word is actually executed (control register set) one CPU cycle after it is addressed. When the first MPX channel microprogram word is addressed, the control register is prevented from setting by the any-priority-pulse. This creates an inactive cycle as far as the execution of a microprogram word is concerned. You might also notice that the W-register is not set to a particular address. The W-register is allowed to reset to zero at its normal set time (T1), and the W-register parity latch is turned on by the any- priority-pulse.

Although the any-priority-pulse becomes active for a second time, it does not perform any function because it is turned off at T3 time by the H-6 latch which is turned on by the first MPX channel microprogram word.

The X6 and X7 branch conditions are stored in buffer latches because the branch conditions of the last major microprogram word may be dynamic conditions, such as the Z-bus-zero. These conditions would not be present when this microword is executed which is upon the resumption of the major microprogram. Therefore, the X6 and X7 branch conditions are stored in the MPX channel X6 and X7 buffer latches and are gated to the X-bus when the major microprogram is resumed.

MULTIPLEXOR TRAP

- Turns on H-register-6 latch, masking off the initiation of any further multiplexor share traps from other I/O units.
- Allows the I/O unit to connect to the interface.
- Stores the contents of the CPU data-flow registers in local storage.
- Loads the UCW operating information into CPU registers:
  1. Count into L-and D-registers.
  2. Data address into UV-registers.
  3. Op and flags into G-register.
- Services the I/O unit.
- Updates the UCW.
- Turns off the H-register-6 latch.
- Restores the CPU data-flow registers to their original value.
- Restarts the microprogram at the point it was discontinued.
- See Figure 3-7.

The first microword of the multiplexor trap turns on the H-register-6 latch. This prevents the request-in signals on the multiplexor channel from initiating any multiplexor share-request traps until the H-register-6 latch turns off. This also allows break-in by higher priority trap routines.

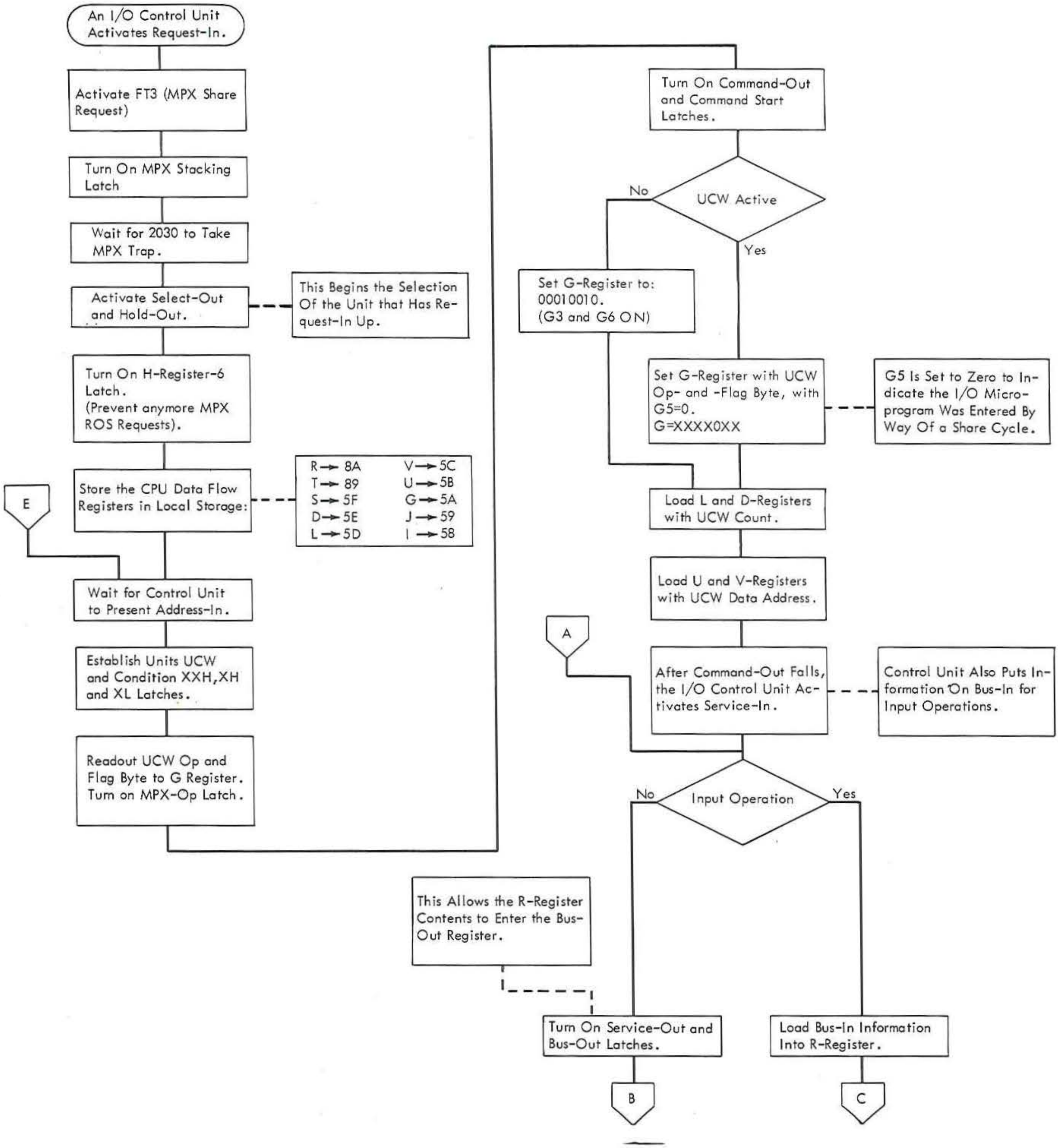
The multiplexor trap now stores the contents of the CPU data-flow registers in local storage.

When the I/O control unit receives select-out, it activates its operational-in line, places its address and the address of the I/O device on bus-in, and activates address-in. When the microprogram detects the rise of address-in, it stores the unit address in local storage and determines from the unit address the address of the associated UCW.

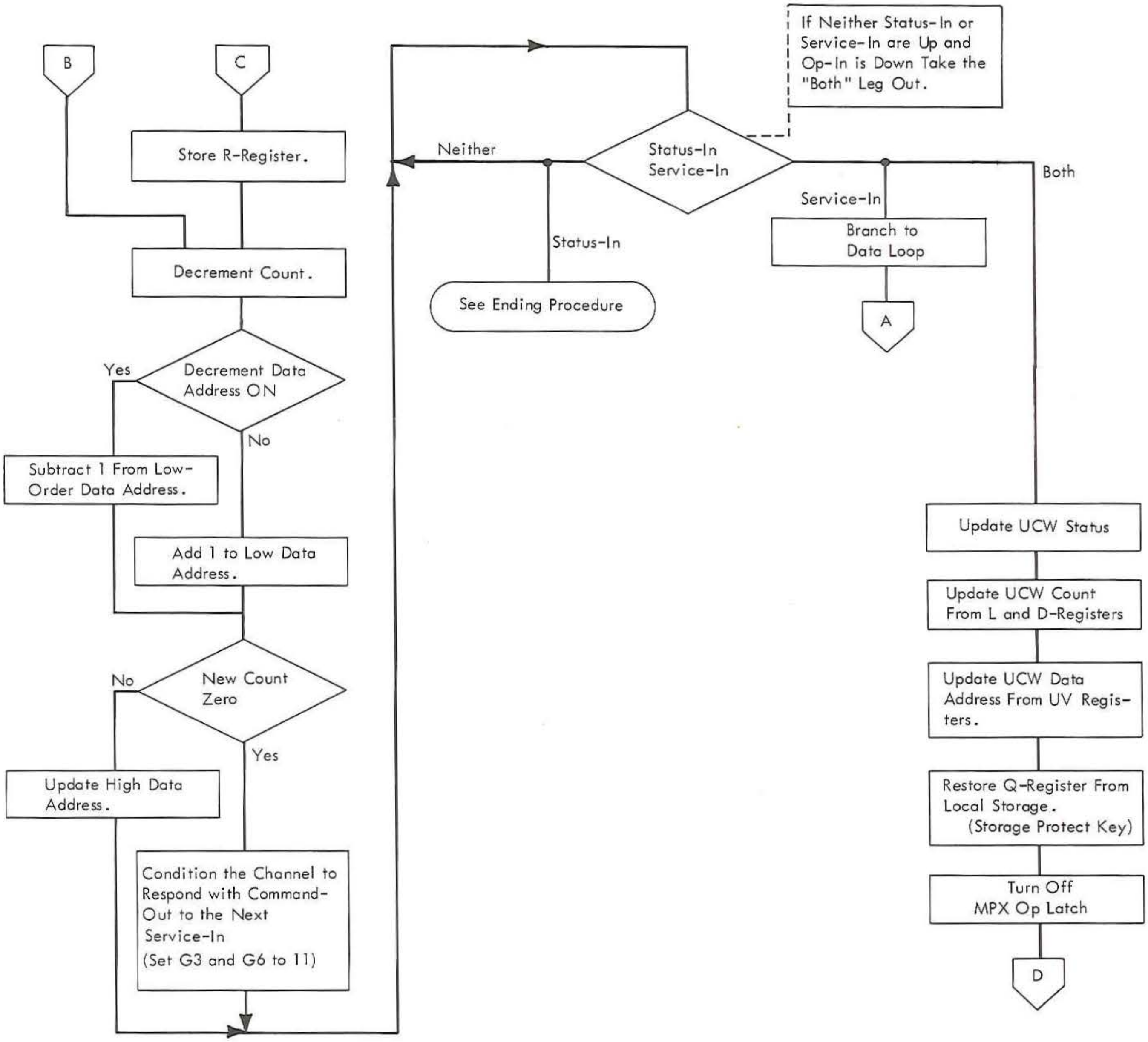
The microprogram now proceeds to load the CPU data-flow registers with the UCW operating information. The microprogram:

1. Loads the G-register with the UCW op-and-flag byte.
2. Sets the Q-register with the multiplexor memory-protect tag.
3. Loads the L and D-registers with the UCW count.
4. Loads the U and V-registers with the UCW data address.

After the I/O unit deactivates address-in as a result of command-out, it activates service-in. If the I/O unit is executing an input CCW it will also place information on bus-in at this time.


An I/O Control Unit Activates Request-In.
Activate FT3 (MPX Share Request)
Turn On MPX Stacking Latch
Wait for 2030 to Take MPX Trap.
Activate Select-Out and Hold-Out.
This Begins the Selection Of the Unit that Has Request-In Up.
Turn On H-Register-6 Latch. (Prevent anymore MPX ROS Requests).
Store the CPU Data Flow Registers in Local Storage:
R → 8A
T → 89
S → 5F
D → 5E
L → 5D
V → 5C
U → 5B
G → 5A
J → 59
I → 58
E
Wait for Control Unit to Present Address-In.
Establish Units UCW and Condition XXH, XH and XL Latches.
Readout UCW Op and Flag Byte to G Register. Turn on MPX-Op Latch.
Turn On Command-Out and Command Start Latches.
UCW Active
No
Yes
Set G-Register to: 00010010. (G3 and G6 ON)
Set G-Register with UCW Op- and -Flag Byte, with G5=0. G=XXXX0XX
G5 Is Set to Zero to Indicate the I/O Microprogram Was Entered By Way Of a Share Cycle.
Load L and D-Registers with UCW Count.
Load U and V-Registers with UCW Data Address.
A
After Command-Out Falls, the I/O Control Unit Activates Service-In.
Control Unit Also Puts Information On Bus-In for Input Operations.
Input Operation
No
Yes
This Allows the R-Register Contents to Enter the Bus-Out Register.
Turn On Service-Out and Bus-Out Latches.
Load Bus-In Information Into R-Register.
B
C

● Figure 3-7. MPX Share Cycle: Service-In and No CCW Flags (Part 1 of 3)


B
C
Store R-Register.
Decrement Count.
Yes
Decrement Data Address ON
No
Subtract 1 From Low-Order Data Address.
Add 1 to Low Data Address.
No
New Count Zero
Yes
Update High Data Address.
Condition the Channel to Respond with Command-Out to the Next Service-In
(Set G3 and G6 to 11)
If Neither Status-In or Service-In are Up and Op-In is Down Take the "Both" Leg Out.
Neither
Status-In Service-In
Both
Status-In
Service-In
Branch to Data Loop
See Ending Procedure
A
Update UCW Status
Update UCW Count From L and D-Registers
Update UCW Data Address From UV Registers.
Restore Q-Register From Local Storage. (Storage Protect Key)
Turn Off MPX Op Latch
D

● Figure 3-7. MPX Share Cycle: Service-In and No CCW Flags (Part 2 of 3)

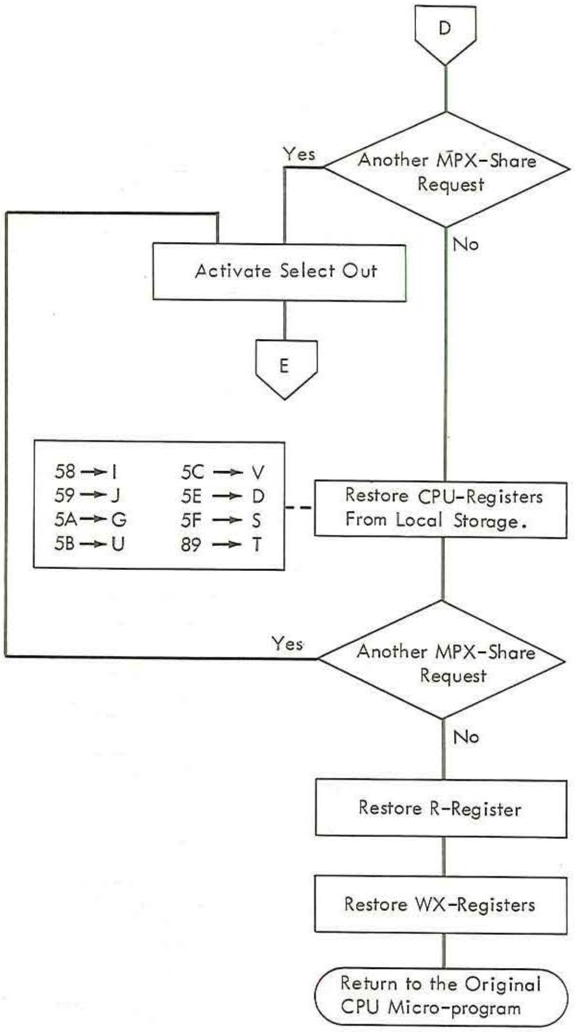

D
Yes
Another MPX-Share Request
No
Activate Select Out
E
58 → I
59 → J
5A → G
5B → U
5C → V
5E → D
5F → S
89 → T
Restore CPU-Registers From Local Storage.
Yes
Another MPX-Share Request
No
Restore R-Register
Restore WX-Registers
Return to the Original CPU Micro-program

Figure 3-7. MPX Share Cycle: Service-In and No CCW Flags (Part 3 of 3)

When the microprogram detects the rise of service-in, it reads out the core storage position referenced by the UCW data address, places the information in the R-register, and decrements the low-order count (U-register) by one. The microprogram now determines if the skip flag is on and acts accordingly. (refer to Multiplexor CCW Flag Operations for skip.)

Assume that skip is not specified by the UCW flag. The microprogram determines if the operation is input or output. For input operations the microprogram loads the R-register with bus-in information, writes this information into the core-storage position referenced by the UCW data address, and activates service-out. If the operation is an output operation, the microprogram turns on the service-out and bus-out latches. The bus-out latch loads the bus-out register with the R-register information. Thus, a byte has been transferred between the CPU and the I/O unit.

The microprogram now updates the data address. It modifies the data address -1 for read-backward operations, and +1 for all other operations.

The microprogram now samples the operational-in, service-in and status-in lines. As long as operational-in remains active, the I/O unit remains connected to the interface and to the channel. Because the CPU actually becomes the multiplexor channel for multiplexor operations, it must continue functioning as a multiplexor channel until operational-in falls.

When the I/O unit holds the channel for transmissions of multiple bytes of data the microprogram executes only the data-loop portion of the multiplexor trap. This consists of either sending or receiving information as specified by the UCW Op byte, and updating the count and data addresses which are in the LD and UV-registers, respectively.

An I/O control unit may keep its operational-in line active for one byte, several bytes, or for the entire record specified by the UCW count. If operational-in remains active, service cycles must occur at a specified rate, or an interface check will occur after a time-out sequence.

To follow this description, assume the I/O unit has requested service several times and now deactivates its operational-in line. When operational-in falls, it forces both the service-in and status-in branching conditions (not interface signals). This signals the microprogram to update the unit control word in MPX storage with the updated count and data-address from the LD- and UV-registers, respectively. It also sets the Q-register with the original storage-protect key. When this is accomplished, command-start turns off to allow further share-requests before the microprogram restores the CPU registers to their original value.

The multiplexor microprogram tests the FT bus for the rise of request-in before returning the CPU to the execution of the original microprogram . If the microprogram determines that request-in is active during the test for request-in, select-out and hold-out are activated if command-start is off and the microprogram tests for the rise of address-in. When address-in becomes active, the microprogram reads out the UCW associated with the I/O unit that is sending the request-in signal and executes the data loop. In this case, the microprogram did not have to store the CPU-registers in local storage, because this information was already in local storage from the previous request-in.

When a microprogram determines that it has honored all outstanding request-in and service-in signals on the multiplexor channel, it turns off the H-register-6 latch and gates the back-up read- only address register (ROBAR; FWX-register) into the ROAR. This allows the 2030 to return to the execution of the microprogram that was discontinued by the multiplexor share-request.

ORIGINAL MICROPROGRAM RESUMPTION

- The ROBAR enters the ROAR.
- The H-register-6 (H6) latch turns off.
- The X-6 and X-7 buffer latches condition the X-bus X-6 and X-7 lines.

To return the CPU to its original microprogram after a multiplexor ROS trap, the multiplexor ROS trap has, as its last instruction, an instruction to transfer the contents of the FWX register to the WX register. The last multiplexor trap ROS word also turns off H6.

During T3 time of the ROS word that specifies WX=FWX, the SALS become effective. The SAL outputs allow the FWX register contents to condition the W-and X-buses.

At T1 time of the next CPU cycle, the W-and X-bus conditions set into their respective registers. This addresses the last original program ROS word addressed, before the execution of the multiplexor ROS trap. Remember that although this ROS word was addressed, its execution was prevented because the CD- registers did not set. During this same CPU cycle, the CD-registers for the last ROS word of the multiplexor ROS trap are set, thereby executing the microprogram instruction.

At T4 time of this cycle, the ROAR restore buffer latch turns on. This latch gates the X-6 and X-7 buffer latches (holding branch condition) onto the X-bus 6 and 7 lines. The executing of the original microprogram is now at the point at which the multiplexor-share-request discontinued it.

ENDING PROCEDURE

- A control unit may present three types of ending status:

  1.Channel-end; data-transfer complete.

  2.Device-end; CCW-execution is complete.

  3.Control unit-end; control unit was busy but is now free.

- When a channel receives ending status, it may obtain a new CCW or cause an interrupt.

- To cause an interrupt, the channel puts the unit address and status in the interrupt buffer and turns on the interrupt latch.

- If the interrupt buffer is full, the channel queues (stacks) the status if not command chaining.

- The channel always queues device-end status when not command chaining.

- See Figure 3-8.

An I/O control unit signals the channel when it has executed the data transfer portion of a CCW command. It also signals the channel when the selected I/O device has fully executed the CCW command. These two signals are channel-end (bus-in bit 4) and device-end (bus-in bit 5). Channel-end and device-end are status conditions and are, therefore, 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. This depends on the CCW command and the I/O unit involved.

Besides these two types of ending status, a control unit may also present control unit-end status. It does this when it becomes free after it has been busy to a new command, or when it finds unusual status after presenting its previous status.

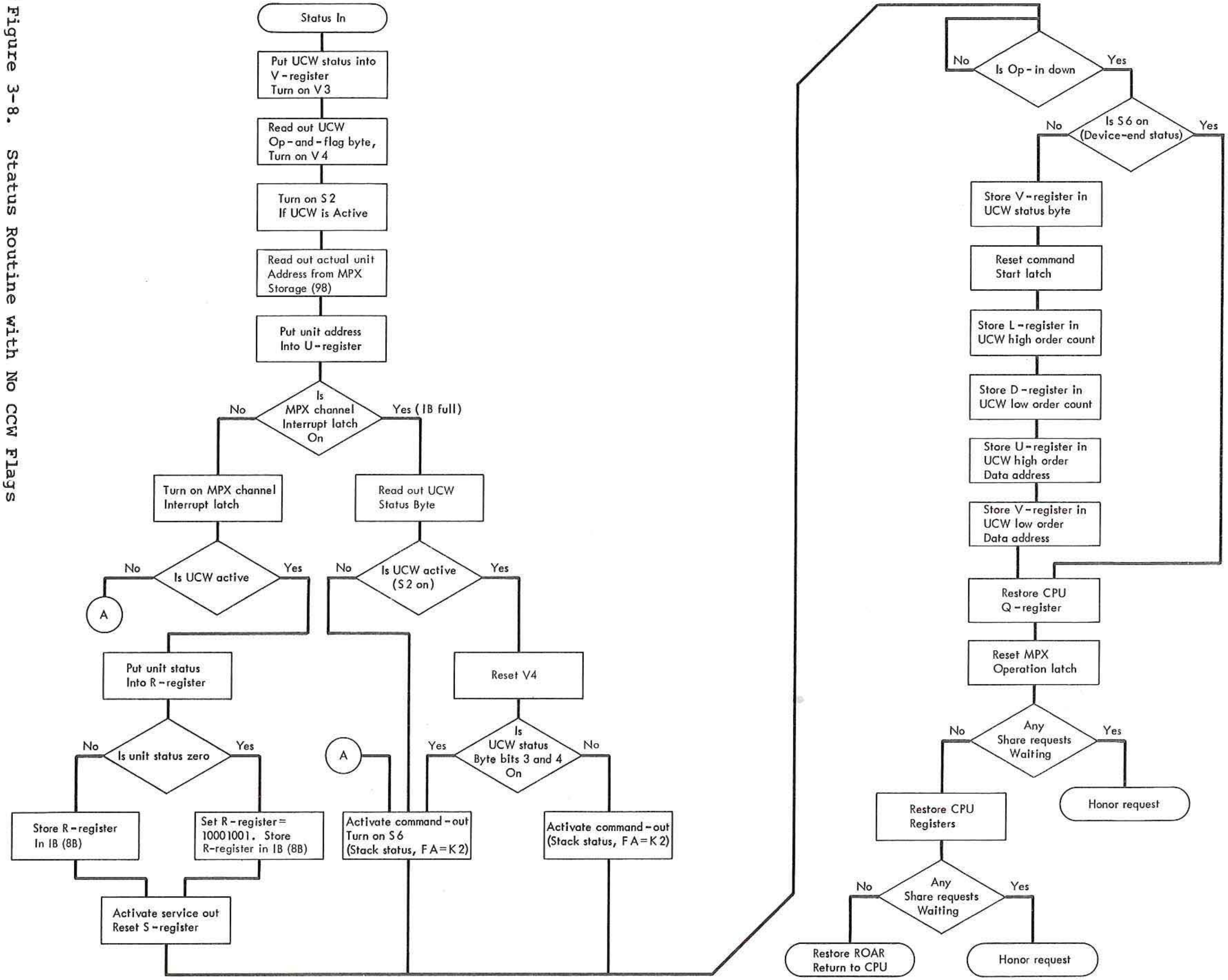
A control unit may present ending status three different ways:

1. During initial selection.
2. Immediately following a data cycle. (This is the case for devices that hold operational-in up until an entire record is transmitted.)
3. By raising request in and raising status-in after address-in falls.

Ending status may appear alone in a status byte or it may be accompanied by some other status such as unit-check status or status-modifier status.

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 Channel Status Word (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. For example, when a UCW op-and-flag byte contains the command chaining bit, the UCW status is status-next or zero, and status-in is not unit-check, unit-exception, control unit-end, or busy status, the channel obtains a new CCW. This is explained in CCW Flag Operations.

● Figure 3-8. Status Routine with No CCW Flags


Status In
Put UCW status into V-register Turn on V3
Read out UCW Op-and-flag byte, Turn on V4
Turn on S2 If UCW is Active
Read out actual unit Address from MPX Storage (98)
Put unit address Into U-register
Is MPX channel Interrupt latch On
No
Yes (IB full)
Turn on MPX channel Interrupt latch
Read out UCW Status Byte
Is UCW active
No
Yes
A
Is UCW active (S2 on)
No
Yes
Put unit status Into R-register
Reset V4
Is unit status zero
No
Yes
Store R-register In IB (8B)
Set R-register = 10001001. Store R-register in IB (8B)
Activate service out Reset S-register
A
Is UCW status Byte bits 3 and 4 On
Yes
No
Activate command-out Turn on S6 (Stack status, FA=K2)
Activate command-out (Stack status, FA=K2)
Is Op-in down
No
Yes
Is S6 on (Device-end status)
No
Yes
Store V-register in UCW status byte
Reset command Start latch
Store L-register in UCW high order count
Store D-register in UCW low order count
Store U-register in UCW high order Data address
Store V-register in UCW low order Data address
Restore CPU Q-register
Reset MPX Operation latch
Any Share requests Waiting
No
Yes
Restore CPU Registers
Honor request
Any Share requests Waiting
No
Yes
Restore ROAR Return to CPU
Honor request

To follow this description, assume the channel receives request-in. The control unit presents status-in in response to command-out with no bits on bus-out (proceed). When the channel receives request-in, it stores the CPU-registers in local storage, polls (selects) the unit, and loads the units UCW into the CPU registers. The control unit now presents status-in.

When the multiplexor microprogram recognizes the rise of status-in, it tests the G-register to determine if the G3 and G6 latches are on. These conditions are set in the UCW op-and-flag byte during the multiplexor share cycle in which the count reduces to zero. Assume both G3 and G6 are on. The microprogram reads out the UCW status byte and begins to assemble a new status byte in the V-register. The microprogram next reads out the UCW Op-and-flag byte and turns on S2 if the UCW is active. The microprogram now reads out the unit address from local storage K8 (hex 98) and places it in the U-register. The microprogram now tests the FT-bus to determine if the multiplexor channel interrupt latch is on (IB full). If the IB is full, the microprogram activates command out. If the IB is full and the UCW is active the channel must determine if the IB contains the channel-end status of the device now presenting status. If this condition does exist the channel microprogram turns S6 on. S6 on indicates device end type of status.

When Op-in falls in response to command-out, the channel microprogram updates the UCW and restores the CPU registers. If S6 is on when Op-in falls the channel does not update the UCW. When a control unit receives command- out, it queues (holds the status in its status register) and deactivates operational-in.

Assume the IB is not full. In this case the microprogram places the unit's address in the IB (K0). When this is accomplished, the microprogram branches on S2 on (UCW active). If the UCW is active, it means 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 IB (K3) and activates service out. When operational-in falls, the microprogram restores the UCW using the contents of the V-register for the UCW status byte.

When S2 is off (UCW not active), it means that the unit is presenting status a second time, normally device-end status. In this case the microprogram sets the IB status (K3) to 0 and, having turned on the IB-full latch, activates command out. This stacks the status at the control unit. This is necessary to prevent multiple data path control units that are connected to two channels from starting an operation on a second channel while the control unit has an interrupt waiting in the multiplexor channel. When operational-in falls, as a result of command out, the microprogram restores the CPU registers to their original state. Remember that if a UCW is not active when the channel receives status-in, the same I/O unit has presented status one time previously; therefore the UCW was updated as a result of the previous status-in and does not have to be updated a second time.

The last microword of the multiplexor trap contains an instruction that turns off the H-register-6 (H6) latch and allows the read-only back-up address-register contents to enter the read-only address register. This branches the microprogram to the point it was discontinued by the request-in.

I/O INTERRUPT

- The multiplexor channel sets up an interrupt when it receives ending status, and command chaining is not specified by the UCW.
- To set up an interrupt, the multiplexor channel turns on its interrupt latch.
- At the end of the current PSW- the CPU honors the multiplexor channel-interrupt.
- The I/O interrupt microprogram routine:
  1. Stores the current PSW in the I/O old PSW location.
  2. Generates the CSW.
  3. Loads the I/O new PSW into the current PSW locations in local storage.
  4. Frees the UCW by resetting the ACTIVE bit
- The CPU executes the I/O program under control of the I/O new PSW.

The multiplexor channel initiates an I/O interrupt when it receives ending status from a control unit, and the UCW op-and-flag byte bit 1 is off (not command chaining). It does this by turning on the multiplexor channel interrupt latch. The CPU executes the I/O interrupt at the end of the current E-phase if I/O interrupts are not masked.

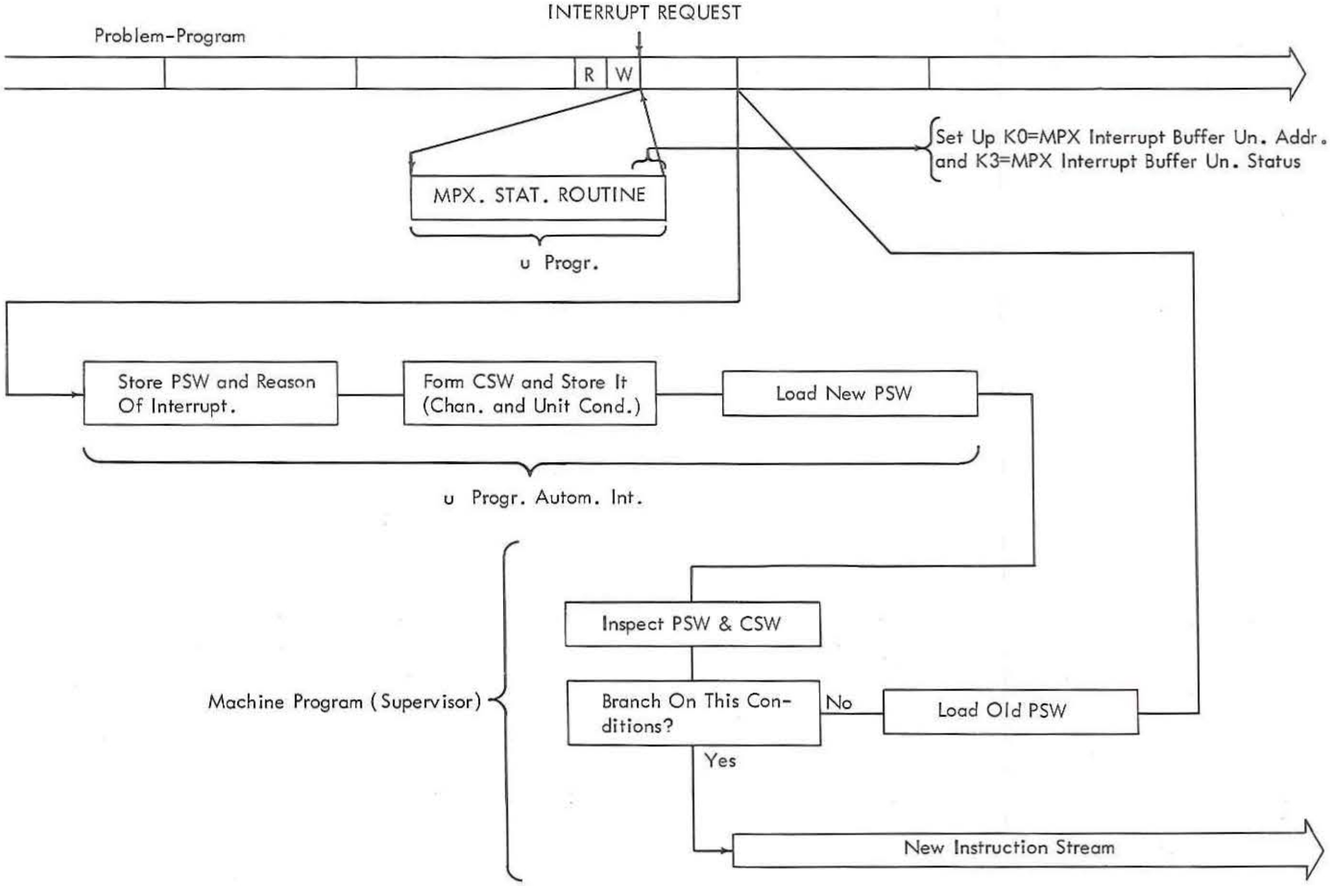

INTERRUPT REQUEST
Problem-Program
R
W
Set Up K0=MPX Interrupt Buffer Un. Addr.
and K3=MPX Interrupt Buffer Un. Status
MPX. STAT. ROUTINE
u Progr.
Store PSW and Reason Of Interrupt.
Form CSW and Store It (Chan. and Unit Cond.)
Load New PSW
u Progr. Autom. Int.
Inspect PSW & CSW
Machine Program (Supervisor)
Branch On This Conditions?
No
Load Old PSW
Yes
New Instruction Stream

Figure 3-9. I/O Interrupt Concepts

An I/O interrupt causes the CPU to:

1. Discontinue the execution of the program being executed under control of the current PSW.
2. Store information in the CSW.
3. Load the I/O new PSW from main storage location 120 (hex 78) into the current PSW locations.
4. Execute the new I/O PSW program.

The CPU cannot honor an interrupt-request until it has completed the execution of the current instruction. Also, the current PSW system-mask bit that corresponds to the channel requesting the interrupt, must be on.

When the CPU has completed the execute phase of the current instruction, the CPU interrupt circuits honor the multiplexor channel interrupt. The interrupt microprogram stores the current PSW in the I/O old PSW location (main-storage position 56), loads the channel status word (CSW, main-storage location 64) with the required information, and loads the I/O new PSW into local storage and CPU registers as the current PSW. After this is accomplished, the CPU returns to I-cycles and starts the execution of the program controlled by the contents of the I/O new PSW. At some later time, the program must load the I/O old PSW into the current PSW, to return to the original program. Figure 3-9 shows the general concept of an I/O interrupt.

PSW STORE

- The PSW store microprogram stores the current PSW in the I/O old PSW location.

The last E-cycle ROS word of each instruction branches the microprogram to the interrupt microprogram routine if an interrupt exists (interrupt latch on).

The first ROS word of the interrupt routine has the instruction Test Interrupt with both X6 and X7 branch indicators set to 1. This instruction is a test to determine the source of the interrupt:

1. External or timer interrupts prevent both the X6 and X7 branches (X6 = 0, X7 = 0).
2. Selector channel 1 interrupt prevents the X7 branch (X6 = 1, X7 = 0).
3. Selector channel 2 interrupt prevents the X6 branch (X6 = 0, X7 = 1).
4. Multiplexor channel allows both X6 and X7 branches (X6 = 1, X7 = 1).

Assume there is a multiplexor channel interrupt. The microprogram now reads out the unit address from local storage, places it in the L-register, and sets the U-register to the address of the multiplexor channel. After the S4 latch is turned on to indicate an I/O interrupt, the microprogram stores the current PSW in the I/O old PSW location (main-storage location 56, address 38).

The interrupt code is the first item stored in the I/O old PSW. Because this is an I/O interrupt, the addresses of the channel and I/O unit, causing the interrupt, are stored as the interrupt code. The microprogram next reads out the storage-protect key and system mask and stores them in their respective positions in the I/O old PSW.

The Instruction Length Code (ILC) condition code (CC), and program mask are not contained in one byte in the current PSW. Therefore, the microprogram must determine what the ILC is by testing the two high-order bits (0 and 1) of the G-register which contains the op code of the last executed instruction. The microprogram also decodes the four-position condition register (local storage K27) into two bits.

After this is accomplished, the microprogram assembles the program mask, condition code, and instruction length code into one byte and stores this byte in the I/O old PSW. The current PSW is now fully stored in the I/O old PSW location. The microprogram now stores the required information in the CSW, located at main-storage location 64 (hex 40).

CSW STORE

- The CSW store microprogram is part of the test I/O execute cycle.
- The G-3 latch being on identifies this routine as a part of an interrupt routine.

The first microprogram word of this routine turns on the G-register 3 and 7 latches. The G-register-7 latch makes the I/O interrupt appear as a test I/O instruction. The microprogram determines the difference between a true test I/O instruction and a I/O interrupt by branching on the G3 latch which is on for interrupts.

The first objective of this routine is to determine if the UCW associated with the unit requesting the interrupt is active. The microprogram determines the address of the units UCW by testing the V-register which contains the unit address. After setting the T-register and the XXH, XH, and XL latches, the microprogram reads out the units UCW op-and-flag byte and tests it for the active bit On. If the UCW active bit is on, the unit's first status is in the interrupt buffer. In all probability, this status will be channel-end status either alone or with other status.

If the UCW active bit is off, the unit has presented status a second time. In all probability, this status will be device-end status, either alone or with other status.

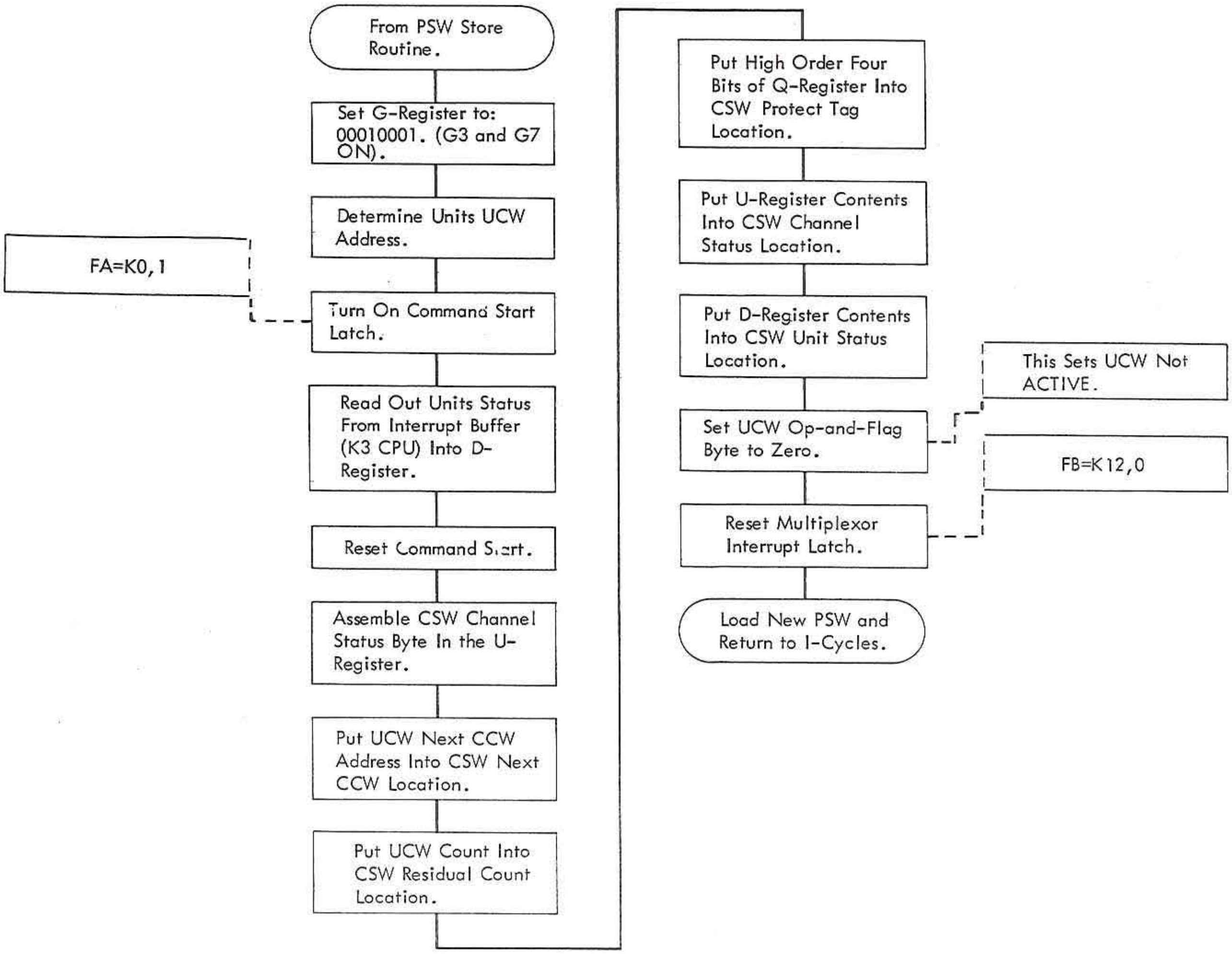

From PSW Store Routine.
Set G-Register to: 00010001. (G3 and G7 ON).
Determine Units UCW Address.
FA=K0,1
Turn On Command Start Latch.
Read Out Units Status From Interrupt Buffer (K3 CPU) Into D-Register.
Reset Command S.art.
Assemble CSW Channel Status Byte In the U-Register.
Put UCW Next CCW Address Into CSW Next CCW Location.
Put UCW Count Into CSW Residual Count Location.
Put High Order Four Bits of Q-Register Into CSW Protect Tag Location.
Put U-Register Contents Into CSW Channel Status Location.
Put D-Register Contents Into CSW Unit Status Location.
This Sets UCW Not ACTIVE.
Set UCW Op-and-Flag Byte to Zero.
FB=K12,0
Reset Multiplexor Interrupt Latch.
Load New PSW and Return to I-Cycles.

Figure 3-10. CSW Store; UCW Active

UCW ACTIVE

- When the UCW is active, the CSW store microprogram:

  1. Assembles a CSW channel-status byte.
  2. Loads the CSW location with the required information.
  3. Resets the multiplexor-interrupt latch.

- The unit status is obtained from the interrupt buffer.

- Figure 3-10 shows the CSW store routine when the UCW is active.

Bit	Op-and-Flag Byte	UCW-Status Byte
0	Chain Data Address (CDA)	Count Zero (1050 Only)
1	Chain Command (CC)	Channel Control Check
2	Suppress Length Indication (SLI)	Interface Control
3	SKIP	First Status Received
4	Program Controlled Interrupt (PCI)	Bit Not Named
5	Active	Incorrect Length
6	Output	Program Check
7	Decrement Data Address	Protection Check

34
00 - Handling Data; Expect data
01 - Device Told to Stop; Expect Status
10 - Interrupt Stacked
11 - Interrupt In IB

Figure 3-11. UCW Status

When the interrupt microprogram determines that the UCW is active, it reads out the unit status from the interrupt buffer (local storage K3) and puts it in the D-register. The microprogram next assembles the CSW channel status byte in the U-register. It does this by combining parts of the UCW op-and-flag and channel-status bytes. Figure 3-11 shows the format of the UCW op-and-flag byte, and the UCW channel status byte.

The CSW store routine now loads the CSW location with its required information:

1. The high order of the next CCW address is set to zero and the two low-order bytes set with the UCW next CCW address.

2. The residual count positions set to the value of the UCW count.

3. The memory-protect-tag position sets to the contents of the Q-register.

4. The channel-status position sets to the contents of the U-register.

5. The unit-status position sets to the contents of the D-register.

When this is accomplished, the microprogram sets the UCW op-and-flag byte to zero, turns off the multiplexor-interrupt latch, and proceeds to load the I/O new PSW into the current PSW.

UCW NOT ACTIVE

- An I/O units UCW is not active when the I/O unit initiates a second interrupt for a CCW.
- The multiplexor channel executes a test I/O sequence with the I/O unit to obtain the units status.
- The CSW store routine sets the entire CSW to zero and then stores the unit's status in the unit status position of the CSW.

When a control unit initiates two interrupts during the execution of a CCW, the UCW will not be active for the second interrupt. This is because the UCW op-and-flag byte is set to zero during the first interrupt. For example, suppose a control unit initiated an interrupt by presenting channel-end status alone. When the CPU executes this interrupt, it sets the UCW to inactive. Therefore, when the CPU executes an interrupt initiated by device-end status from the control unit, the UCW is inactive.

Because the multiplexor channel always stacks the status from a control unit whose UCW is not active and because it does not store the units status in the interrupt buffer, the CSW store routine selects the unit to obtain its status.

Figure 3-12 shows the general sequence of events for the CSW store routine when the UCW is not active.

After the microprogram captures polling on the multiplexor channel by turning on the command-start latch, it reads out the address of the unit that requested the interrupt from the interrupt buffer. The microprogram puts the address on bus-out and activates address-out. Address-out activates select-out. The I/O unit decodes the address on bus-out and activates operational-in.

The multiplexor channel and the I/O unit continue the initial selection sequence with the channel issuing a command-out that is zero. The control unit decodes this as a test I/O.

When the channel receives status-in during this initial selection sequence, the CSW store routine puts the status in the D-register and activates the multiplexor channel service-out line.

When the CSW store routine determines that the I/O control unit has disconnected from the channel by deactivating operational-in, it allows the channel to resume polling.

After setting the IJ-register to the address of the CSW, the CSW store routine sets the R-register to zero and stores the R-register at every position in the CSW. This sets the entire CSW to zero.

The CSW store routine addresses the unit status position of the CSW a second time and stores the contents of the D-register (unit status) in this position. This completes the CSW store portion of the I/O interrupt execution.

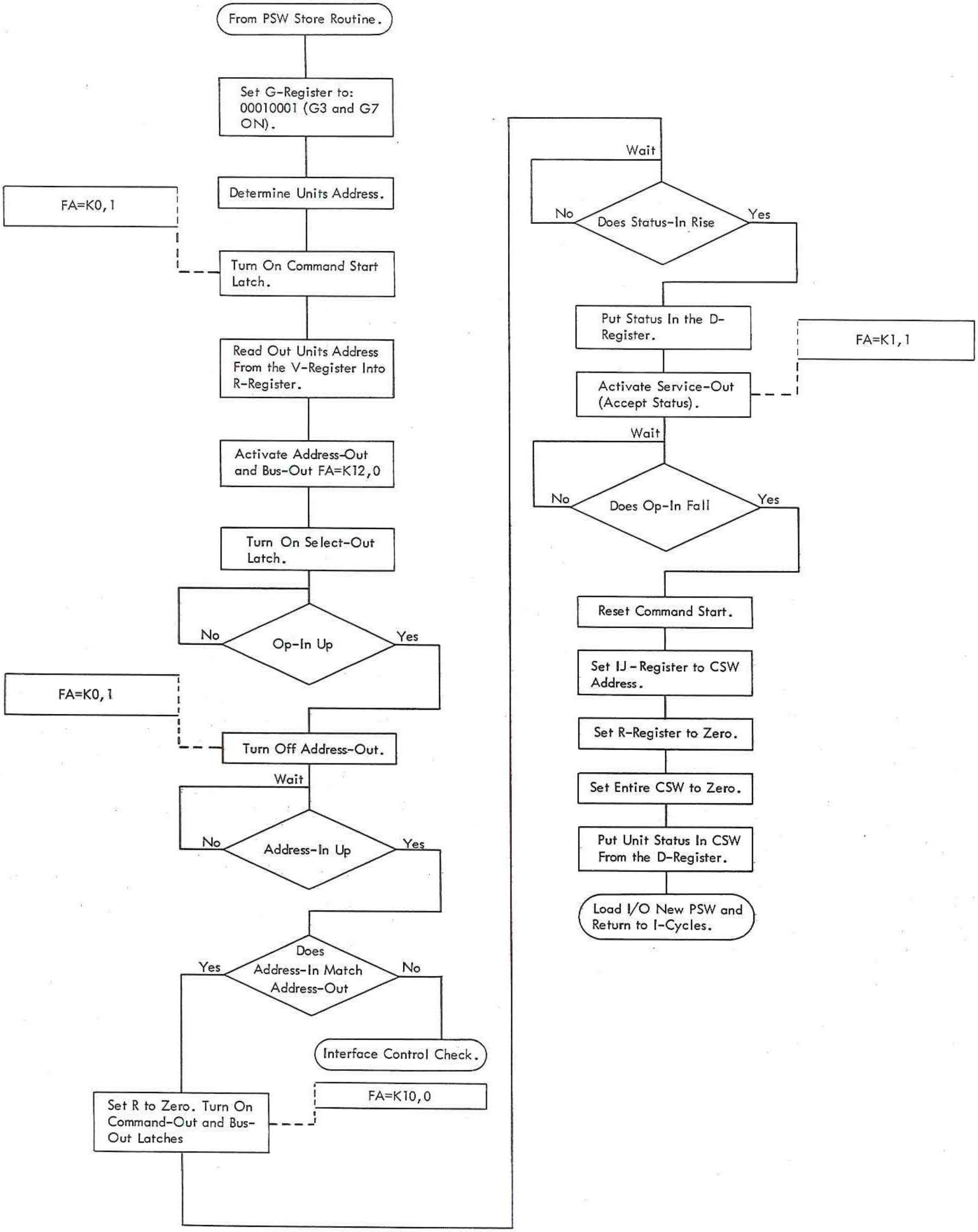

From PSW Store Routine.
Set G-Register to: 00010001 (G3 and G7 ON).
Determine Units Address.
FA=K0,1
Turn On Command Start Latch.
Read Out Units Address From the V-Register Into R-Register.
Activate Address-Out and Bus-Out FA=K12,0
Turn On Select-Out Latch.
No
Op-In Up
Yes
FA=K0,1
Turn Off Address-Out.
Wait
No
Address-In Up
Yes
Yes
Does Address-In Match Address-Out
No
Interface Control Check.
FA=K10,0
Set R to Zero. Turn On Command-Out and Bus-Out Latches
Wait
No
Does Status-In Rise
Yes
Put Status In the D-Register.
FA=K1,1
Activate Service-Out (Accept Status).
Wait
No
Does Op-In Fall
Yes
Reset Command Start.
Set IJ-Register to CSW Address.
Set R-Register to Zero.
Set Entire CSW to Zero.
Put Unit Status In CSW From the D-Register.
Load I/O New PSW and Return to I-Cycles.

Figure 3-12. CSW Store; UCW Not Active

NEW PSW LOAD

- The I/O new PSW loads into the current PSW local storage locations.

At the same time the I/O new PSW loads into local storage certain latches associated with the PSW turn on.

1. The mask latches turn on when the system mask reads out of the I/O new PSW and into local storage (K24).
2. The malfunction-suppress latch turns on if the machine- check mask-bit is 0 (off).
3. The ASCII latch turns on if ASCII is designated by the I/O new PSW.
4. The Q-register sets with the information in the PSW protect key area.

The instruction address enters the IJ-registers and does not enter the local storage. The two-bit new PSW CC decodes into four bits and enters the condition register (local storage K27). If, during this load new PSW routine, the microprogram determines that the high order of the instruction address is not 0, it stores this byte at local storage K16. It also turns on the S-register two latch if the low-order byte of the instruction address is not an even numbered address (position 7 of that J-register is on.

When this PSW is fully loaded, the microprogram branches to I-cycles, and starts the execution of the I/O program.

CCW FLAG OPERATIONS

- Five CCW flags modify the CCW operation.

The CCW contains five flag bits. These bits and their locations in the CCW are:

1. bit 32-CD ; chain data address.
2. bit 33-CC; chain command.
3. bit 34--SLI; suppress length indication.
4. bit 35-SKIP; skip data.
5. bit 36-PCI; program control interrupt.

With the exception of the CC-bit and SLI independently of each other. The CC-bit and SLI bit do not take effect if the CD -bit is on.

The time, in relation to the CCW execution, that a flag bit takes effect varies as follows:

1. The CD -bit takes effect when the count reduces to zero.
2. The CC-bit takes effect when a control unit sends device-end status and no error conditions exists in the channel or device.
3. The SLI-bit takes effect to cause the channel to ignore any wrong length records. This allows command chaining to occur despite any mismatch of specified record length vs actual record length.
4. The SKIP-bit takes effect during the execution of a CCW.
5. The PCI-bit takes effect after its associated CCW is loaded into the CPU registers and the UCW.

CHAIN DATA ADDRESS (CD)

- The CD bit takes effect when the count reduces to zero.
- It allows the channel to fetch the count, data address, and flag byte from the next CCW.
- CD allows the channel to gather information from storage positions that are not adjacent or to place input information into storage locations that are not adjacent.

When the chain-data flag bit of the CCW is on, it is an indication to the channel to fetch the count, data address, and flag byte from the next CCW when the count in the UCW is reduced to zero. This permits the channel to use core storage locations that are not adjacent positions, in the execution of a CCW. This linking together of data address is called data chaining.

Data chaining is controlled by the G-register-0 latch. This latch turns on when the UCW loads into the CPU registers and the UCW op-and-flag CD bit is on.

During any data loop cycle in which the working count has been reduced to zero, a test is made by the data loop microprogram to determine if the CD flag is on (G0=1). If this condition exists, the (CD routine is entered for the purpose of obtaining replacements for the working count, data address and flags from the next CCW.

This microprogram routine reads out the next CCW-address from the UCW and places it in the IJ-registers. The CD microprogram uses the IJ-registers to read out portions of the next CCW. The microprogram checks the command byte of the next CCW. If it is a transfer-in-channel command, the microprogram adds eight to the transfer-in-channel command address and begins to read out the <u>next</u> CCW.

Assume the CCW command is not a transfer-in-channel command. In this case, the microprogram:

1. Reads the CCW count into the LD-register.
2. Reads the CCW data address into the UV-registers.
3. Reads the CCW flag byte into the high positions (0-4) of the G-register.
4. Determines the next CCW address and stores it in the UCW.

The CPU registers now contain new multiplexor channel operating information. The microprogram returns to the data loop entry STI, SVI branch.If operational-in is inactive, the microprogram begins to store the UCW and restore the CPU as it does when operational-in falls at the end of a normal multiplexor share cycle.

If operational-in is still active at the end of the CD routine, the microprogram waits for operational-in to fall or service-in or status-in to rise. When one of these conditions occurs, the microprogram branches to the associated routine.

A program check occurs during the CD routine if the new CCW count is zero. the data address is invalid, the three low-order bits of the flag byte are not zero, or the next CCW-address is zero (memory wrap).

COMMAND CHAINING

- 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 chain data address flag bit off and the command chaining flag bit On.
- When the channel successfully executes a CCW without errors and receives its associated device-end status, the channel reads out the next CCW and begins its execution.
- The presence of status-modifier status along with device-end status requires the channel to bypass the next CCW in favor of the CCW following it.

- The channel suspends polling while command chaining.
- The channel activates suppress out while command chaining.

The multiplexor channel may execute a series of CCW's that apply to one particular control unit and device. When the execution of one CCW is complete, the channel reads out the next CCW in ascending order and begins its execution. This linking together of CCW's is called command chaining.

Command chaining is controlled by the CCW flag byte. Command-chaining is specified when CCW flag byte chain-data- address bit (CD ; bit 0) is off and the chain command bit (CC; bit 1) is on. This condition turns on the UCW command- chaining bit when the channel generates the UCW associated with the CCW. The UCW command chaining bit is bit 1 of the UCW op-and-flag byte, and is held in G1.

Figure 3-13 shows the format of the a UCW. Notice that the last two bytes of a UCW contain the address of the next CCW. This information is put into the UCW by the channel microprogram at the start of every CCW execution.

When the channel successfully executes a CCW that has specified command chaining, the channel reads out the next CCW and begins its execution.

If the channel does not successfully execute a CCW that has specified command chaining, it ignores the command chaining specification. That is, it sets up an interrupt if the multiplexor interrupt latch is off (IB empty) or stacks the status at the control unit if the IB is full.

Another alternative that may occur is the channel may be required to bypass the next CCW and execute the CCW following it. This condition occurs when the channel receives status-modifier status along with device-end status and command chaining is specified by the UCW.

The channel does not read out and begin the execution of the next CCW until it receives device-end status from the I/O device. If the channel receives channel-end alone, it responds to the I/O unit with service-out along with suppress-out. The channel waits for device-end status before beginning the command chaining routine.

When the channel receives device-end status and begins to read out the next CCW, it activates suppress-out. This is the channel's indication to an I/O device that it is command chaining. The channel also suspends polling while it is obtaining a new CCW.

Figure 3-14 shows the general logic flow of the command- chaining sequence.

UNIT CONTROL WORD
UCW

000	001	010	011	100	101	110	111
Channel Status	Op. and Flags	Count High	Low	Data Address High	Low	Next CCW Address High	Low

Channel Status Byte

0 - Count Zero (1050 Only)
1 - Channel Control Check
2 - Interface Control Check
3 - First Status Received Coded explanation below.
4 - (No Name)
5 - Incorrect Length
6 - Program Check
7 - Protection Check

Op. and Flags Byte

0 - Chain Data Address (CDA)
1 - Command Chaining (CC)
2 - Suppress Length Indication (SLI)
3 - Skip
4 - Program Controlled Interrupt (PCI)
5 - Active
6 - Output (Write)
7 - Decrement Data Address

3	4	
0	0	Handling Data; Expecting Data.
0	1	Device Instructed to Stop; Expecting Status.
1	0	Status Stacked at Control Unit.
1	1	Status Is in Interrupt Buffer (IB).

● Figure 3-13. Unit Control Word Format

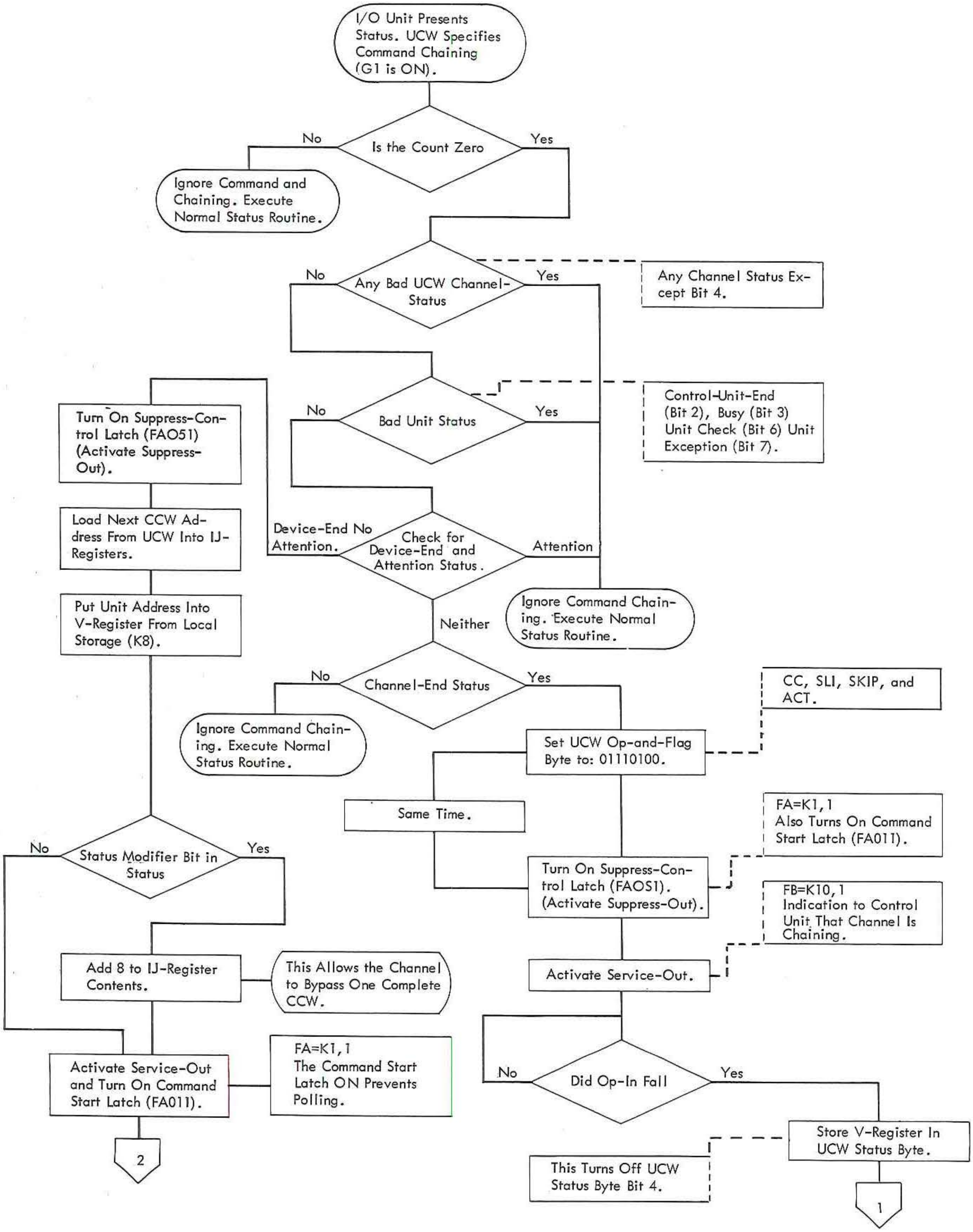

I/O Unit Presents Status. UCW Specifies Command Chaining (G1 is ON).
Is the Count Zero
No
Yes
Ignore Command and Chaining. Execute Normal Status Routine.
Any Bad UCW Channel-Status
No
Yes
Any Channel Status Except Bit 4.
Bad Unit Status
No
Yes
Control-Unit-End (Bit 2), Busy (Bit 3) Unit Check (Bit 6) Unit Exception (Bit 7).
Turn On Suppress-Control Latch (FA051) (Activate Suppress-Out).
Load Next CCW Address From UCW Into IJ-Registers.
Put Unit Address Into V-Register From Local Storage (K8).
Device-End No Attention.
Check for Device-End and Attention Status.
Attention
Ignore Command Chaining. Execute Normal Status Routine.
Neither
Channel-End Status
No
Yes
Ignore Command Chaining. Execute Normal Status Routine.
CC, SLI, SKIP, and ACT.
Set UCW Op-and-Flag Byte to: 01110100.
Same Time.
FA=K1,1 Also Turns On Command Start Latch (FA011).
Turn On Suppress-Control Latch (FAOS1). (Activate Suppress-Out).
FB=K10,1 Indication to Control Unit That Channel Is Chaining.
Status Modifier Bit in Status
No
Yes
Add 8 to IJ-Register Contents.
This Allows the Channel to Bypass One Complete CCW.
Activate Service-Out.
Activate Service-Out and Turn On Command Start Latch (FA011).
FA=K1,1 The Command Start Latch ON Prevents Polling.
Did Op-In Fall
No
Yes
Store V-Register In UCW Status Byte.
This Turns Off UCW Status Byte Bit 4.
2
1

Figure 3-14. Command Chaining Sequence (Part 1 of 2)

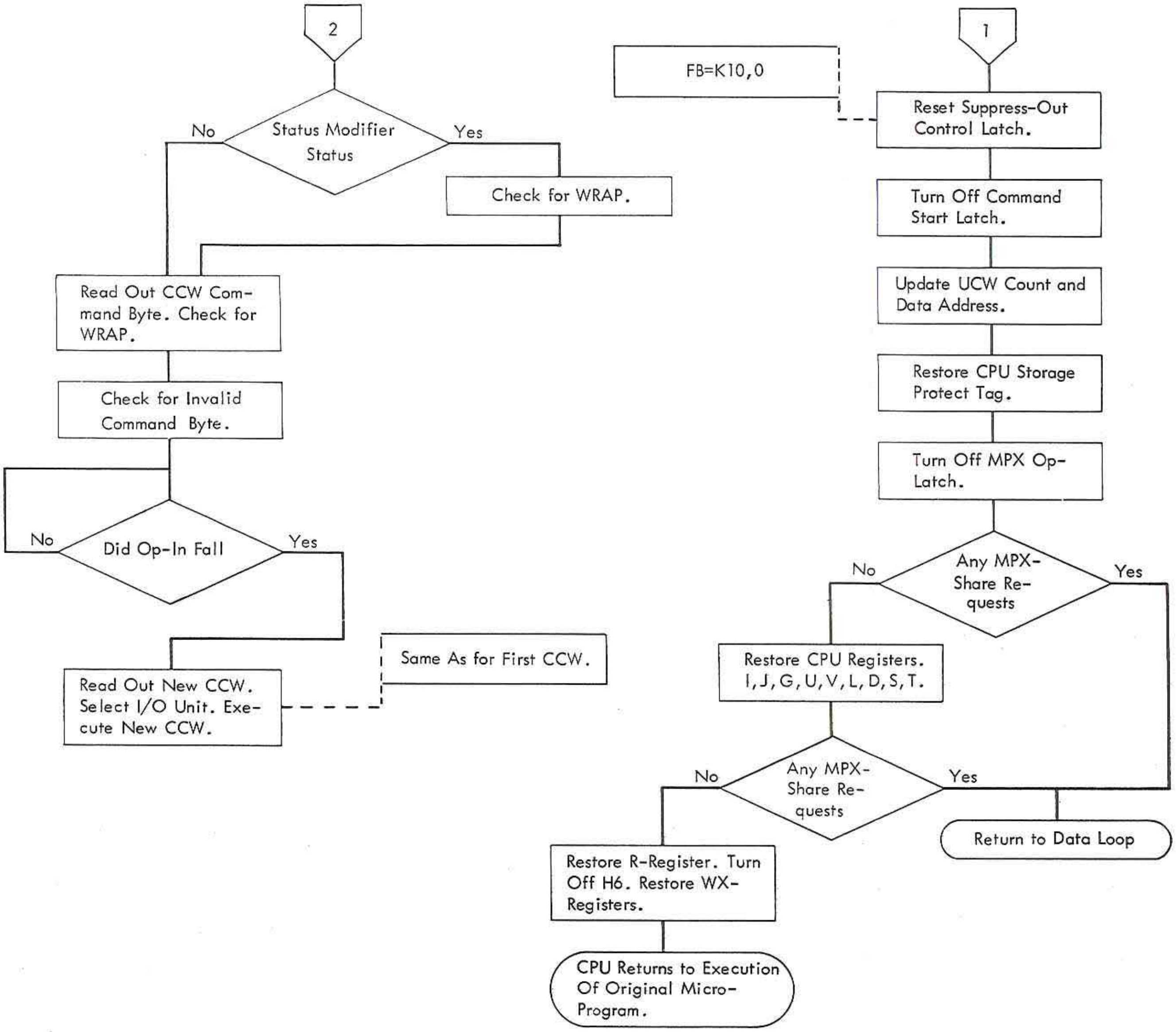

2
Status Modifier Status
No
Yes
Check for WRAP.
Read Out CCW Command Byte. Check for WRAP.
Check for Invalid Command Byte.
Did Op-In Fall
No
Yes
Read Out New CCW. Select I/O Unit. Execute New CCW.
Same As for First CCW.
FB=K10,0
1
Reset Suppress-Out Control Latch.
Turn Off Command Start Latch.
Update UCW Count and Data Address.
Restore CPU Storage Protect Tag.
Turn Off MPX Op-Latch.
Any MPX-Share Requests
No
Yes
Restore CPU Registers. I,J,G,U,V,L,D,S,T.
Any MPX-Share Requests
No
Yes
Return to Data Loop
Restore R-Register. Turn Off H6. Restore WX-Registers.
CPU Returns to Execution Of Original Micro-Program.

Figure 3-14. Command Chaining Sequence (Part 2 of 2)

SUPPRESS LENGTH INDICATION (SLI)

- The SLI-bit allows the channel to ignore incorrect length conditions that occur when a unit whose UCW count is Not zero activates status-in.
- The SLI bit is effective only if the CD -bit is off.
- Figure 3-15 shows how the SLI-bit functions.

The SLI-bit is bit-2 of a CCW flag byte. When the channel generates a UCW from the CCW information, bit-2 of the UCW op-and-flag byte becomes the SLI-bit. SLI is held in the G-register 2 latch (G2) during subsequent channel operations.

The SLI-bit is effective only if the UCW CD -bit (bit-0 of the op-and-flag byte) is off.

To understand the function of the SLI-bit, first consider the channel's operation when the SLI-bit is off. In this case, when an I/O device presents status-in to the channel and the I/O units' UCW count is Not zero, the incorrect length bit of the UCW channel-status byte turns on. This condition is subsequently placed in the CSW, and prevents command chaining from occurring if CC is specified.

When a UCW has its SLI-bit on, this condition is suppressed. The channel responds to the status in with service-out and initiates an interrupt if not command-chaining, or it begins the command-chaining sequence if the CC-bit of the UCW op-and-flag bit is on.

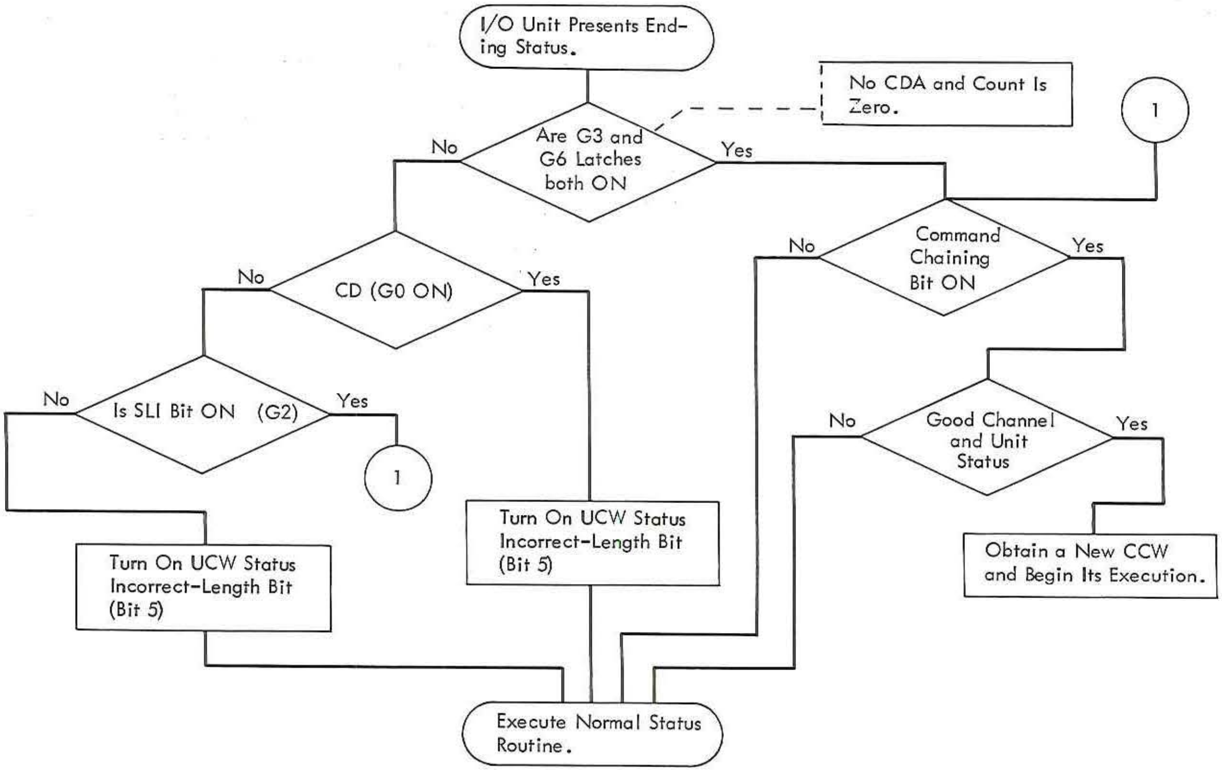

I/O Unit Presents Ending Status.
No CDA and Count Is Zero.
1
No
Are G3 and G6 Latches both ON
Yes
No
Command Chaining Bit ON
Yes
No
CD (G0 ON)
Yes
No
Is SLI Bit ON (G2)
Yes
1
No
Good Channel and Unit Status
Yes
Turn On UCW Status Incorrect-Length Bit (Bit 5)
Turn On UCW Status Incorrect-Length Bit (Bit 5)
Obtain a New CCW and Begin Its Execution.
Execute Normal Status Routine.

Figure 3-15. SLI-Bit Functions

SKIP

- The SKIP-bit is effective only on input operations.
- The SKIP-bit does not become a part of the UCW op-and-flag byte for output operations.
- It is effective during the CCW-execution.
- It suppresses the transfer of information to storage.
- The SKIP-bit allows the channel to store specific input records in CPU storage.
- Figure 3-16 shows how the SKIP-bit modifies an input operation.

The SKIP-bit, bit-3 of the CCW flag byte and UCW op-and-flag byte allows the channel to receive data form a unit but not store the data in CPU storage.

When the channel receives service-in and the SKIP- bit is on (G-register 3 latch on), the channel does function normally except for the data loop. Instead of executing the data loop, the channel updates the count and responds with service-out. This continues until the count is exhausted at which time the normal status sequence takes place.

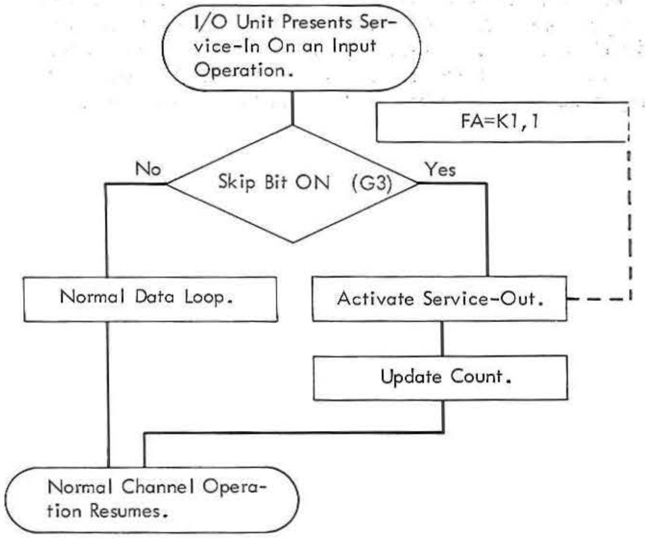

I/O Unit Presents Service-In On an Input Operation.
FA=K1,1
No
Skip Bit ON (G3)
Yes
Normal Data Loop.
Activate Service-Out.
Update Count.
Normal Channel Operation Resumes.

Figure 3-16. Skip-Bit Functions

PROGRAM CONTROLLED INTERRUPT (PCI)

- The PCI-bit allows the program to initiate an interrupt.
- The PCI-bit sets up an I/O interrupt when the interrupt latch is off (IB empty).
- When the CPU executes an interrupt initiated by the PCI-bit, the PCI-bit becomes part of the channel-status byte that is stored in the CSW.

The program-controlled interrupt bit is bit-4 of the CCW flag byte. It also becomes bit-4 in the UCW op-and-flag byte that is generated from the CCW. The PCI bit in a CCW initiates an on-the-fly interrupt without, in any other way, effecting the channel operation. It may be used by the program to initiate an interrupt at any point in a chain of CCW's to determine how the operation is progressing. For example, the multiplexor channel may be executing a chain of ten CCW's. Normally no interrupts will occur until after the tenth CCW is executed. By the use of the PCI-bit, the program can force an interrupt at any point in the chain to determine the status of the operations up to that point.

The PCI-bit takes effect two times:

1. It takes effect during the execution of the CCW in which it occurs, (Figure 3-17).
2. It takes effect during the CSW store routine of an I-O interrupt. 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. Remember the main purpose of an I/O interrupt is to generate the CSW. The CSW is then interrogated by the I/O program and action is taken according to the conditions found in the CSW.

When the CPU executes an interrupt initiated by the multiplexor channel, it resets the PCI-bit in the UCW associated with the unit that initiated the interrupt. This occurs whether or not the interrupt was caused by the PCI-bit.

During the course of chaining (CC or CD), the PCI-bit is propagated to the new UCW if it is present in the UCW just completed. This will be the case when the CPU has not executed the PCI-initiated interrupt, or the PCI-bit has been unable to initiate an interrupt. The latter condition occurs when a burst mode device is operating on the multiplexor channel.

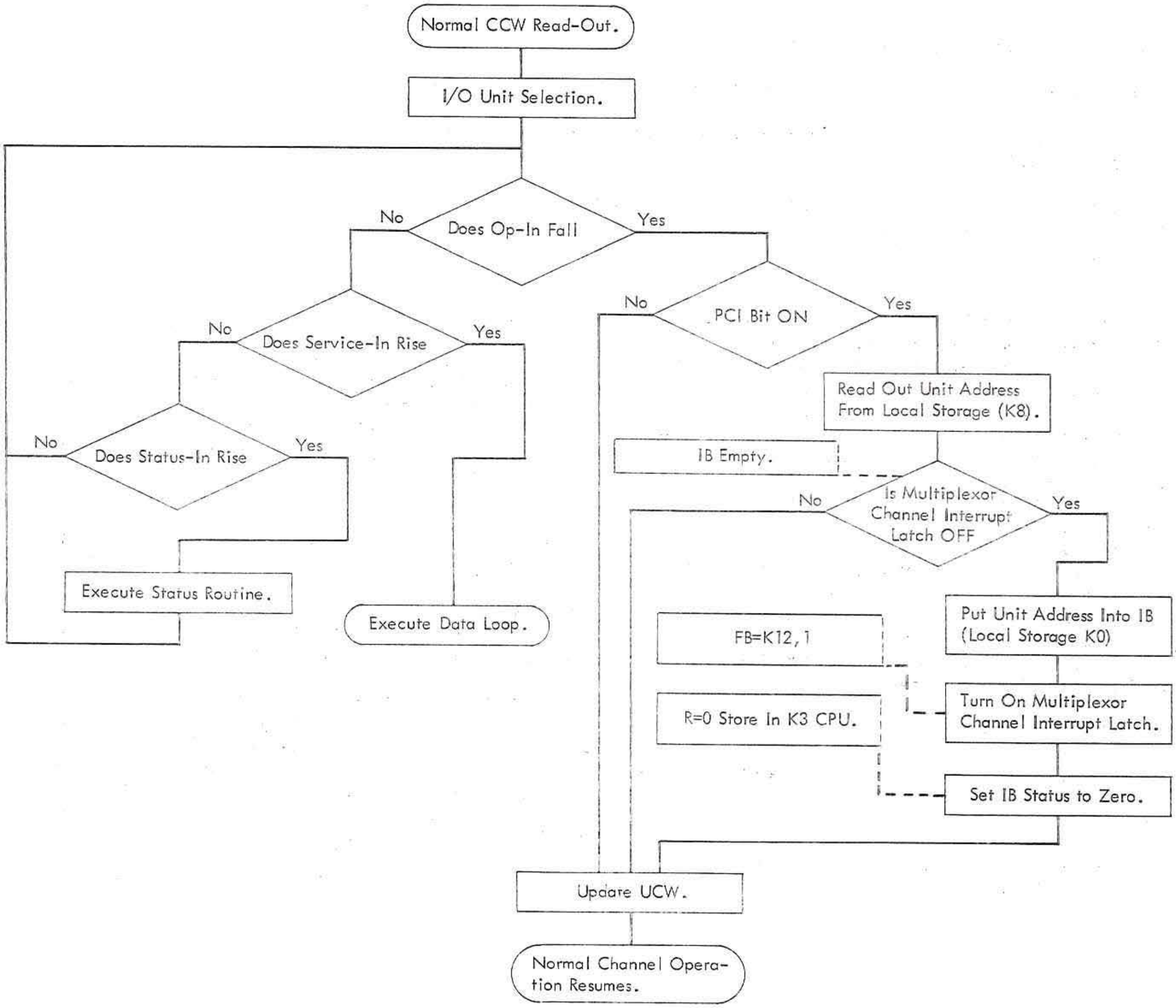


Figure 3-17. PCI-Bit Operation

START I/O TRANSFER-IN-CHANNEL (TIC) COMMAND

- The TIC command does not involve any I/O device operation.
- The channel obtains the CCW referenced by the TIC CCW data address.
- A TIC CCW cannot be the first CCW in a chain.
- Two TIC CCW's in succession cause a program check.

The transfer-in-channel (TIC) command is a command to the channel and does not involve any I/O device operation. It is a command to the channel to obtain and execute the CCW whose high order byte location is referenced by the TIC CCW data address.

The TIC command is used only in a chain of commands and may not be the first command of the chain. Also two TIC commands in succession cause a program check.

The TIC command is used mainly in conjunction with the search command.

For example, look at the following chain of CCW's.

CCW1, Search

CCW2,TIC (data address is address of CCW1)

CCW3, Write

CCW1 is a search. If the search is not successful, CCW2 causes CCW1 to be executed again. This loop will continue until the search is successful. When this happens, the I/O device presents status-modifier status along with channel-end status. This allows the channel to bypass the TIC CCW and execute the write CCW.

Command Chaining to a TIC command begins the same way as for any other CCW command. When the TIC command is recognized during the CCW check sequence its data address is placed in the I J registers. The channel now reads out the CCW addressed by the I J - registers and executes a normal CCW check, CCW load operation.

INITIAL PROGRAM LOAD (IPL)

- The purpose of IPL is to initiate processing after power is turned on or when the contents of storage are unsuitable for further processing.
- IPL is started by selecting an input device and pushing the Load button.

The IPL procedure resembles a start-I/O 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, the command-chain flag on, the SLI flag on, and a command address of zero.

The IPL routine reads new information into the first three double words of storage. The remainder of the IPL program can be placed in any desired location of storage.

The IPL procedure is started by selecting a channel with console switch G and an input device with the HJ connecting switches and pushing the Load button. This causes a system reset; turns on the Load light, and initiates a read operation from the selected input device.

The system reset stops the CPU and suspends all instruction processing, interruptions, and timer updating. It also resets all channels and on-line control units and I/O devices. The contents of the General and Floating-Point registers remain unchanged.

Following the system reset, the selected input device starts reading. The first 24 bytes read enter storage locations 0 to 23. Storage-protection and possible incorrect-length errors are ignored. The double-word read into location eight is used as the CCW for the next input command. When chaining is specified in this CCW, the operation proceeds with the CCW in location 16. The double word at location 0 becomes the IPL PSW.

After the input operation is completed, the channel and device address is stored in the IPL PSW interrupt code (bits 21 to 31 of the first double word in storage). Bits 16 to 20 are zeros, and bits 0 to 15 remain unchanged. The CPU then fetches the IPL PSW from location zero as a new PSW and changes from the stopped to the operating state. Operation proceeds under control of the new PSW and the Load light turns off. If the I/O operations and PSW loading are not satisfactory, the CPU stops and the Load light remains on. The MPX channel's execution of the IPL routine is the same as it's execution of a start I/O routine with the exception of the initial set up and the ending procedure.

When the load key is pressed, it causes a system reset. When the load key is released the load-key latch (KC 041) turns on. The output of the load-key latch turns on the load-indicator latch (KC 051). The output of this latch turns on the load-request stacking latch (KM 062). If no higher priority stacking latches are on, the load-request stacking latch generates an IPL pulse which in turn forces the first micro word address of the IPL routine (002) into the ROSAR (WX-registers).

The IPL routine first generates a CCW and places it in main storage locations 0-7. A read command is placed in byte 0, a count of 24 (hexadecimal 18) enters byte 7, and the CC and SLI flags enter byte 4. Also during this routine the channel and unit address enter the U and V-register from the HJ switches respectively and all of the H-registers latches with the exception of the H6 latch (MPX hold) are turned on to prevent any but MPX traps.

The channel now reads out this CCW from storage locations 0-7, and checks it for validity the same as for a start I/O instruction. When this is complete the channel microprogram loads the CCW into the CPU data flow register while selecting the unit specified by the HJ-switches. From this point the channel and I/O device operate normally until the unit deactivates operational-in.

When op-in falls, the channel exits from the data loop and updates the UCW as for a start I/O. When this is complete the channel micro program does not attempt to restore the CPU as for a start I/O but enters the IPL wait loop, microprogram word 682. IPL MPX share requests are handled by the channel in the same way as for start I/O operations. When the selected device presents status-in, the channel tests the status for device-end or channel-end status. Any other status results in an IPL stop, microprogram word 6FF. When the status is good (channel-end or device-end) the channel resets the IPL latch and H-registers latches. When Op-in falls, the channel stores the unit and channel address into the interrupt code location of the IPL PSW (main storage byte locations 3 and 4. When this is complete, the channel loads the IPL PSW into the current PSW locations, after which the CPU begins executing instructions under control of the IPL PSW.

If for some reason, a device other than the device specified by the HJ switches presents status, the channel issues command-out. This response stacks the status at the control unit. The 2030 enters the IPL wait loop until the next MPX share request.

TEST I/O

- The object of a test I/O instruction is to store a CSW.
- It does not require a CCW.
- The supervisor program issues test I/O instructions to obtain status from I/O devices when the channel interrupts are masked off by the PSW system mask.

The supervisor program issues the test I/O instruction to obtain the status of a particular I/O device. The main use of the test I/O instruction is to obtain the status of I/O devices when channel interrupts are masked off by the PSW system mask.

The test I/O instruction has the same format as a start I/O instruction and its execution is begun in the same manner as the start I/O instruction. After the CAW has been read out the channel differentiates between a start I/O and test I/O and begins the test I/O routine.

In order to execute a test I/O instruction, the channel first determines if the UCW associated with the specified device is ACTIVE or AVAILABLE. It does this by reading out the UCW op-and-flag byte and testing its active bit

UCW AVAILABLE: When the UCW is not active it means that either the I/O device involved has status stacked or the I/O device 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 zero command. When the device responds to command-out with status-in, the channel tests the status for a zero condition. If the status is zero the channel sets PSW condition code 0 and returns to I-cycles. When the status is not zero, the channel puts the status in the D-register and activates service-out. After the I/O device deactivates operational-in, the channel sets the entire CSW to zero, addresses the CSW unit-status byte location a second time and places the unit status, now in the D-register, into this byte location. When this is complete the channel sets PSW condition code 1 and returns the CPU to I-cycles.

When the channel receives status-in in response to address-out (control unit busy), it puts the status in the D-register and activates service out. When status-in falls, the channel sets the CSW to zero and then sets the CSW unit status byte with the D-register contents (status-in). When this is complete the channel sets PSW condition code 1 and returns the CPU to I-cycles.

UCW ACTIVE: When the UCW is active it means one of three things:

1. The subchannel (UCW) is busy either with the specified device or another device.
2. The subchannel has an interrupt condition waiting in the interrupt buffer. This may be an interrupt from the specified I/O device or another I/O device.
3. Status is stacked at the I/O device.

When the UCW is ACTIVE, the channel reads out the UCW op-and-flag byte and tests it to determine if bit 3 is on or off. When bit 3 is off it indicates that the specified device has not presented status and is therefore working. In this case the channel sets PSW condition code 2 and returns the CPU to I-cycles.

If bit 3 is on, indicating a device associated with the UCW has presented status, the channel reads out the UCW hi-order data address and compares this address with address in the V-register I/O control-unit-and device address). An equal comparison means that the device specified by the test I/O instruction has presented status.

Note: During status-in MPX share-cycles, the address of the unit presenting status is placed in the UCW data address.

This comparison is necessary because a UCW may be associated with more than one I/O device and 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 in the PSW (subchannel working) and returns the CPU to I-cycles.

Assume bit 3 of the UCW op-and-flag byte is on, and the address in the UCW data address is the address of the I/O device specified by the test I/O instruction. In this case the channel must find out where the status is. The status may be in the interrupt buffer or it may be stacked at the I/O control unit.

When the status is in the IB, the channel puts the status in the D-register and generates the CSW from the UCW and the unit status that is now in the D-register.

After storing the CSW, the channel turns off the MPX interrupt latch (in FB-reg), sets the UCW op-and-flag byte to zero and returns the CPU to I-cycles.

When the status is not in the IB, the channel executes an initial selection sequence with the I/O device specified by the test I/O instruction. When the I/O device presents status to the channel in response to Command-out the channel executes the CSW store routine, storing a full CSW:

1. protect key
2. Next CCW address
3. Unit status
4. Channel status

After storing the CSW the channel sets the PSW condition code to 1 and returns the CPU to I-cycles.

HALT I/O

- The halt I/O instruction discontinues operation between a particular I/O device and the channel.
- The channel executes the halt I/O instruction before returning the CPU to I/cycles
- The halt I/O instruction does not require a CCW.
- The channel issues a halt I/O (stop) to an I/O device by activating the address-out tag line while the device has its operational-in active, and select-out is down.

The halt I/O instruction is issued by the supervisor program to discontinue operation between a particular I/O device and the channel. Response of I/O devices to a halt I/O instruction varies from one device to another and is covered in the Field Engineering manual of instruction for that device.

The channel begins the execution of a halt I/O instruction in the same manner as the start I/O instruction. That is, the PSW is checked to insure that the CPU is in supervisory state, and part of the CAW is read out. This occurs before the channel makes the test to determine which I/O instruction is being issued to the channel.

When the channel recognizes the halt I/O instruction, it discontinues reading out the CAW and branches to its microprogram routine that develops a UCW address from the I/O control unit and device address. The channel now reads out the UCW op-and-flag byte to determine if the UCW is active.

UCW ACTIVE. When the UCW active bit is on, it indicates that the CPU has not executed an interrupt (stored a CSW) initiated by the I/O device associated with the UCW. Therefore, the channel must read out the UCW status byte and test bits 3 and 4 to determine if the I/O device has an interrupt waiting in the interrupt buffer. This is the case when both bits 3 and 4 of the UCW status byte are on.

When an I/O device has an interrupt waiting in the interrupt buffer, it is not operating with the channel. Therefore, the channel does not execute the halt I/O interface sequence, instead the channel sets the PSW condition code to zero and returns the CPU to I/cycles.

When the I/O device does not have an interrupt waiting, the channel selects the device, and activates address-out after the I/O device has activated operational-in and address-in. Address-out and operational-in both active at the same time while select-out is down at the I/O device means halt I/O (STOP).

The halt I/O interface sequence is:

1. Channel activates address-out and bus-out.
2. Channel activates select-out and hold-out.
3. The I/O device activates operational-in.
4. The channel deactivates address-out.
5. The channel deactivates select-out.
6. The I/O device activates address-in.
7. The channel activates address-out.
8. The I/O device deactivates operational-in and address-in.
9. The channel deactivates address-out.

When the interface sequence is complete, the channel turns off the UCW op-and-flag byte CC and CD bits, and sets the UCW status byte to 10001000 (88).

The channel now sets the UCW op-and-flag byte to zero, sets the PSW condition code to 1, and returns the CPU to I/cycles

UCW NOT ACTIVE: The halt I/O routine for an available UCW is the same as for an active UCW with this exception: The channel does not have to test the UCW status byte for bits 3 and 4 On (interrupt waiting in IB). This is because, for a UCW to become available, the I/O device must have had an interrupt executed by the CPU and, therefore, cannot have an interrupt waiting in the I.B.

When the channel finds the UCW available, it executes the halt I/O interface sequence, sets the UCW status to 10001000 (88), stores all zeros in the CSW, sets the PSW condition code to 1, and returns the CPU to I-cycles.

The channel does not set the UCW op-and-flag byte to zero for an available UCW, because it is already zero.

TEST CHANNEL

- Executed to determine the status of a particular channel.
- Sets a condition-code in the PSW.
- The condition-code set indicates the status of the addressed channel.

The test-channel command determines the status of a particular channel. It is executed only with the CPU in the monitor state and has no effect on the channel. The test-channel command sets a condition-code in the PSW that indicates the current state of the addressed channel.

The test-channel operation begins like a start-I/O; however, when the addressed channel is selected, a microprogram branch occurs to the test-channel routine. In this routine the channel is checked for a busy status or an interrupt condition, and a condition code is set.

Condition codes for the test-channel instruction are:

Code 0 - Channel available.
Code 1 - Interrupt condition in channel,

MPX ERRORS

- The MPX channel can present four types of errors to the CPU:
  1. Channel-control check (CCC)
  2. Interface-control check (ICC)
  3. Memory-protection check (MPC)
  4. Program check (PC)

- The Error routine:
  1. Logs the error for CCC and ICC errors.
  2. Clears the interface
  3. Gets a CSW stored with the error information

When an error occurs that is chargeable to the MPX channel, the channel makes every attempt to disconnect the I/O device from the interface and store either a partial CSW or a full CSW. In some cases an error occurs before a unit becomes connected to the interface. In this case, the channel must merely store the CSW. The time at which a CSW is stored depends on when the error occurs. In some cases the CSW is stored by the MPX error routine, but in other cases the MPX error routine creates a condition that makes the I/O device initiate an interrupt, which is handled in a normal manner.

The MPX error routine must also log the error when a channel-control or interface control check occurs. The MPX log area is bytes 129, 130, 131 of main storage (hexadecimal addresses 81, 82, and 83). The MPX error routine stores the following information in the log locations.

1. Byte location 129: The MPX error routine loads byte 129 with either the contents of the machine check register or the contents of the L-register which is the catalog number. The contents of the machine-check register enter byte 129 when the MPX error routine is entered from the machine -check trap (channel-control checks). The contents of the L-register (catalog number) enter byte 129 on interface control checks. The catalog number is formed in the L-register whenever a program, or interface control check occurs. It indicates where the error occurred (at what microprogram word).

2. Byte 130: Byte 130 is the indicator byte. The four high-order bits serve as a counter of the number of errors logged by the MPX error routine. It is updated each time an error is logged. Bit 4 is turned on by the first error. It indicates that at least one error has occurred. When the MPX error routine stores the catalog number in byte 129, it turns bit 5 on. When bit 5 is off the contents of the machine check register is in byte 129. Bits 6 and 7 together inform you of when Op-in becomes inactive. When an error occurs when op-in is down, the MPX error routine sets both bits 6 and 7 off. When op-in falls only after the MPX error routines executes a halt I/O sequence, the error routine sets bit 6 and 7 on. If op-in falls only after a selective reset by the MPX error routine, the error routine sets bit 6 on bit 7 off. When op-in doesn't fall after all efforts of the MPX error routine to disconnect the I/O device, the error routine sets both bits 6 and 7 on.

3. Byte 131: Whenever the MPX error routine logs an error it stores the address of the I/O control unit and device in byte location 131.

ERRORS WHICH ARE NOT PROGRAM CHECKS

- These errors are:
  1. Channel-control checks
  2. Interface-control checks
  3. Protection checks

- For these errors the MPX error routine:
  1. Logs the error (no logout on protection checks).
  2. Clears the interface when possible and creates an interrupt.

Channel-control checks, interface control checks and protection checks are all handled in a similar manner by the MPX error routine. The basic difference between these three checks is their cause.

CHANNEL CONTROL CHECKS: When the MPX error routine is entered from the machine check trap a channel control check has occurred. These are the same type of errors detected by the CPU for CPU errors but when the H-6 latch is on, the MC trap branches to the MPX error routine. A channel control-check causes the contents of the machine check register and not the catalog number to be stored at byte 129 (81).

INTERFACE CONTROL CHECKS: These errors are detected by microprogramming only. An interface control check indicates an invalid response, slow response or no response from an I/O device to a tag-out signal. A catalog number is stored at byte 129 (81) for interface control check.

PROTECTION CHECKS: These errors indicate that an address outside of the protect key area has entered the MC-register. These errors are detected by error checking circuits only and cause a catalog number of 1111 1111 to be stored at byte 129 (81).

MPX ERROR ROUTINE

- Stores the unit address at byte 131(83)
- Stores the L-register contents at byte 129(81). This is either a catalog number or the contents of the machine-check (MC) register.
- Updates the high-order of the indicator byte, byte 130(82).
- Sets bit 4 of the indicator byte on.
- Sets bit 5 of the indicator byte on if the catalog number is stored at byte 129 (81).
- Attempts to clear the interface
- Either stores the CSW channel and unit status or creates an interrupt.

After logging bytes 131, and 129, the MPX error routine sets up the L and S register to log byte 130 (bits 0-5). Before doing so, however, the Error routine attempts to clear the interface.

If op-in is down to begin with, the MPX error routine sets both bits 6 and 7 of the indicator byte OFF. Otherwise, the error routine executes the halt I/O interface sequence (activates address-out). If op-in falls as a result of this, the error routine sets indicator byte bit 6 off and bit 7 on. If op-in does not fall, the error routine activates suppress-out and deactivates op-out (selective reset). If op-in falls as a result of this, the error routine sets indicator byte bit 6 on and bit 7 off. If op-in still doesn't fall the error routine sets indicator byte bit 6 and bit 7 both on.

When the interface is OK (op-in was down to begin with or fell as a result of the halt I/O sequence) and the MPX operations latch is on, the error routine stores the error condition in the UCW status byte from the U-register, and sets the UCW op-and-flag byte skip, output and active bits on. This causes the channel to respond to the next service vice-in associated with the UCW, with command out: This means stop and causes the I/O device to present status, initiating an interrupt. After storing the UCW op-and-flag byte the error routine branches to the normal CPU restore routine.

When op-in does not fall in response to the halt I/O interface sequence, the interface is not OK. In this case, the error routine determines if the error occurred during the first CCW. If so, it stores the CSW, storing the status only.

When op-in does not fall in response to the halt I/O interface sequence and the error occurs on the second or later CCW, the error routine test the interrupt buffer. If the IB is full the error routine branches the CPU back to the machine check trap. If the IB is not full, the error routine stores the I/O device address in the IB, sets the IB status byte to 00010000 (Busy), and turns on the MPX interrupt latch. When this is accomplished the error routines sets the UCW status byte with the error code and sets the UCW op-and-flag byte skip, active and output bits on.

LOGOUT PROCEDURES: Information to the CE and programmer, about errors that occur within the channel, is provided by a logout of error oriented data. The logout area for the multiplexor channel is main storage bytes 128, 129, and 130 (81, 82, 83). Interface and channel control checks are the only errors that cause logout.

In addition to error logout, the controls of the system take steps necessary to set up information to be stored in the CSW, through immediate or interrupt means. They also attempt to free the interface of an offending unit (force op-in down). If the interface is not free, an interface disconnect is attempted to remove the unit. If this does not work, a selective reset is given. When this fails, no further action is taken and manual intervention is necessary.

Following logout and interface reset, an attempt is made to store a CSW immediately if the error is during start I/O interrupt or a programmed test-I/O. If this is possible, regular processing is resumed.

Some situations exist in handling interface and channel control checks making it impossible to guarantee notification of the error through the CSW storage mechanism. This unqueable interrupt situation forces a CPU machine check trap if machine checks are allowed (PSW Bit 13=1).

If machine checks are not allowed:

1. Entry to the CPU machine check trap is prevented.
2. The error condition is recorded in the machine check register.
3. The system continues to function under existing control conditions.

When, by programming, the machine check trap is allowed, a CPU machine check trap occurs.

On an intermittent error, detecting a machine check causes a hardware forced microprogram trap to the machine check control program. This routine determines which system element was using the data flow at the time the error occurred. Control passes from this routine to the error microprogram for the system element indicated (CPU, multiplexor, or selector channel). Error traps are not allowed for the time necessary to clear error conditions and perform the logout. The machine check register is recorded as soon as possible in the routine, and the first-error latch is set on. Any errors detected from this time until the error routine is completed, stop the system. At the end of the error routine, the first-error latch is reset. Microprogram recovery, with a reasonable amount of information supplied concerning the nature and extent of the malfunction, is possible if the time between errors is more than the length of time for the error routine.

For more information on machine checking and error logging, refer to Chapter 2 in the FE Maintenance Manual, form 225-3390.

CHANNEL CONTROL CHECKS (FIGURE 3-18): Any machine check trap identified as a channel error is a channel control check. A channel control check causes the contents of the machine check register to be stored at byte 129 (81). This is the only source of channel control check errors in the multiplex channel. These errors use the same checking circuits as the CPU hardware, but when the H-6 latch is on, the machine check trap branches to the multiplexor check routine. This check routine takes the following action:

1. An interlock latch sets to prevent further traps until the microprogram progresses for enough to be useful.
2. The H-register indicator latches are inspected to determine if the error occurred under CPU, selector channel, or multiplexor channel control.
3. A branch is then taken to further handle these conditions.

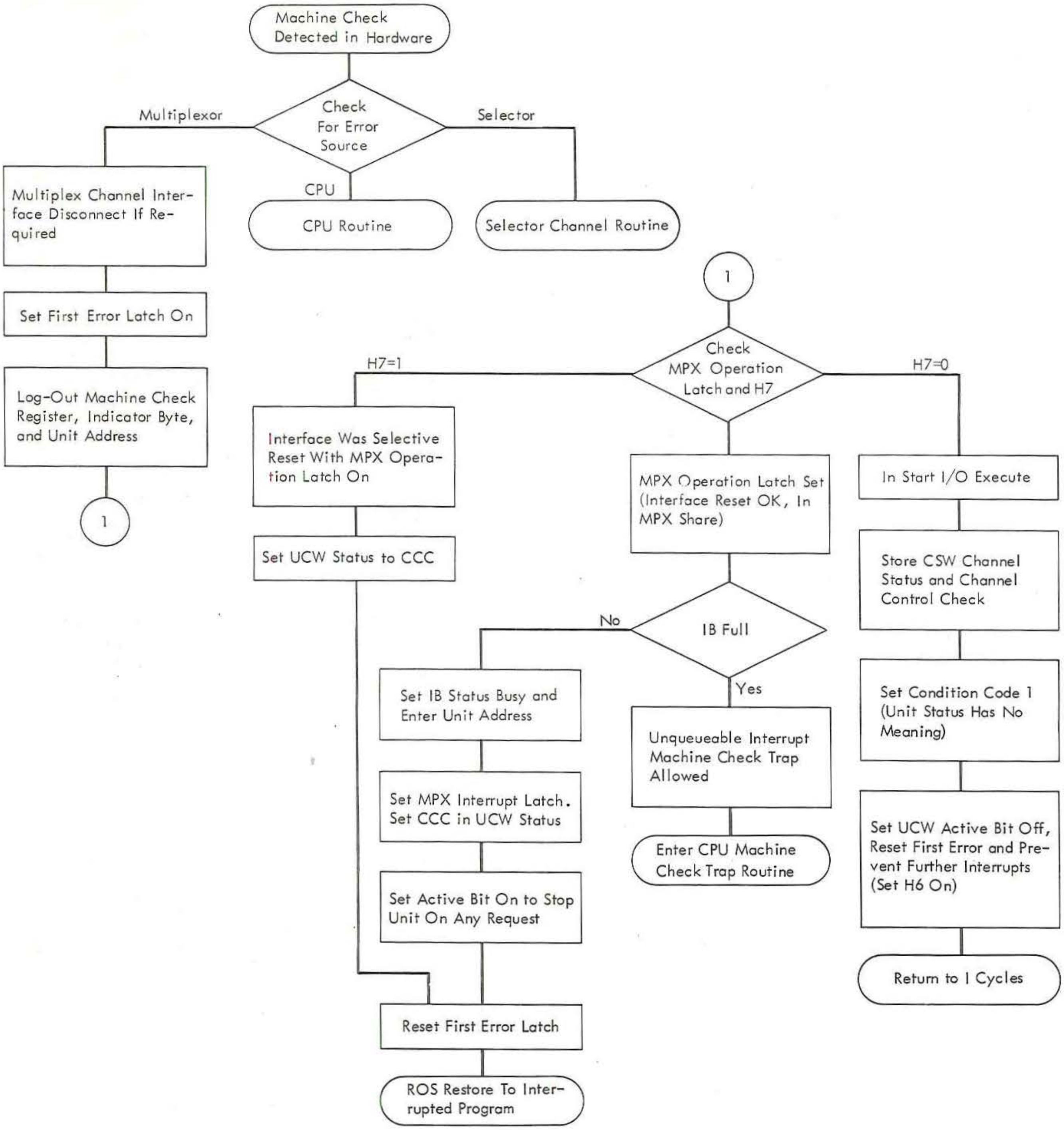

Machine Check Detected in Hardware
Check For Error Source
Multiplexor
Selector
CPU
Multiplex Channel Interface Disconnect If Required
CPU Routine
Selector Channel Routine
Set First Error Latch On
Log-Out Machine Check Register, Indicator Byte, and Unit Address
1
1
Check MPX Operation Latch and H7
H7=1
H7=0
Interface Was Selective Reset With MPX Operation Latch On
MPX Operation Latch Set (Interface Reset OK, In MPX Share)
In Start I/O Execute
Set UCW Status to CCC
Store CSW Channel Status and Channel Control Check
IB Full
No
Yes
Set IB Status Busy and Enter Unit Address
Unqueueable Interrupt Machine Check Trap Allowed
Set Condition Code 1 (Unit Status Has No Meaning)
Set MPX Interrupt Latch. Set CCC in UCW Status
Enter CPU Machine Check Trap Routine
Set UCW Active Bit Off, Reset First Error and Prevent Further Interrupts (Set H6 On)
Set Active Bit On to Stop Unit On Any Request
Return to I Cycles
Reset First Error Latch
ROS Restore To Interrupted Program

Figure 3-18. Channel Control Check

If machine check traps are not allowed (PSW Bit 13 = 0) no trap will occur even though one of the machine check latches is set on, and operation will continue.

INTERFACE CONTROL CHECKS (FIGURE 3-19): These errors are detected by microprogramming only and are caused by an excessive time lapse in an expected interface signalling sequence. These sequence timeouts are microprogram count loops, approximately 128 or 256 control cycles long. The timeouts check for excessive delay in interface response. Each data sequence (service-in, service-out) is timed out to approximately 30 seconds. This long delay time allows for tape gaps that can exist as a result of tape programming.

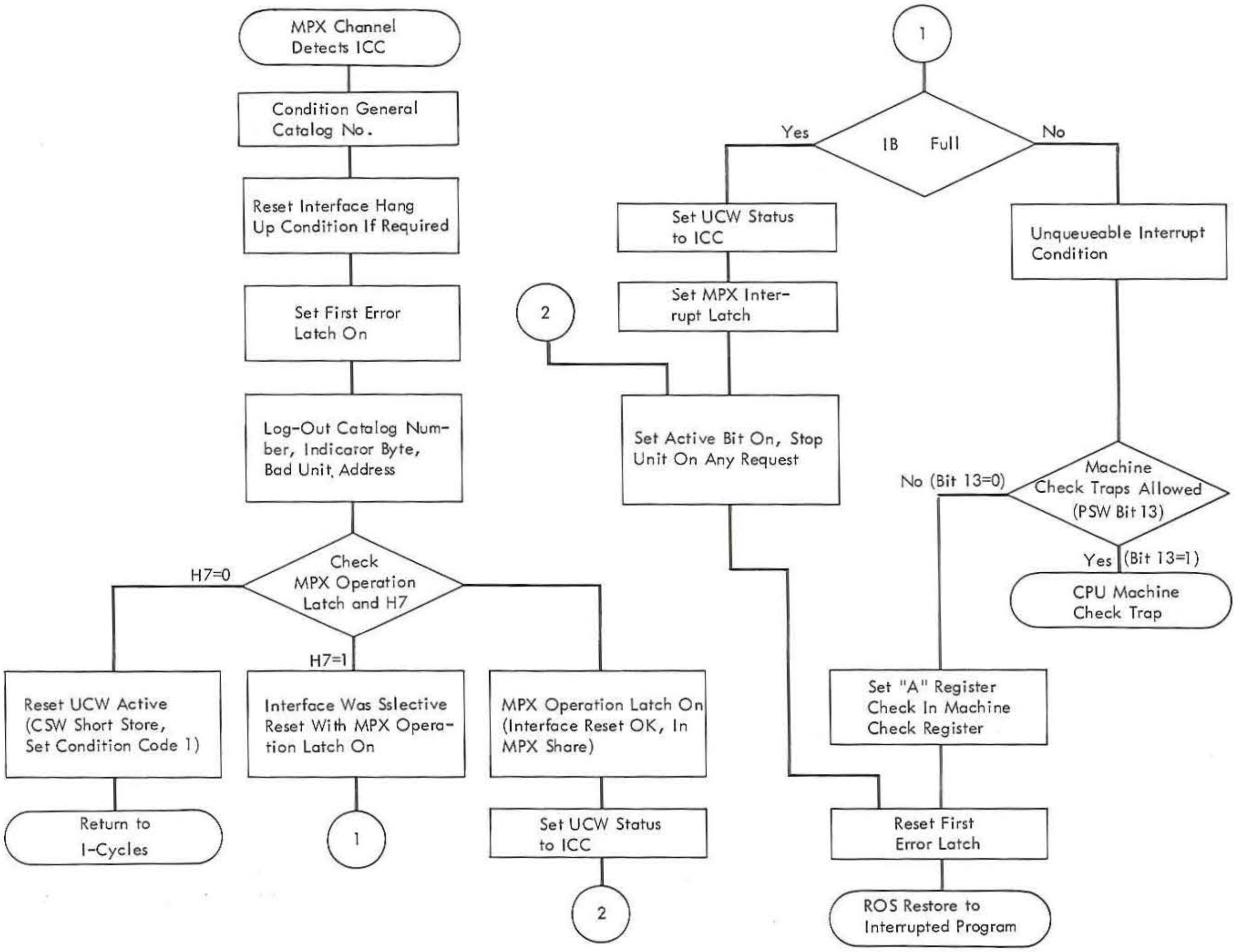

MPX Channel Detects ICC
Condition General Catalog No.
Reset Interface Hang Up Condition If Required
Set First Error Latch On
Log-Out Catalog Number, Indicaror Byte, Bad Unit Address
Check MPX Operation Latch and H7
H7=0
H7=1
Reset UCW Active (CSW Short Store, Set Condition Code 1)
Interface Was Sslective Reset With MPX Operation Latch On
MPX Operation Latch On (Interface Reset OK, In MPX Share)
Return to I-Cycles
1
Set UCW Status to ICC
2
1
IB Full
Yes
No
Set UCW Status to ICC
Set MPX Interrupt Latch
2
Set Active Bit On, Stop Unit On Any Request
Unqueueable Interrupt Condition
Machine Check Traps Allowed (PSW Bit 13)
No (Bit 13=0)
Yes (Bit 13=1)
CPU Machine Check Trap
Set "A" Register Check In Machine Check Register
Reset First Error Latch
ROS Restore to Interrupted Program

Figure 3-19. Interface Control Check

Interface control checks are stored in the CSW formed at the end of a start I/O execute cycle, or as a result of an I/O interrupt. A catalog number is stored at byte 129 (81) for interface control checks.

PROTECTION CHECKS (FIGURE 3-20): A protection check occurs when the key assigned to a subchannel does not match the protection key assigned to a section of main storage into which data is to be stored. A protection check ends the operation on the subchannel, stores the CSW channel status, and sets the protection check bit. If the error is detected before completion of the start I/O instruction, the CSW is "short stored" and condition code 1 is set. If the error is detected after completion of the start I/O instruction, an I/O interrupt is taken and the CSW is stored. The CSW unit status has no meaning in this instance.

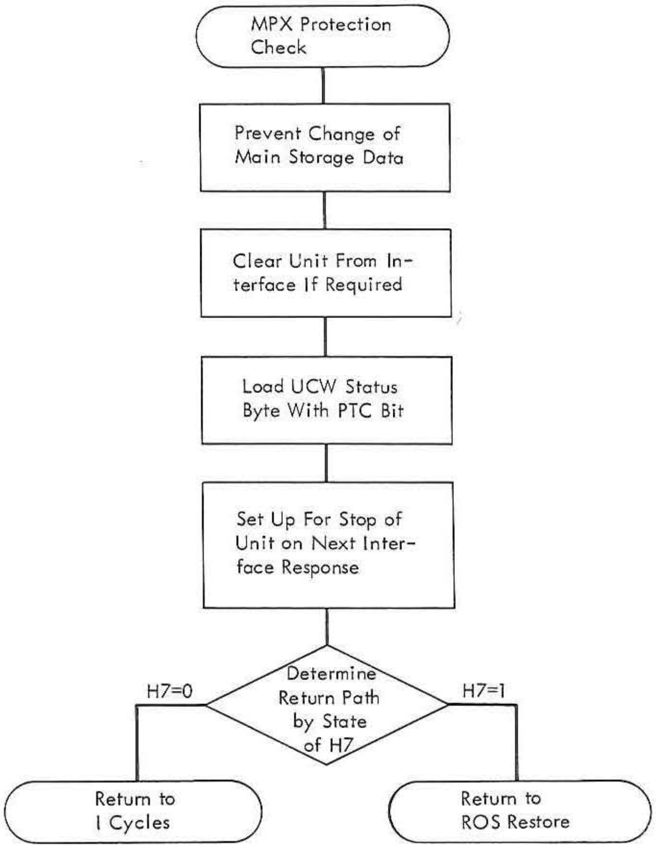

MPX Protection Check
Prevent Change of Main Storage Data
Clear Unit From Interface If Required
Load UCW Status Byte With PTC Bit
Set Up For Stop of Unit on Next Interface Response
Determine Return Path by State of H7
H7=0
H7=1
Return to I Cycles
Return to ROS Restore

Figure 3-20. Protection Check

PROGRAM CHECKS

- Program checks are caused by:
  1. Invalid CAW
  2. Invalid CCW
  3. Memory wrap

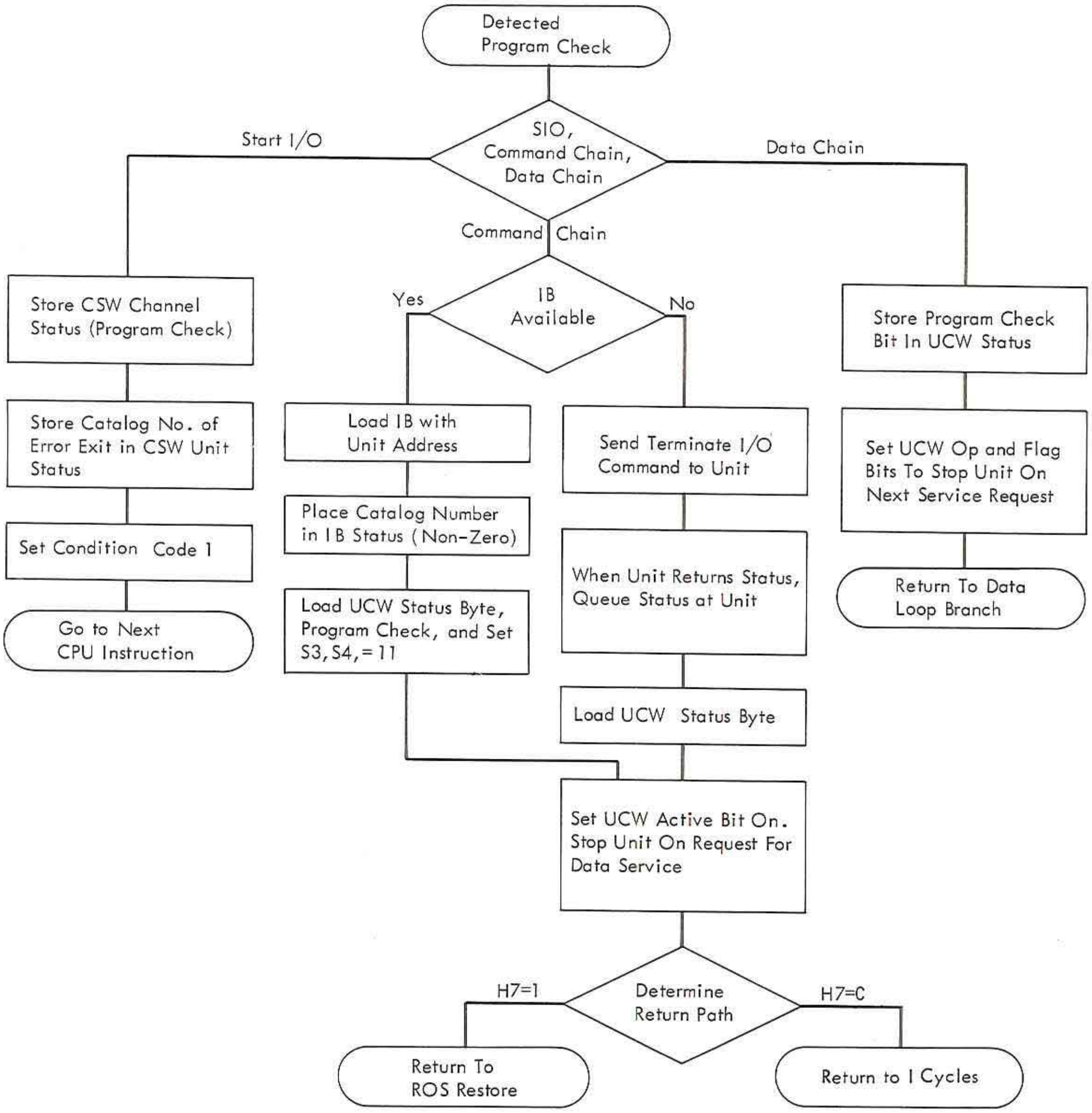
- Program checks are detected by microprogramming and by error checking circuits

- The MPX error routine does not log program check errors.

Program checks (Figure 3-21) are not logged by the MPX error routine and for the most part the interface is already clear. Two types of program checks require the MPX error routine to clear the interface, invalid address or memory wrap during MPX channel operations. In these two causes the error routine tries to clear the interface as for channel control, interface control and protect checks, and gets a CSW stored in the same way.

When other program checks occur during initial set-up of an I/O instruction, the error routine stores the channel status in the CSW, sets condition code 1 and returns the CPU to I-cycles.

If a program check occurs during command chaining , the MPX error routine executes a test I/O sequence with the I/O device involved and issues a command out when the device activates status-in. This allows the device to initiate an interrupt. Besides this the error routine sets the UCW status with the program check bit, and sets the UCW op and flag bits 3, 5, and 6 on.


Detected Program Check
SIO, Command Chain, Data Chain
Start I/O
Data Chain
Command Chain
IB Available
Yes
No
Store CSW Channel Status (Program Check)
Store Catalog No. of Error Exit in CSW Unit Status
Set Condition Code 1
Go to Next CPU Instruction
Load IB with Unit Address
Place Catalog Number in IB Status (Non-Zero)
Load UCW Status Byte, Program Check, and Set S3, S4, = 11
Send Terminate I/O Command to Unit
When Unit Returns Status, Queue Status at Unit
Load UCW Status Byte
Store Program Check Bit In UCW Status
Set UCW Op and Flag Bits To Stop Unit On Next Service Request
Return To Data Loop Branch
Set UCW Active Bit On. Stop Unit On Request For Data Service
Determine Return Path
H7=1
H7=C
Return To ROS Restore
Return to I Cycles

● Figure 3-21. Program Check

COMPREHENSIVE INTRODUCTION

- IBM 2030 may have a maximum of 2 selector channels.
- Selector channels take one memory cycle (2.0 microseconds M2, 1.5 microseconds M2-I) of CPU time to handle each byte of I/O data.
- Each selector channel has associated registers that contain I/O operating information, such as:
  1. Count
  2. Data Address
  3. Command
  4. Flags
  5. Channel Status
  6. Protect Key
- Priority assignment, with all channels requesting service, guarantees:
  1. Selector channel 1, 50% of the time.
  2. Selector channel 2, 25% of the time.
  3. Multiplexor channel, 25% of the time.

NOTE: Model M2 has a 2.0 microsecond memory cycle and Model M2-I has a 1.5 microsecond memory cycle.

The IBM 2030 may have up to two selector channels. A selector channel is a high-speed channel that takes one memory cycle of CPU time to handle each byte of data and does not disturb the contents of the CPU registers. It contains registers that hold the Channel Command Word (CCW) for the entire data transfer portion of an I/O operation. The selector channel also contains a clock to provide timing during a selector share cycle.

Figure 4-1 shows the general data flow of a selector channel. When the selector channel is addressed by a start I/O command, its registers are loaded with the operating information of the CCW, under microprogram control. After a selector channel has initially selected the correct I/O control unit, also under microprogram control, it waits for a service-in from that control unit while the CPU is processing other information. When the selector channel receives a service-in from the control unit, a service-out is returned immediately if:

1. On an output operation, the GR register is full and a data byte can be sent.

2. On an input operation, the GR register is empty and ready to receive data.

If a service-in is received while GR is empty for an output operation or full for an input operation, the selector channel:

1. Waits for the present storage cycle to be completed.
2. Assumes control of core storage.
3. Handles the data.
4. Updates the count and data address.
5. Responds with a service out.
6. Returns storage control to the CPU

Priority assignment for storage cycles or ROS cycles (multiplexor channel) is such that selector channel 1 has priority fifty percent of the time; selector channel 2 has priority twenty-five of the time: and the multiplexor channel has priority twenty-five percent of the time. This priority takes effect only when all three channels are requesting storage cycles at the same time. Any time a channel does not use a cycle, the cycle may be used by the other channels according to their priorities. Any cycles not used by the I/O channels are utilized by the CPU.

When any channel requests a storage cycle, it has priority over the CPU, and will be honored as soon as the CPU finishes the current cycle. Selector share cycles, which are storage cycles, can interrupt a multiplexor channel that is using ROS cycles to handle data. Once a storage cycle has been started, other requests will not be honored until the storage cycle is complete.

All registers associated with selector channel 1 are identified by a prefix G.

All registers associated solely with selector channel 2 are identified by a prefix H. For example, the GR register is the data register of selector channel 1, and the HR register is the data register for selector channel 2. A single micro program is used for both selector channels. The micro program always uses the G prefix. The controls are gated to selector channel 1 if SX-1 gate is on, and to selector channel 2 if SX-2 gate is on.

Because selector channel 1 and selector channel 2 are identical in operation, except for the register used, the following descriptions are for selector channel 1. Apply the descriptions to selector channel 2 by mentally inserting a prefix H, where there is prefix G in the description.

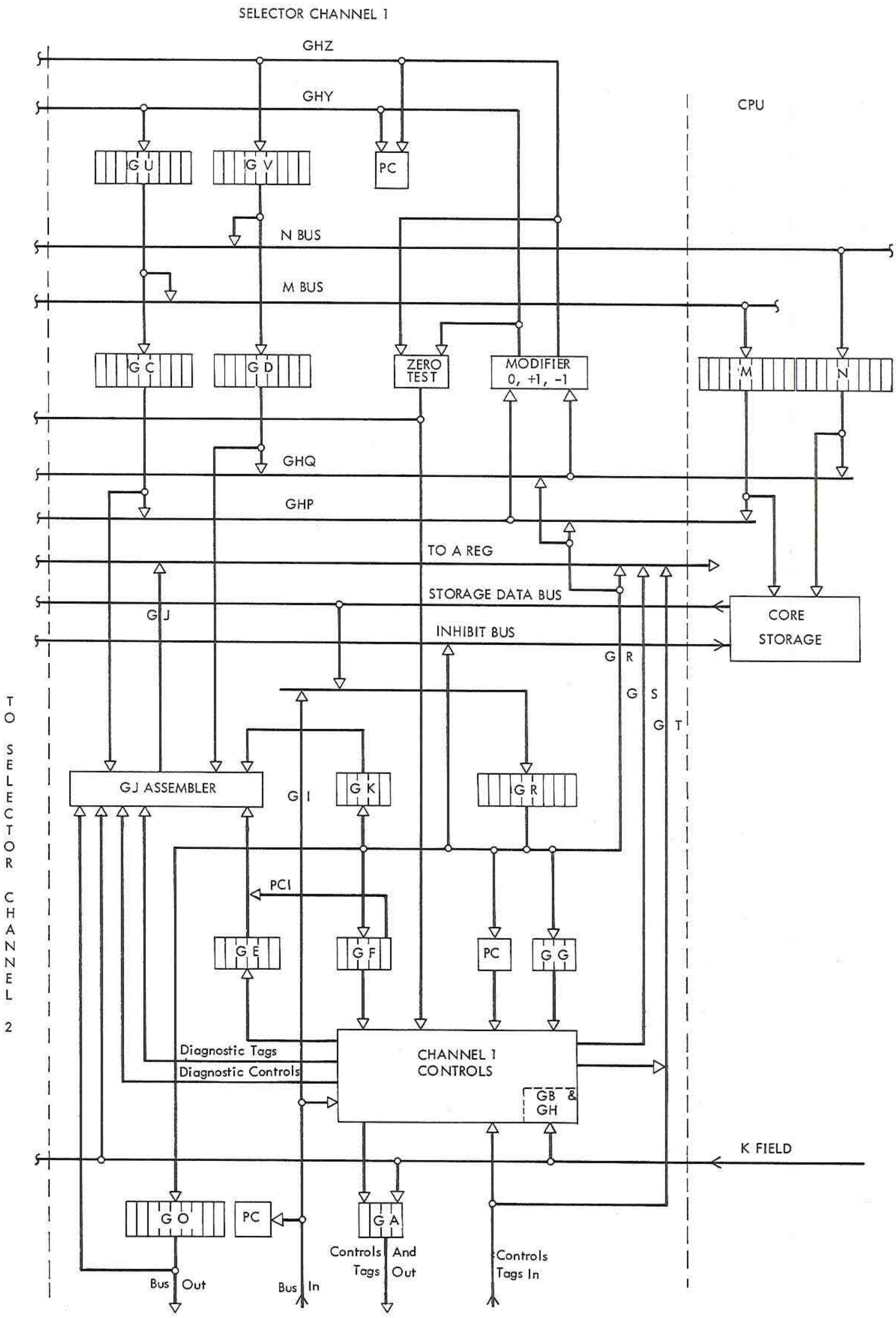


Figure 4-1. Selector Channel 1 Data Flow

GA REGISTER (TAG LINE CONTROL)

- Controls out-bound tag lines.
- Controls the gating of information to out-bound bus lines.
- Sets under microprogram control during initial selection sequence, and under hardware control during data handling sequence.
- Sets outbound tag response to in-bound tag line.

The GA register is a four-position register that controls the out-bound tag lines and the out-bound bus lines. These latches turn on under microprogram control or in response to an in-bound tag line, or combinations of both.

The microprogram turns on the GA register latches under control of the CS field. The CS field must be GA=K (Gate CK field bits to GA register) where the value of the CK field may be from 0 to 15 with or without the parity bit.

Figure 4-2 shows the four GA register latches and the conditions that turn them on.

LATCH NAME AND PURPOSE	TURN ON CONDITIONS	RESETS
Bus-out control; gate the GO register to Bus-out.	1. Gate K to GA, and P1, CK Sal 0 bit. 2. Output operation, GR full, and service-in and not service out.	1. Not service-in, not status-in, and not address-out. 2. Gate K to GA, and P1 and CPU T1. 3. Machine reset.
Address Out; raise the address-out tag line.	Gate K to GA, and P1, CK Sal 1 bit, and CPU T2 pulse.	1. Gate K to GA, and P1, CPU T1, and not halt I/O. 2. Not poll-control, not Op-in, and not Halt I/O. 3. Op-in, select-out, and not Halt I/O. 4. Machine reset.
Command Out; raise the command-out tag line.	1. Gate K to GA, and P1, CK Sal2 bit and CPU T2 pulse. 2. Service-in, status stop condition, and not check stop mode.	1. Gate K to GA, and P1, and CPU T1. 2. Not address-in, not service-in, and not status-in. 3. Machine reset.
Service Out; raise the service-out tag line.	1. Gate K to GA, and P1, CK Sal3 bit. 2. Channel-end alone and command chaining. 3. Service signal and not set bus-in to GR.	1. Gate K to GA, and P1, and CPU T1. 2. Not address-in, not service-in, and not status-in. 3. Machine reset.

Figure 4-2. GA Register

GB CONTROLS

- Turned on under microprogram control.
- CS field is GB = K.

The GB controls are 16 general-purpose controls which provide a variety of functions in the selector channel.

The microprogram turns on the GB controls under control of the CS field. The CS field must be GB = K, where K is equivalent to the value of the CK field. The value of the CK field (K) can be 0 to 15, with or without the parity bit (P). The P-bit is used as an additional control bit. The CK field, when used for these controls, can be odd or even parity since this field is not individually parity checked.

Figure 4-3 shows the GB controls and the purpose of each.

GB	NAME	CK FIELD DECODE*	REASON
0.	Set Program Check	KO	Zero Count Except for TIC, Memory Wrap, Three Low Order Flag Bits Not Zero, First CCW is a TIC, Two TIC's in Succession, CCW Not on Word Boundaries, or Invalid Command.
1.	SX2 Selection	K1*(0 or 1)	0 = SX1, 1 = SX2
2.	Operational-Out Reset	K2	
3.	Reset PC1	K3	
4.	Set Selector Interrupt	K4	
5.	Set Channel Control Check	K5	A Machine Check Trap when H-Register 5 Latch is On Indicates Hardware Failure during selector channel chaining or interrupt routine.
6.	Set GR to Zero	K6	
7.	Not Used	K7	
8.	Count Ready-Not Zero	K8	
9.	Channel Reset	K9*(0 or 1)	0 = Reset Everything Except Poll Control. 1 = Reset Everything Including Poll Control.
10.	Suppress-Out	K10*(0 or 1)	0 = Reset, 1 = Set
11.	Poll Control	K11	0 = Reset, 1 = Set
12.	Reset Select-Out	K12	
13.	Set Channel Busy	K13	
14.	Set Halt I/O Latch	K14	
15.	Set Interface Control Check	K15	Address Mismatch, No Response, Time Out, Unit Busy on Chaining.

*(0 or 1) Refers to the CK Field Parity Bit.

●Figure 4-3. GB Controls

GC AND GD REGISTERS

- Both are nine-position registers.
- They keep a count of the data transfers.
- They are initialy set to the value of the CCW count.
- They are decremented every selector channel 1 share cycle.
- They always set from the GU and GV registers.

These nine-position registers (See Figure 4-1) maintain a count of data transfers during the I/O operation on selector channel 1. The GC register maintains a high-order count, while the GD register maintains the low-order count.

The GC and GD registers set to the value of the CCW count during the CCW loading sequence. They actually set from the value of the GU and GV registers, through which the count must pass before entering the GC and GD registers.

During selector share-read cycles (data-transfer cycles between main storage and selector channel), the GC and GD registers contents are gated into the modifier, modified minus one, then set into the GU and GV registers

During T1 time of selector share write cycles, this modified count is transferred from the GUV registers to the GCD registers.

GE REGISTER (CHANNEL STATUS)

- Set by error conditions during channel operation.
- Reads out under microprogram control.
- Controls premature ending of I/O operations when errors are detected.

The GE register contains six latches which are set by error conditions which occur when loading the channel with the CCW, or by error conditions during a channel operation.

1. Incorrect length
2. Program check
3. Protection check
4. Channel data check
5. Channel control check
6. Interface control check

Program controlled interrupt (PCI) is inserted on the GE bus from the GF register.

The GE register and the PCI flag bit is read out on the GE bus to the GJ assembler when channel status is being stored in the CSW. The latches in the GE register detect errors associated with channel operation and provide premature ending of I/O operations as required, when errors are detected.

GF REGISTER (FLAG REGISTER)

- Five-position register.
- Set when the CS field of a microprogram instruction is GF = GR
- Set with CCW flag information.

This five-position register (See Figure 4-1) sets with the CCW flag information during the CCW loading sequence under microprogram control. The five flag bits are:

1. Chained data (CD), bit 0.
2. Chained command (CC), bit 1.
3. Suppress length indication (SLI), bit 2.
4. Skip, bit 3.
5. Program control interruption (PCI), bit 4.

The flag byte of the CCW enters the GF register when the CS field of a microprogram instruction is GF = GR.

GG (COMMAND) REGISTER

- Set under microprogram control, CS is GG = GR.
- Four latches will set to the four low order bits of the CCW command code.
- The four latches are decoded into control lines:

  1. Read backward.
  2. Input.
  3. Output.

This four-position register sets with the four low-order bits on the CCW command code which specified the commands:

1. Read
2. Write
3. Sense
4. Control
5. Read backward

It sets under microprogram control; the CS field of the microprogram instruction is GG = GR.

The output of the GG register latches is decoded to furnish control signals:

1. Read backward.
2. Input.
3. Output.

GH CONTROLS

- Turned on under microprogram control.
- CS field is GH=K.
- General purpose controls used for diagnostic microprograms, and for various channel micro-program controls.

GH=K provides 16 possible controls (GK field equals 0 through 15) used for diagnostic purposes and for micro program control of the channel. See Figure 4-4. The GH controls are additional general-purpose controls similar to the GB controls.

GH	NAME	REASON
0	SX1-SX2 Machine Reset	Used by diagnostics only
1	Set Diagnostic Mode and Tag Control	Used by diagnostics only
2	Reset Diagnostic Tag Control	Used by diagnostics only
4	Channel Switch	Used to Switch Microprogram Control to Other Channel
7	Set Chain Detect	Used During a PCI Interrupt
11	Set Count Ready-Zero	Used by Diagnostics Only
12	Set Select-Out	
13	Chain Reset	Used to Reset Various Latches and Set Poll Control When Starting a Chain Operation

Note: 3,5,6,8,9,10,14 and 15 Bits Not Used.

Figure 4-4. GH Controls

GI BUS

- Nine bus lines.
- Connects bus-in to the GR register
- Connects bus-in to the channel-control circuits.

The GI bus is a set of nine lines connecting bus-in to the GR register and the channel-control circuits. Entry to the GR register is under microprogram control (CS field is GR=GI), or in the case of data transfer, under hardware control. Entries to the channel control circuits are ANDed with the status-in line to provide control unit and device status for the channel.

A parity check is made on bus-in to determine if address and status information from the control unit has valid parity.

GJ BUS

- A nine-position bus that provides access to the A-register.
- Information is put on the bus under microprogram control.

The GJ bus is a nine-position bus receiving information from both selector-channel 1 and selector-channel 2. It provides an additional access to the A-register for information from the selector channels.

All information is placed on the GJ bus under microprogram control, as specified by the CK field in the microword which names GJ as the A-register source.

GJ ASSEMBLER (SELECTOR CHANNEL 1)

- A nine-position assembler accepting inputs from the following sources under microprogram control:

  GC register

  GD register

  GE bus

  GK register

  GO register (diagnostic)

  Channel tags (diagnostic)

  Channel controls (diagnostic)

- Provides an input to the GJ bus for information transferred between selector channel 1 and the A-register.

The GJ assembler is a nine-position gate to the GJ bus for the following information (specified by the CK field) when the microprogram names GJ as the A-register source:

K1 is GJ = GC

K2 is GJ = GD

K3 is GJ = GK

K4 is GJ = GE

K6 is GJ = Controls (diagnostic)

K7 is GJ = Tags (diagnostic)

K8 is GJ = GO (diagnostic)

The output of the GJ assembler (the GJ bus) provides one of the inputs to the A-register

GK REGISTER (MEMORY PROTECT KEY)

- Set by microprogram; CS is GK = GR.
- Loaded from CAW (Loc 48 hex).
- Output compared against the storage key read out when it is addressed by the data-address during a selector-share input cycle.

The GK register is a five-position memory-protect register. It is set under microprogram control from the GR register, when the CS field of a microprogram instruction is GK = GR.

The memory-protect register (GK) is optional on the IBM 2030. When the memory-protect feature is installed on the 2030, selector channel 1 will contain the GK register. The output of the GK register compares to the storage key of main storage that is addressed by the data address during a selector-share-input cycle. If the keys are not equal, and the protect key (GK) is not zero, the location in memory is not changed, and a protection-check is indicated in the subsequent I/O interrupt.

GO REGISTER

- A nine-position register used as a buffer register for bus-out information.
- Input to GO register is from GR register.
- Frees GR for other operations during information transfer on bus-out.
- Sets during address-out, command-out, or service-out.

The GO register is a nine-position register used to supplement the GR register during information transfers on bus-out. Normally, any information in the GR register feeds through the GO register latches to the bus-out control gate, however, when the address-out, command-out, or service-out tag is raised by the selector channel, the information contained in the GR register is latched up in the GO register. This releases the GR register for other selector-channel operations for the time period required to send a command, an address, or a data byte to an I/O control unit or device and receive a reply. During output data cycles, when service-out latches up the data byte in the GO register, a share-request is generated to pre-fetch the next data byte from storage and place it in the GR register thus enabling faster channel data rates. Similarly, during data chaining, while the last data byte is held in the GO register, the GR register is used to read out the chained CCW and load it into the channel.

GR REGISTER (DATA REGISTER)

- Nine-position register.
- Intermediate storage for information transferred between selector channel 1 and main storage.
- May be set by:
  1. Main storage output.
  2. Microprogram.
  3. Bus-in information.

The GR register (See Figure 4-1) is a nine-position register that is similar to the R-register. It is intermediate storage for all information transferred between selector channel 1 and main storage.

Like the R-register, the GR-register latches may be turned on by several conditions:

1. Whenever main storage is addressed and GR is specified, or during Selector Channel data cycles.
2. From the I/O interface bus-in lines.
3. By microprogram, set GR to zero.

GR, GS, AND GT BUSES

- Three buses for address, status, error, and control information to the A-register.
- GR bus: interface address and status.
- GS bus: status and control conditions.
- GT bus: in-bound tags and control information.

The GR, GS, and GT buses provide a function similar to the GJ bus, whereby the microprogram is used to test for information concerning the interface and the selector channels. Figure 4-5 shows several buses and the functions provided.

All entries to the A-register from these buses are under microprogram control.

Direct Input to A Register

Bit	GS Bus	GT Bus
P	None	None
O	GR Full	Select In
1	Chain Detect	Service-in Not Service-Out
2	Not Used	Poll Control or Halt I/O Stop
3	Interrupt Condition	Channel Busy
4	CD	Address In
5	1=Chnl 1, 0=Chnl 2	Status In
6	Not Used	SX1 Interrupt Latch
7	Chain Request	Op-In

Input to GJ Assembler

Bit	GE Bus	Diagnostic Controls Bus	Diagnostic Tags Bus
P	None	None	None
O	PCI	Count Ready, Not Zero	Input
1	Incorrect Length	SLI Flag	Suppress Out
2	Program Check	Output	SX1 ROS Request
3	Protection Check	Count Ready, Zero	Address Out
4	Channel Data Check	Not Used	Command Out
5	Channel Control Check	CC Flag	Service Out
6	Interface Control Check	Read Backward	Bus-Out Control
7	Not Used	Skip Flag	Operational Out

● Figure 4-5. Buses

GU AND GV REGISTERS

- Both are nine-position registers.
- They maintain the data address.

These registers hold the address during selector channel 1 operations. GU contains the high-order address and GV the low order. A count also enters these registers, momentarily, during the initial loading of the CCW into the channel registers, and during selector- share cycles, after it is decremented by the modifier.

The GU and GV register latches are set from the output of the modifier. During selector share cycles (data transfer between the selector channel and main storage) the GU and GV register contents enter the MN register to address main storage. During the write portion of the selector-share cycle, the contents of MN is modified and set into the GU and GV registers.

GW, GX REGISTERS

- Two registers combined to serve as a back-up ROAR to store contents of WX registers during a selector-channel ROS request.
- Stores location of last microprogram word addressed.
- Used during CC chaining, CD chaining, or selector- channel interrupts.

The GW and GX registers contain fifteen latches (combined) and are used as temporary storage for the ROS address in the WX registers. When the selector channel makes a request for ROS control and breaks into the CPU microprogram routine, the location of the last microprogram word addressed is put into the GWX registers. When the selector-channel operation requiring ROS is completed, the contents of GWX are returned to the WX registers under microprogram control (gate GWX to WX). The CPU program now starts at the location (address) where the interruption occurred.

A selector channel ROS request is initiated when:

1. The CC (command chaining) flag bit is on and the end of present operation has been reached with no errors or unusual conditions.

2. The CD (data chaining) flag bit is on and the count has gone to zero for the present operation, and no errors have been detected in the channel.

3. A selector-channel interrupt occurs.

These are the only times when the GWX registers are used as the back-up ROAR.

MODIFIER

- Modifies +1, -1, or 0.
- Count always modifies -1 during the read portion of a selector-share cycle.
- Modification is always 0 on initial loading of the CCW
- The modifier output parity is determined independently from the modifier data output providing a check on modifier operation.
- It may modify the data address +1 or -1 during the write portion of a selector-share cycle (+1 for a read operation, -1 for a read-backward operation).

Information sent through the modifier is modified +1, -1, or 0. The modifier is shared by both selector channels (see Figure 4-1). It receives information from:

1. The GC and GD registers; selector channel 1 (count).
2. The HC and HD registers; selector channel 2 (count)
3. The MN register; both selector channels (data address).
4. The GR register; selector channel 1 (loading of the CCW).
5. The HR register; selector channel 2 (loading of the CCW.

During initial loading of the CCW into selector-channel 1, the modifier accepts information from the GR register and provides an unmodified output to the GU and GV registers.

During selector-share read cycles, the modifier receives the count and modifies it by -1. This output then enters the GU and GV registers.

During selector-share write cycles, the modifier receives the data address from the MN register and modifies it +1 or -1 according to the command the selector channel is executing. During the early part of this cycle, the modified count is transferred from the GU and GV registers into the GC and GD registers. Near the end of this cycle, the modifier output (data address) then enters the GU and GV registers.

The input to the low order of the modifier is labelled the GHQ input. The high order is the GHP input. The low-order output is GHZ, and the high-order is GHY.

Note that the modifier output parity bits are developed independent of the results of the data modification. Similarly, the high order carry-in is developed independent of the results of the low order modification. This provides a means of checking the modifier and also reduces the time required for results to propagate through the modifier.

Exclusive OR's control the modifier outputs (GHY and GHZ). A GHY or GHZ bit is on when the exclusive-OR output is minus. This occurs when the inputs are both of the same level (both plus or both minus). See Figure 4-6.

The carry-in line of the low-order modifier circuit becomes plus whenever either +1 or -1 modification is required. The high-order carry-in line becomes plus when there is a carry from the low order of the modifier; the GHQ (low-order) input has all 8 bits on for a +1 modification and all bits off for a -1 modification.

Each bit has an AND-OR-Inverter (AOI) associated with it. This AOI may furnish a <u>carry</u> to the next succeeding position. A carry is indicated by a minus from the AOI's for the odd bits, and by a plus from the AOI's for the even bits.

An exclusive-OR's output will be minus when its input is either:

1. GHQ = 1 and no carry entry.
2. GHQ = 0 and a carry entry.

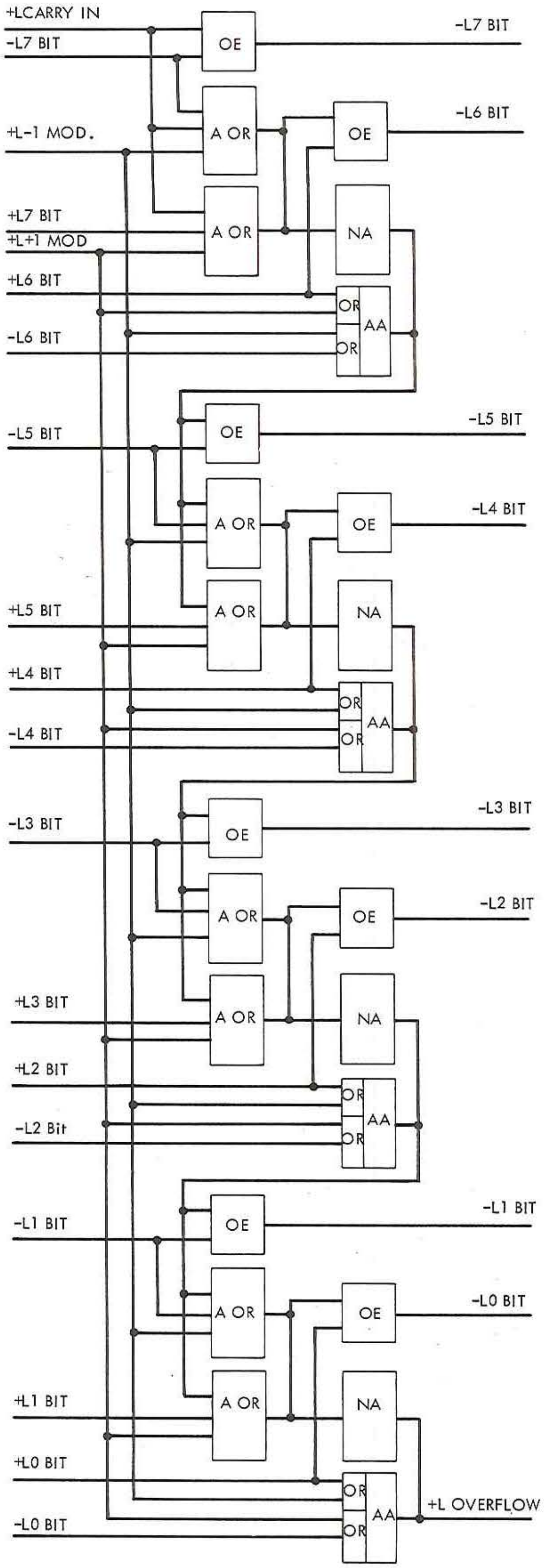


Figure 4-6. +0, +1 Modification

Circuit Objectives

Assume a read command and a selector-read cycle with a count of seven in the GC - GD registers.

1. Turn on -1 modification GHPQ
   Selector Read Cycle
2. Activate reverse GHQ Parity
   Mod-1
   GHQ 7-bit
3. Generate GHZ Parity
   Reverse GHQ Parity
   Not GHQ P-bit
4. Generate GHQ carry-in.
   Mod-1
5. Decondition GHZ 7-bit.
   Carry-In
   GHQ 7-bit
6. Condition GHZ 6-bit.
   No carry from bit 7 (GHQ 7-bit with-1 mod prevents a carry)
   GHQ 6-bit
7. Condition GHZ 5-bit.
   No carry from bit 6
   GHQ 5 bit
8. GHZ bits 4, 3, 2, 1, and 0 are not conditioned.
   No Carry
   No GHQ 4, 3, 2, 1, or 0 bits
9. GHY bits 0 through 7 are not conditioned
   No GHQ carry-out

   No GHP bits 0 through 7

CLOCK (SELECTOR CHANNEL)

- Four latches advanced in sequence by a 2 megacycle oscillator (M2) or a 2.667 megacycle oscillator (M2-I).
- Latch outputs provide basic clock pulses P1, P2, P3, and P4.
- Basic pulses ANDed to furnish selector timings T1, T2, T3, and T4.
- Clock runs only during selector read/write cycles.

The clock used to develop timing pulses for selector channel operation is a four-stage latch ring (Figure 4-7.). A 2 megacycle oscillator (M2) or a 2.667 megacycle oscillator (M2-I) supplies the input pulses to the ring. This is the same oscillator that drives the CPU clock. The clock latch outputs are four 500 nanosecond (M2), or 375 nanosecond (M2-I), basic pulses designated selector P1, P2, P3, and P4. These four pulses are then ANDed together to provide the four 250 nanosecond (M2) or 187.5 nanosecond (M2-I) timing pulses designated selector T1, T2, T3, and T4. These T-pulses are used to time all selector channel read/write operations.

The four latches comprising the clock ring are coupled so that clock-1 on, gates on clock-2 and gates off clock-4 (see Figure 4-7). Clock-2 on, gates on clock 3 and gates off clock 1. When clock-4 comes on, it gates clock-1 on and clock-3 off. Each latch, in conjunction with the oscillator pulses, turns on and off in sequence. This sequence is repeated until the clock is stopped.

The timing relationship between the clock P-pulses and the selector T-pulses is shown in Figure 4-8. Note that an overlap of the P-pulses is necessary to satisfy the AND conditions which develop the T-pulses (Figure 4-7).

The selector clock runs only during selector read/write cycles. At this time the CPU clock is stopped long enough to complete the information transfers to or from the selector channels. At the completion of selector read/write cycles, the CPU clock is restarted.

The selector clock is started normally by an SX1-or SX2- share-request (Figure 4-9). The share request, (not) allow-write, and T3 pulse turn on the selector share-cycle latch. With this latch on, the selector-share-hold latch is turned on stopping the CPU clock and starting the selector clock. The selector clock runs for two cycles (read and write) and stops if there are no other share-requests. The selector clock continues to run if another share-request is received before the end of T3 time of the selector write cycle.

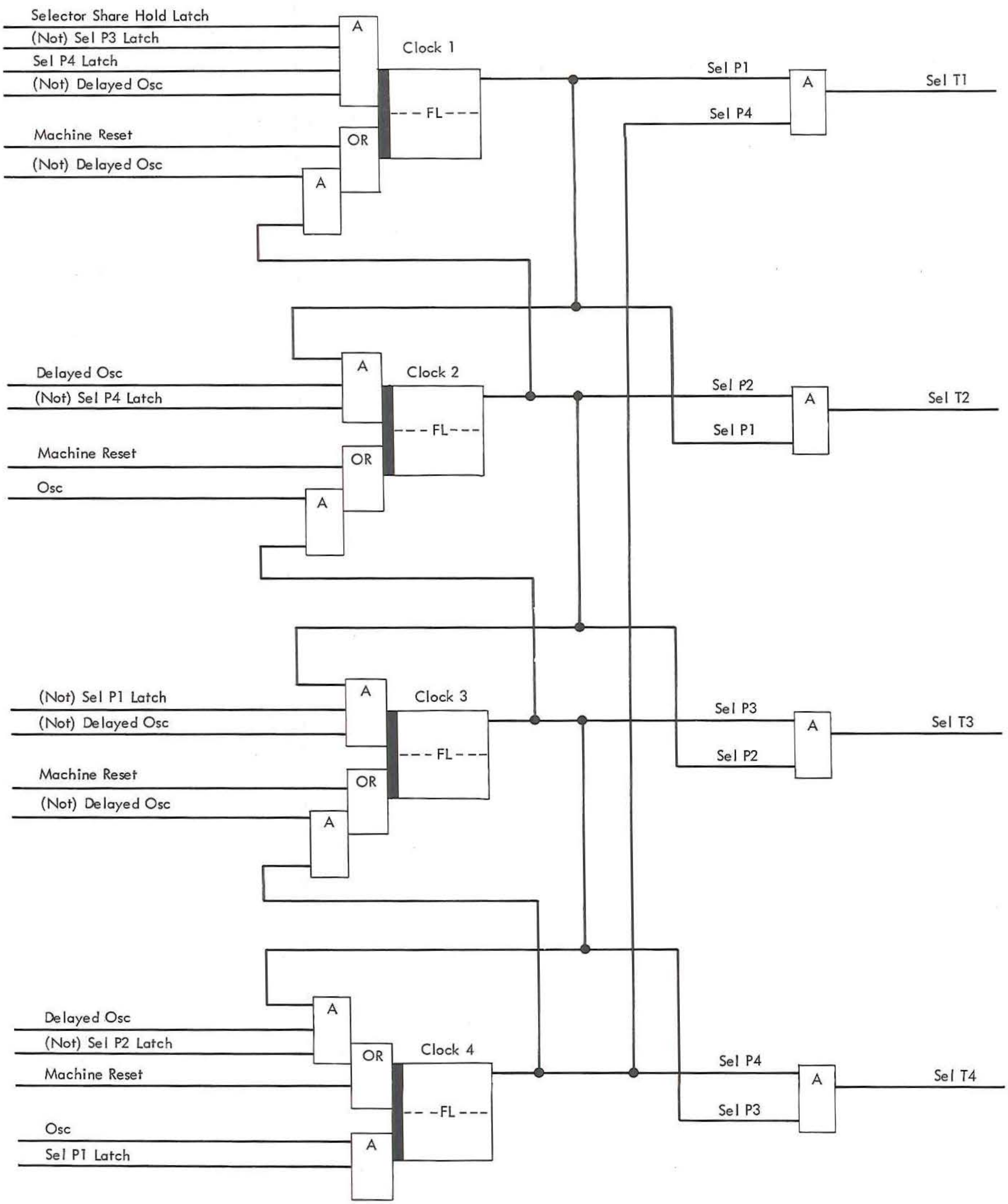
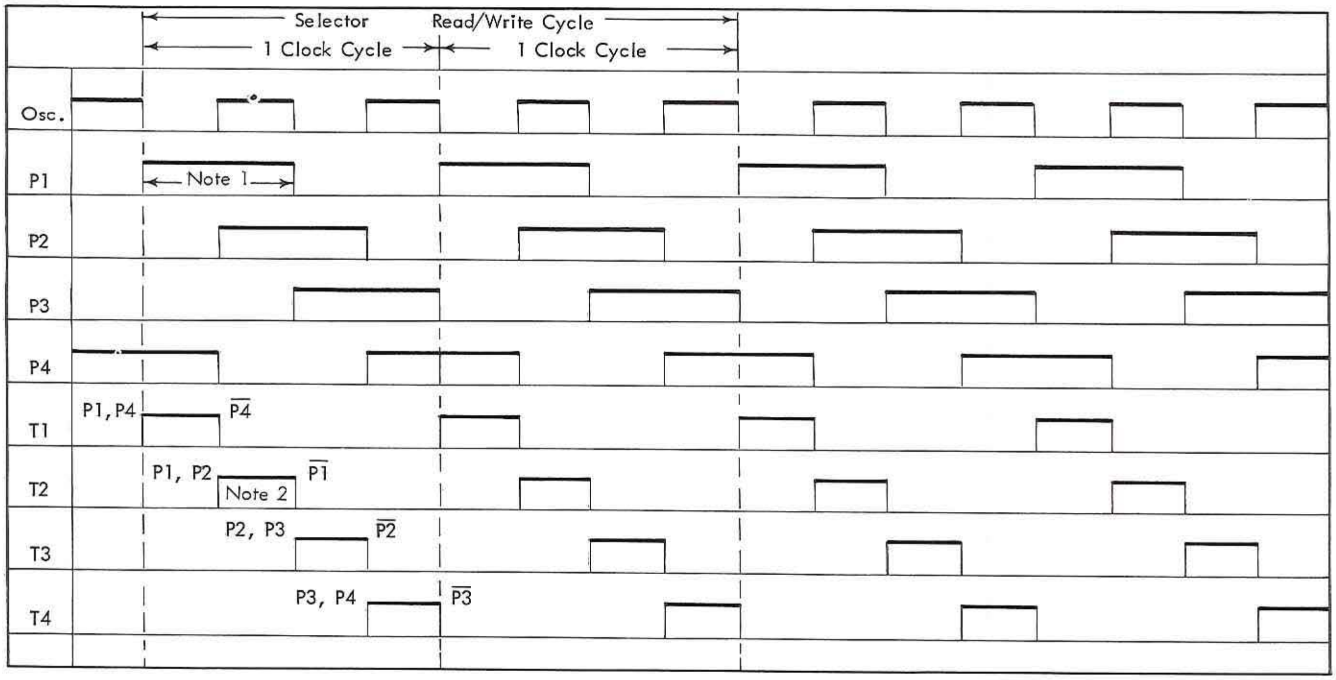


Figure 4-7. Selector Clock



Note 1 - 500 ns Mod 2, 375 ns Mod 2 I
Note 2 - 250 ns Mod 2, 187.5 ns Mod 2 I

Figure 4-8. Selector Clock Timings

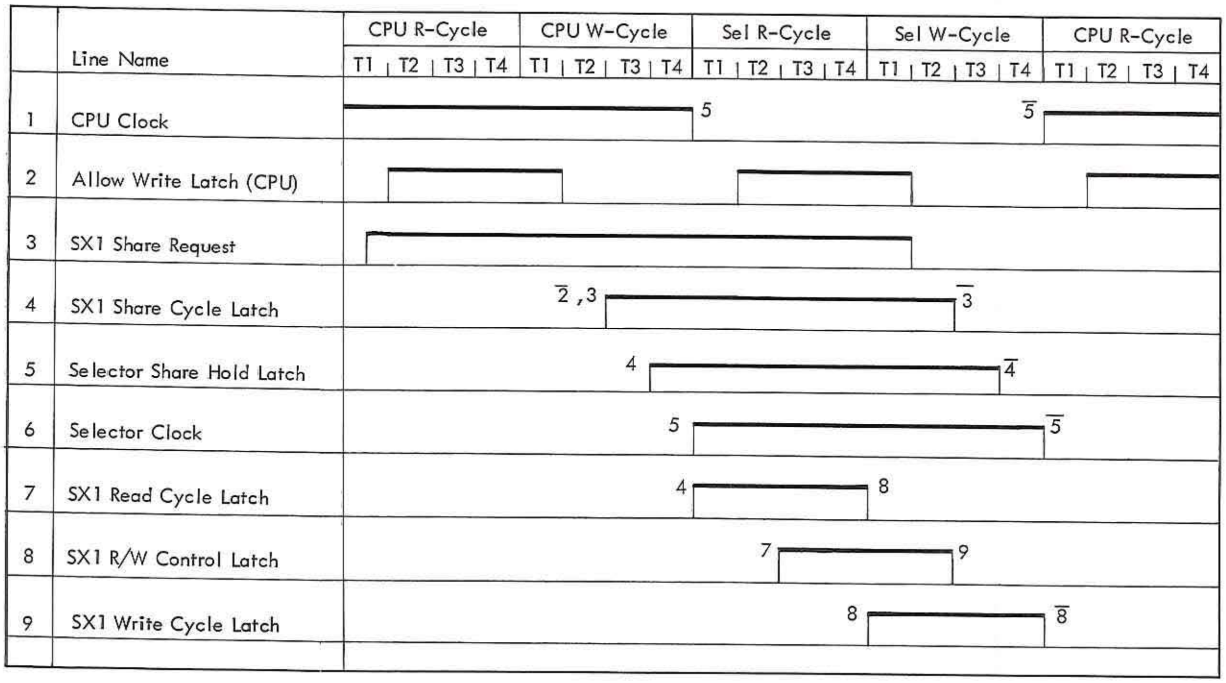


Figure 4-9. Selector Clock Start and Stop Sequence

START I/O; COMMANDS TO CONTROL UNITS.

- A selector channel start I/O instruction has five phases:
  1. Read and load the channel address word (CAW).
  2. Selects a channel.
  3. Loads the CCW into channel registers and selects an I/O control unit and device.
  4. Executes the CCW command.
  5. End the operation.

During I-cycles of a start I/O instruction, the control unit and device address enter the V-register. The channel address enters the U-register. Because this is a description of selector channel 1 start I/O operation, the contents of the U-register is 00000001.

A selector channel start I/O instruction is initialized under microprogram control. When the 2030 recognizes an instruction as a start I/O instruction during I-cycles, it branches the microprogram to the I/O routine. This routine reads out the CAW and then branches to the selector channel microprogram routine, which reads the CCW out of core storage and loads it into selector channel registers. This microprogram routine also controls the initial selection sequence of a control unit and device. After completing the initial selection sequence, the microprogram returns to I-cycles of the next instruction, while the selector channel, the selected control unit, and device execute the CCW command overlapped with CPU processing. Whenever the control unit requires service from the channel, it raises service-in. The channel stops the CPU clock, honors the request for service using its own clock and registers, and restarts the CPU clock.

When the channel has transferred a complete record between the IBM 2030 and an I/O device, the control unit presents ending status to the channel. This generates an I/O interrupt unless the CCW specifies command chaining.

READ AND LOAD CHANNEL ADDRESS WORD (CAW)

- The same microprogram routine reads out the CAW for multiplexor and selector channel operations.
- Check for monitor state.
- Check validity of CCW address.
- Place CCW address in IJ register.
- Store CAW memory protect key in GK register.

The CAW is one word located in main storage at location 72(48Hex). It reads out of storage under microprogram control. This microprogram is common to multiplexor channel operations and selector-channel operations. Besides reading out the CAW, this routine also tests the current PSW to be sure the 2030 is in monitor state. This is necessary because a start I/O instruction is privileged and must be issued only in monitor state.

The microprogram routine reads out the current PSW monitor bit from local storage. If the 2030 is not in monitor state, the microprogram branches to an error routine.

The microprogram now stores the contents of the instruction counter (IJ register) in local storage and places the address of CAW byte 4 (4B Hex) in the T-register. When a start I/O instruction is being executed, the instruction counter has previously been stored by the Execute instruction. When memory is addressed from the T-register, the low order of the CCW address reads out of storage and into the J-register and is checked to see that the CCW is on a double word boundary (3 low-order bits are 0). The T-register decrements by one each cycle and reads out the remainder of the CAW. Byte 3 enters the I-register. In the IBM 2030, byte 2 is checked for all bits zero because storage size does not permit addresses greater than two bytes. The four low order bits of byte 1 (memory protect byte) are checked for zero bits.

When the CAW read out is complete, the microprogram reads out the command byte of the CCW and branches to either the selector-channel fetch-CCW microprogram routine or the multiplexor-channel fetch-CCW microprogram routine.

CHANNEL SELECTION

- Test for valid channel address.
- Turn on selector channel 1 gate.
- Turn on poll control.
- Store unit address in local storage.
- Store next CCW address in local storage.
- Set protect-key in GK register.

This phase of the selector-channel microprogram determines which selector-channel registers to load by examining the channel address in the three low order bits of the U-register. If this address is not valid for the IBM 2030, the microprogram sets the current PSW condition code to three and returns to I-cycles. If the channel address is valid, the microprogram turns on a selector-channel gate. This is a description of selector-channel 1 start I/O operation; therefore, selector-channel-1 (SX1) gate is turned on by the statement GB=K1,0. With SX1 gate on, any subsequent micro-words affecting selector channels will be directed to selector channel 1. This allows selector channel 1 registers (G prefix) to be loaded by the microprogram.

The microprogram now tests the GT lines to determine if the addressed channel is busy. If it is busy, the microprogram sets the current PSW condition code to two and returns to I-cycles.

Assuming the channel address is valid and the addressed channel is not busy, the microprogram attempts to capture polling, that is, not to honor any request-in signals. If polling is not captured within 768 microseconds (M2) or 576 microseconds (M2-I), a time-out error occurs. Poll control cannot be turned on if there is a select-out or an operational-in. After setting on poll control, the microprogram stores the V-register (unit address) in local storage for later use in a command chaining and setting the interrupt code of I/O PSW. When selector-channel 1 is the addressed channel, this local storage position is addressed by the statement K5CPU. It is K21CPU for selector channel 2.

The microprogram also stores the address of the next CCW in local storage for later use in command or data chaining and CSW store. The statements for storing this address in selector-channel 1 local storage is K6CPU and K7CPU. They are K22CPU and K23CPU for selector-channel 2.

If a 2030 has the memory-protect feature the storage protection key of the CAW is stored in the GK register. When any CCW in the start I/O chain is an input CCW, the storage protection key (GK) is compared with the storage key that is read out with each data storage cycle. If they are not equal and the memory protect key is not zero, the location in storage is not changed, and a protection-check is indicated in the subsequent I/O interrupt.

NOTE: The microprogram documents specify locations K5, K6, and K7. If SX2 gate is on, the controls automatically address locations K21, K22, and K23.

INITIAL SELECTION AND CCW LOAD

- Check for a valid TIC on a TIC command (if the first CCW is a TIC, a program check occurs).
- Check for a valid command (not zero).
- Check for a valid count (not zero).
- Load the CCW count into the GCD registers.
- Check for a valid flag byte (three low order bits zero).
- Place the flag byte into the GF register.
- Select the I/O control unit and device.
- Place the command into the GG register.
- Send command-out.
- Load the data address into the GUV registers.

This phase of the microprogram routine performs two major functions:

1. It loads the CCW information into the channel registers.

2. It selects an I/O control unit and device.

The CCW reads out of main storage into the GR register. The flag byte reads into the GR register and then into the GF register. The count reads out, enters the GR register and then the GUV registers momentarily. The microprogram now reads out the unit address from local storage and places it into the GR register. The count in the GUV registers enters the GCD registers. At P1 time of the next cycle, bus-out control is raised gating the contents of the GR register to the bus-out interface lines. During this same cycle, the unit address is placed into the R-register for comparing with the address from the device. At T2 time of this cycle, the address-out latch is turned on. This sets the GO register latches with the contents of the GR register. At T2 time of the next microword, the select-out latch is turned on, raising the hold-out tag line.

While doing this, the microprogram tests the count to insure that it is not zero, even on operations that do not require a data transfer. It also checks the validity of the CCW address (high order byte must be 00) and tests the flag byte to insure that this three low order bits are zero. If either the command is invalid, the count is not zero, the data-address is invalid, or the three low order bits of the flag byte are not zero, the microprogram branches to an error routine (set program check).

When the control unit raises operational-in, the address-out latch resets, dropping address-out. The control unit waits for address-out to fall at its input, raises address-in, and places its address and the address of the selected device on bus-in. When the channel receives address-in, the microprogram places the bus-in information into the GR-register. The microprogram then compares the R-register contents to the GR-register contents. If they do not match, an incorrect control unit or device has been selected and a mismatch error occurs (interface-check latch is turned on GB=K15).

If there were no errors, the CCW command byte now reads out of storage into the GR register and is gated to bus-out. Also at this time, the low order four bits of the command byte are placed into the GG register. The microprogram then turns on the GA register command-out latch, raising command-out. When the control unit resets address-in, the channel command-out latch is reset. While this is taking place, the microprogram places the data address into the GUV registers and tests for status-in from the control unit.

Next, the microprogram tests for zero status and turns on the service-out latch; raising service-out. If the status is zero, the microprogram sets the count-ready-not-zero latch and the current PSW condition code to zero, and returns to I-cycles of the problem program while the selector channel executes the CCW command.

SELECTOR SHARE CYCLE

- Stops the CPU clock and takes one storage cycle.
- Controlled only by selector-channel clock and registers; it does not involve any microprogram routines.
- Does not disturb the contents of any CPU registers.

A selector-share cycle is 2 microseconds (M2) or 1.5 microseconds (M2-I) long. It consists of a 1.0 microsecond (M2) or a .75 microsecond (M2-I) read cycle and write cycle. A selector- share cycle is performed under control of the selector channel clock and registers with no microprogramming involved.

For input, a selector share cycle is initiated by the rise of service-in, which signals the arrival of a byte of data. For output, a byte of data is pre-fetched from storage and placed into the GR register providing the count is ready and not zero, and the GR register is free to accept the data.

Input Selector-Share Cycle

- Read, Read Backward and Sense commands.
- Control unit places information on bus-in and raises service-in.
- The information enters the GR-register if the GR-register is not in use.
- After the information has entered the GR register, the selector channel raises service-out.
- The selector channel stops the CPU clock and starts the selector clock at the end of the current microword or, if a storage cycle is in process, at the end of that read/write or read/compute/write cycle.
- Selector channel stores the input information, and updates the count and data address.
- Selector channel stops selector clock and starts CPU clock.

Figure 4-10 shows the timing sequence of an input selector-share cycle. To follow this timing chart, assume that the GR-full latch is off when service-in rises. This indicates that the information on bus-in may enter the GR register immediately.

The rise of the service-in and not service-out line activates the SX1-share-request line and the set bus-in to GR pulse. After a delay of 125 nanoseconds, the GR-full latch turns on, and service out is sent to the I/O device to indicate that a byte of data has been accepted. The SX1-share-request line remains on until a selector-share-cycle is initiated. It is an indication that the channel will assume control of core storage at the end of the present read/write or read/compute/write sequence.

At T3 time of the CPU write or unmasked cycle the selector-share-cycle latch turns on providing positive indication that the channel will assume control of core storage at the end of the write cycle. At T4 time of the same cycle, the selector-share-hold latch turns on. This latch stops the CPU clock at the end of T4 time and starts the selector clock.

With the selector-share-cycle latch on, the GUV registers (data-address) are gated to the MN registers and at selector T1 time, this address is set into the MN registers. Also at selector T1 time, the SX1-read-cycle latch turns on indicating a selector read cycle,, and a selector read-call is generated and sent to main storage. The SX1 read cycle gates the GCD registers (count) to the GHP and GHQ buses (modifier entry).

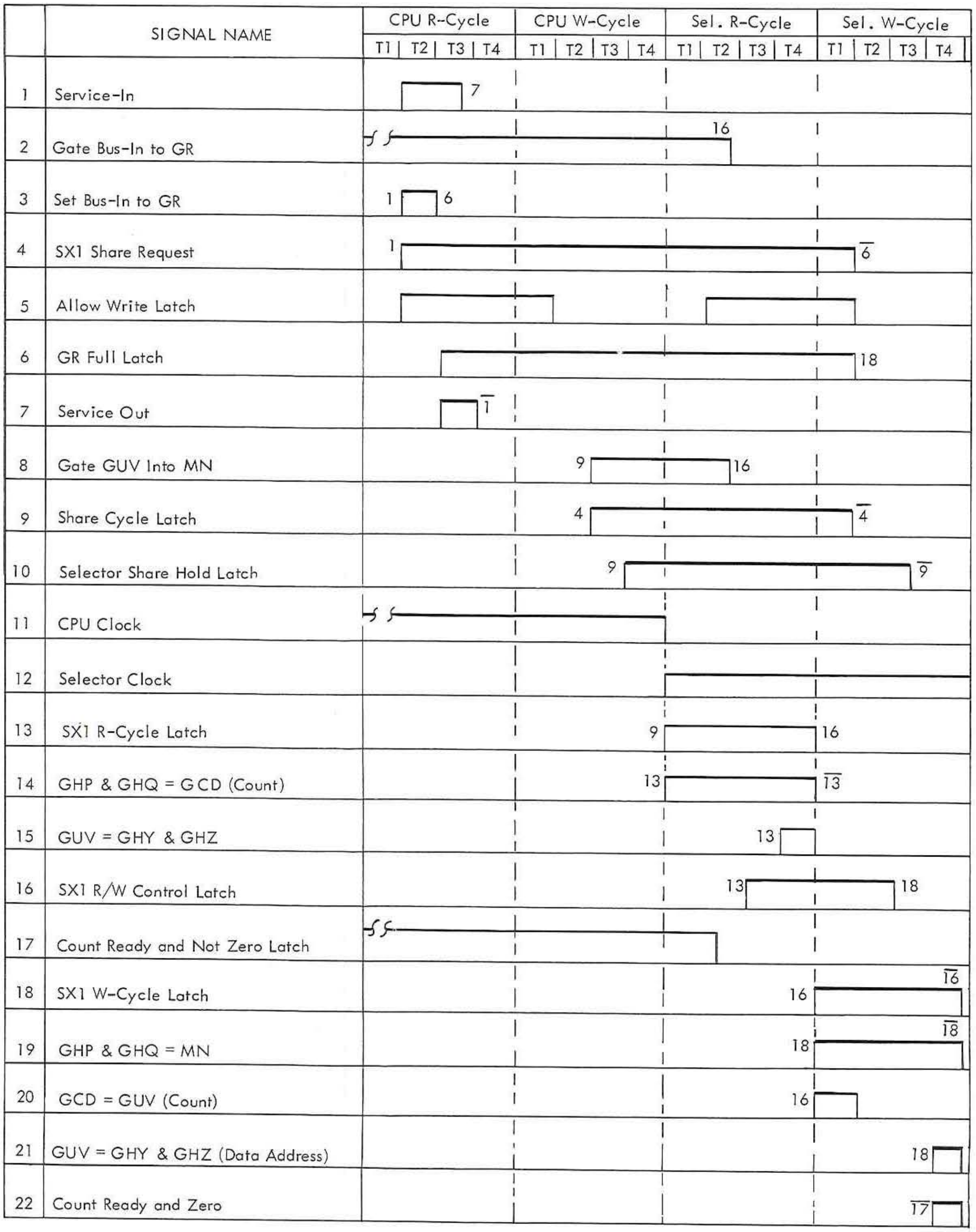


Figure 4-10. Input Selector Share Cycle

During the read cycle, the modifier subtracts one from the count and places this modified count on the GHY and GHZ buses. At T4 time of the selector read cycle this modified count is placed into the GUV registers. At the same time, the decremented count is tested for a zero value. If it is zero, the count-ready-and-not-zero latch is reset.

At T3 time of the read cycle, the SX1 read-write control latch turns on. The purpose of this latch is to eliminate timing conditions during the transition from a selector read cycle to a selector write cycle. The latch is reset at T3 time of the write cycle.

The AND condition of SX1 read-write control latch and T1 time provides the reset for the SX1-read-cycle latch and the set for the SX1-write-cycle latch. This condition also sets the contents of the GUV registers (modified count) into the GCD registers.

The SX1-write-cycle latch, when on, gates the contents of MN (data-address) to the GHP and GHQ buses. The modifier then updates the data-address either +1 or -1 (-1 for a read-backward operation). At T4 time of the write cycle, the modified data-address is set into the GUV registers. If at this time the new count is zero, the count-ready-zero latch is turned on.

If service-in rises before the end of T3 time of the write cycle, the selector-share-hold latch remains on and the selector channel proceeds with another share cycle. If service-in does not rise before the end of T3 time, the selector-share-hold latch turns off at T4 time and the CPU clock restarts at the end of selector T4 time.

Circuit Objectives

1. Initiate a selector-share cycle.

   a. A control unit raises service-in to indicate a request for service.

   b. Store bus-in information in GR register.

      (1) Generate the gate bus-in to GR line. Input, count ready and not zero, not poll control, and SX1 R/W control.

      (2) Generate the set bus-in to GR line

         (a) Service-in not service-out delayed

         (b) Not Selector Channel 1 write cycle

         (c) GR-full latch off

         (d) Input

         (e) Count ready not zero.

      (3) Place Bus-in information into GR register.

         (a) Bus-in information

         (b) Gate bus-in to GR

         (c) Set bus-in to GR

   c. Raise service-out

      (1) Generate service signal

         (a) Set bus-in to GR delayed 125 nanoseconds.

      (2) Turn on GR-full latch

         (a) Input

         (b) Service signal

      (3) Turn on service-out latch

         (a) Service signal

         (b) (Not) set bus-in to GR

   d. Generate selector channel 1 (SX 1) share request

      (1) Service-in not service-out, not Halt-I/O, and count ready not zero (This generates SX1 share-request until GR-full comes on and service-out is given).

      (2) Input operation, GR full, and count ready not zero (This holds SX1 share-request on after service out is given).

      (3) Output operation, GR not full, count ready not zero. This holds SX1 share-req on after service out is given).

2. Request a selector-share cycle

   a. Stop CPU clock at the end of a write or unmasked cycle and start the selector-share cycle.

      (1) Turn on selector-share-cycle latch

         (a) SX1 share request

         (b) Not selector channel 2 (SX2) share request or not SX1 write cycle (this is part of the priority control. Assume the lines are not active).

(c) Allow-write latch off (turned off at CPU write T2 time).

(d) Selector-share count-3 latch off or not H6 (part of priority control)

(e) CPU T3, Sel T3, or not oscillator pulse (in case CPU clock is off)

(2) Turn on selector-share-hold latch

(a) Selector-share-cycle latch on

(b) CPU T4, Sel T4, or oscillator pulse (in case CPU clock is off).

(3) Turn off CPU clock

(a) Selector-share-cycle latch (on) pulls down the clock-start line.

(b) CPU clock continues to the end of CPU T4 time then stops

(4) Turn on selector channel clock

(a) Selector-share-hold latch and not osc. pulse turns on Sel-P1 latch

3. Execute selector-share read cycle.

a. Turn on read-cycle latch

(1) SX1-share-cycle latch on

(2) Sel T1 Time

(3) SX1-read/write (R/W) control latch off

b. Address core storage

(1) Generate gate GUV to MN

(a) Selector-share-cycle latch on

(3) SX1-read/write

(b) SX1 R/W control latch off

(2) Generate Set GUV to MN

(a) Selector T1 time

(b) SX1 R/W control off

c. Decrement Count

(1) Gate GC and GD registers to the GHP and GHQ buses

(a) GC, GD register output

(b) Selector-1 read cycle latch on.

d. Generate selector read-call

(1) SX1 R/W control latch off

(2) SX2 R/W control latch off

(3) Sel T1 time

e. Generate selector auxiliary read call

(1) SX1 R/W control latch off

(2) SX2 R/W control latch off

(3) Sel T2 time

f. Turn on SX1 R/W control latch

(1) SX1 read cycle latch on

(2) Sel T3 time

g. Place decremented count into GUV registers

(1) GHY and GHZ buses (modifier output)

(2) Selector read cycle

(3) Sel T4 time

h. If Count is decremented to zero turn off count ready and not zero latch

(1) SX1 read cycle

(2) GHYZ zero (count equals zero)

(3) Sel T4 time

i. Turn on channel control check latch if modifier parity check or carry-out check is detected during the selector read cycle.

(1) SX1 modifier check

(a) Modifier parity even, or mismatch between CHQ carry-out and GHZ carry-out

(b) Sel read cycle

(c) Sel T4 time

(2) Turn on channel control latch if even parity in M or N register

(a) MN register parity check

(b) T3 time of selector read cycle

j. If a channel control check was detected, reset count ready and not zero latch (This latch normally stays on for the complete data transfer).

k. Status stop condition

(1) Channel control check

(2) Interface control check

(3) Channel data check

(4) Protection check

(5) Program check

4. Execute selector write cycle

a. Turn on SX1 write cycle latch

(1) SX1 R/W control latch on

(2) Sel T1 time

b. Place count into GCD registers

(1) Generate GC and GD sets

(a) Sel T1 time

(b) SX1 R/W control

(2) Place count into GCD registers

(a) GC and CD sets

(b) GUV outputs

c. Update data-address

(1) Gate MN register to the GHP and GHQ buses

(a) MN register outputs

(b) SX1 write cycle

d. Generate selector write-call

(1) SX1 R/W control latch on

(2) Sel T1 time

e. Generate auxiliary write-call

(1) SX1 R/W control latch on

(2) Sel T2 time

f. Reset GR-full latch

(1) Input

(2) SX1 write cycle

(3) Sel T2 time

g. Turn on count ready and zero latch

(1) Count ready and not zero latch off

(2) SX1 write cycle

(3) Sel T4 time

(4) Not status stop condition

h. Reset selector-share-cycle latch

(1) Not SX-1 share request

(2) Sel T3 time

(3) Allow write latch off

i. Reset selector-share-hold latch

(1) Sel T4 time

(2) Selector 1 share cycle latch off

(3) Selector 2 share cycle latch off

j. Place updated data-address in GUV registers

(1) GHY and GHZ bus information (modifier output)

(2) GU and GV sets

(a) SX1 write cycle

(b) Sel T4 time

k) Stop selector clock and start CPU clock

(1) Selector-share-hold latch off

(2) Clocks-start line up.

Output Selector-Share Cycle

- Write and control commands.
- A share-request is initiated by the GR-full latch off; information in the GR-register has been transmitted to an I/O control unit.
- Selector channel initiates a selector-share cycle upon completion of initial selection.
- When service-in rises:
  1. The bus-out control latch turns on to gate the GO-register to bus-out.
  2. Service-out rises (after a delay of 125 ns).
  3. Service-out latches up the GO register and frees GR for another byte of data.
  4. GR-full latch turns off.
- When the GR-full latch is off:
  1. A selector share-request is generated and the CPU clock stops. The selector clock starts when the present CPU R/W or R/C/W sequence is complete.
  2. Data address enters the MN register, and selector read call is sent to memory.
  3. Data from memory enters GR register, and if service-out is reset, the data feeds from GR into the GO register.
  4. Modifier updates the count and data address.
  5. The selector clock stops and CPU clock starts.
- Figure 4-11 shows output selector-share cycle timing.

For output operations, the selector channel attempts to have a byte waiting in the GR and GO registers (same byte in both registers) before a control unit raises service-in calling for that byte. Because of this requirement to look ahead, the selector channel initiates a selector-share cycle immediately following the initial selection of a control unit.

When the selected control unit raises service-in, requesting a byte of information, the channel turns on the bus-out control latch which gates the GO-register information to bus-out. After a delay of 125 nanoseconds, the channel resets the GR-full latch and raises service-out. This identifies the information on bus-out as data (either record data or control data). Service-out also allows the GO-register latches to latch, holding the data on bus-out until service-out falls. This allows the channel to reset the GR-register and load it with new data while maintaining data on bus-out.

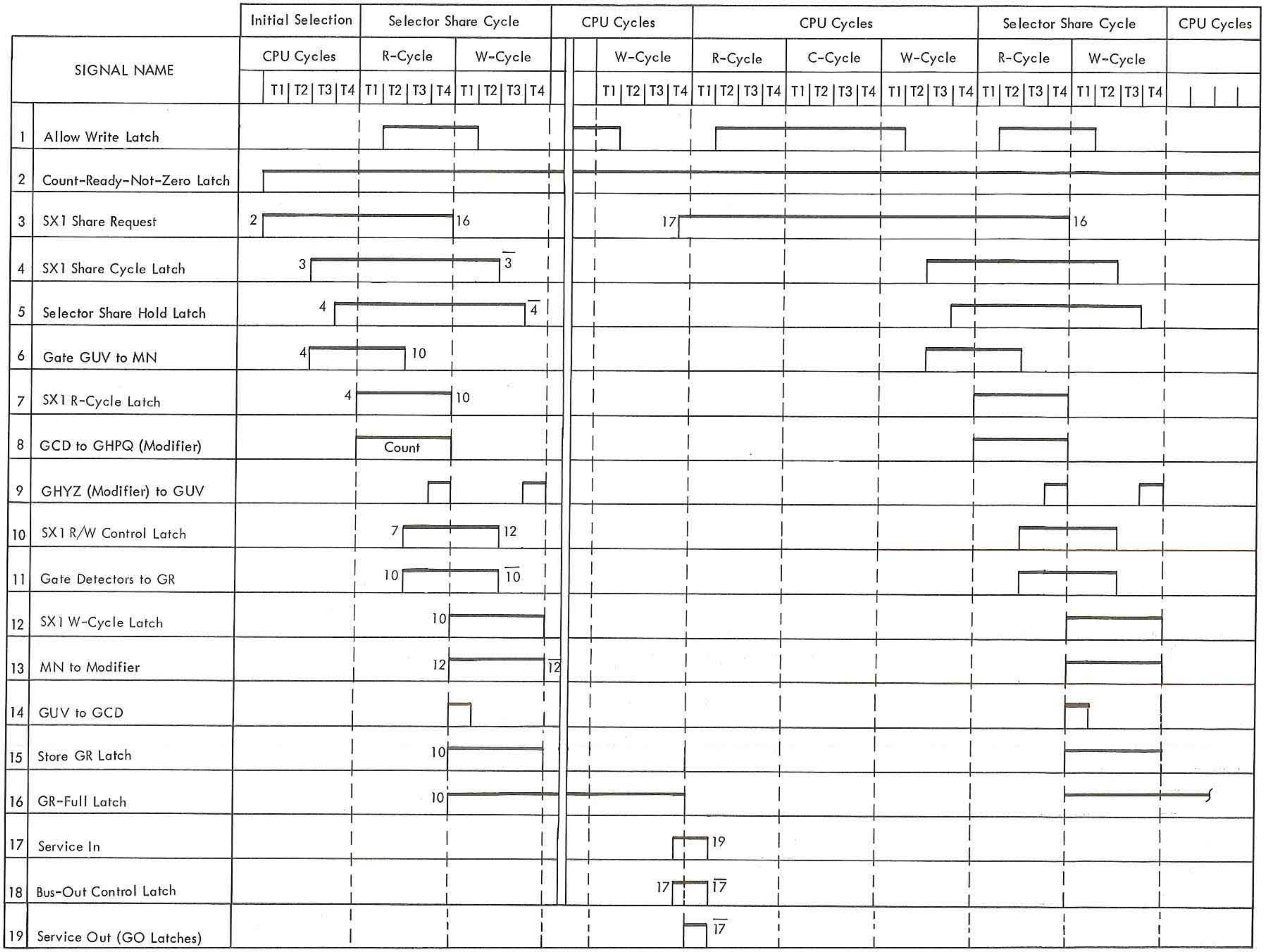


Figure 4-11. Output Selector Share Cycle

When the GR-full latch is off, and the count ready and not zero latch is on, it initiates a selector-share request. A selector share-request may also be initiated by service-in not service-out, count ready and not zero, and not halt I/O. The GR-full latch turns off at the same time the channel raises service-out. If the GR-full latch turns off before the end of T3 time of either a CPU write cycle, an unmasked cycle, or a selector-share write cycle, the selector-share cycle begins at the end of T4 time.

A selector-share cycle consists of a one microsecond read cycle (R-cycle) and a one microsecond write cycle (W-cycle). For M2-I, the read and write cycles are .75 microseconds long.

During the R-cycle, the data address enters the MN register, addressing core storage. The information in the core-storage position addressed enters the GR-register (does not become stable until W-cycle). Also during the R-cycle, the count enters the modifier, is decremented by one, and enters the data-address registers (GUV).

During the W-cycle, the count enters the count registers (GCD) from the data-address registers while the data address enters the modifier from the MN register. The data address is modified by plus one. Also, during a selector-share W-cycle, the GR information regenerates the addressed core-storage position. At the end of the W-cycle the updated data-address enters the data-address registers (GUV) from the modifier. The 2030 returns to CPU cycles at the end of a selector-write cycle unless either selector-channel 1 or 2 has a request for service by the end of selector write cycle T3 time.

Circuit Objectives

1. Load GR-register with the first output byte.

   a. Stop CPU clock and start selector clock

      (1) Generate SX1 share-request

         (a) Output operation

         (b) GR-full latch off

         (c) Count-ready-not zero latch on (turned on by GB=K8,1 of start I/O microprogram routine).

      (2) Turn on selector-1 share-cycle latch

         (a) Allow write latch off (turns off at T2 time of a write cycle).

         (b) Not SX2 share-request, or not SX1 write cycle

         (c) CPU T3 time

         (d) SX1 share-request.

      (3) Turn on selector-share-hold latch

         (a) Selector 1 share-cycle latch on

         (b) CPU T4 time.

      (4) Stop CPU Clock

         (a) Selector-share-cycle latch on prevents the CPU clock from starting another cycle.

      (5) Start Selector Clock

         (a) Selector-share-hold latch on.

         (b) Oscillator pulse.

   b. Place data-address in MN-register.

      (1) Generate Gate-GUV-to-MN Register

         (a) Selector 1 share cycle latch on

         (b) SX1 R/W control latch off

      (2) Generate MN register set pulse

         (a) Sel T1 time

         (b) SX-1 R/W control latch off

   c. Execute selector share read cycle.

      (1) Turn on SX-1 R-cycle latch

         (a) SX-1 share cycle latch on.

         (b) SX-1 R/W control latch off.

         (c) Selector T1 time.

      (2) Modify count by minus one.

         (a) Generate - 1 MOD GHPQ

            SX1 read cycle

         (b) Place count (GCD register) on GHP and GHQ buses (modifier input).

            GCD register output

            SX-1 R-cycle latch on

      (3) Turn on SX-1 R/W control latch

         (a) SX-1 R-cycle latch on

(b) Selector T3 time.

(4) Place storage data in GR register

(a) Generate gate detectors to GR line.

Output operation

SX-1 R/W control latch on

(b) Place storage data in GR register.

Storage data out

Gate detectors to GR

(Memory) data ready

(5) Place count in GUV-registers.

(a) Generate set-GV and set-GU lines

Selector T4 time

SX1 read cycle

d. Execute selector share write cycle.

(1) Turn on SX-1 W-cycle latch.

(a) SX-1 R/W control latch on

(b) Selector T1 time

(2) Turn off SX-1 R-cycle latch

(a) Selector T1 time

(b) SX-1 R/W control latch on

(3) Place count (GUV registers) in GCD registers.

(a) Generate GC and GD set lines.

SX-1 R/W control latch on

Selector T1 time

(b) Place GUV-register in GCD register

GUV-register output

GC and GD set lines

(4) Modify data address plus one (assume a write CCW).

(a) Generate + 1 Mod GHPQ line

SX1 write cycle

Not SX1 read backward

(b) Place MN register on GHP and GHQ buses (modifier input buses).

MN register outputs

SX1 write cycle

(5) Regenerate core storage.

(a) Turn on store-GR latch

SX1 R/W control

Sel T1 time

(6) Place modifier output in GUV-register.

(a) GHY and GHZ buses (modifier output)

(b) Selector T4 time

(c) SX-1 write cycle

(7) Turn off R/W control latch

(a) SX-1 W-cycle latch on

(b) Selector T3 time

(8) Turn off W/cycle latch.

(a) SX-1 R/W control latch off

(b) Not selector P3 time

e. Stop selector clock and start CPU clock

(1) Reset selector-share-hold latch

(a) Selector T4 time

(b) Selector 1 share-cycle latch off (turned off at selector W-cycle T3 time.

(c) Selector 2 share-cycle latch off

(2) Selector clock stops and CPU clock starts at the end of selector W-cycle T4 time.

2. Honor a request for service.

a. Service-in rises.

b. Generate SX1 share request

(1) Service-in not service-out

(2) Count-ready-not-zero latch on

(3) Not halt I/O instruction

c. Place GO-register contents on bus-out.

(1) Turn on bus-out control latch

(a) Halt I/O stop latch off

(b) Output operation

(c) GR-full latch on

(d) Service-in not service out

(e) Not SX1 status stop condition

(2) Place GO-register on bus-out.

(a) GO-register output

(b) Bus-out control latch on

d. Raise Service out.

(1) Generate service signal

(a) Bus-out control latch delayed 125 nanoseconds

(2) Turn on service-out latch

(a) Not Set bus-in to GR

(b) Service signal

(3) Turn off GR-full latch

(a) Output operation

(b) Service signal

(4) Hold SX1 share request line.

(a) GR-full latch off

(b) Output operation

(c) Count-ready-not zero latch on

(5) Set GO-register latches

(a) Service-Out

Note: The GR-register contents may now be changed with no effect on bus-out.

Channel Share Priority

- Determines which channel can originate a share cycle.
- With all channels requesting service, priority circuits allow:

  Selector channel 1, 50% of the time.

  Selector channel 2, 25% of the time.

  The multiplexor channel 25% of the time.

The priority assignment of the channels is fixed by circuit design and can not be changed. Selector-channel 1 has the highest priority followed by selector-channel 2, then the multiplexor channel.

With all channels requesting service, selector-channel 1 assumes control and turns on the SX1 share-cycle cycle latch (Figure 4-12). Selector-channel 1 is prevented from taking two share cycles if the SX2 share-request line is active.

The counting circuit in Figure 4-12 determines multiplexor channel priority.

With all channels requesting service, three selector-channel cycles may be taken before the multiplexor channel obtains priority for a R/W, a R/C/W, or an unmasked micro-program cycle.

The priority circuits are effective only when more than one channel requests service. In all instances where only one channel requests a storage cycle, it has priority over the CPU and the request is honored at the end of the CPU write or unmasked cycle.

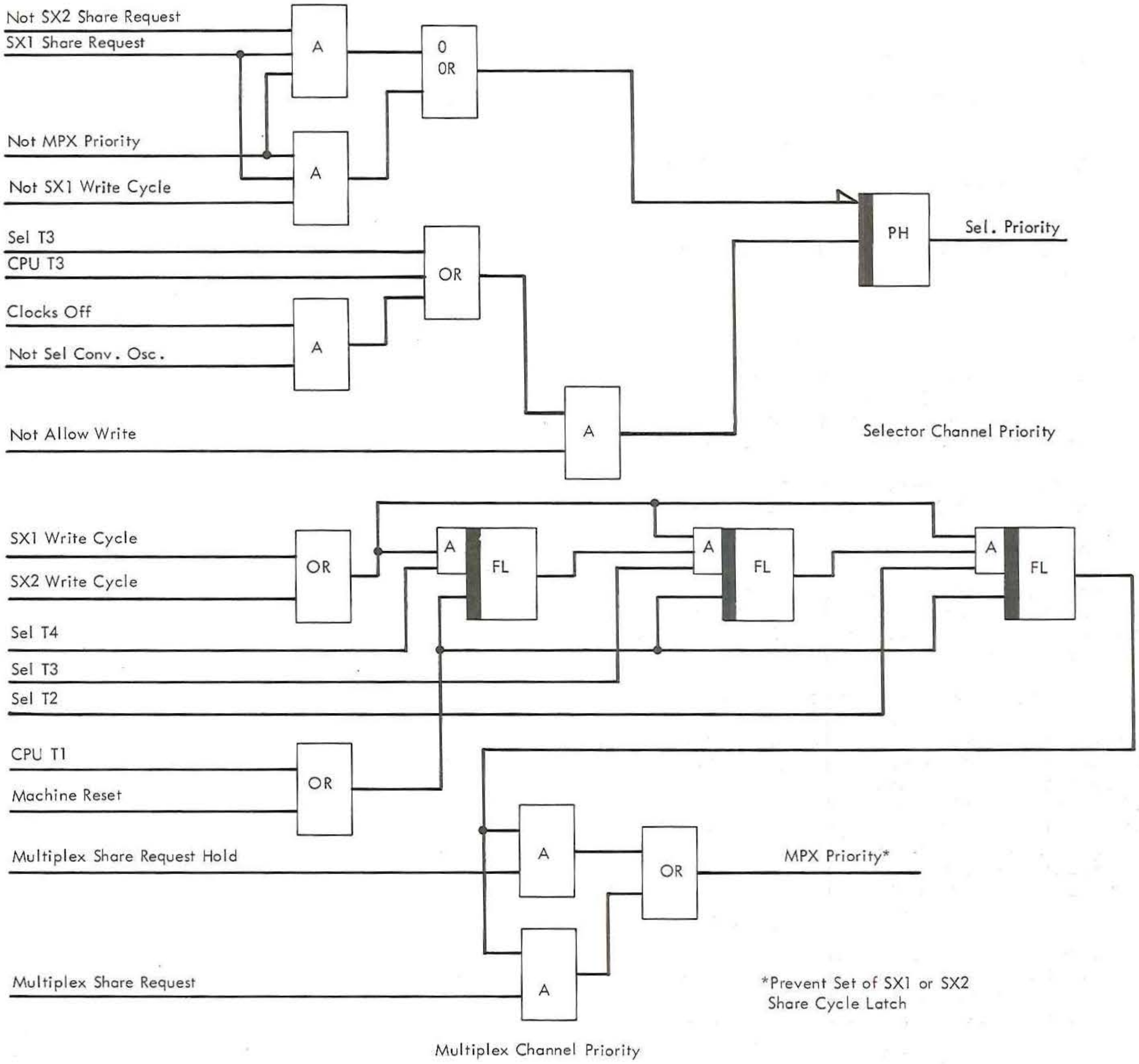


Figure 4-12. Priority

CCW ENDING PROCEDURE

- The ending procedure is initiated by ending status presented by a control unit.
- Ending status may initiate a request for a new CCW or a request for an interrupt (a new CCW if command chaining).

A control unit may present ending status to the channel:

1. during initial selection.
2. when the number of bytes specified by the CCW count have been transferred or when the device cannot send or accept more data (channel-end status).
3. when an I/O device has completed the execution of a CCW, and is ready to accept a new command (device-end status).

During initial selection, a control unit may present channel-end and device-end status to the channel to indicate a command that requires no data transfer.

A control unit presents channel-end status to the channel when the number of bytes specified by the CCW count have been transferred, or when it cannot send or accept any more data. A control unit may also present device-end status at this time. This is the case when the I/O device is in a position to accept a new command. Tape units and file-control units are good examples of I/O devices that keep pace with the data transferred on the interface.

A control unit may present device-end status to the channel when a selected device completes the execution of a CCW at some time after the control unit has presented channel-end status to the channel. For example, a buffered device in an output operation presents channel-end status when the data transfer is complete, operates on the data in the buffer, and presents device-end to the channel when the CCW execution is complete.

The channels respond to status-in in different ways, depending on whether the command-chaining bit of the CCW is on or off. If the CCW specifies command chaining, the channel initiates a request for the ROS chaining microprogram routine. This routine loads the next CCW into the channel registers and goes through an initial selection sequence after which the interrupted microprogram is restarted at the point where it was discontinued.

When the channel receives ending status, and the CCW does not specify command chaining, the channel activates interrupt-ROS-request. This causes a trap to a selector channel microprogram which turns on the SX-1 interrupt latch in the channel. It accepts the device status if the channel-busy latch is on, or stacks the status back to the control unit if the channel busy latch is off. Now, if the current PSW has the I/O interrupt mask-bit for this channel set to one, the channel calls for an I/O interrupt. When the CPU honors this requests for an interrupt, it stores the current PSW (presently in local storage in the old-I/O-PSW location, loads the Channel Status Word (CSW) with the required information, and loads the PSW from the I/O-new-PSW location into the current PSW location in local storage.

When the channel receives ending status, and the CCW specifies command chaining, the channel proceeds with the chaining only if no errors have been detected to this point in the operation. If any check latches are on, the channel generates interrupt-ROS-request as described earlier.

Refer to CCW Flag Operations for a description of command chaining.

I/O INTERRUPT

- The selector channel initiates an interrupt when it receives ending status, and command chaining is not specified by the CCW.
- To initiate an interrupt, the selector channel must first turn on its interrupt latch under microprogram control. It breaks into (traps) the microprogram to do this. (A microprogram address is forced by hardware into the WX registers)
- At the end of the current PSW E-phase, the CPU honors the selector-channel interrupt.
- The I/O interrupt microprogram routine:

  1. Stores the current PSW in the I/O old PSW location.
  2. Generates the channel status word (CSW).
  3. Loads the I/O new PSW into the current PSW locations in local storage
- The CPU executes the I/O interrupt program under control of the I/O new PSW.

The selector channel initiates an I/O interrupt when it receives ending status from a control unit, and the GF-register CC-latch is off (not command chaining).

An I/O interrupt is a discontinuance of the program (instruction sequence) controlled by the current PSW, and the execution of a new program sequence, which is under control of a new PSW. In this case, the new PSW is the I/O new PSW located at main storage position 120 (78 Hex). The CPU cannot honor an interrupt request until it has completed the execution of the current instruction (G-register). Also, the current PSW system mask bit that corresponds to the channel requesting the interrupt, must be on.

To initiate an interrupt, the selector channel must turn on its interrupt latch. It does this under microprogram control. Therefore, it must obtain ROS control and break into (trap) the current microprogram. A break in the microprogram can occur at the end of any CPU cycle if no write cycles are pending (the allow-write latch off).

When the selector-channel interrupt trap microprogram routine is complete (remember this routine only turns on the selector channel interrupt latch and accepts or stacks the device status), the microprogram restarts at the point it was discontinued by the selector channels request for ROS control. (The contents of GW GX is gated to the WX registers)

When the CPU has completed the execute phase of the current instruction (G-register), it honors the selector-channel interrupt provided the system mask bit for the channel is on. It does this by storing the current PSW (located in local storage in the I/O old PSW location (main storage position 56 or 38 Hex), loading the channel status word (CSW, main storage location 64 or 40 Hex) with the required information, and loading the I/O new PSW into local storage as the current PSW. After this is accomplished, the CPU returns to I-cycles and starts the execution of the program controlled by the current PSW, which is in this case the I/O new PSW. At some later time, the program must load the I/O old PSW into local storage as the current PSW, to return to the original program. Figure 4-13 shows the general concept of an I/O interrupt.

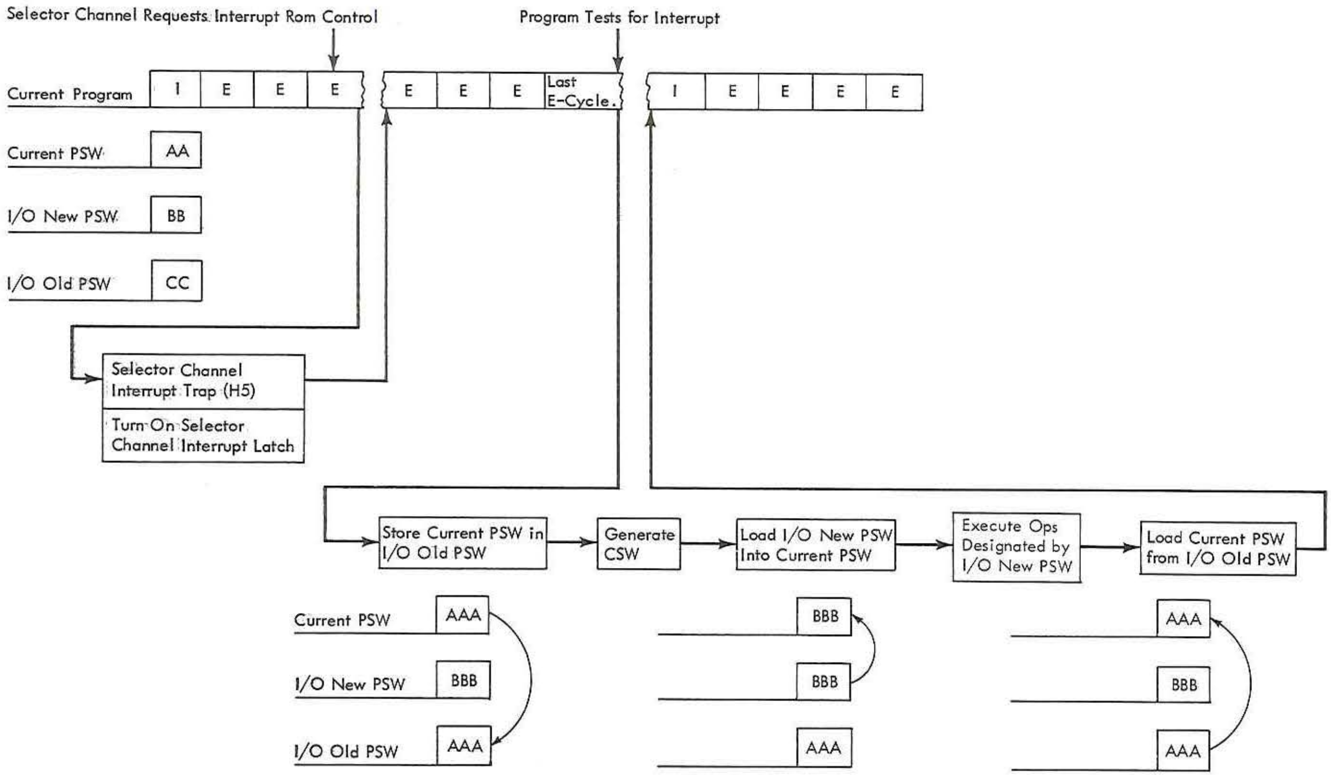

Selector Channel Requests Interrupt Rom Control
Program Tests for Interrupt
Current Program
I
E
E
E
E
E
E
Last E-Cycle.
I
E
E
E
E
Current PSW
AA
I/O New PSW
BB
I/O Old PSW
CC
Selector Channel Interrupt Trap (H5)
Turn On Selector Channel Interrupt Latch
Store Current PSW in I/O Old PSW
Generate CSW
Load I/O New PSW Into Current PSW
Execute Ops Designated by I/O New PSW
Load Current PSW from I/O Old PSW
Current PSW
AAA
I/O New PSW
BBB
I/O Old PSW
AAA
BBB
BBB
AAA
AAA
BBB
AAA

Figure 4-13. I/O Interrupt Concept

Selector Channel Interrupt ROS Request

- The selector channel requests ROS control when it receives ending-status and the CCW does not specify command chaining.
- The CPU allows the selector channel to take ROS control following a cycle which has no write cycles pending and when there are no higher priority requests for ROS control.
- The selector channel stores the ROAR in its back-up ROAR (GWX-register) before it assumes ROS control.
- The selector channel stores the X6 and X7 branch conditions in its X6 and X7 buffer latches.

To initiate an I/O interrupt, the selector channel must first turn on the selector-channel-interrupt latch. This is accomplished by the selector-channel-interrupt microprogram routine (interrupt trap).

When the channel receives ending status from a control unit, and the GF-register CC latch is off, it generates a request for ROS control. This request turns on the selector channel stacking latch at T3 time if the allow-write latch is off and there are no higher priority requests for ROS control.

When the selector channel stacking latch turns on, it:

1. Stores the read-only address register (ROAR) contents in the selector channel back-up ROAR (GWX-register).
2. Conditions the X-register to set to the address of the first ROS word of the selector-channel-interrupt microprogram routine.
3. Stores the X6 and X7 branch conditions in their respective selector-channel buffer latches on the next CPU cycle.
4. Prevents the set of the control register on the next cycle (dead cycle).

Figure 4-14 shows the timing sequence of the transition period from the current microprogram routine to the selector channel microprogram routine. Notice that the ROS address of the last microprogram word addressed, not the next microprogram word address, is stored in GWX. Normally, a ROS word is executed (control register set) one CPU cycle after it is addressed. When the first selector-channel-interrupt microprogram word is addressed, the control register is prevented from setting by the any-priority pulse. This creates an inactive cycle as far as the execution of a microprogram word. The WX-register is conditioned to set to the address of the first ROS word of the selector channel interrupt microprogram routine. The W-register is allowed to reset to zero at its normal set time (T1) and the W-register parity latch is turned on by the any-priority-pulse. The X-register four bit is conditioned to turn on.

The X6 and X7 branch conditions are stored in buffer latches because the branch conditions of the last major microprogram word may be dynamic conditions, such as the Z-bus-zero. These conditions would not be present when this microprogram is executed which is upon the resumption of the major microprogram. Therefore, the X6 and X7 branch conditions are stored in the selector channel X6 and X7 buffer latches and are gated to the X-bus when the major microprogram is resumed.

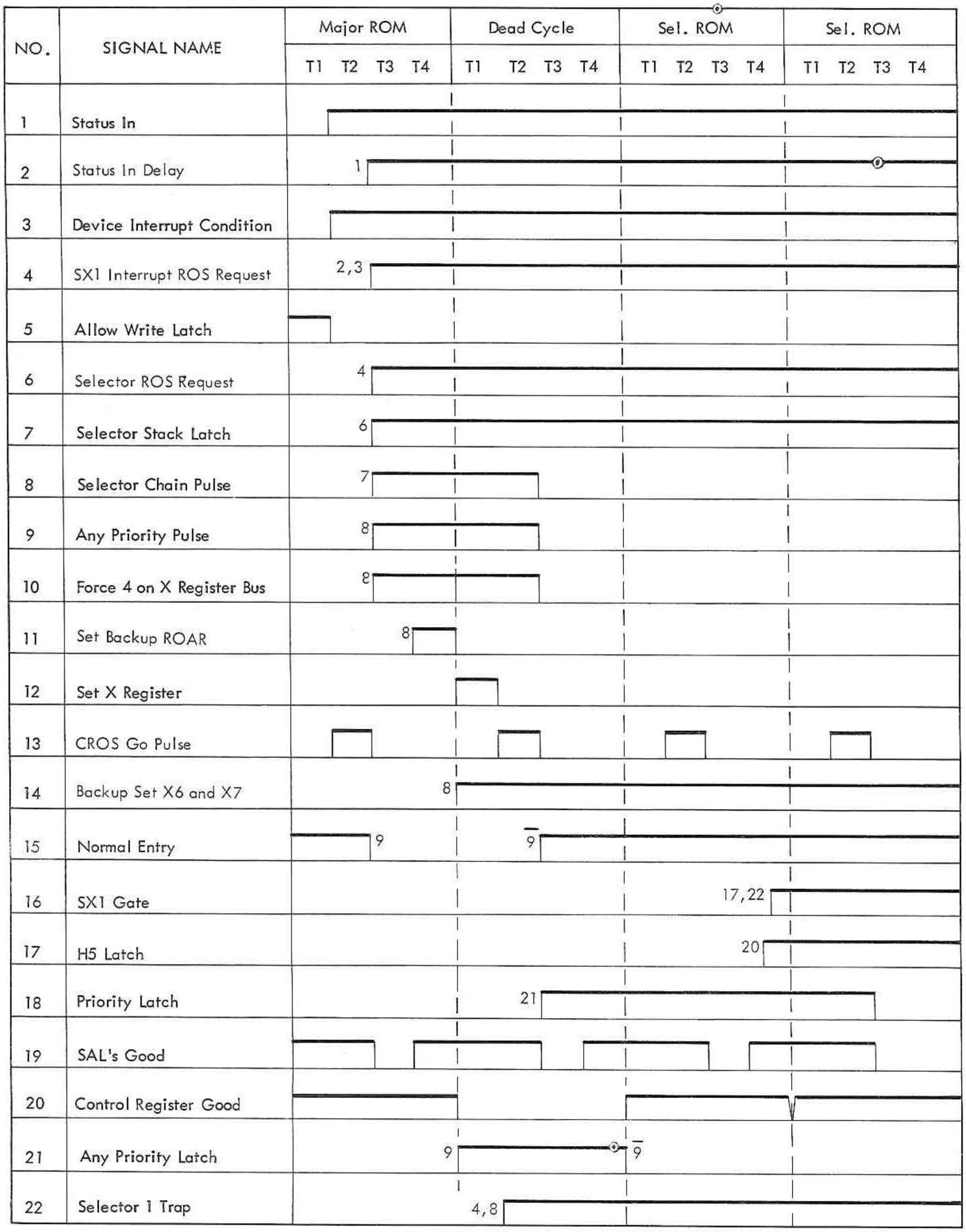


Figure 4-14. Selector Channel Interrupt Trap Timing

Circuit Objectives

1. Control unit presents ending status

   a. Status-in

   b. Bus-in bit 4 (channel-end)

   c. Bus-in bit 5 (device-end)

2. Turn off count ready and not zero latch.

   a. Status-in, not service-out delayed 375 ns

      (1) Not poll control

      (2) Not set count ready

      (3) Not share-request

      (4) Not R/W control

3. Generate selector channel ROS request

   a. Generate device-interrupt-condition

      (1) Status-in

      (2) Channel-end

      (3) Not CC

   b. Generate interrupt condition

      (1) Status-in

      (2) Device-interrupt condition

   c. Generate SX1 interrupt ROS request

      (1) Status-in, not service-out delayed 375 nanoseconds

      (2) SX1 interrupt latch off.

      (3) Interrupt condition.

      (4) Halt I/O stop latch off.

      (5) H6 off (MPX CHANNEL NOT WORKING)

   d. Generate selector-ROS-request

      SX1 interrupt ROS request.

4. Assume ROS Control

   a. Turn on selector-chain-request latch

      (1) Selector ROS request

      (2) Not H-register-5 bit

      (3) Not suppress A reg. check latch

      (4) Not allow write latch

      (5) T3 pulse

   b. Generate selector chain pulse (KM081)

      (1) H-register-5 latch off

      (2) Selector chain request latch on

      (3) Priority latch off

      (4) No higher priority latches on

   c. Generate any-priority pulse

      (1) Selector-chain pulse

   d. Place address of first selector-channel-interrupt microword on X-bus and W-bus (ROS address 008).

      (1) Condition X-bus 4 line

         (a) Selector-chain pulse

         (b) Any-priority pulse

      (2) Condition W-bus parity bit line

         (a) Any-priority-pulse

         (b) Not gate-switches-to-WX latch

         (c) Not 1401 mode

   e. Place ROAR into Back-up ROAR

      (1) Set X-register contents into GX register

         (a) X-register output

         (b) Selector-chain pulse

         (c) T4 pulse

      (2) Set W-register contents into GW register

         (a) W-register output

         (b) Selector-chain pulse

         (c) T4 pulse

   f. W-and-X-buses set into the WX register at T1 time. This addresses the first selector-channel-interrupt ROS word which is an instruction to turn on the H-register-5 latch.

   g. Turn on the any-priority latch

      (1) Any-priority pulse

      (2) T1 time

h. Turn on selector-channel 1 trap latch

(1) Selector-chain pulse

(2) SX1 interrupt ROS request

(3) Not CC ROS request.

(4) CPU T2 time

Note: This latch will stay in its present state until the next selector channel ROS request.

i. Set selector channel X6 and X7 buffer latches

(1) X6 data

(2) X7 data

(3) Selector-chain pulse

(4) T1 time

j. Turn on priority latch

(1) Any-priority latch on.

(2) T3 time

k. Deactivate selector-chain pulse and any-priority pulse.

(1) Priority latch on

l. Turn off priority latch

(1) Not any-priority pulse

(2) T3 time

(3) H-register is Z-bus destination

Note: This is two CPU cycles later than step K. (one cycle after the selector channel interrupt microprogram calls the H-register as the Z-bus destination).

m. The execution of the selector-channel-interrupt microprogram has begun.

Selector Channel Interrupt Trap Microprogram

- This trap is executed as a result of a control units presentation of status-in or address-in (not initial selection).
- It turns on the interrupt latch for status-in and not IPL.
- It checks the status during IPL.
- It stores the unit address for address-in.

Figure 4-15 is a flow chart of the selector-channel-interrupt microprogram (trap) routine. The selector channel requests the CPU to execute this microprogram routine for one of two reasons:

1. The channel has received status-in from a control unit (not initial status).
2. The channel has received a request for selection from a control unit (request-in, select-out, address-in).

In both cases, the microprogram turns on the H-register-5 latch which identifies the microprogram as one associated with the selector channel. The H-register-5 latch also prevents the other selector channel or the multiplexor channel from obtaining ROS control while it is on . The microprogram also resets the select-out latch when it is being executed for either reason. When the H-register-5 latch turns on, it generates the selector-channel gate for the selector channel requesting the execution of this microprogram (SX1 gate or SX2 gate).

When the selector-channel-interrupt trap microprogram is being executed as a result of a control unit's presentation of status-in, and the CCW execution was not a result of initial program load (IPL)the microprogram:

1. Accepts the status from the control unit if the channel is busy, and stacks it if the channel is not busy.
2. Turns on the selector-interrupt latch associated with the selector channel making the ROS request.
3. Turns off the H-register-5 latch.
4. Allows the CPU to resume the execution of the original microprogram.

Request-in cannot turn on the select-out latch as long as the interrupt latch remains on. Therefore, select-out remains inactive, and status-in remains active until the CPU stores the CSW at which time the interrupt latch is turned off by a microprogram word specifying channel-reset.

When the selector-interrupt trap microprogram is being executed as a result of status-in, and the CCW was executed as a result of IPL, the microprogram:

1. Places the unit status into the R-register.

2. Check channel status.

3. Turns off the H-register-5 latch.

4. Returns to the IPL operation.

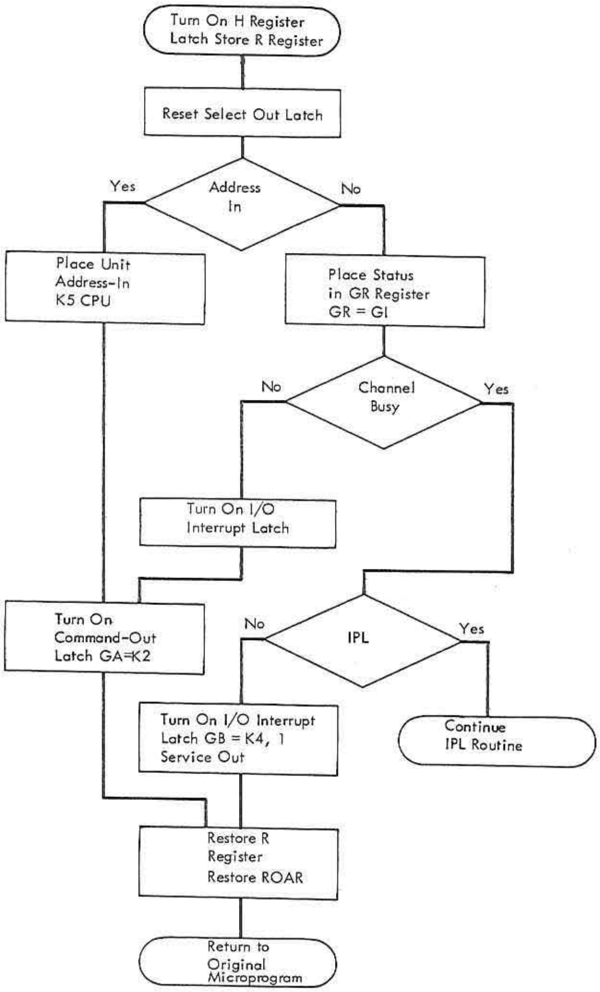

Turn On H Register Latch Store R Register
Reset Select Out Latch
Yes
Address In
No
Place Unit Address-In K5 CPU
Place Status in GR Register GR = GI
No
Channel Busy
Yes
Turn On I/O Interrupt Latch
Turn On Command-Out Latch GA=K2
No
IPL
Yes
Turn On I/O Interrupt Latch GB = K4, 1 Service Out
Continue IPL Routine
Restore R Register Restore ROAR
Return to Original Microprogram

Figure 4-15. Selector Channel Interrupt Trap Flow Chart

Original Microprogram Resumption

- The backup ROAR enters the ROAR (The GWX registers are transferred to the WX registers).
- The H-register-5 latch turns off.
- The X-6 and X-7 buffer latches condition the X-bus X-6 and X-7 lines (The interrupted microprogram dynamic branching conditions).

To return the CPU to its original microprogram after a selector ROS trap, the selector ROS trap microprogram has as its last instruction an instruction to transfer the contents of the GWX register to the WX register. The last selector trap ROS word also turns off the H-register-5 latch.

The last ROS word of the selector trap microprogram specifies WX=FWX. Because the H-register-5 latch is on, the FWX portion of the instruction is decoded as GWX. The SAL output for this instruction becomes good prior to T1 time of the last word, and gates the GWX register contents to the W and X buses.

At T1 time of the last word, the W-and X-bus conditions set into their respective registers. This addresses the last original program ROS-word addressed before the execution of the selector interrupt ROS trap. Remember that although this ROS word was addressed, its execution was prevented because the control-register did not set. During the same CPU cycle the control-register sets for the last ROS word of the selector ROS trap thereby executing the microprogram instruction.

At T4 of this cycle, the ROAR-restore-buffer latch turns on. This latch gates the X-6 and X-7 buffer latches (holding branch condition) onto the X-bus 6 and 7 lines. The execution of the original microprogram is now at the point at which the selector ROS trap discontinued it.

I/O INTERRUPT EXECUTION

- Store current PSW in I/O old PSW location.
- Store required information in CSW location.
- Load new I/O PSW into local storage, making it the current PSW

The last E-cycle ROS word of each instruction branches the microprogram to the interrupt microprogram routine if an interrupt exists.

The first ROS word of the interrupt routine has the instruction: Test Interrupt with both X6 and X7 branch indicators set to 1. This instruction is a test to determine the source of the interrupt:

1. External or timer interrupts prevent both the X6 and X7 branches (X6=0, X7=0).
2. Selector-channel 1 interrupt prevents the X7 branch (X6 = 1, X7 = 0).
3. Selector-channel 2 interrupt prevents the X6 branch (X6 = 0, X7 = 1).
4. Multiplexor channel allows both X6 and X7 branches (X6 = 1, X7 = 1).

Assuming a selector-channel 1 interrupt, the microprogram branches to ROS word 112. The microprogram turns on the associated gate (SX-1 or SX-2) and reads out the unit address from local storage, places it in the L-register, and sets the U-register to the address of selector-channel 1. After turning on the S4 latch to indicate a selector-channel interrupt, and setting the Q-register to zero to deactivate memory protect, the microprogram stores the current PSW in the I/O old PSW location (main-storage location 56, Hex address 38).

The interrupt code is the first item stored in the I/O old PSW. Because this is an I/O interrupt, the addresses of the channel and I/O unit causing the interrupt, are stored as the interrupt code. The microprogram next reads out the storage protect key and system mask and stores them in their respective positions in the I/O old PSW. The microprogram then takes the current instruction counter and stores it in the low order two bytes of the I/O old PSW.

The instruction length code (ILC), condition code (CC), and program mask are not contained in one byte in the current PSW. Therefore, the microprogram must determine what the ILC is by testing the two high-order bits (0 and 1) of the G-register which contains the command code of the last executed instruction. Both the current condition register, and the program mask are located in local storage (K27CPU). The microprogram also reads these out and decodes the four-position condition register into two bits. After this is accomplished, the microprogram assembles the program mask, condition code, and instruction length code into one byte and stores this byte in the I/O old PSW. The current PSW is now fully stored in the I/O old PSW location. The microprogram now stores the required information in the channel status word (CSW), located at main storage location 64, Hex address 40.

The CSW store portion of the I/O interrupt begins on CAS page Q0. The first ROS word of this phase, turns on the G-register G3 and G7 latches. This makes the I/O interrupt appear as a test I/O instruction. The microprogram determines the difference between a true test I/O instruction and an I/O interrupt by the condition of the S4 latch, which is <u>on</u> for I/O interrupts. After branching on the S4 latch, the microprogram determines which selector channel has initiated the interrupt, and turns on the associated gate (SX1 or SX2). When this is accomplished, the microprogram stores the required information in the CSW:

1. Count from the GCD registers.

2. Memory protect key from the GK register.

3. Unit status from the GR-register if the channel is busy, or from the device if the channel is not busy.

4. Channel status from the GE register.

5. Next CCW address from local storage.

When the CCW is stored, the microprogram resets the channel that initiated the interrupt. The microprogram now starts loading the I/O new PSW into local storage as the current PSW (CAS page D5).

At the same time the I/O new PSW loads into local storage, certain latches associated with the PSW turn on:

1. The mask latches turn on when the system mask reads out of the I/O new PSW and into local storage (K24).

2. The malfunction-suppress latch turns on if the machine-check mask-bit is 0 (off).

3. The ASCII latch turns on if ASCII is designated by the I/O new PSW.

4. The Q-register sets with the information in the PSW protect key area.

The instruction address enters the IJ-registers and does not enter local storage. The two-bit new PSW CC decodes into four bits and enters the condition register (K27 CPU). If, during this load new PSW routine, the microprogram determines that the high order of the instruction address is not 0, it stores this byte at K16 CPU. It also turns on the S-register-two latch if the low-order byte of the instruction address is not an even numbered address (position 7 of the J register is ON).

When this PSW is fully loaded, the microprogram branches to I-cycles, and starts the execution of the I/O interrupt program.

CCW FLAG OPERATIONS

- Five CCW flags modify the CCW operation.

The CCW contains five flag bits. These bits and their locations in the CCW are:

1. bit 32--CD; chain data address.

2. bit 33--CC; chain command.

3. bit 34--SLI; suppress length indication

4. bit 35--SKIP; skip data.

5. bit 36--PCI; program control interrupt.

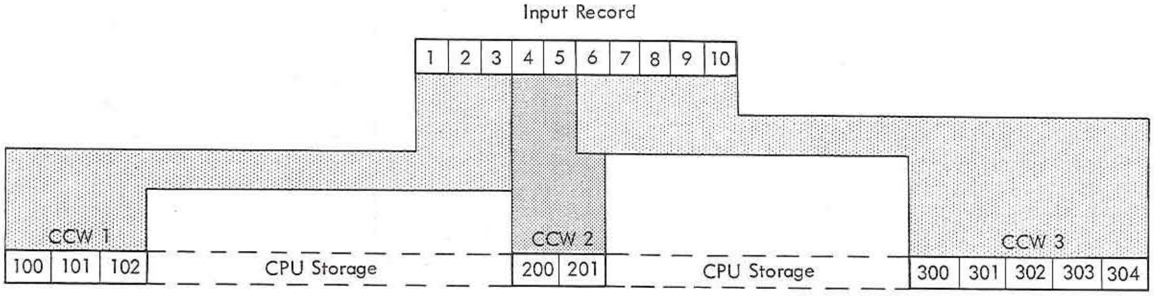
The flag bits function independently of each other except that the CC and SLI flags do not take effect if the CD flag is on.

The time, in relation to the CCW execution, that a flag bit takes effect, varies as follows:

1. The CD bit takes effect when the count reduces to zero.
2. The CC-bit takes effect when a control unit sends ending status and the CD bit is off.
3. The SLI bit takes effect any time during the operation an incorrect length situation arises; that is, the control unit sends status-in and the count is not zero, or it sends an additional service-in after the count has reached zero.
4. The SKIP-bit takes effect during the execution of a CCW specifying input (read, read backward, or sense).
5. The PCI-bit takes effect after its associated CCW is loaded into the selector channel registers, and the system mask allows I/O interrupts.

CD FLAG BIT

- The CD-bit takes effect when the count in the GCD registers reduces to zero.
- It allows the selector channel to fetch the count, data address, and flag byte from the next CCW.
- CD allows the selector channel to gather output information from the storage positions that are not adjacent or to place input information into storage locations that are not adjacent.



CCW 1	Read Operation	Count = 3, Data Address 100
CCW 2		Count = 2, Data Address 200
CCW 3		Count = 5, Data Address 300

Figure 4-16. Data Address Chaining (CD)

When the chain-data flag bit of the CCW is on, it is an instruction to the channel to fetch the count, data address, and flag byte from the next CCW when the count in the count register is reduced to zero. This permits the channel to use core-storage locations that are not adjacent in the execution of a command (two or more chained CCW-s). This linking together of data addresses is called chaining data addresses. Figure 4-16 illustrates the principles of chaining data addresses.

Chaining data addresses is controlled by the CD latch of the GF-register. This latch turns on whenever the CCW loads into the selector channel registers and the CD-bit of the CCW is <u>on</u> (in a logical one state).

During the selector share cycle in which the count becomes zero, the count-ready-zero latch turns on. This latch on, together with the CD latch, generates a selector chain ROS request. This is a request for the CPU to trap into the current microprogram routine and execute the selector chain microprogram routine.

The selector chain microprogram routine stores the contents of the R-, S-, V-, and U-registers into local storage and reads out the next CCW address from local storage. It then reads out the next CCW from storage and loads the count into the count registers (GCD), the CCW flags into the selector-channel flag register (GF), and loads the CCW data address into the selector channel data address registers (GUV). It also determines the next CCW address and places it in local storage. After this new CCW information is loaded into the selector channel registers, the microprogram restores the R-, S-, V- and U-registers to their original state, and the microprogram continues at the point where it was discontinued by the selector chain ROS request.

A program check occurs if any part of the CCW (command code, data address, flag, or count) is invalid.

Circuit Objectives

Assume the CD-latch in the GF-register turned on when the CCW was entered into selector channel registers. This specifies that data chaining will take place when the CCW count goes to zero, if no channel checks have been detected.

1. Recognize a count of zero.

   a. Count reduces to 0 on selector share read cycle.

   b. Count-ready-not-zero latch resets

      (1) Sel T4 time

      (2) SX1 read cycle

      (3) GHY & GHZ = Zero (modifier output)

   c. Count-ready-zero latch turns on

      (1) SX1 write cycle

      (2) Selector T4 time

      (3) Count-ready-not-zero latch off

      (4) Not status-stop condition

2. Request a selector ROS trap.

   a. Generate Selector ROS REQUEST

      (1) Generate CD chain request.

         (a) CD-latch on

         (b) Count-ready-not-zero latch on

         (c) GR-full latch off

      (2) Generate SX1 chain ROS Request

         (a) CD chain request

      (3) Generate selector ROS request

         (a) SX1 chain ROS Request

3. Execute Selector chain microprogram.

   a. Turn on H-register 5 latch.

   b. Store contents of CPU registers R, S, U, and V.

      k20 CPU

      (1) Store R-register in local storage

      (2) Turn off latches that were on due to previous operations and turn on poll control

         (a) GH=K13 (chain-reset)

      (3) Store S-register in local storage

         (a) K29 CPU

         (b) R=S

      (4) Store V-register in local storage

         (a) K31 CPU

         (b) R=V

      (5) Store U-register in local storage

         (a) K30 CPU

         (b) R=U

   c. Update next-CCW-address.

      (1) Read out low-order of next CCW-address
         (a) K7 CPU

      (2) Increment it by 8 and replace in K7 of local storage. Load low order address in V-register.

         (a) RC=R+K8L

      (3) Read out high order of next CCW-address
         (a) K6 CPU

      (4) Update high order of next CCW address and replace in K6 of local storage. Load high order address in U-register.

         (a) RC=R+C (this adds 1 if an address carry occurred when the low order of the next CCW address was updated)

d. Fetch next CCW.

(1) Read out op byte and check for a transfer-in-channel (TIC). Assume it is not

(2) Read out flag byte

(3) Place flag byte in GF register

(a) GF=GR

(4) Read out low-order count

(5) Place low-order count in GV-register.

(a) GV=GR

(6) Read out high-order count

(7) Place high-order count in GU register.

(a) GU=GR

(8) Place count in GCD registers

(a) GCD=GUV

(9) Read out low order of CCW data address into the GV-register

(a) GV=GR

(10)Read out high-order of CCW data address into the GU-register

(a) GU=GR

(11)Turn on count ready and not zero latch

(a) GB=K8, 1

e. Restore CPU registers.

(1) Restore U-register

(a) K30 CPU

(b) U=R

(2) Restore V-register

(a) K31 CPU

(b) V=R

(3) Restore S-register

(a) K29CPU

(b) S=R

(4) Restore R-register

(a) K20 CPU

f. Restart microprogram at the point where it was discontinued.

(1) Restore ROAR

(a) WX=FWX

(2) Turn H5 off

(a) H=H-K4L
Note: The FWX part of the instruction is decoded as GWX when the H-register-5 latch is on.

COMMAND CHAINING (CHANNEL-END AND DEVICE-END)

- The CC flag bit allows the selector channel, control unit, and I/O device to execute several CCW's with one start I/O instruction. The CCW command byte may be different than the one in the first CCW.
- When the control unit presents device-end status, the channel requests ROS control.
- The selector channel ROS request breaks into the current microprogram routine after any pending write cycles.
- After storing CPU register information, the selector channel reads out a new CCW, and executes a unit selection sequence.
- After completion of the load CCW and unit selection microprogram routine, the microprogram restores the contents of the CPU-registers and restarts the current microprogram where it was discontinued.
- An I/O interrupt is not taken until the last CCW has been completed, unless a channel check or device check is detected. In this case, chaining is suppressed and an interrupt is taken.

When the CC-bit of a CCW is on, and the CD bit is off, the channel executes the CCW and fetches the next CCW. This is accomplished under microprogram control.

The channel requests ROS control after it has received channel-end and device-end status from the control unit and the device that executed the CCW. This request for ROS control enters the stacking latches at T3 time (When the CPU cycles can be interrupted), that is if no write cycles are pending. The stacking latches determine the priority of ROS requests.

If no higher priority stacking latches turn on when the selector ROS request stacking latch turns on, the CPU stores the read-only-address-register (ROAR) in the backup ROAR and signals the channel that it will honor its request for ROS control. When the channel receives this signal, it provides a signal to the CPU that loads the ROAR with the address of the first word of the selector channel chaining microprogram routine.

The first word of the selector-channel chaining microprogram routine, turns on the H-register-5 latch. This latch prevents the CPU from honoring a ROS-control request from the other selector channel or the multiplexor channel until it turns off. It also turns on the decode gate (SX1 or SX2) for the channel that requested ROS control (selector-channel 1 or 2). After a chain-reset, the microprogram stores the contents of the R-,S-, V-, and U-registers in local storage. It also reads out the next CCW address from local storage, updates it by eight or sixteen, and places the updated next CCW address in local storage. The microprogram now starts a unit selection, then loads the selector channel registers with the CCW information and finishes the unit selection. The microprogram now checks for another selector channel chain-request. If a request is waiting, the chain routine is entered without restoring the CPU registers.

When this is complete, the microprogram branches to a routine that restores the U-, V-, S-, and R-registers to their original state. It also restores the ROAR with the address in the backup ROAR, and turns off the H-register-5 latch.

The microprogram now continues at the point where it was discontinued by the selector channel ROS request.

Circuit Objectives

1. Control unit presents ending-status to the channel.
   a. Status-in
   b. Bus-in 4-bit (Channel-end)
   c. Bus-in 5-bit (Device-end)
2. Request read only storage (ROS) control.
   a. Generate CC gate
      (1) CC latch of GF-register on
      (2) Status-in delayed

(3) Not interrupt condition

(4) CD latch of GF-register off

(5) Halt I/O stop latch off

b. Turn on suppress-out latch (raise suppress out).

(1) CC chain request

<u>Note</u>: This is the channel's indication to the selected control unit that it is chaining.

c. Generate CC-chain request

(1) CC-gate

(2) Bus-in 5-bit (device-end)

(3) Not service-out

d. Turn off select-out latch

(1) CC chain request

e. Turn on poll control latch

(1) GH=K13 (chain-reset)

f. Generate SX1 chain ROS request

(1) Chain request

(1) SX1-chain-ROS-request

3. Address first word of selector chain microprogram routine

a. Turn on selector chain request stack latch

(1) Selector ROS request

(2) H-register-5 latch off

(3) Allow write latch off (No writes pending).

(4) CPU T3 time

b. Generate Selector chain pulse (T3 time)

(1) Selector-chain-request stack latch on

(2) No higher priority stack latch on

(3)H-register-5 latch off

c. Generate any-priority pulse at T3 time

(1) Selector-chain-request stack latch on

d. Force X-bus 4 line

(1) Selector chain pulse

(2) Any-priority pulse

e. WX-register (ROAR) contents in GWX register (Back up ROAR) and X6, X7 branches enter buffer latches.

(1) Selector chain pulse

(2) CPU T4 time

f. Turn on Selector CC ROS request latch.

(1) SX-1 chain ROS request, Not SX-1 interrupt ROS request, and Not SX-2 interrupt ROS request.

(2) SX-1 chain ROS request, Selector chain pulse, and CPU T4 time.

g. Force X-bus 6 and 7 lines

(1) Selector CC ROS request latch on

(2) Selector (SX)-chain-pulse

(3) Any-priority pulse

h. Generate X-register set

(1) Not inhibit ROAR set

(2) CPU T1 time

i. Set X-register with address of first word of the selector chain microprogram routine.

(1) X-bus

(2) X-register set

j. Turn on selector 1 trap latch

(1) Selector chain pulse

(2) Selector-CC-ROS-request latch on

(3) SX1 chain ROS request

(4) CPU T2 time

k. Turn off Selector-CC ROS request latch

(1) Not SX-1 chain ROS request

l. Turn on priority latch

(1) Any-priority latch on (selector chain latch is on).

(2) T3 time

m. Decondition selector chain pulse.

(1) Priority latch on

4. Execute selector chain microprogram routine.

a. The channel trap word 00B is now present in the ROS SALs and control register. This word issues an instruction to turn on the H-register-5 bit. H = H$K4L

(1) Turn on H-register-5 bit

(a) Z-bus 5 bit

(b) Microprogram CD field combination in control register 0101 (Hex 5)

(c) CPU T4 time

b. Generate SX1 gate

(1) Selector 1 trap latch on

(2)H-register 5 latch on

c. Store CPU registers R,S,U, and V in local storage.

(1) Store R-register

K20 CPU

(2) Place status in GR register

GR=GI

(3) Store S-register
K29 CPU

(4) Turn off latches which may be on due to previous CCW
KH=K13

(5) Turn on service-out latch

GA=K1

(6) Store V-register

K31 CPU

(7) Store U-register

K30 CPU

d. Update next CCW address

(1) Read out low order of next CCW address.

K7 CPU

(2) Add 8 or 16 to low-order of next CCW address and replace in K7 CPU

(3) Read out high-order of next CCW address.

(4) Add 1 to high-order of next CCW address if low-order forced an address carry and replace in K6 CPU

(5) Check that the control unit dropped off the interface

k. Fetch next CCW ; select unit. Some of this is the same routine that loaded the first CCW into the selector channel registers and initially selected a control unit and I/O device. The unit selection is started before the next CCW is fetched.

Note: The suppress-out latch turns off during this routine.

f. Restore CPU.

(1) Restore U-register.

K30 CPU
U = R

(2) Restore V-register.

K31 CPU
V = R

(3) Restore S-register.

K29 CPU
S = R

(4) Restore R-register.

K20 CPU
Write

g. Continue microprogram at the point it was discontinued.

(1) Turn off H-register-5 latch.

H = H - K4L

(2) Place back-up ROAR (GWX) in ROAR.

WX = FWX

COMMAND CHAINING (CHANNEL-END AND DEVICE-END SEPARATELY)

- Channel responds with suppress-out, then after a delay of 375 nanoseconds, service out.
- When device-end status arrives at the channel, it initiates a selector-channel ROS request.
- Select-out remains active until reset by the selector-chain microprogram.

When the GF-register command chaining latch is on and a control unit presents channel-end to the channel, the channel responds with suppress-out and after a 375 nanosecond delay, service-out. The channel does not request ROS control at this time and select-out remains active. The suppress-out latch being on at this time indicates to the control unit that the channel is chaining commands.

When the I/O device completes the execution of the CCW, the control unit presents device-end status to the channel. When the channel receives the device-end status it generates CC chain-request. From this point, the operation is the same as for selector channel chaining, when the channel-end and device-end are presented together to the channel.

SLI (SUPPRESS LENGTH INDICATION) FLAG BIT

- Prevents indication of an incorrect length condition.
- Effective only when CD flag is off.

The SLI flag bit (bit-34 of the CCW), when on, prevents the turn-on of the incorrect-length latch. This bit is effective only if the CD flag bit is off in the same CCW. The SLI bit is used in all CCWs except those in which the count agrees exactly with the number of service-in signals expected from the control unit for that operation.

Length check must always be suppressed for devices in which block length is not defined. For example, when writing on magnetic tape, the amount of data written is controlled only by the count in the CCW. Every operation terminated under count control causes an incorrect length indication, unless suppressed by the SLI flag.

SKIP FLAG BIT

- Suppresses transfer of information to storage.
- Takes effect during execution of the CCW.
- Used for Read, Read Backward, or Sense Operations.

The SKIP flag bit is turned on to suppress the transfer of information to CPU storage during read, read backward, or sense operations. This flag bit is effective only for the CCW having the bit on. In this way, data can be selectively stored in the CPU.

The SKIP flag affects only the handling of information by the channel. The operation of the control unit and device proceeds normally and all data is transmitted to the channel. The channel updates the count but does not place the data into main storage.

In the case of command chaining or data-address chaining, normal operation resumes if the skip-flag in the new CCW is not on.

No checking for invalid or protected data addresses takes place during skipping, except that the initial data address in the CCW cannot exceed the largest address of the system.

PROGRAM CONTROLLED INTERRUPT (PCI)

- The PCI-bit in a CCW causes an I/O interrupt during the execution of an I/O operation.
- The PCI-bit can be in the first, or any other CCW in a chained routine.
- The program-controlled interrupt sets channel-status bits in the CSW.

The PCI flag bit, when on, causes an I/O interrupt at some time during the execution of an I/O operation (if the I/O interrupt is not masked off). This flag bit can be in the first or any other CCW in a chain of CCW's. When the PCI-bit is detected, the channel attempts an interrupt as soon as possible after fetching the CCW by bringing up the selector channel interrupt line to the CPU. This indicates the normal interrupt procedure as described under <u>I/O Interrupts.</u>

When the interrupt occurs, a new CSW is stored, and the PCI-bit (GE bit 0) and any other existing channel-status bits are set in the CSW. The presence of the PCI-bit in the CSW reflects the progress of the operation at the time the CSW is stored. In this way, the status of the channel can be determined while a chained operation is being executed. The PCI-bit has no effect on the actual execution of the operation.

When the PCI-bit is presented by an interrupt before the operation is completed, the channel-end and device-end bits in the CSW will be off. If the channel detects data errors in the operation, the channel-data-check bit is on and is set into channel status when the CSW is stored. The condition is not reset, however, and is indicated again at the end of the operation.

Presence of the channel-end and device-end bits with the PCI-bit indicates that the operation is ended. The CSW in this case has its regular format with the PCI-bit added.

If chaining occurs before the PCI interrupt takes place, the PCI indication is carried over to the new CCW. This occurs on both data-address and command chaining, and in either case the condition is propagated through a transfer-in-channel command.

PCI conditions are not accumulated. If a CCW containing the PCI-flag is fetched, before the interrupt because of a previous PCI-flag takes place, only one interrupt is taken.

The status of the PCI flag-bit is inspected in all CCW's except those specifying a transfer-in-channel. In a CCW specifying transfer-in-channel, the setting of the PCI flag is ignored.

TRANSFER IN CHANNEL (TIC)

- Causes the channel to fetch the next CCW from the location specified by the data-address field of the CCW.
- Provides chaining between nonadjacent CCW's.
- Can occur in both data-chaining and command chaining.

When a CCW containing the TIC reads out, and the TIC-command is recognized (Figure 4-17) it causes the channel to read out a new CCW from the location specified by the data-address field and perform the operation in the new CCW.

The TIC-command does not initiate an I/O operation at the channel or the I/O device. The transfer-in-channel command provides chaining between nonadjacent CCW's.

The first CCW in a chain of CCW's cannot specify transfer-in-channel. When this condition is detected during the execution of the start-I/O instruction, no I/O operation is initiated and a CSW is stored with a program-check indication. Similarly, a CCW specifying a TIC cannot be fetched from a location specified by a preceding transfer-in-channel. When two successive transfer-in-channel commands are detected, a program-check is again indicated. If this condition is detected during data-chaining, the I/O device is signalled to end the operation.

To address a CCW on integral boundaries for double words, a CCW specifying a transfer-in-channel must contain zeros in bit positions 29, 30, and 31. When this restriction is violated, or if an invalid address is encountered, a program check is indicated. Either of these errors, detected during data-chaining, end the operation at the device. During command chaining, they cause an I/O interrupt.

In a CCW containing a TIC, bit positions 0 to 3 and 32 through 63 are ignored.

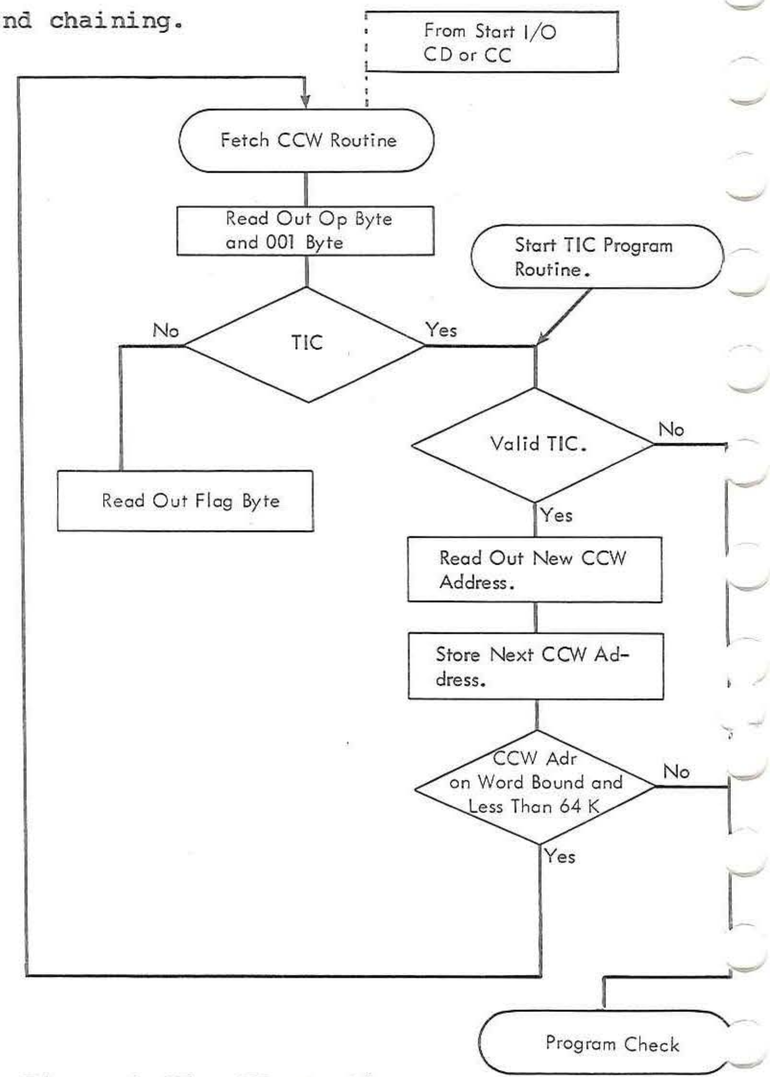


Figure 4-17. TIC Routine

TEST I/O

- A programmed request for channel and control unit status.
- Requires an initial selection sequence only.
- Control unit status is checked and a condition code is set in the PSW. In some cases the CSW is also stored.

During I-cycles, the CPU recognizes the instruction as a test I/O and performs three functions:

1. Stores the control unit address in the V-register.
2. Stores the channel address in the U-register.
3. Branches to the I/O microprogram routine.

The address in the U-register determines to which selector-channel the test-I/O command is directed. Assume selector-channel 1 is specified. SX1 gate is turned on by the statement GB=K1,0. The microprogram also tests the GT lines to determine if the addressed channel is busy (Figure 4-18). If the channel is busy and the interrupt latch is off, the program sets the current PSW condition code to two and returns to I-cycles.

With the interrupt latch on, however, the control-unit address is read out and compared to the address of the functioning control unit. An address mis-match causes condition code two to set in the current PSW, and the program returns to I-cycles. This indicates that the channel is busy with another control unit. Therefore, the device associated with the test I/O cannot be selected on the interface

If the addresses match, the microprogram branches to the CSW store routine and sets condition code one.

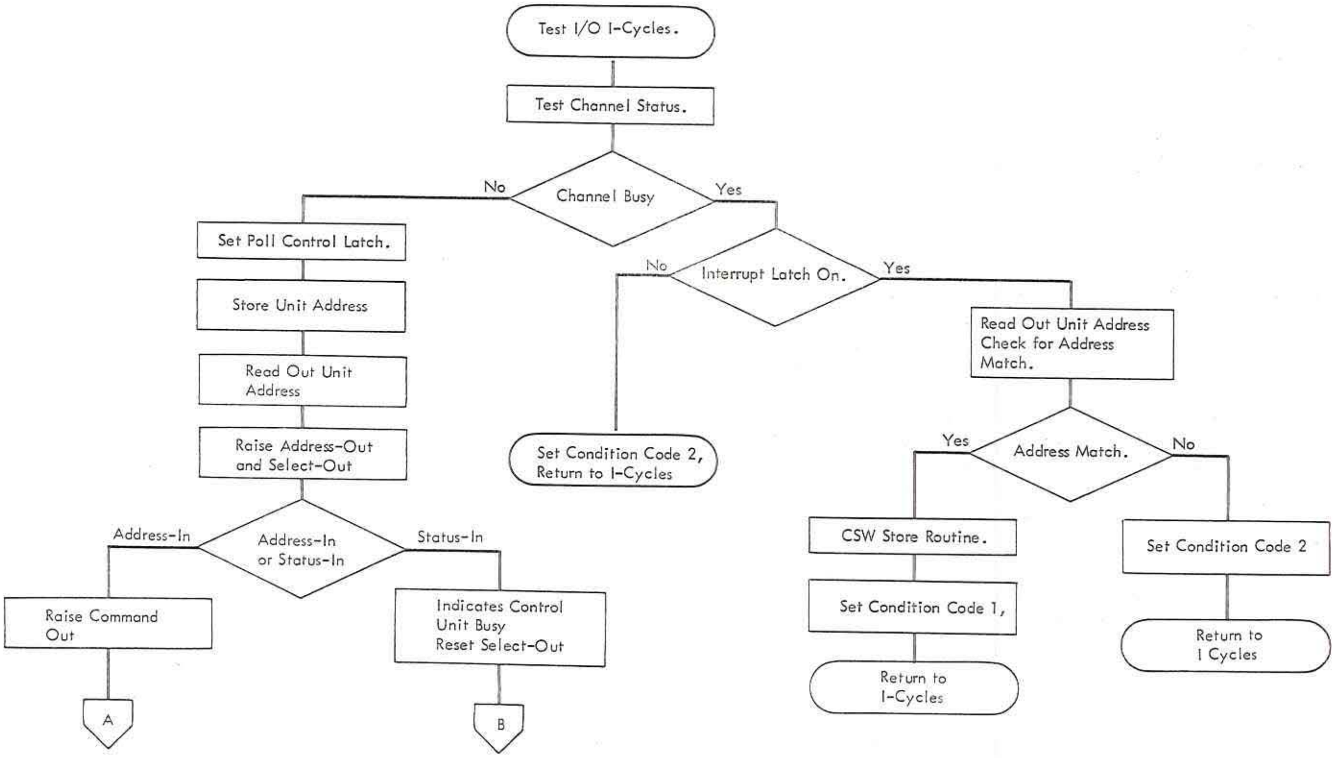


Figure 4-18. Test I/O (Part 1 of 2)

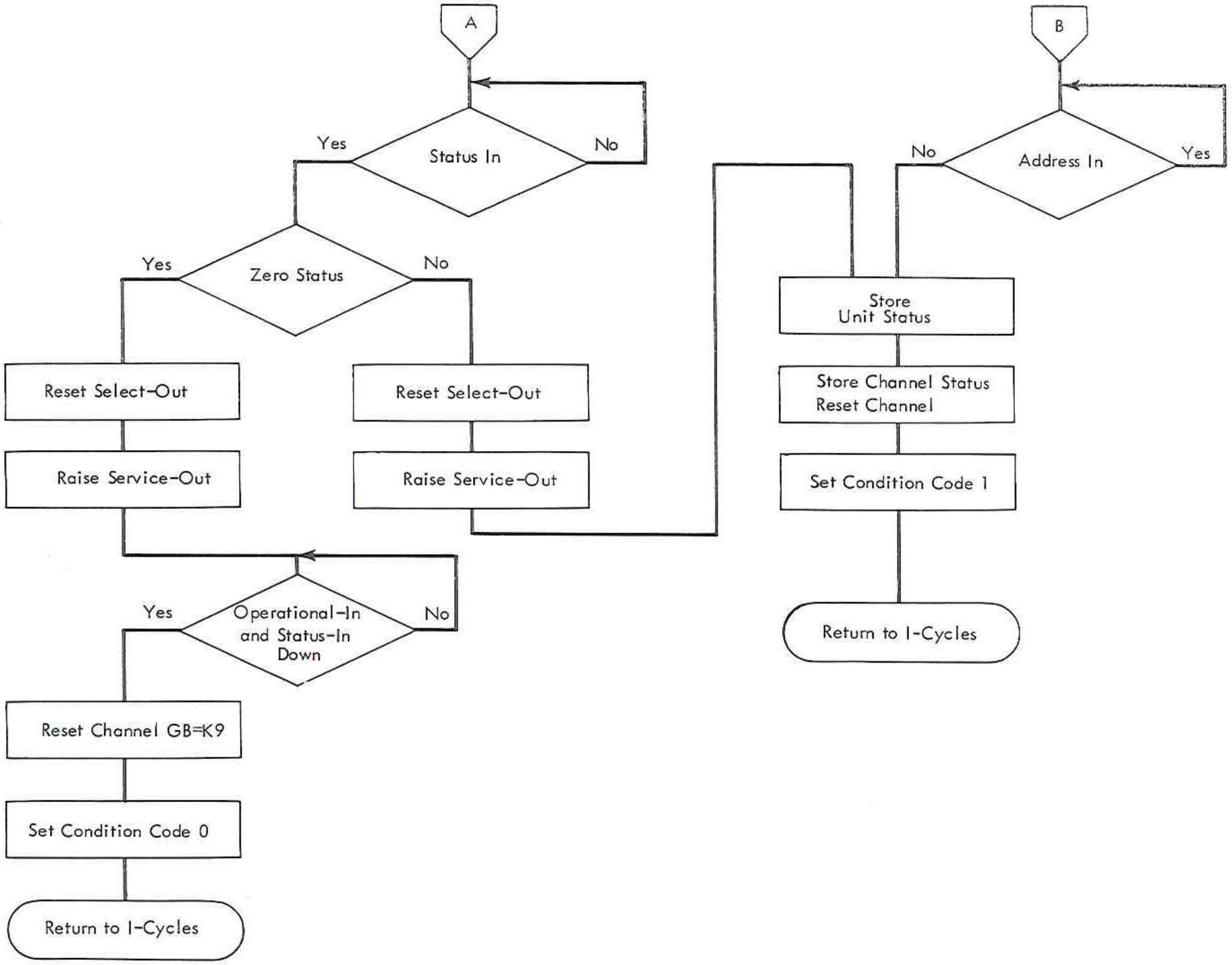


Figure 4-18. Test I/O (Part 2 of 2)

If the channel is not busy when the GT lines are tested, the microprogram sets the poll-control latch to capture polling. The microprogram then stores the control unit address in local storage position K5 and does a channel reset excluding poll control. It then reads out the address to the GR-register, raises the address-out and select-out out tag lines, and waits for a reply from the control unit. If the reply from the control unit is status-in (indicating control unit busy), the microprogram branches to a store-status routine. With the address-in in reply (addresses match), the channel raises the command-out line and sends the test-I/O command to the control unit (no bits on bus-out).

The control unit replies to command-out with the status-in tag line and places any existing status bits on the bus-in lines. The channel resets select-out, responds to service-out, and checks the incoming status for the presence of status bits.

If the status byte on bus-in contained no bits, the channel:

1. Sets condition zero in the PSW.
2. Returns to I-cycles.

If the status byte contained any status bits, the channel:

1. Stores the control unit status in the CSW.
2. Zeros the CSW.
3. Stores the channel status in the CSW.
4. Resets the channel.
5. Sets condition code 1 in the PSW.
6. Returns to I-cycles.

For more detailed information, and circuit objectives, refer to Start I/O sections.

HALT I/O

- Executed when address-out is up, select-out is down, and operational-in up.
- Causes the I/O device operating with the channel to stop data transfers and drop operational-in.
- Causes an addressed I/O device to stop processing data and reset.

The halt I/O instruction is issued by the channel to cause:

1. The control unit currently on the channel to disconnect from the interface.
2. A control unit that is not connected to the interface to stop.

Execution of the halt-I/O instruction, by the channel, is through a sequence of signals rather than a command sent to the I/O device. The microprogram routine used depends on the channel-busy status, therefore, each is discussed separately.

On entering the halt-I/O microprogram routine, the channel is tested for busy status (Figure 4-19). If the channel is busy (an I/O device is on the interface with op-in up) the halt I/O latch is turned on. Following this, the select-out line is reset, and address-out is raised.

The operating I/O device receiving this signal sequence must drop its operational-in line, disconnect from the interface and reset. With the fall of the op-in line, the microprogram then turns on the SX1 interrupt latch requesting an interrupt, sets condition code two in the PSW, and returns to I-cycles.

If the channel was not busy when tested, the halt-I/O instruction follows a different routine. In this case, the I/O device in question is not on the interface and does not have its operational-in line up. However, the device must be reset. The microprogram routine followed is the same as for the start-I/O routine and a normal initial-selection sequence takes place. Once the device is selected, the halt-I/O latch turns on and the same signal sequence is used causing the selected device to drop op-in, disconnect, and reset. Condition code one is set , and the CSW status bytes are set to zero.

Refer to the Start-I/O section for more detailed information and circuit objectives.

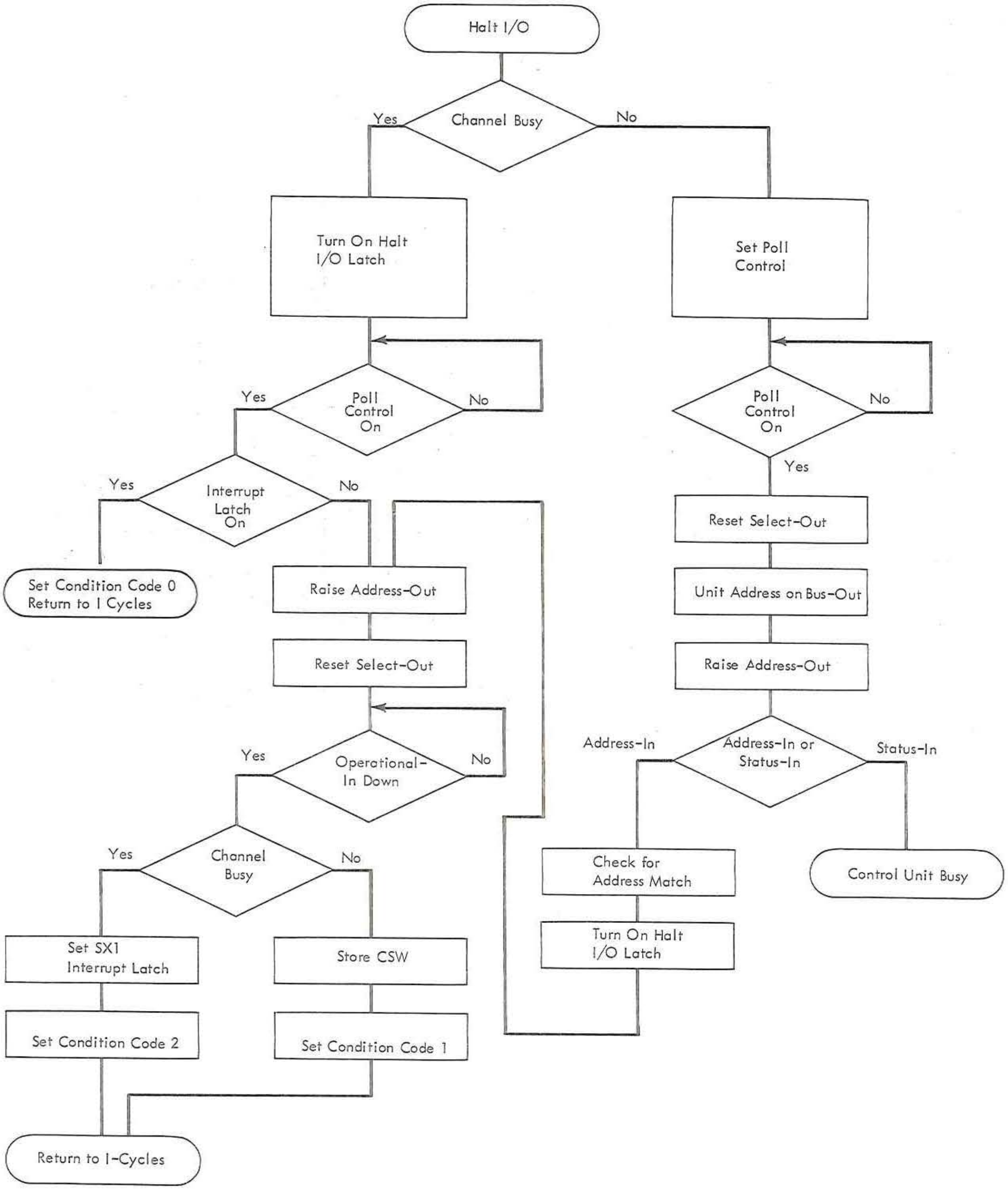


Figure 4-19. Halt I/O

TEST CHANNEL

- Executed to determine the status of a particular channel.
- Sets a condition-code in the PSW.
- The condition-code set indicates the status of the addressed channel.

The test-channel command determines the status of a particular channel. It is executed only when the CPU is in the monitor state and has no effect on the channel. The test-channel command sets a condition-code in the PSW that indicates the current state of the addressed channel.

The test-channel operation begins as a normal start-I/O; however, when the addressed channel is selected, a microprogram branch occurs to the test-channel routine. In this routine (Figure 4-20), the channel is checked for a busy status or an interrupt condition and a condition code is set.

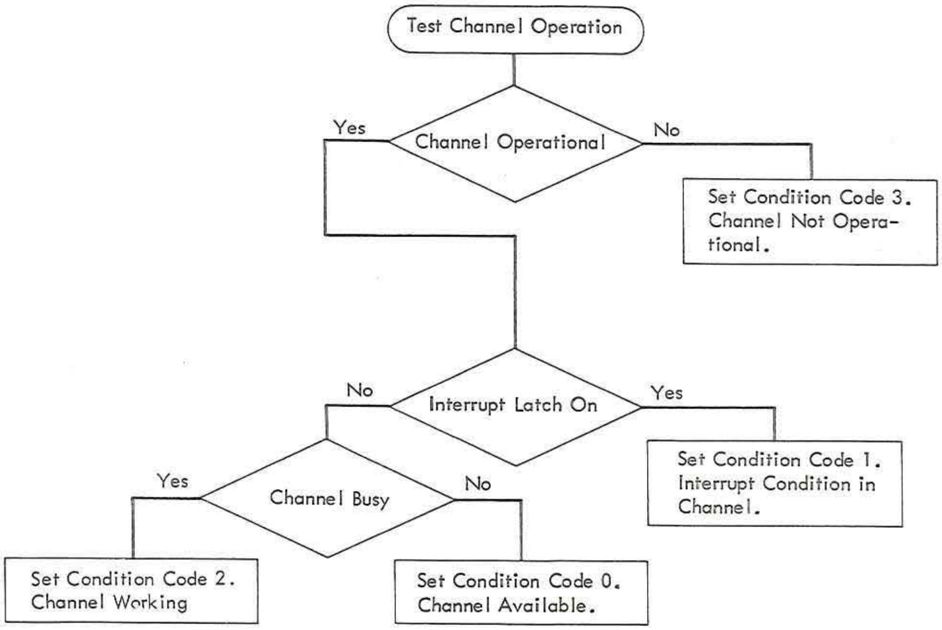
Condition codes for the test-channel instruction are:

Code 0 - Channel available.

Code 1 - Interrupt condition in channel.

Code 2 - Channel working.

Code 3 - Channel not operational.


Test Channel Operation
Channel Operational
Yes
No
Set Condition Code 3. Channel Not Operational.
Interrupt Latch On
No
Yes
Set Condition Code 1. Interrupt Condition in Channel.
Channel Busy
Yes
No
Set Condition Code 2. Channel Working
Set Condition Code 0. Channel Available.

●Figure 4-20. Test Channel

INITIAL PROGRAM LOAD (IPL)

- Initiates processing after power is turned on or a new program is desired in storage.
- Started by selecting an input device and pushing the Load button.

The IPL procedure is provided to initiate processing after a power-on condition, or when the contents of storage is not suitable for further processing.

The IPL procedure resembles a start-I/O 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, the command-chain flag on, the SLI flag on, and a command address of zero. This CCW is forced by microprogram.

The IPL reads new information into the first six words of storage. The remainder of the IPL program can be placed in any desired location of storage.

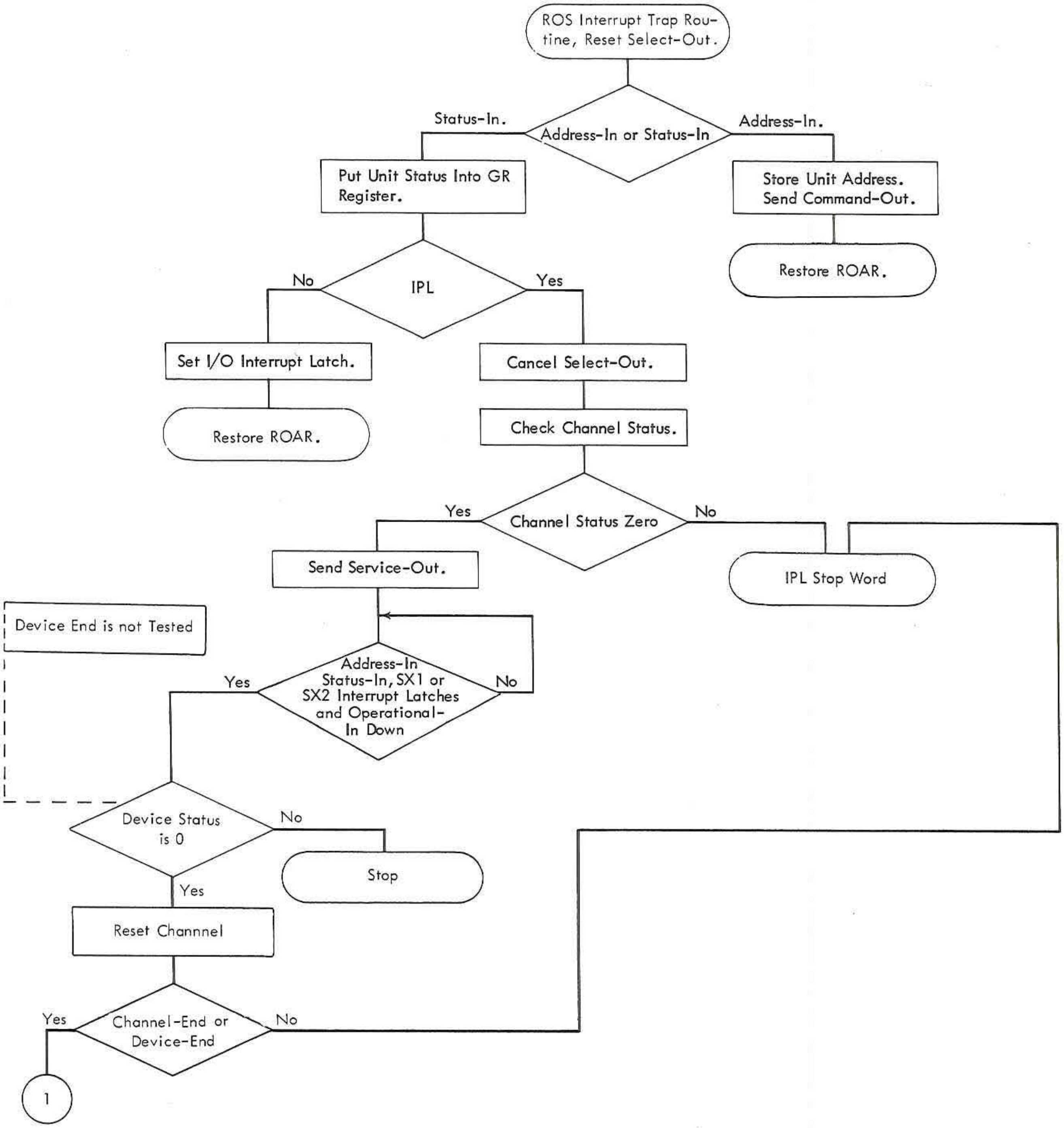
The IPL procedure is started by selecting an input device with the load-unit switches and pushing the Load button. This causes a system reset, turns on the Load light, and initiates a read operation from the selected input device. When command chaining is completed, the CPU starts operating and the Load light turns off.

The system reset stops the CPU and suspends all instruction processing, interruptions, and timer updating. It also resets all channels and on-line control units and I/O devices. The contents of the general and floating-point registers normally remain unchanged.

After system reset, the selected input device starts reading. The first 24 bytes read are placed in storage locations 0 to 23 (IPL PSW, IPL CCW 1, IPL CCW 2). Storage-protection and possible incorrect-length errors are ignored. The double-word read into location eight is used as the CCW for the next input command. When chaining is specified in this CCW, the operation proceeds with the CCW in location 16.

After the input operation is completed, the channel and device address is stored in bits 21 to 31 of the first word in storage. Bits 16 to 20 are zeros and bits 0 to 15 remain unchanged. The CPU then fetches the double-word in location zero as a new PSW and changes from the stopped to the operating state. Operation proceeds under control of the new PSW. If the I/O operations and PSW loading are not satisfactory, the CPU stops and the Load light remains on.

The microprogram for the IPL routine is shown in Figure 4-21. The start of the IPL routine is similar to the normal start-I/O routine and is not duplicated on this chart. During the ending procedure, the ROS interrupt trap microprogram branches out into the IPL program to check the unit and channel status prior to the execution of the Load-PSW routine.



● Figure 4-21. Initial Program Load (Part 1 of 2)

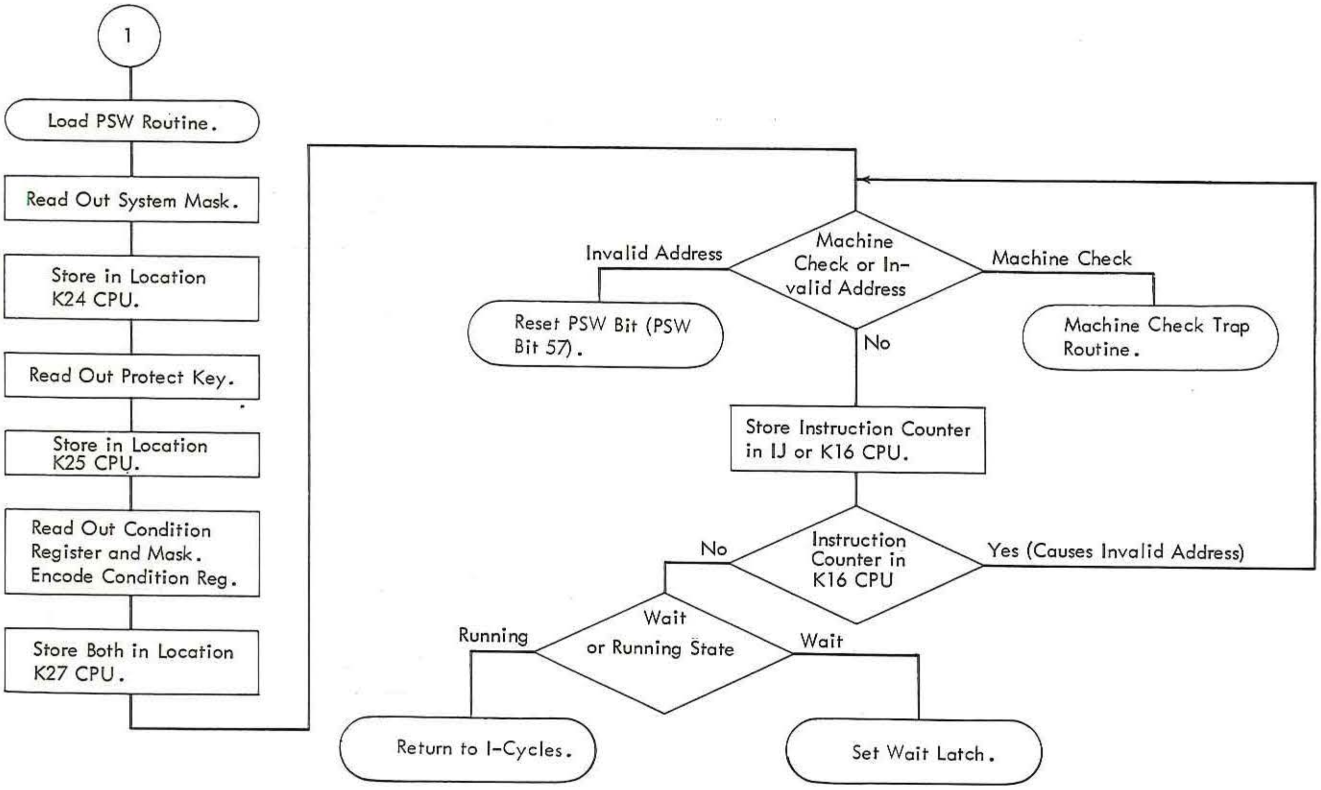

1
Load PSW Routine.
Read Out System Mask.
Store in Location K24 CPU.
Read Out Protect Key.
Store in Location K25 CPU.
Read Out Condition Register and Mask. Encode Condition Reg.
Store Both in Location K27 CPU.
Machine Check or In- valid Address
Invalid Address
Machine Check
Reset PSW Bit (PSW Bit 57).
Machine Check Trap Routine.
No
Store Instruction Counter in IJ or K16 CPU.
Instruction Counter in K16 CPU
No
Yes (Causes Invalid Address)
Wait or Running State
Running
Wait
Return to I-Cycles.
Set Wait Latch.

Figure 4-21. Initial Program Load (Part 2 of 2)

ERROR DETECTION

- During channel operation, the interface, the channel, and the CPU are checked for errors.
- Detected errors set channel status bits in the CSW.
- The operation is either suppressed, halted, or completed depending on the type and location of the error.

During normal selector-channel operation, the channel is connected on one side to the processor and on the other to an I/O control unit or device. It must then be able to detect errors that occur in the channel itself, on the interface, or in the CPU.

All errors detected by the channel during operation with the interface or the CPU set channel-status bits in the CSW when the CSW is stored. The action taken by the channel when an error is detected depends on the operation being performed and the type of error encountered.

The chart in Figure 4-22 lists the errors that are detectable by the channel and their causes.

Error Indication	Cause of Error	Reason	Action Taken
Incorrect Length	1. Long record on Input or Output. 2. Short record on Input or Output.	SLI flag off and CCW count reduced to zero and service-in is received. SLI flag off and I/O device ends the operation before CCW count reaches zero or a halt I/O is signaled by the channel.	Sets channel status bit 1. Ends operation on detection.
Program Check	1. Memory Wrap (Invalid data address) 2. Invalid Count. 3. Invalid Memory Protect Key 4. Invalid Format. 5. Invalid Command. 6. Invalid CCW Address.	Overflow from modifier, or M register contains address greater than memory size during a share cycle. Not a TIC and CCW count is zero. Key bit positions not zero in unprotected machines. Three low order flag bits in CCW not zero. Command not acceptable to I/O device or channel. CCW address not on valid boundaries.	Ends operation on detection and sets bit 2 in channel status.
Protection Check	1. Protect key mismatch.	Memory protect keys do not match and protect key is not zero. (An attempt to place data into a protected location).	Ends operation on detection; Sets channel status bit 3.
Channel Data Check	1. GR parity check on data (SX1	Parity error on data byte in GR register from interface or storage.	Sets channel status bit 4, suppresses chaining and ends operation when current CCW is finished.
Channel Control Check	1. MN register parity (during share cycle). 2. GHYZ Parity (During share cycle for other than data). 3. GR register parity (CPU store cycle or BUS IN to GR). 4. A-register parity (GR or valid GJ bytes to A reg). 5. Any machine check (during SEL ROS Request Mode). 6. Storage key parity (during input share cycle).	Parity errors on CCW or data addresses, or contents of CCW. Parity errors detected at modifier output. Parity errors detected at GR register output when handling CCW, CSW, chaining, etc. Parity errors detected at A register (Under ROS control). CPU hardware checks (usually detected as a parity error), while Selector Channel is using ROS (H5 on). Memory Protect stack parity check (Q-Lo Register) during selection input share cycles when protection key is not zero.	Ends operation on detection and sets bit 5 in channel status.
Interface Control Check	1. Bus-in parity check (Address or Status) 2. Address mismatch 3. Time out 4. No response 5. Unit busy on chaining.	Parity error detected on unit address or status byte transfer from bus-in to GR. Address from device and address sent on bus-out do not match. Response to an outbound tag not received within time limit. Select-in is received in response to address-out and select-out. Control Unit responds with the busy bit during status-in.	Sets bit 6 in channel status and ends the operation on detection.

Figure 4-22. Selector Channel Errors

MEMORY PROTECT

- Provides protection for the contents of specified areas of storage.
- Achieved by identifying blocks of storage with a storage key, and comparing this key against the protect key.
- Detection of a mismatch results in a protection interruption.

The memory-protect feature is provided to protect the contents of certain areas of storage from destruction due to the storing of erroneous information while executing a program. To achieve protection, blocks of storage are assigned a storage key which is compared against the protect key. For I/O operation, the protect key supplied in the CAW is used. Other operations affecting memory use the protect key supplied in the PSW. If the keys do not match during an input share cycle, a protection-check is set and the operation is terminated.

In the selector channels, the memory protect key is obtained from bits 0 to 3 in the CAW, and is carried in the GK-register when the I/O operation is started. It is stored in bits 0 to 3 of the CSW when an I/O interrupt occurs.

A protection mismatch during channel operation terminates the operation in such a way as to leave the protected storage location unchanged. The mismatch is indicated in the CSW which is stored as a result of the operation. This can only occur on an input operation.

When the memory-protect feature is not installed on the system, the protection key in the CAW and in the PSW must be zeros or a program check error is generated.

COMPREHENSIVE INTRODUCTION

- The 1050 console attachment will provide a basic keyboard and printer for the operator-to-program communications.
- The system (Figure 5-1) consists of:

1. 1051 Console Attachment (located in 2030).
2. 2030 to 1051 Interface (connecting cable).
3. CPU Attachment (in 1051).
4. 1050 System (1051 and associated I/O equipment).

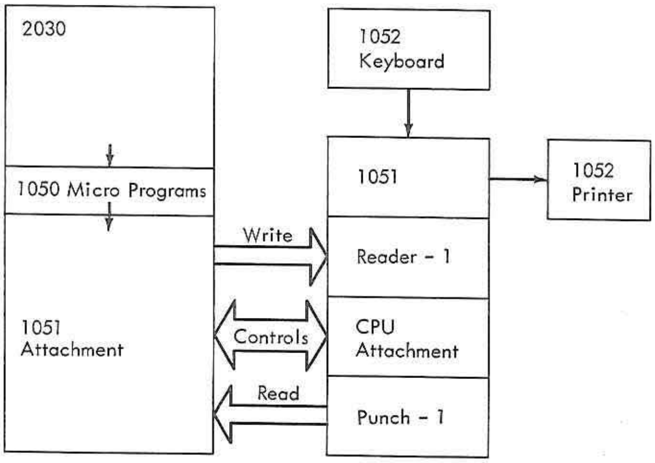


Figure 5-1. 2030-1050 Console System Units

An important link in any type of computer operation is communications between the program and the operator. The 1050 with the 2030 provides this link. Some uses of this communications link are:

1. Error message printout.
2. Instructions to the operator.
3. Program-controlled changes to data or program.
4. Job logging.

Input to the 2030 can be from a keyboard (1052) and output at the 1050 can be a printer (1052). Thus, with a 1051 and a 1052, the operator has a device that provides the equipment needed to communicate with the 2030.

OPERATION

- Operation of the 1050 is similar to a device on the multiplexor (MPX) channel.
- Operation to the 2030 is through the native interface.
- The 1050 home loop is used to send or receive data to or from the 2030.

The 1050 is programmed and operates similar to an equivalent device on the multiplexor channel. Some controls used in the MPX channel are used when operating the 1050. These controls interlock the MPX channel and alter the CPU to channel operation. The microprogram for the 1050 is similar to the MPX channel program in theory. However, because the actual hardware used in the attachment differs from the channel, a unique microprogram is used to handle data flow.

The attachment of the 1050 to the CPU is on the native interface. This term means that all the controls and connecting lines to adapt the 1050 are located in the 2030, or are native to it.

When the 1050 system is attached to the 2030 as a console, the 1050 must have the

CPU Attachment special feature. This attachment allows control of the 1050 home-loop for data handling to and from the 1050. It also makes use of the reader-1 and punch-1 interfaces in the 1051. Reader-1 is used to receive data from the CPU and punch-1 is used to send data to the CPU.

PTT CODE/8 (PERFORATED TAPE AND TRANSMISSION)

- Data sent to and from the 1050 will be in the PTT Code/8.

The 1050 system operates with the PT&T/8 code. Translation to and from this code is handled in the 2030-1051 attachment. The PT&T/8 code is made up of the standard BCD bits, B, A, 8, 4, 2, and 1. The C-bit is used to maintain odd parity. This standard code is shifted (upper case or lower case) to give more possible code combinations. Figure 5-2 shows both the upper-case and lower-case designations of the PT&T/8 characters used for console operation to the 2030.

BCD BITS		A	B	BA
	Space	@	-	&
1	1	/	j	a
2	2	s	k	b
2 1	3	t	l	c
4	4	u	m	d
4 1	5	v	n	e
4 2	6	w	o	f
4 2 1	7	x	p	g
8	8	y	q	h
8 1	9	z	r	i
8 2	0			
8 2 1	#	,	$	.
8 4	PUNCH ON	BYPASS	RE STORE	PUNCH OFF
8 4 1	READER STOP	LINE FEED	NEW LINE	HORIZ TAB
8 4 2	UPPER CASE SHIFT	EOB	BKSP	LOWER CASE SHIFT
8 4 2 1	EOT	PREFIX	IDLE	DELETE

LOWER CASE

	A	B	B A
Space	¢	—	+
=	?	J	A
<	S	K	B
;	T	L	C
:	U	M	D
%	V	N	E
'	W	O	F
>	X	P	G
*	Y	Q	H
(	Z	R	I
)			
"	|	!	¬
PUNCH ON	BYPASS	RESTORE	PUNCH OFF
READER STOP	LINE FEED	NEW LINE	HORIZ TAB
UPPER CASE SHIFT	EOB	BKSP	LOWER CASE SHIFT
EOT	PREFIX	IDLE	DELETE

UPPER CASE

Figure 5-2. PTT Code /8

EXTENDED BINARY-CODED DECIMAL INTERCHANGE (EBCDI)

- Data from the 1050 will be stored in the 2030 using the EBCDI code.

The EBCDI code is used to store data or send data to the 1050. Translation is necessary to convert the PTTC/8 to EBCDI, and EBCDI to PTTC/8. The EBCDI code in the 2030 makes the PTTC/8 characters compatible with the 8-bit code used by the CPU. Figure 5-3 shows the EBCDI code and the bit structure of the PTTC/8 characters as they would be stored in the CPU as a byte.

Bit Positions Within Byte

4567	00				01				10				11			
	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NUL				BLK	&	-									0
0001							/		a	j			A	J		1
0010									b	k	s		B	K	S	2
0011									c	l	t		C	L	T	3
0100	PCH OFF	RE-STORE	BYPASS	PCH ON					d	m	u		D	M	U	4
0101	HORIZ TAB	NEW LINE	LINE FEED	RDR STOP					e	n	v		E	N	V	5
0110	LOWER CASE SHIFT	BKSP	EOB	UPPER CASE SHIFT					f	o	w		F	O	W	6
0111	DELETE	IDLE	PREFIX	EOT					g	p	x		G	P	X	7
1000									h	q	y		H	Q	Y	8
1001									i	r	z		I	R	Z	9
1010					¢	!	■	:								
1011					.	$	,	#								
1100					<	*	%	@								
1101					(	)	—	'								
1110					+	;	>	=								
1111					|	¬	?	"								

Figure 5-3. EBCDI Code

CODE TRANSLATION

- 93 code combinations are translated on input to 2030.
- 105 code combinations are translated on output to 1050.

Input Translation PTTC/8 to EBCDI

The PTTC/8 Code Characters that will be translated on input are:

26 Upper Case Alpha: A-Z

26 Lower Case Alpha: a-z

10 Numerics: 0-9

26 Special Characters

1 Space or blank

4 Control codes: New Line, Tab, Line Feed, and Backspace

Figure 5-4 shows the character code and the characters that are entered into the 2030 storage.

BCD BITS		A	B	BA
	Space	@	-	&
1	1	/	j	a
2	2	s	k	b
2 1	3	t	l	c
4	4	u	m	d
4 1	5	v	n	e
4 2	6	w	o	f
4 2 1	7	x	p	g
8	8	y	q	h
8 1	9	z	r	i
8 2	0			
8 2 1	#	,	$	.
8 4	PUNCH ON	BYPASS	RE STORE	PUNCH OFF
8 4 1	READER STOP	LINE FEED	NEW LINE	HORIZ TAB
8 4 2	UPPER CASE SHIFT	EOB	BKSP	LOWER CASE SHIFT
8 4 2 1	EOT	PREFIX	IDLE	DELETE

LOWER CASE

	A	B	B A
Space	¢	—	+
=	?	J	A
<	S	K	B
;	T	L	C
:	U	M	D
%	V	N	E
'	W	O	F
>	X	P	G
*	Y	Q	H
(	Z	R	I
)			
"	|	!	¬
PUNCH ON	BYPASS	RESTORE	PUNCH OFF
READER STOP	LINE FEED	NEW LINE	HORIZ TAB
UPPER CASE SHIFT	EOB	BKSP	LOWER CASE SHIFT
EOT	PREFIX	IDLE	DELETE

UPPER CASE

Figure 5-4. PTT Code/8 to EBDIC Code TRANSLATION

Output Translation-EBCDI to PTTC/8

The EBCDI code characters used with the 1050 are:

26 Upper Case Alpha: A-Z

26 Lower Case Alpha: a-z

10 Numerics: 0-9

26 Special Characters

1 Space or blank

16 1050 Control Codes

Figure 5-5 shows all the code characters that will translate to PTTC/8 during output to the 1050.

Bit Positions Within Byte

0 1	00				01				10				11			
4567 \ 2 3	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
0000	NUL				BLK	&	-									0
0001							/		a	j			A	J		1
0010									b	k	s		B	K	S	2
0011									c	l	t		C	L	T	3
0100	PCH OFF	RE-STORE	BYPASS	PCH ON					d	m	u		D	M	U	4
0101	HORIZ TAB	NEW LINE	LINE FEED	RDR STOP					e	n	v		E	N	V	5
0110	LOWER CASE SHIFT	BKSP	EOB	UPPER CASE SHIFT					f	o	w		F	O	W	6
0111	DELETE	IDLE	PREFIX	EOT					g	p	x		G	P	X	7
1000									h	q	y		H	Q	Y	8
1001									i	r	z		I	R	Z	9
1010					¢	!	■	:								
1011					.	$	,	#								
1100					<	*	%	@								
1101					(	)	—	'								
1110					+	;	>	=								
1111					|	¬	?	"								

Figure 5-5. EBCDI Code to PTT Code/8 Translation

IBM 1051, MODEL N1

- A 1050 system with home-loop only operation.
- Reader-1 and punch-1 attachments combine to provide a data interface with the 2030.
- Model N1 can operate on-line to the CPU or off-line with standard 1050 I/O equipment.
- Home Error Correction is the only home-loop optional feature that will not work with the 2030.

The 1050 Model N1 is a stand-alone device using the 1050 home-loop operation only. It has no line-loop capabilities (sending or receiving data to or from remote 1050 terminals). Reader-1 and punch-1 attachments are reserved for use by the 2030. Figure 5-6 shows the maximum system configuration. With the configuration shown and with the home-component-recognition feature, the 2030 program can:

1. Select and read from a 1054 or 1056 reader.

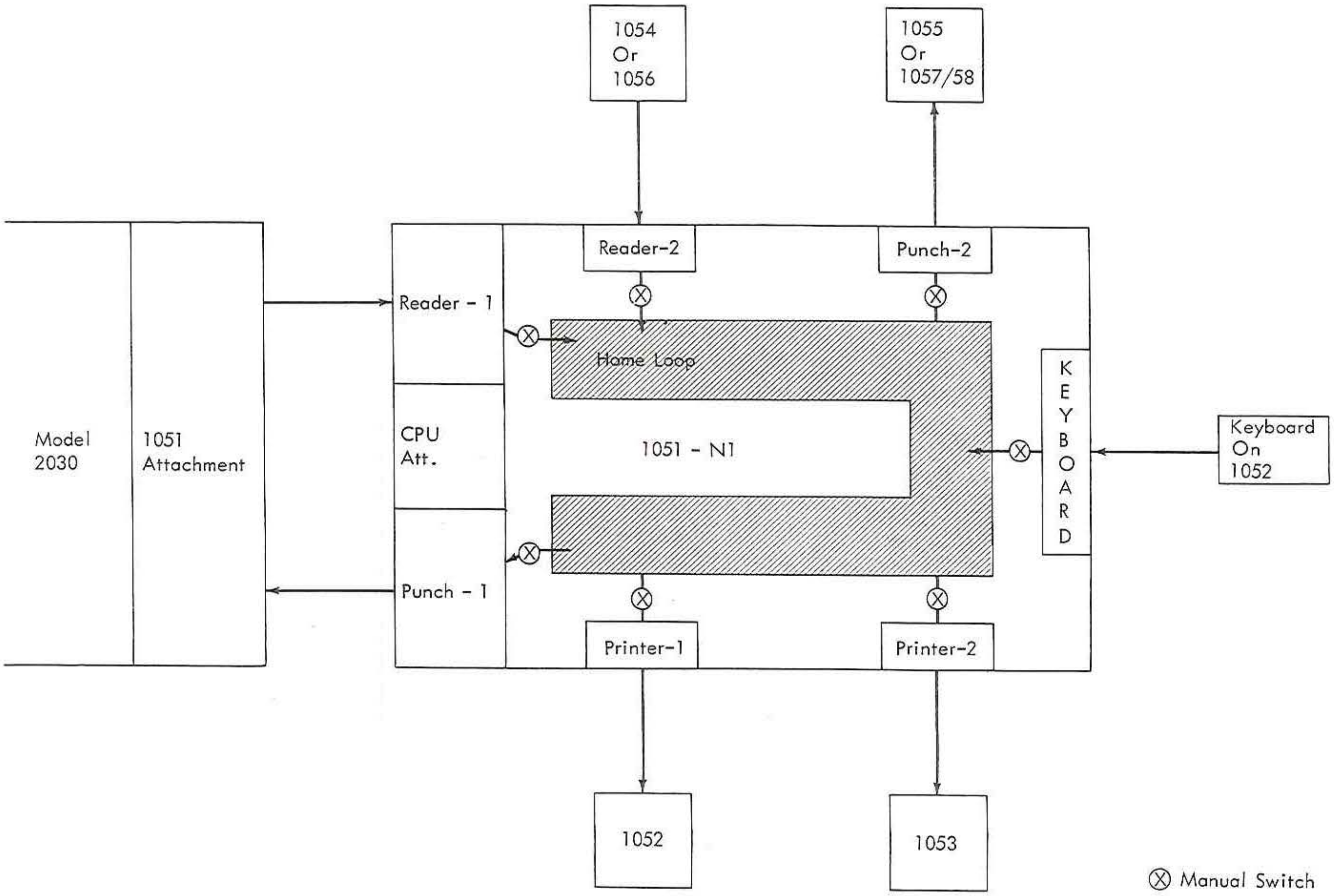


Figure 5-6. Maximum 1050 N1 System Configuration

2. Accept an operator initiated input from the keyboard, 1054, or 1056 reader.

3. Accept an operator initiated split input alternately from keyboard and from a 1054 or 1056 reader during a single read command.

4. Selects any one, or a combination of three outputs.

5. Change output devices when desired by control characters in the data stream.

6. Change ribbon color by control characters in the data stream.

7. Change line-feed spacing by control characters in the data stream.

All optional features available to the 1050 home loop will work with the 2030 except home error correction. See Figure 5-7 for the features available and on-line or off-line functions.

1050 HOME LOOP FEATURES	1051 Model N1	1051 Model 1	FUNCTIONAL		BASIC PREREQUISITES FOR 2030 ATTATCHMENT
			ON LINE TO 2030	OFF LINE	
2nd Printer Attachment	yes	yes	yes	yes	
1st Reader Attachment	yes	yes	yes	no	yes
2nd Reader Attachment	yes	yes	yes	yes	
1st Punch Attachment	yes	yes	yes	no	yes
2nd Punch Attachment	yes	yes	yes	yes	
CPU Attachment	yes	yes	yes	no	yes
Home Component Recognition	yes	yes	yes	yes	
Auto Fill Character Generation	yes	yes	yes	yes	
Auto Ribbon Shift	yes	yes	yes	yes	
Selective Line Feed	yes	yes	yes	yes	
Home Error Correction	yes	yes	no	yes	
Switch Unit	yes	yes	yes	yes	Needed only if a 1052 is not part of system
Extended Graphics	yes	yes	yes	yes	Yes - only if card equipment is attached.

Figure 5-7. 1050 Home Loop Features

Keys, Lights, and Switches

- Punch-1 switch becomes CPU connect switch.
- Request Key and Proceed Light used for operator initiated input to CPU. Figures 5-8A and 5-8B show the 1050 N1 keyboard and switch panel. The switches, keys and lights that pertain to CPU operation are circled.

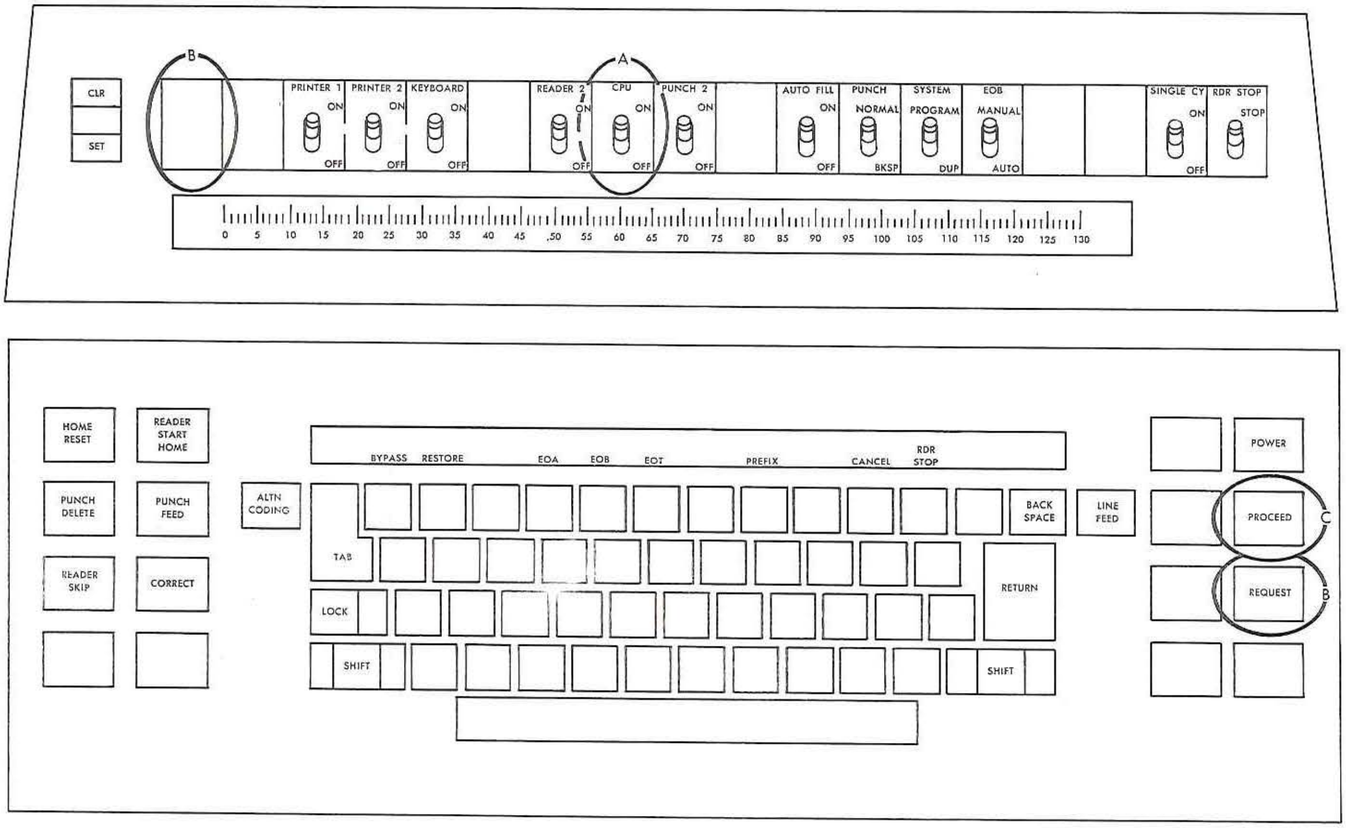


Figure 5-8. Keyboard and Switch Panel for 1052 N1 (Part 1 of 2)

CPU SWITCH: With the CPU attachment feature, the punch-1 switch becomes the CPU connect switch. In the CPU On position, the 1050 is on-line to the 2030. In the CPU Off position, the 1050 is off-line to the 2030.

REQUEST KEY: The operator presses this key when data is to be sent to the 2030.

PROCEED LIGHT: When on, this light signifies that a request has been granted and data can be sent.

All other keys, lights and switches retain their same function as described in the 1050 SRL Manual, Form A24-3020.

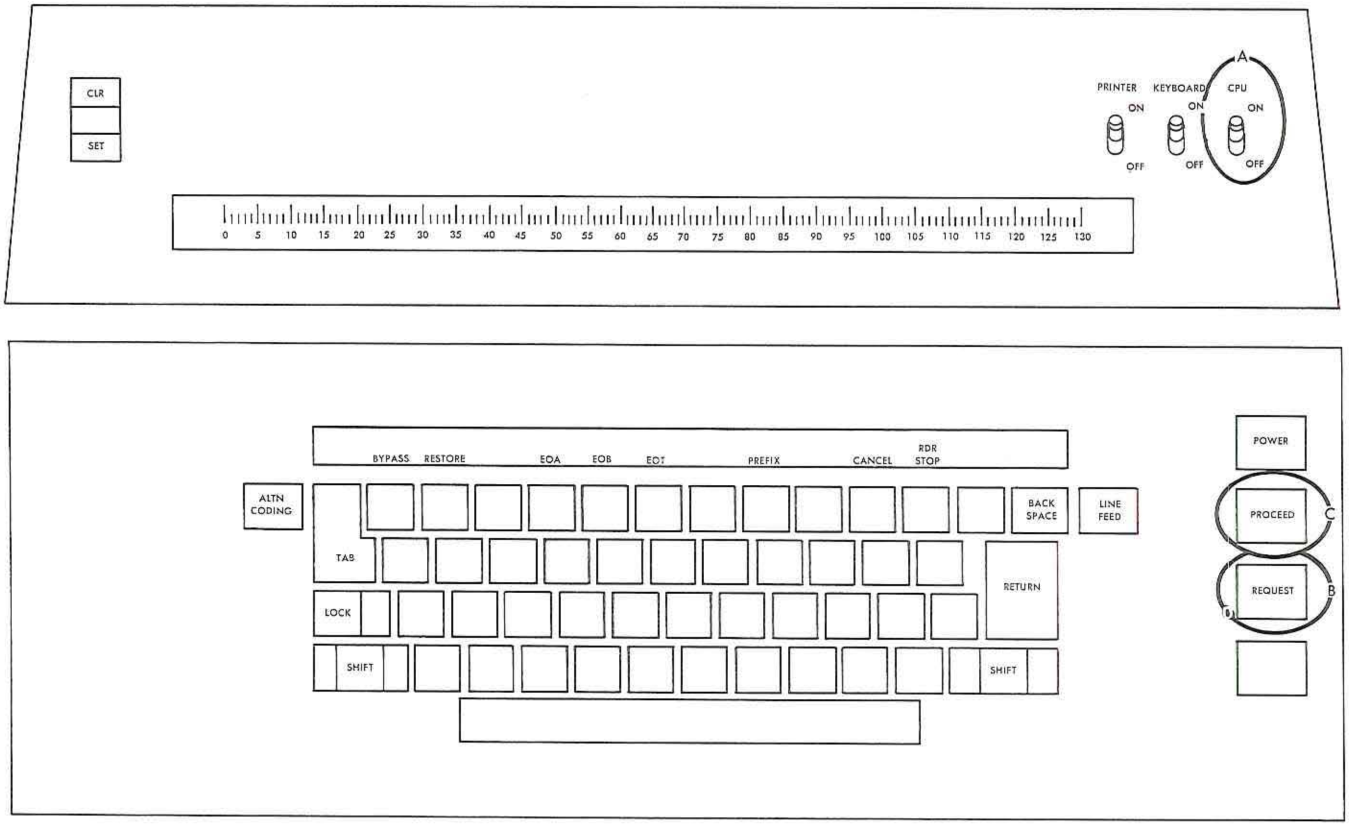


Figure 5-8. Keyboard and Switch Panel for 1052 N1 (Part 2 of 2)

IBM 1050, MODEL 1

- A 1050 system with both home-and line-loop operation.
- Reader-1 and punch-1 attachments combine to provide a data interface with the 2030.
- Model 1 can operate on-line to the CPU with the home loop, and with other 1050 terminals using the line loop.
- Home-error correction is the only home-loop optional feature that will not operate with the 2030.

The 1050 Model 1, having both home and line loops, can operate:

1. Home loop to 2030 only.
2. Home loop off-line only.
3. Home loop off-line and line loop to other terminals provided the same I/O device is not needed for both loops.
4. Home loop on-line and line loop to other terminals if the same I/O devices are not needed for both loops.

This flexibility of operation provides a unique console function. The information from the line loop can be assembled in punched cards or paper tape and then sent to the 2030 on the home loop.

As in the N1, the reader-1 and punch-1 attachments are reserved for CPU use. Figure 5-5 shows the home-loop options and their functions for the Model 1.

Figure 5-9 shows the maximum system configuration for the Model 1.

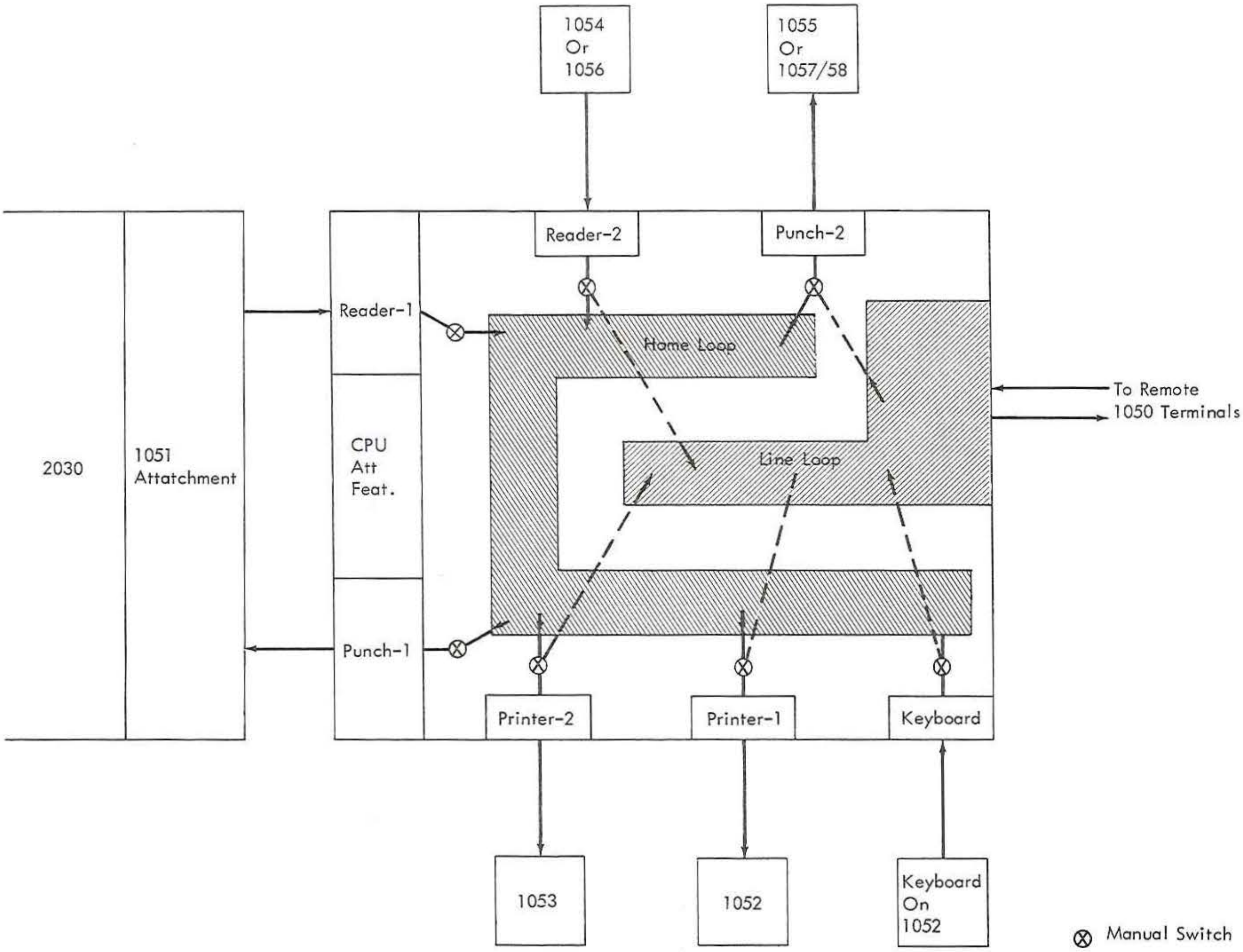


Figure 5-9. Maximum 1050 Model 2 Systems Configuration

KEYS, LIGHTS, AND SWITCHES

- Punch-1 Switch becomes CPU Connect Switch.
- Attend-Unattend switch must be in the Attend position.
- Request key and proceed light are used for operator initiated input.

Figure 5-10 shows the switches that pertain to CPU operation with the 1050 Model 1.

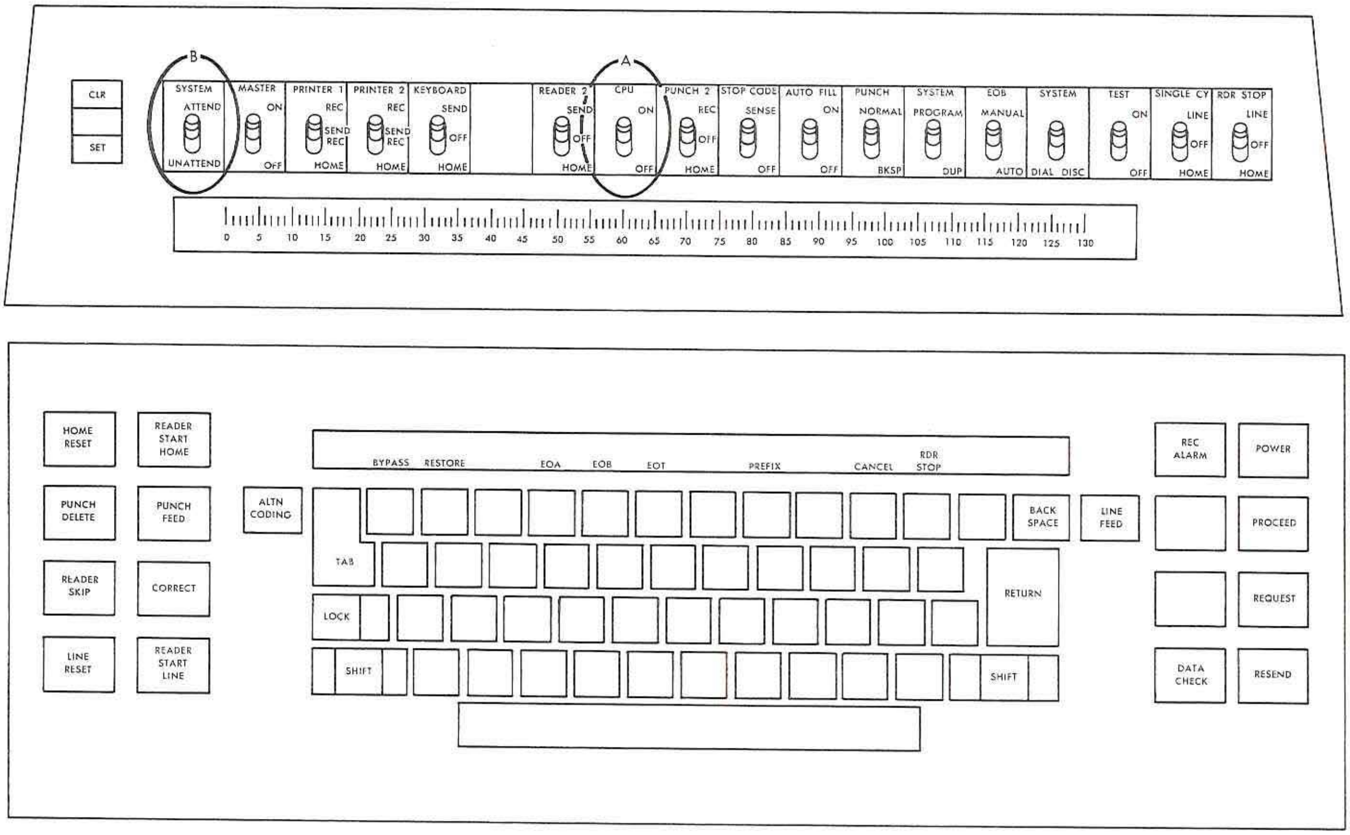


Figure 5-10. 1050 Model 1 Switch Panel

CPU SWITCH: The punch-1 switch becomes the CPU connect switch with the CPU attachment feature. In the On position the 1050 is on-line to the 2030. In the Off position, the 1050 is off-line to the 2030.

ATTEND-UNATTEND SWITCH: must be in the Attend position, because there is no way to turn on motor control in the 1050 using the home loop.

The request key and proceed light serve the same function as in the Model N1. All other keys, lights and switches maintain their function as described in the 1050 SRL Manual.

The reader-1 switch is removed with the CPU attachment feature because the CPU has the reader-1 interface connected, and no other I/O device can operate on it.

CPU ATTACHMENT IN 1051 (M1+N1)

- Contains the controls needed to allow CPU operation of the home loop.

Several basic controls are needed to have the 1050 function on-line and have the operation under control of the CPU. The circuitry necessary is contained on two double SMS cards in the 1051. These circuits will allow:

1. CPU reset of the home loop.
2. Control of keyboard restore.
3. Control of the home-component-recognition latches for

punch-1, reader-1, reader-2, and printer-1 without a prefix operation.

4. The turn-on and turn-off of the home-reader start latch.

5. Control of the reader-2 hold line.

6. Downshift and carrier-return operations when needed at the beginning or end of a print operation.

7. Preventing home-error-correction circuits from functioning when on-line.

Besides controlling the 1051, the circuitry in the attachment feature activates many control lines to the CPU. An explanation of these and other connecting lines to the CPU is covered in Functional Units.

IBM 2030 TO 1051 INTERFACE

- The cable connecting the 2030 to the 1051 is called the interface.
- Contains four basic sections:

  1. Read (Punch-1 Attachment)
  2. Write (Reader-1 Attachment)
  3. Controls to and from the 1051.
  4. Voltage lines and EPO (Emergency Power Off).

Read (Punch-1 Attachment)

The read bus between the 2030 and the 1051 is used to send data from the 1050 to the 2030. The punch-1 attachment in the 1051 is used for this purpose. It contains the 7 bit lines and controls necessary to send data to the CPU.

WRITE (READER-1 ATTACHMENT)

The write bus portion of the interface is used to send data from the 2030 to the 1050. One means of input to the 1050 is the reader-1 attachment. The CPU will use this attachment when sending data to the 1050. In fact, the 2030 looks like a reader to the 1051 controls.

CONTROLS

Many lines are necessary on the interface to send controls to and from the 1051. Basically, in write mode, these controls are used to prevent the 2030 from running faster than the 1051. Also, during read mode, the 1051 is interlocked to prevent it from running faster than the 2030 program.

VOLTAGES AND EPO

Voltages are exchanged between the two units on the interface. The EPO lines prevent the 1051 from powering up if the 2030 is in an EPO condition. A switch in the 1051 allows operation of the 1050 off-line when the CPU is in EPO.

IBM 1051 ATTACHMENT IN 2030 (FIGURE 5-11)

- The attachment is located in the 2030 and contains:

    1. R/W Clock
    2. R/W Register
    3. Controls out-TA
    4. Controls In-TT
    5. R/W Translators
    6. R/W Controls
    7. Share-Request Controls

R/W CLOCK

The clock runs in both read and write operations. During the read operation, the clock basically serves to time just when the CPU will accept data from the 1050. Also, the clock cycles once for each read cycle.

During the write operation, the clock starts at the beginning and runs until the write command is completed. The clock also serves as a timer, limiting the CPU to the speed of the 1050 I/O equipment.

R/W REGISTER

A set of latches that store characters temporarily during input or output operations.

CONTROLS OUT-TA

A set of 8 lines that control operations from the microprogram. The mnemonic designation of these control out lines is TA.

Controls In-TT

A set of 8 lines the microprogram or hardware can use to test or know the status of the 1050 and/or the attachment. Such things as intervention required will be sampled from this control area.

R/W TRANSLATORS

This is a series of circuits that will translate from EBCDI to PT&T/8 and from PT&T/8 to EBCDI.

R/W CONTROLS

A set of controls that analyze the contents of the TA register and establish read, read inquiry, write, etc. They are used to control the clock speed and data flow.

SHARE-REQUEST CONTROLS

These controls are activated when any condition in the attachment signifies that a microprogrammed share-request should take place.

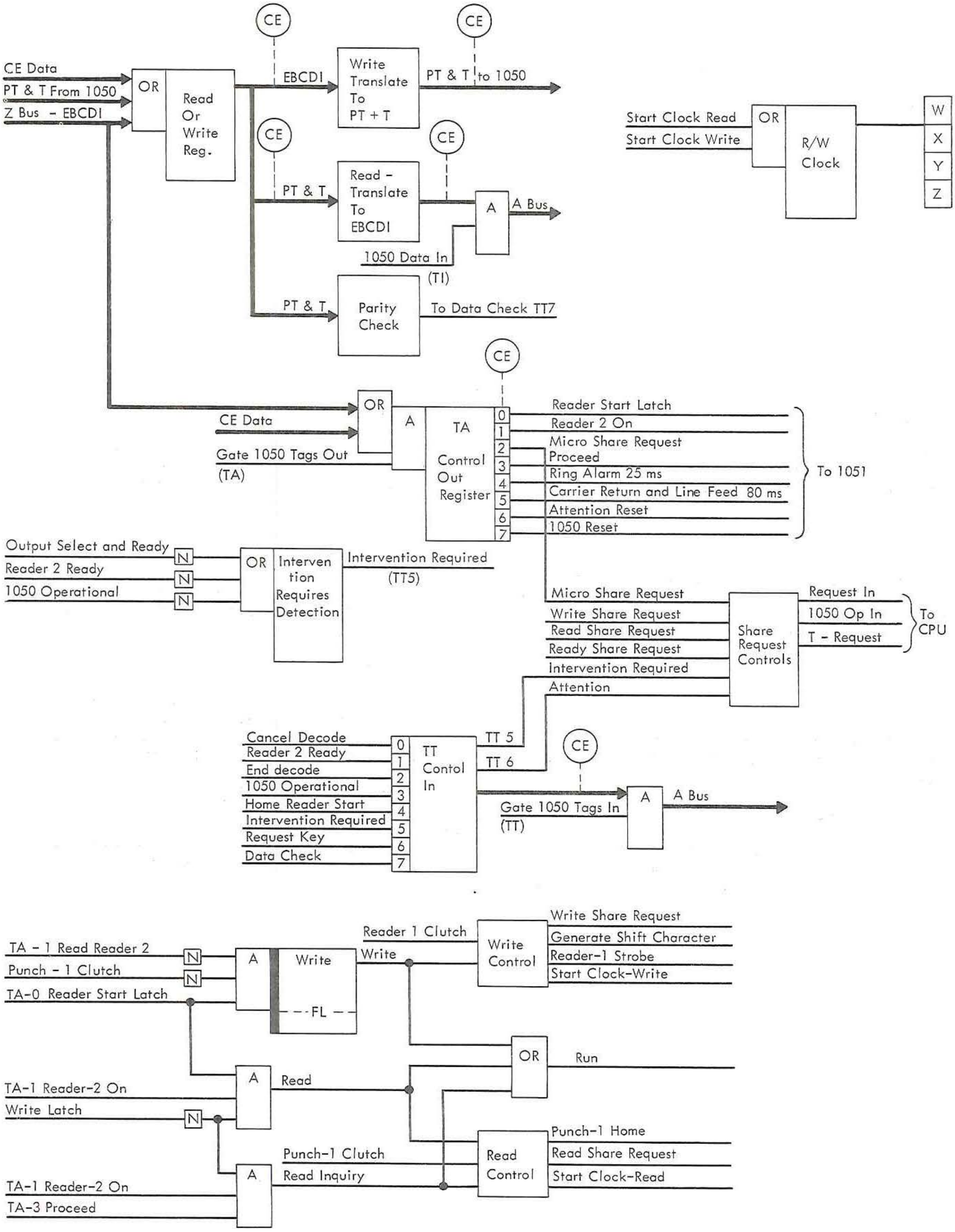


Figure 5-11. 1051 Attachment in 2030

COMMANDS TO THE 1050

- Interrogation of Start I/O CCW Command Byte can initiate these basic commands:

    1. Read
    2. Read Inquiry
    3. Write
    4. Sense
    5. Control

READ

A read operation is defined as a macroprogram initiated read instruction to the 1051.)the microprogram checks for reader-2 and will select the reader, set up controls, and start the attachment clock.

READ INQUIRY

A read inquiry command is normally issued in response to a request-in operation from the 1051. It can be an operator- initiated command, and the microprogram will allow an input from the 1052 keyboard, or reader-2 if the operator so desires, provided the selected device is set to the home position.

WRITE

A write command is initiated by the macroprogram start I/O. The microprogram controls the data flow and checks for any output device at the 1050 being selected and ready.

SENSE

A command issued to the attachment to test the condition of the attachment and the 1051. A byte is assembled and stored in MPX storage, address 04 before issuing the command. The sense operation stores this byte in main storage for further interrogation.

CONTROL

This is command-immediate operation that can be used as a control function. One example is the alarm operation. This command rings an alarm for 2 seconds to alert the operator.

DATA FLOW

- CPU Read Operation is defined as receiving data at the 2030 from the 1050.
- CPU Write Operation is defined as sending data from the 2030 to the 1050.

Figure 5-12 shows the basic data flow path for a CPU write operation. Data from the Z-bus in the CPU enters the 1051 through the reader-1 attachment and prints on printer-1 (1052). This data could have been printed on printer-2 or punched on punch-2 if the 1050 had these options and they were selected to the home loop.

Figure 5-13 shows the basic data flow path for a CPU read operation. The input at the 1051 is reader-2. Input could also be originated at the keyboard. Note that the punch-1 attachment is used to send information to the 2030 and on the A-bus to storage.

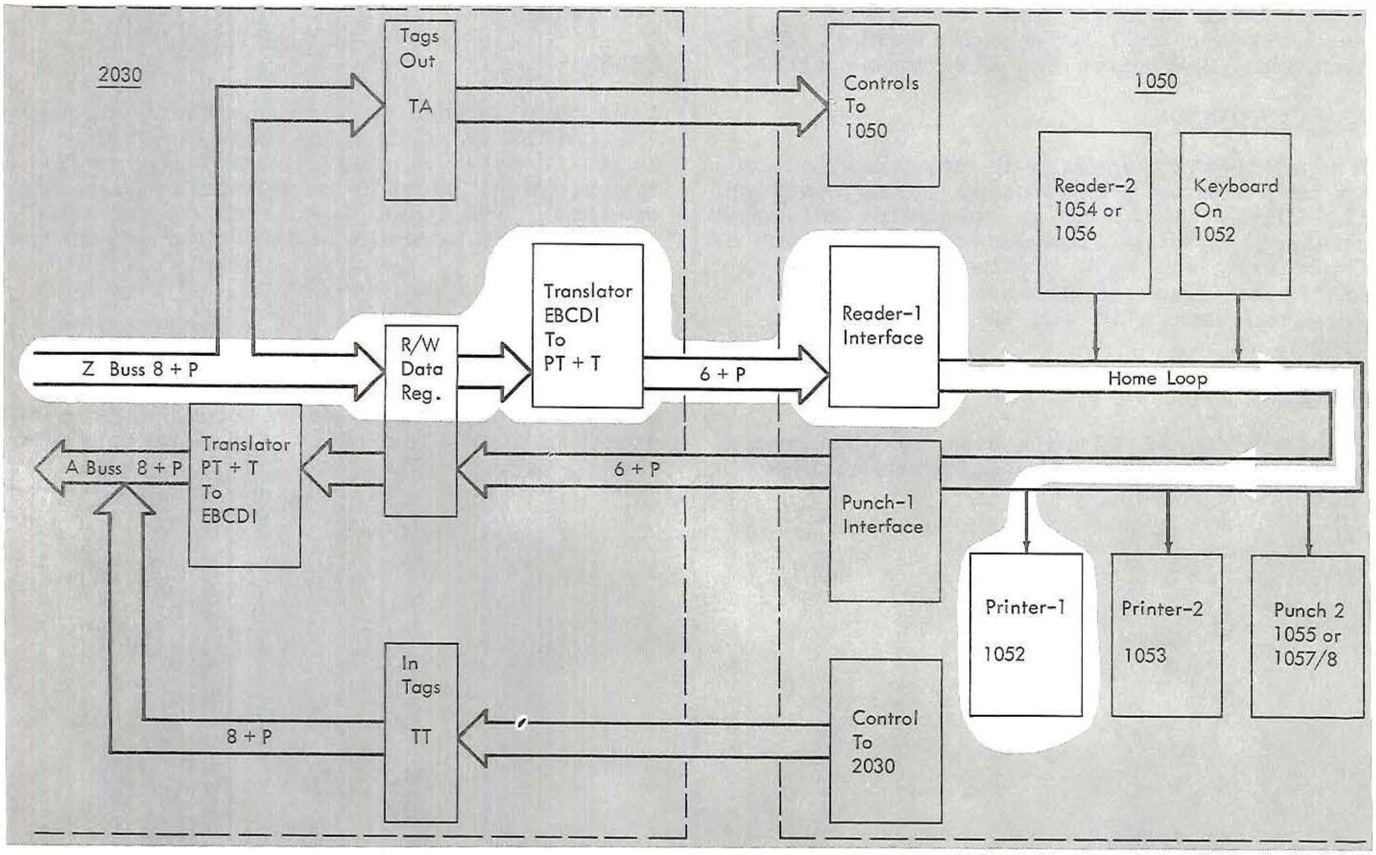


Figure 5-12. Write Data Flow

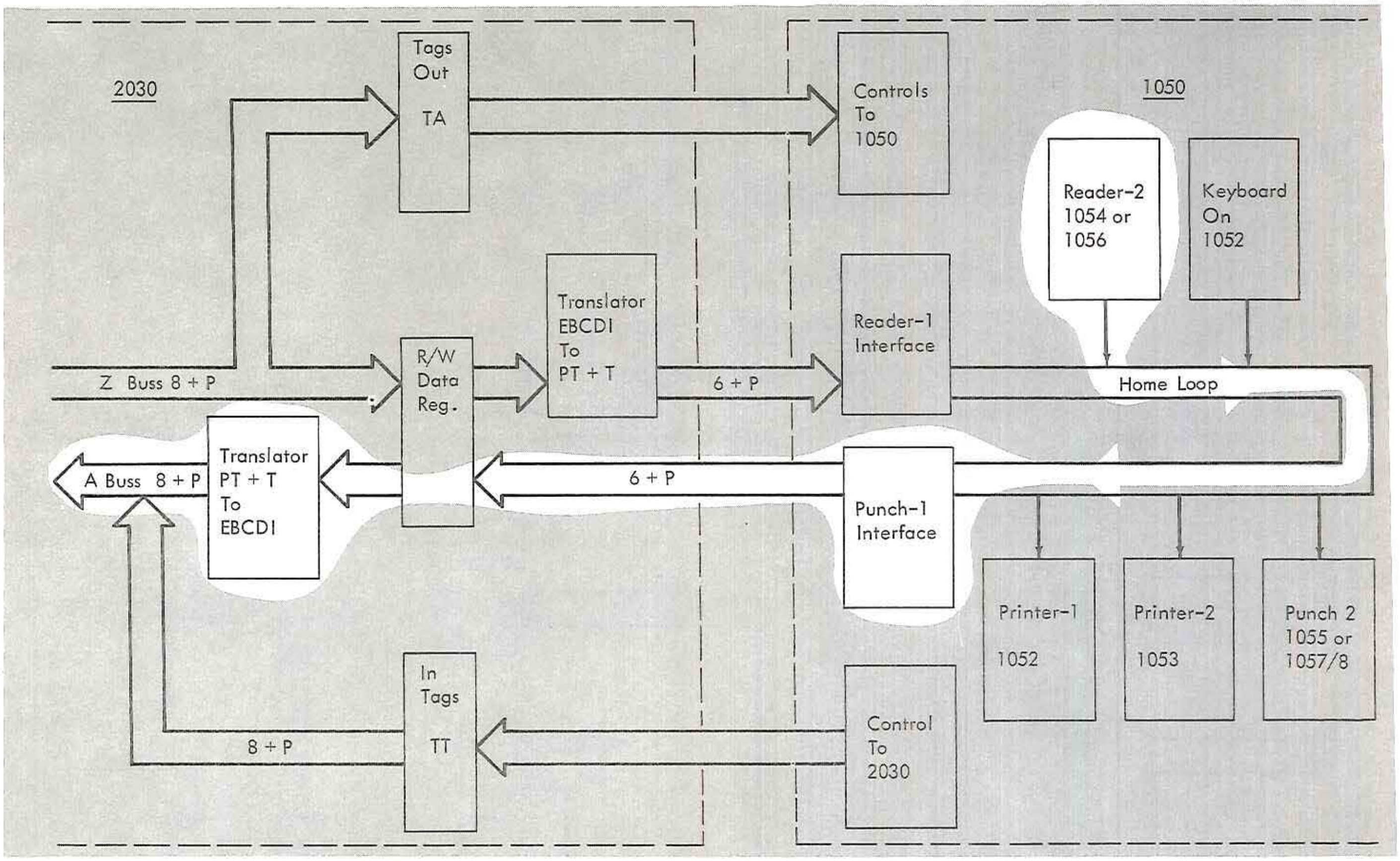


Figure 5-13. Read Data Flow

IBM 1051 ATTACHMENT CLOCK

- Read and write operations are controlled in the attachment by clock timings.
- The clock is oscillator driven.
- The oscillator is clamped off when the clock is stopped.
- One clock cycle results in four timed outputs: w, x, y, and z.

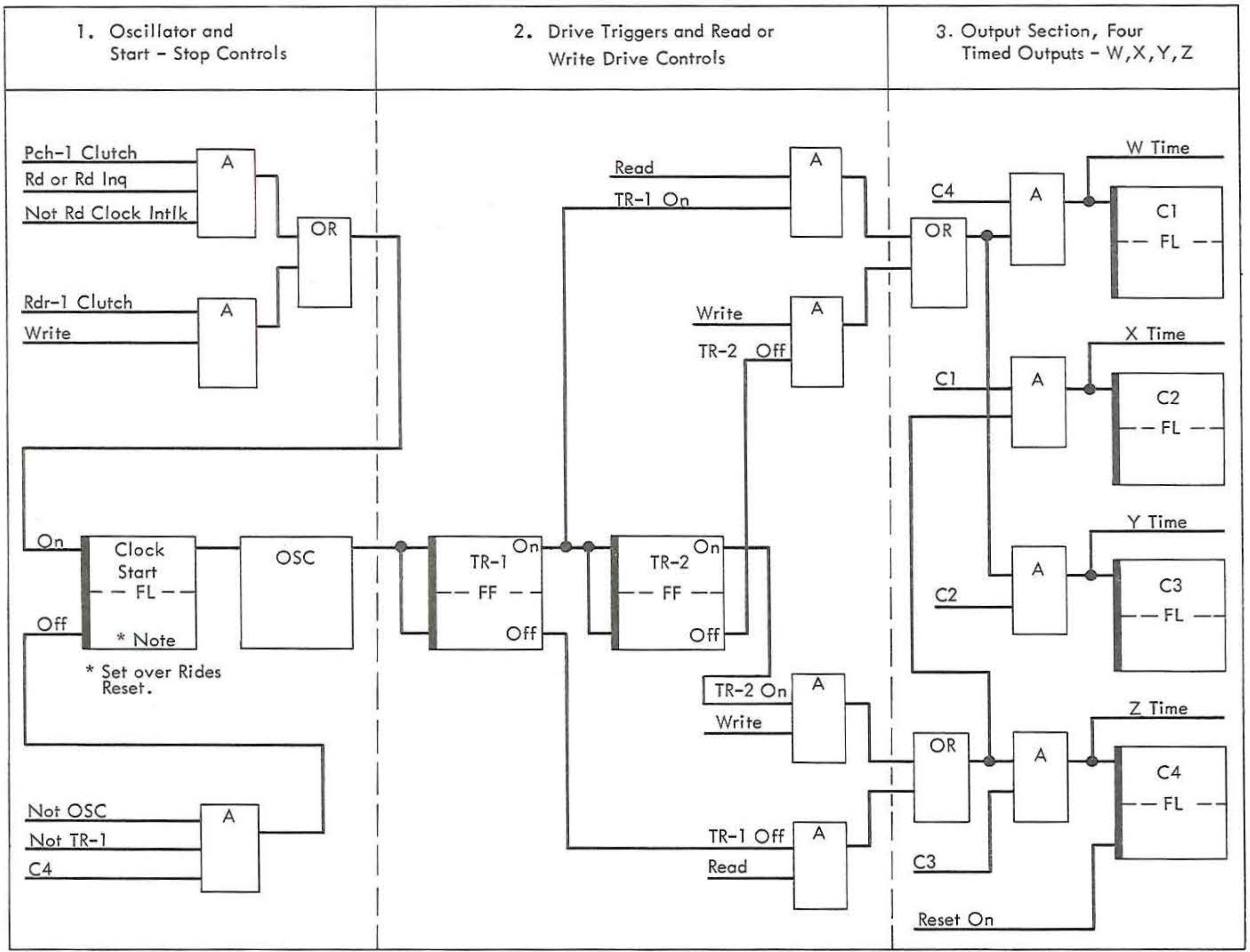


Figure 5-14. Attachment Clock

The attachment clock is oscillator driven and consists of three basic sections (Figure 5-14):

1. Oscillator and start-stop controls.
2. Drive triggers and read-write controls.
3. Output section of four timed pulses.

The oscillator in the clock is always clamped off when the clock is stopped. When the clamp is removed, the output shifts. When the clamp is applied again during a normal operation, it is always applied when the oscillator line is "down" where it remains until unclamped, when the cycle starts all over again.

CLOCK OPERATION DURING WRITE MODE

- The clock is started at the beginning of a write operation and turned off after all data transfers are complete.
- The cycle speed of the clock is limited to the speed of the 1050 output device (67.5 ms).

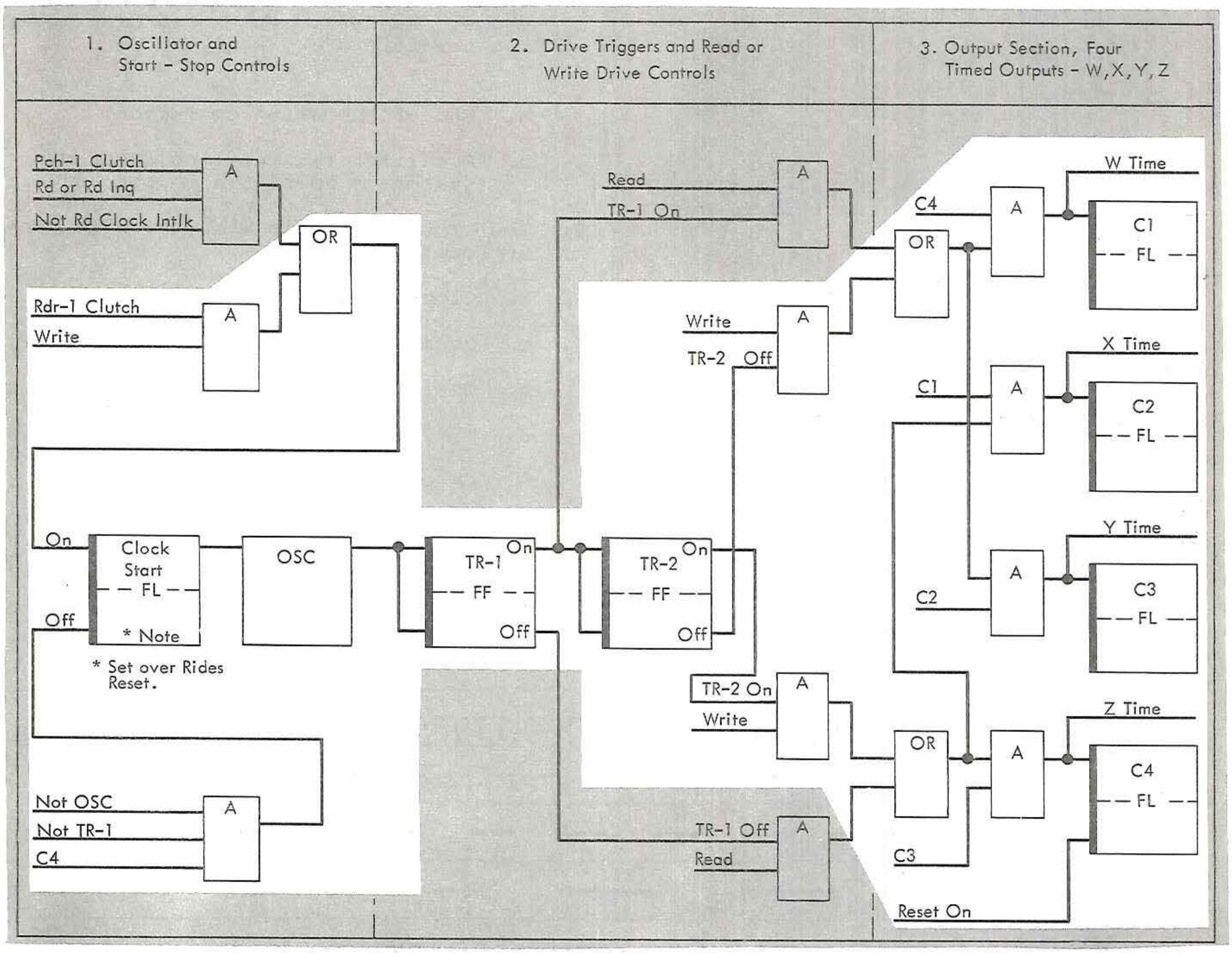


Figure 5-15. Attachment Clock-Write Mode

During a CPU write operation to the 1050, the character repetition rate must be controlled. The output devices at the 1050 operate at a speed of 14.8 characters per second or approximately 67.5 milliseconds per character. The attachment clock provides the control to give a character repetition rate of 68 milliseconds during write mode.

Circuit Objectives

Figure 5-15 shows write controls. The controls shown start the oscillator. The oscillator drives TR-1 which in turn, drives TR-2. In write mode, TR-2 pulses are used to step the control latches (C1, C2, C3, and C4) and also provide the w, x, y, and z clock times. The flipping of TR-2 is timed by the oscillator and TR-1 so that each complete clock cycle results in 68 ms (17 ms per stage).

When TR-2 is on, it supplies a drive pulse to C2 and C4. When TR-2 is off, a drive pulse is available to C1 and C3. The actual stepping and timing controls to the clock are shown in the timing chart (Figure 5-16).

Circuit Description

1. The idle conditions of the clock are:

   a. Oscillator clamped at the off level

   b. TR-1 off

   c. TR-2 on

   d. C4 On

   e. Clock Start Off

2. Reader-1 clutch line is made active by the write command if the 1050 is on-line and ready.

3. Clock start turns on with:

   a. Write

   b. Reader-1 clutch

4. C1 turns on with Clock Start.

5. W-time is available with Clock Start, not TR-2, not C3, and C4 on.

6. Each shift of the TR-2 line will advance the control latches one step and provide a clock-output pulse.

7. Once started, the clock runs continuously unless the reader-1 clutch line drops. Reader-1 clutch can drop under two conditions:

   a. End of the write operation.

   b. A carrier return, line feed, or tabulate operation in the 1050.

8. The clock will stop with the idle conditions:

   a. Clock Start Off

   b. Oscillator clamped off

   c. TR-1 off

   d. TR-2 on

   e. C4 on

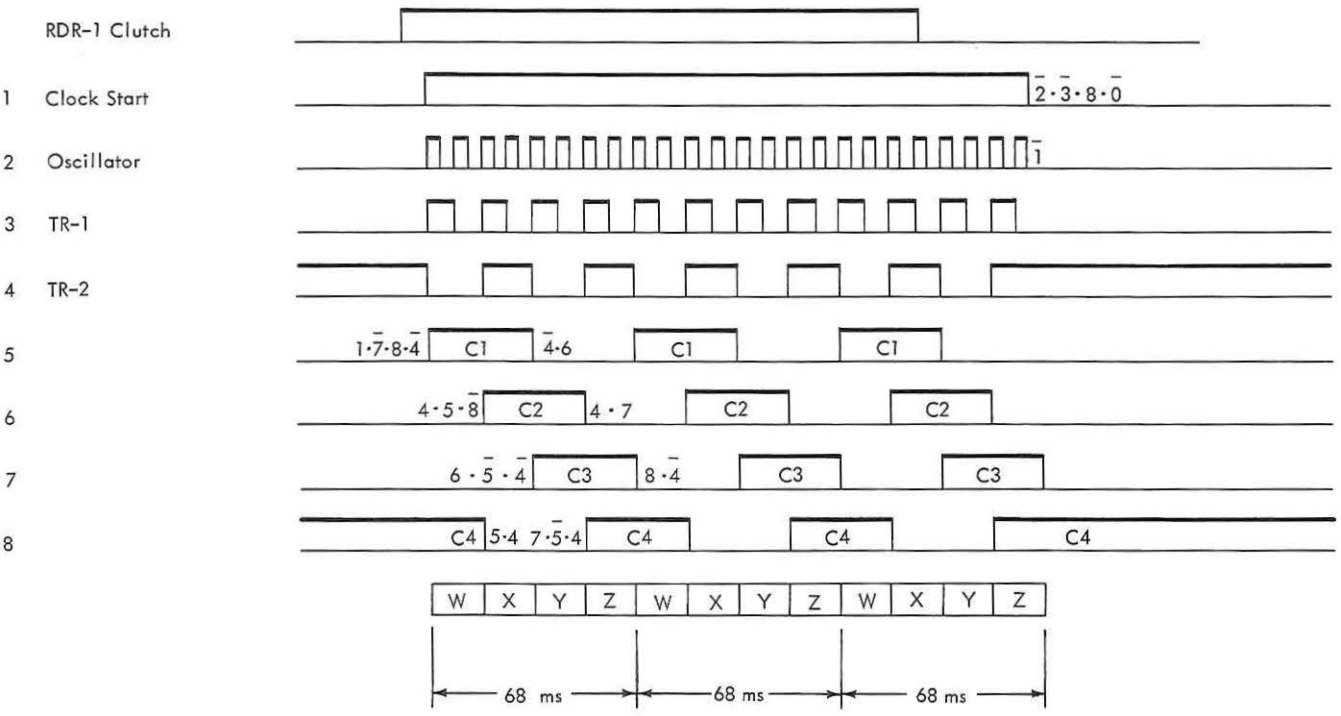


Figure 5-16. Attachment Clock-Write Mode Timing Chart

CLOCK OPERATION DURING READ MODE

- Data originating at the 1050 controls the cycle speed of the attachment.
- Clock is used to set data into the R/W register and initiate a read-share request.
- Clock runs at a cycle rate of 34 ms.
- Each character sent from the 1050 allows the clock to run one cycle and stop.

During a CPU read or read-inquiry operation, data is sent from the 1050, and the character-repetition rate is controlled by the 1050. The clock is used to set the character into the R/W register and initiate a read share request to the CPU. The clock runs at a repetition rate of 34 milliseconds and runs one cycle for each character sent.

Circuit Objectives

The basic read controls are shown in Figure 5-17. The oscillator starts under read control and punch-1 clutch from the 1050. The oscillator drives TR-1 and supplies the drive pulses to the control latches (C1, C2, C3, and C4). It also provides the timed w, x, y, and z clock times. TR-1 flips every 8.5 milliseconds resulting in a clock cycle of 34 milliseconds. When TR-1 is on, a drive pulse is available to the C1 and C3 control latches. When TR-1 is off, a drive pulse is available at the C2 and C4 control latches. Figure 5-18 shows the actual timing and stepping controls.

Circuit Description

1. The idle conditions of the clock are:
    a. oscillator clamped at the off level.
    b. TR-1 OFF
    c. TR-2 on
    d. C4 on
    e. Clock start off.
2. Punch-1 clutch is activated from the 1050 at the time a character is ready to be sent.
3. Clock start turned on by punch-1 clutch and read or read-inquiry operation and not read clock interlock.
4. C1 turns on with oscillator start.
5. The clock steps one stage with each flip of TR-1.
6. At Y time, the read-clock-interlock latch is turned on to allow stopping the clock at the end of Z-time. Read-clock interlock falls with punch-1 clutch.
7. Clock start drops with not oscillator, not TR-1, and C4 on.
8. The clock is now stopped until the rise of punch-1 clutch from the 1050 for the next character.

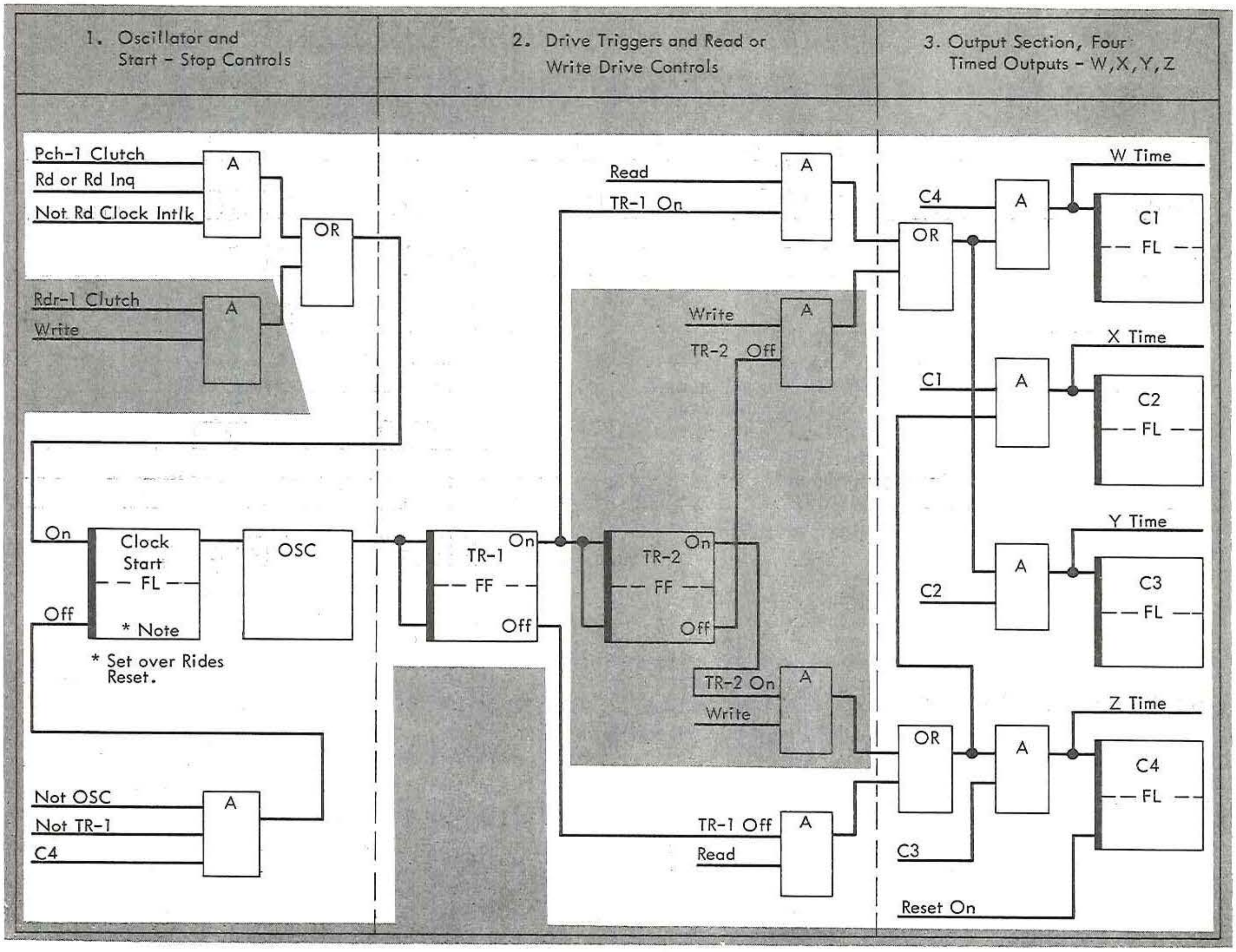


Figure 5-17. Attachment Clock-Read Mode

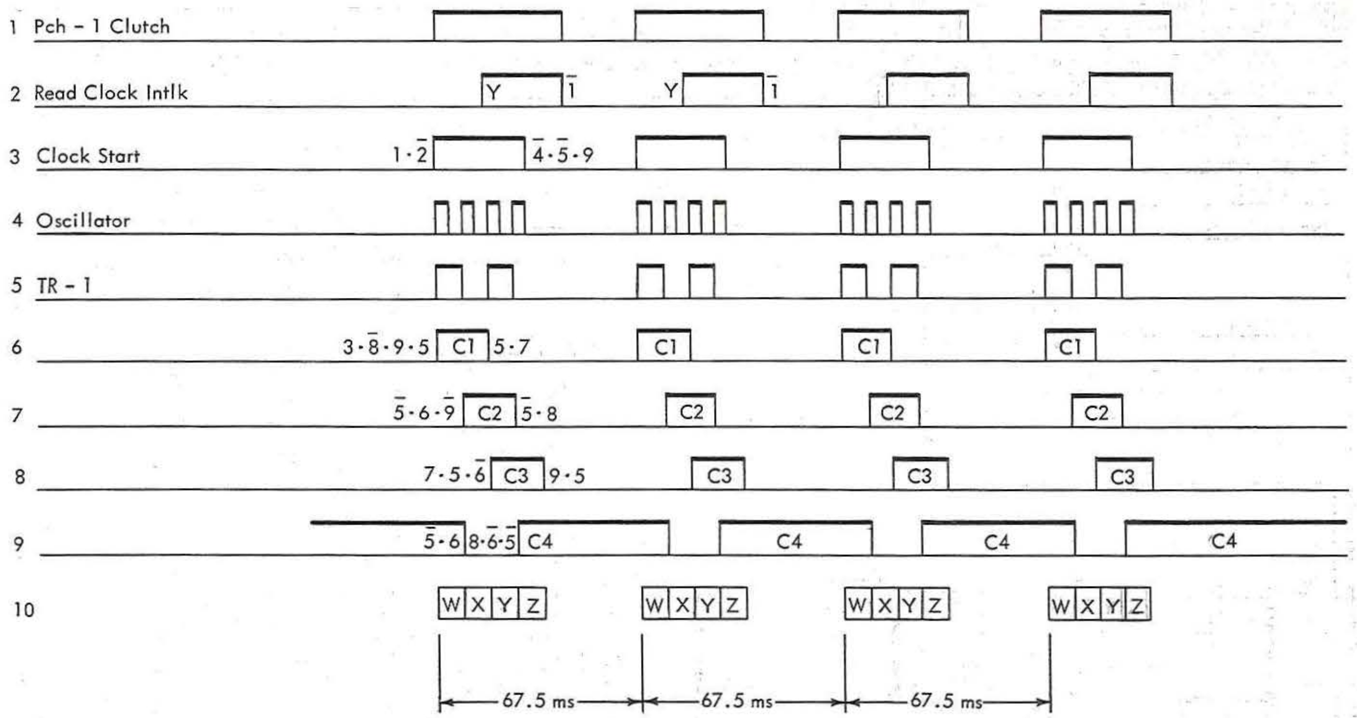


Figure 5-18. Attachment Clock-Read Mode Timing Chart

CONTROLS IN-TT

- A set of eight lines used in the microprogram to test the status of the attachment and the 1050.
- Assembled as a byte on the A-bus with the ROS order, Gate 1050 Tags In.

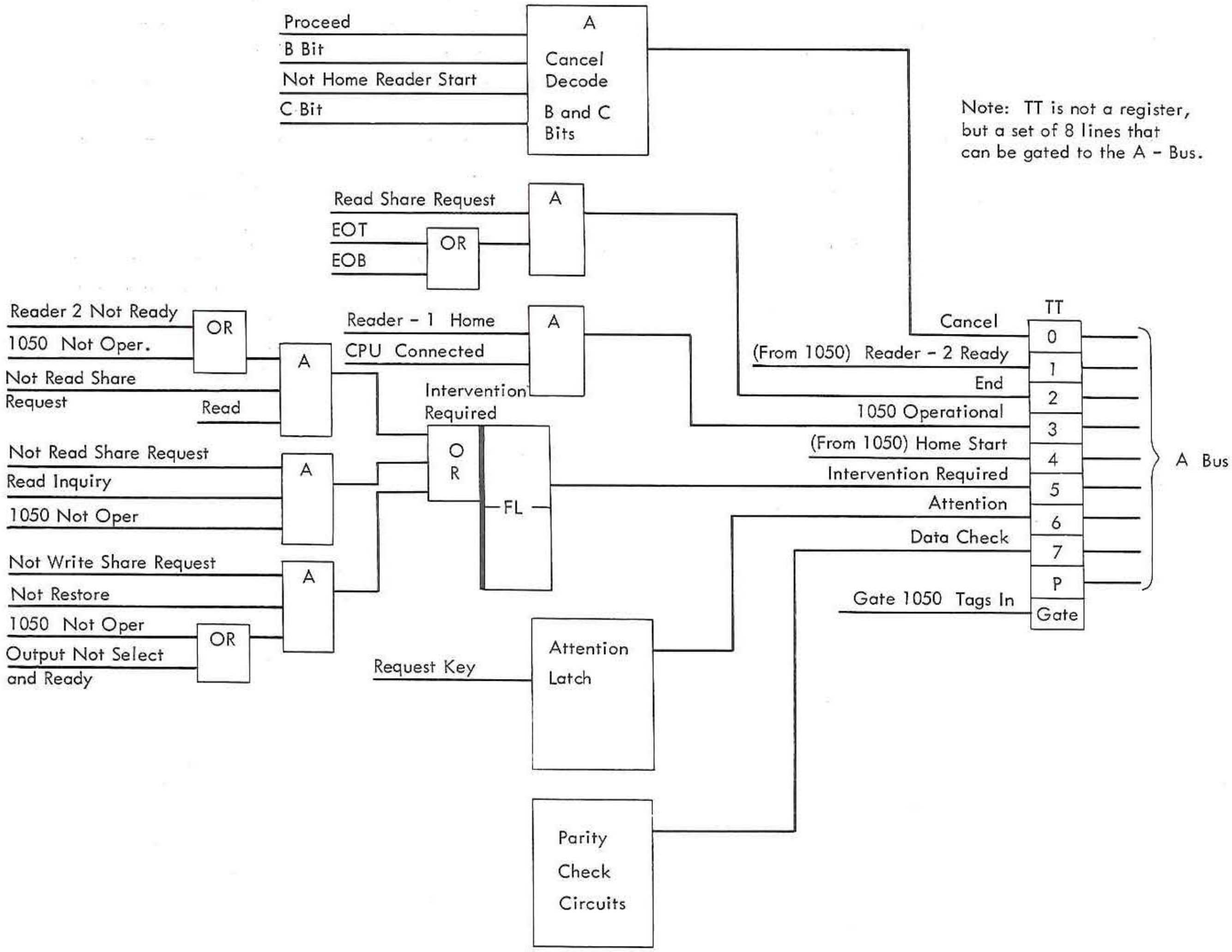


Figure 5-19. Controls In-TT

To operate properly, the microprogram must be able to test conditions within the adapter and in the 1050. Eight lines designated mnemonically as TT provide this control. At any time the program needs the status of these lines, a ROS order, Gate 1050 Tags In, will assemble a byte on the A-bus with a destination specified in the program. Further interrogation of this byte will follow in the microprogram.

The eight lines have been designated (Figure 5-19):

TT0 = Cancel: This line is active on a read-inquiry operation if the character decode is B-bit and C-bit (cancel character from 1050).

TT1 = Reader 2 Ready: This line comes from the 1050 on the interface and signifies the reader is on the home loop and the paper contacts are satisfied.

TT2 = End: This line is activated on a read operation if the character decode is EOB or EOT from the 1050. The ready-share interrupt also sets this line to indicate device end.

TT3 = 1050 Operational: This line is the AND condition of reader-1 home and CPU connected interface lines from the 1050.

TT4 = Home Start: This line is activated when the home-reader-start latch is on in the 1050.

TT5 = Intervention Required: Three conditions can activate this line:

1. Read Operation-reader-2 not ready or 1050 not operational.
2. Read Inquiry Operation-1050 not operational.
3. Write Operation-1050 not operational or Output not Select and Ready.

TT6 = Attention: This is activated by a request from the 1050.

TT7 = Data Check: This is activated on a read operation from the parity-check circuits in the attachment.

CONTROLS OUT-TA

- A set of eight lines is used to control the 1051 attachment during I/O cycles.
- Byte is assembled, it is put on the Z-bus and set into TA by the ROS order Gate 1050 Tags Out.

During any operation with the 1050 or the 1051 attachment, controls must be available to perform the operation; the unit must be told what to do. The TA register performs this function. It consists of eight lines, each representing a control or command to the 1051 attachment.

Each bit-line of the Z-bus is gated to the 1051 attachment with the ROS order: gate 1050 tags out. Figure 5-20 shows the TA register and the Z-bus input. If the zero bit line has a bit, it sets the latch called home reader start. The rest of the Z-bus lines set latches or condition lines with the presence of bits on the Z-bus lines.

Let's briefly discuss each line and its purpose.

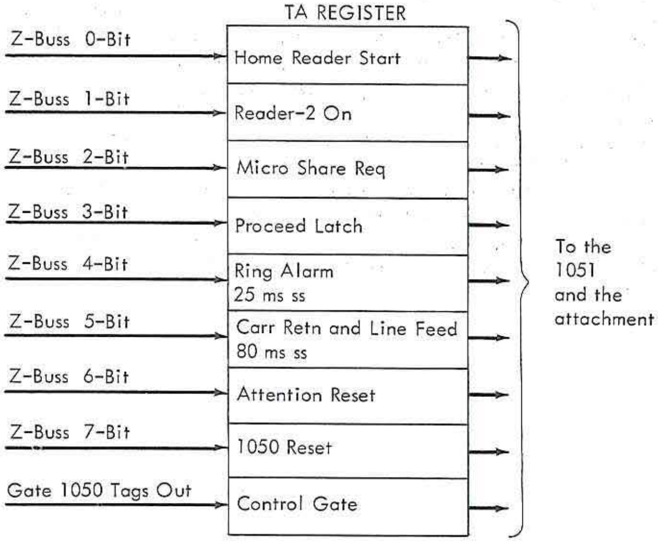


Figure 5-20. Controls Out-TA

TA0-Home Reader Start turns on the home-reader- start latch in the 1051 if the 1050 reader 1 home line is active and the TA proceed latch is off.

TA1-Reader 2 On drops the Reader-2 hold line if no other condition is attempting to hold it up.

TA2-Micro Share-Request causes a share-request cycle in the CPU and allows the 1050 attachment program to take a share cycle.

TA3-Proceed unlocks the 1052 keyboard and lights the proceed light if the attachment is not in a read-share-request cycle.

TA4-Ring Alarm a 25 ms single-shot pulse to ring the alarm if the system is equipped with the feature.

TA5-Carrier Return and Line Feed sends an 80 ms pulse to the 1052 printers to return the carrier and cause a line feed.

TA6-Attention Reset resets the attention latch.

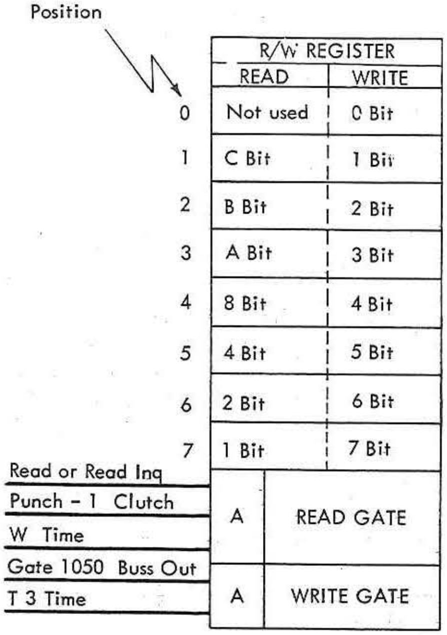
TA7-1050 Reset allows a reset to be sent to the 1051 to reset the home loop if the 1050 is selected to the CPU.

READ-WRITE REGISTER

- A set of eight latches; 8 bit-positions each representing a bit when on and no-bit when off.
- Used both in receiving PT&T/8 code or sending EBCDI code before translation.

Each position of the R/W (Read or Write) register represents a specific bit, depending on the operation being performed. Data can be handled either during a read operation or a write operation. Figure 5-21 shows the register positions and their use during each mode of operation.

The latches used in the R/W register are not reset by a common reset line. Any time the R/W register set line is made active, the latch for each bit position assumes the logic on the bit line input. For example, if the bit 0-line has a bit when the set line is made active, the latch assumes the state of containing a 0-bit. If the bit 0-line does not contain a bit the next time the set line is active, the latch assumes the status of containing no 0-bit. This register, therefore, will always have something in it, and will be allowed to change state only when the set line is made active.



Position	R/W REGISTER	
	READ	WRITE
0	Not used	0 Bit
1	C Bit	1 Bit
2	B Bit	2 Bit
3	A Bit	3 Bit
4	8 Bit	4 Bit
5	4 Bit	5 Bit
6	2 Bit	6 Bit
7	1 Bit	7 Bit

Figure 5-21. Read/Write Register

SHARE-REQUEST CONTROLS-POLL CONTROL

- Will capture polling if the 1050 has requested service.
- Uses MPX channel SEL-O and SEL-I pulses to capture polling when necessary.
- Will force CPU to channel mode and prevent the MPX channel from running until request cycle is over.

The share-request controls are used only when the attachment circuits call for a MPX channel share cycle. The controls will be active when any one of these conditions is satisfied:

1. TA micro share-request latch is on.
2. Request key operation, not run mode, and not TA micro share-request latch on.
3. Read share-request.
4. Write share-request.
5. Intervention required.
6. Ready Share-request

The control circuits are used to interlock the MPX channel polling circuits. Because the attachment is physically in the CPU, install it first or last on the channel poll lines select out (Sel-O) or select in (Sel-I). If the 1050 attachment is installed first, Sel-O is sent through it before going to the MPX channel. If the 1050 is installed last, the Sel-I line is sent through it before returning to the MPX channel poll controls.

Let's consider polling with start I/O and request cycles.

The 1050 is not polled during a start I/O operation. The microprogram recognizes the address as the address of the 1050. This causes an exit from normal start I/O into the 1050 microprogram . During the 1050 start I/O, controls will be set up to cause share-request cycles; the start I/O routine is ended. Even though the 1050 does not require the Sel-O and Sel-I lines during a start I/O, it must be able to handle them the same as any unit on the multiplexor channel during a start I/O to any unit actually on the MPX channel.

POLL CONTROL--START I/O--1050 INSTALLED FIRST

- Sel-O is generated by the start-select-out latch through microprogramming.
- Polling pulse bypasses 1051 attachment controls.

The start I-O instruction turns on the start-select-out latch in the multiplexor or channel poll circuits (Figure 5-22). The Sel-O pulse is sent to the channel, delayed by each unit, and eventually sent back to the CPU as Sel-I. Here, it is determined that the 1050 is first on the channel and the Sel-I pulse will function as in a multiplexor channel operation and turn off the start-select-out latch.

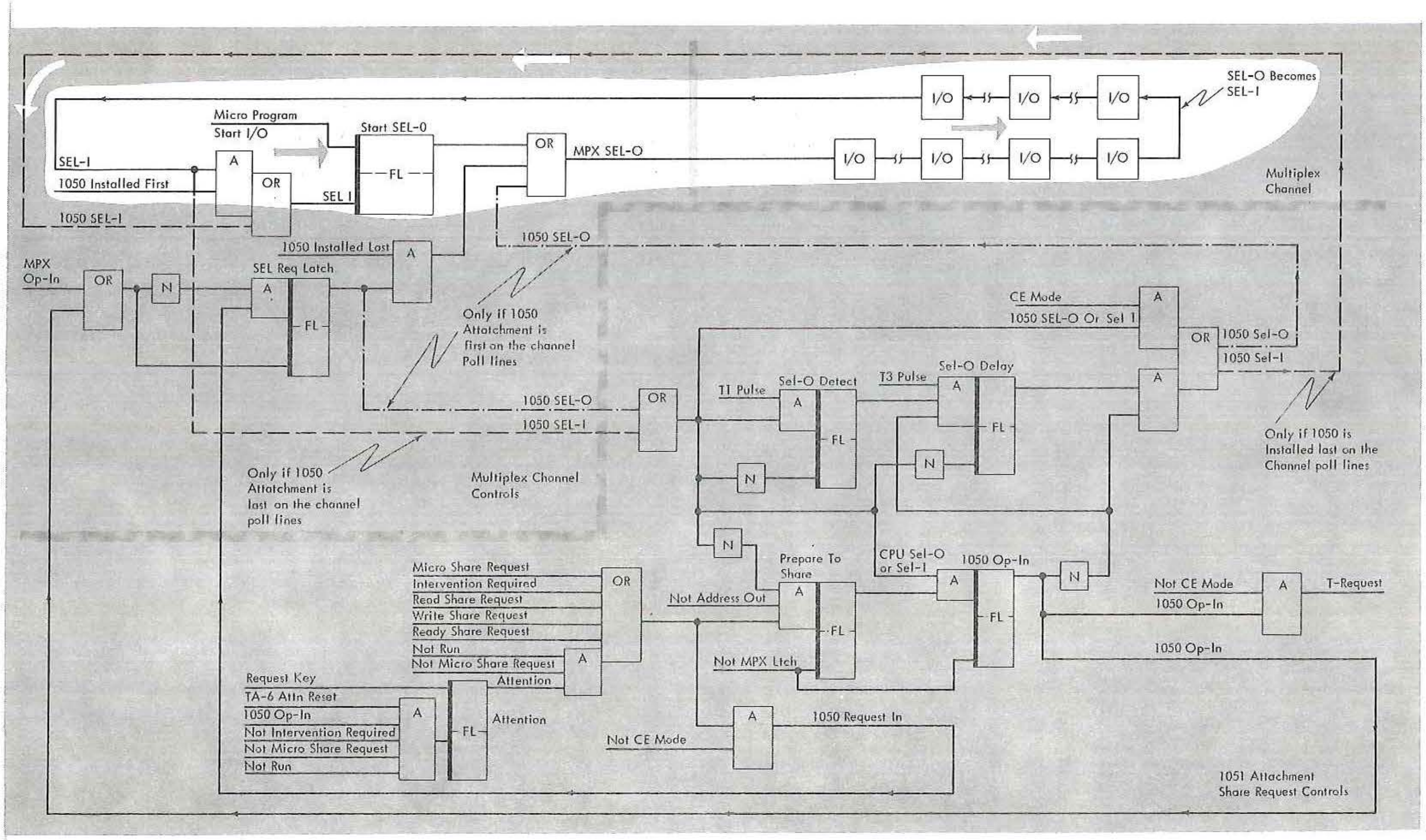


Figure 5-22. Poll Control--Start I/O to MPX Channel with 1050 Installed First

POLL CONTROL--START I/O 1050 INSTALLED LAST

- Sel-O is generated by the start-select-out latch through microprogramming.
- Sel-I is directed through the 1051 attachment circuits and then to turn off the start-select-out latch.

The start-select-out latch is again turned on by the microprogram (Figure 5-23). The Sel-O pulse is sent to the channel and is delayed by each control unit as a test is made to see if that unit can capture the Sel-O pulse. If no unit captures the Sel-O line, the pulse returns to the MPX channel circuits as Sel-I.

An attempt to turn off start-select-out is made, but because the 1050 is installed last, the Sel-I is routed through the 1051 attachment share-request controls. Here the Sel-O detect latch is turned on, and, after a short delay, the Sel-O delay latch is turned on. The original Sel-I line is now called 1050 Sel-I and will now enter the multiplexor controls and turn off the start-select-out latch.

POLL CONTROL--1050 REQUEST KEY OPERATION

- Request key causes a share-request.
- Sel-O is generated to the 1051 attachment by the select-request latch.
- 1051 attachment captures polling by stopping the Sel-O pulse.
- Attachment circuits bring up 1050 operational-in and T-request.

When the operator presses the request key on the 1050, the attention latch is set in the 1051 attachment (Figure 5-24). If the channel is not polling at this time, the prepare-to-share latch is turned on. The same line activates 1050 req-in. The sel-req latch is turned on in the MPX channel controls if the MPX channel is not busy with another device. The sel-out pulse to the 1051 attachment is generated when the select-request latch turns on. This pulse turns on the sel-out detect latch in the 1051 attachment, and at the same time the 1050 op-in latch is able to turn on because the prepare-to-share latch is on from the original request.

The 1050 Op-in latch does three things:

1. Prevents the turn-on of the Sel-O delay latch which captures the polling sequence.
2. Sends 1050 Op-in to the MPX controls, clamps the select-request latch off, and drops Sel-O. This prevents turning on the select-request latch until the 1050 Op-in line falls.
3. Sends T-request in to be used in the microprogram. At this time the microprogram will branch to the 1050 routine and further control will be by the microprogram. The share- request controls simply allowed a share cycle to be initiated and the 1050 to capture the polling sequence.

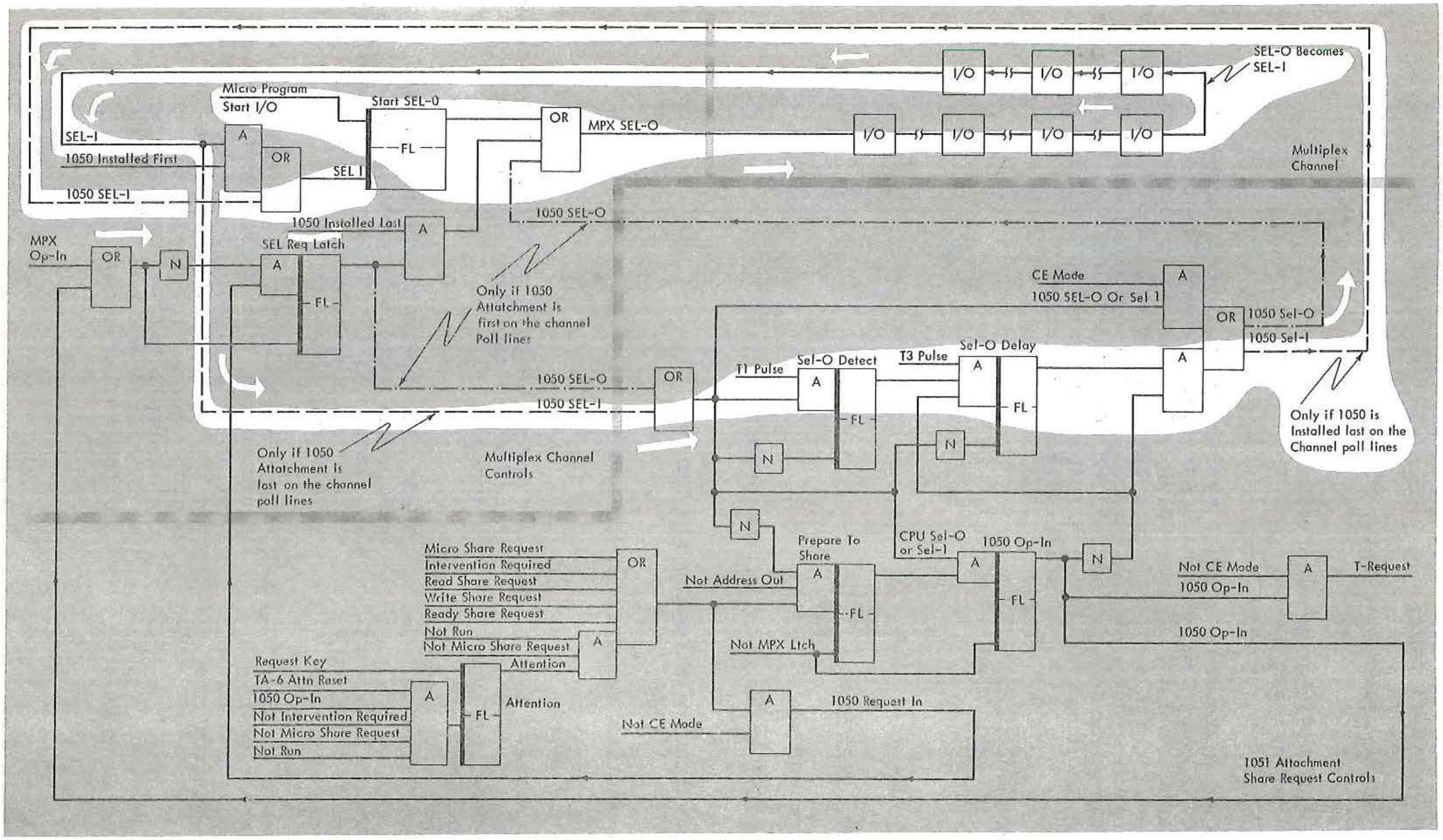


Figure 5-23. Start I/O To MPX Channel with 1050 Installed Last

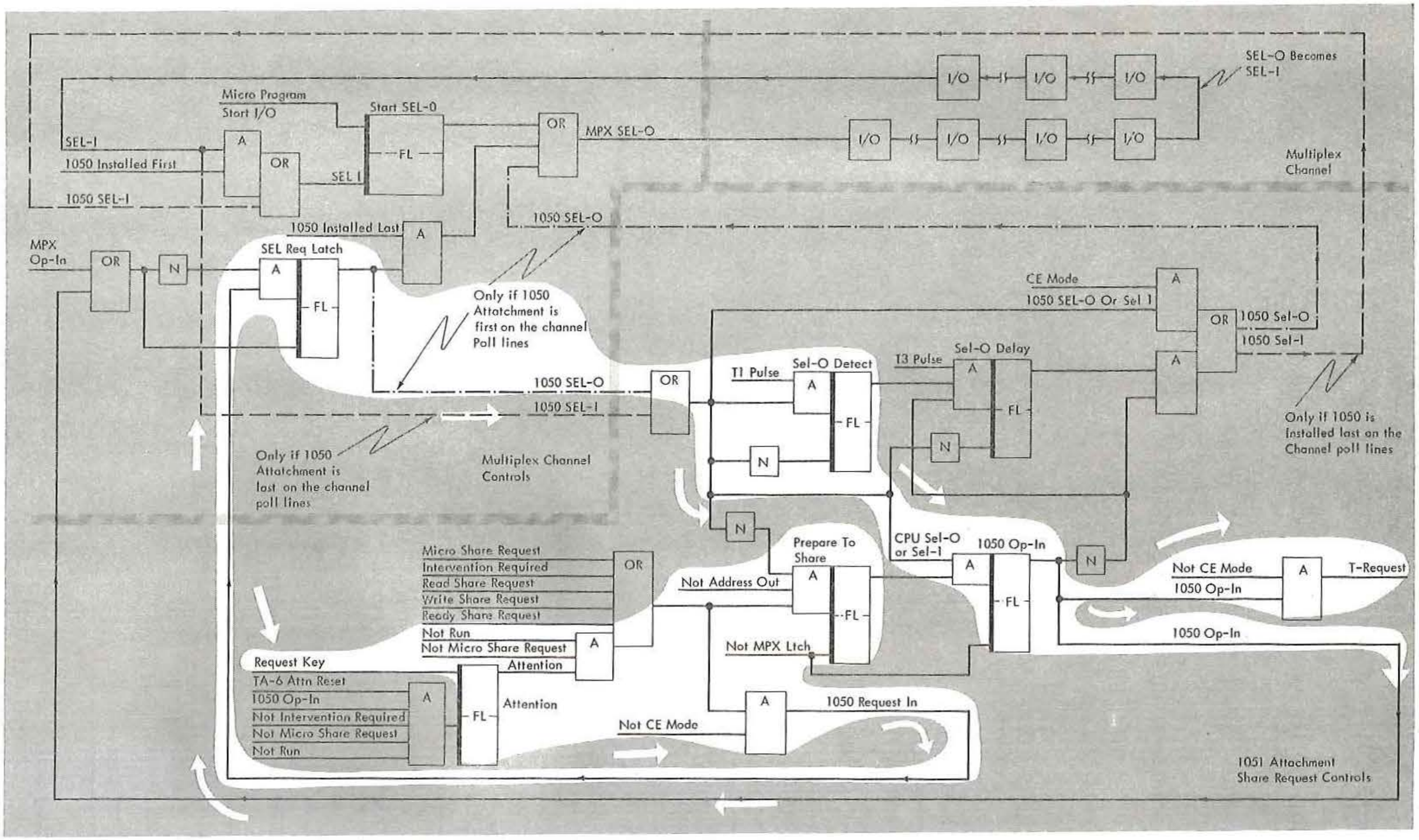


Figure 5-24. Poll Control--Share Request from 1050

IBM 1051 TO 2030 INTERFACE

- Paddle connector C03 (Punch-1) is used for the input to the CPU.
- Paddle connector A03 (Reader-1) is used for the CPU output operation.
- Paddle connector D06 is used to route the control lines to and from the CPU.
- EPO (Emergency Power Off) controls the 1050 power-on sequence.
- Cable will enter the 2030 at 01FA1.

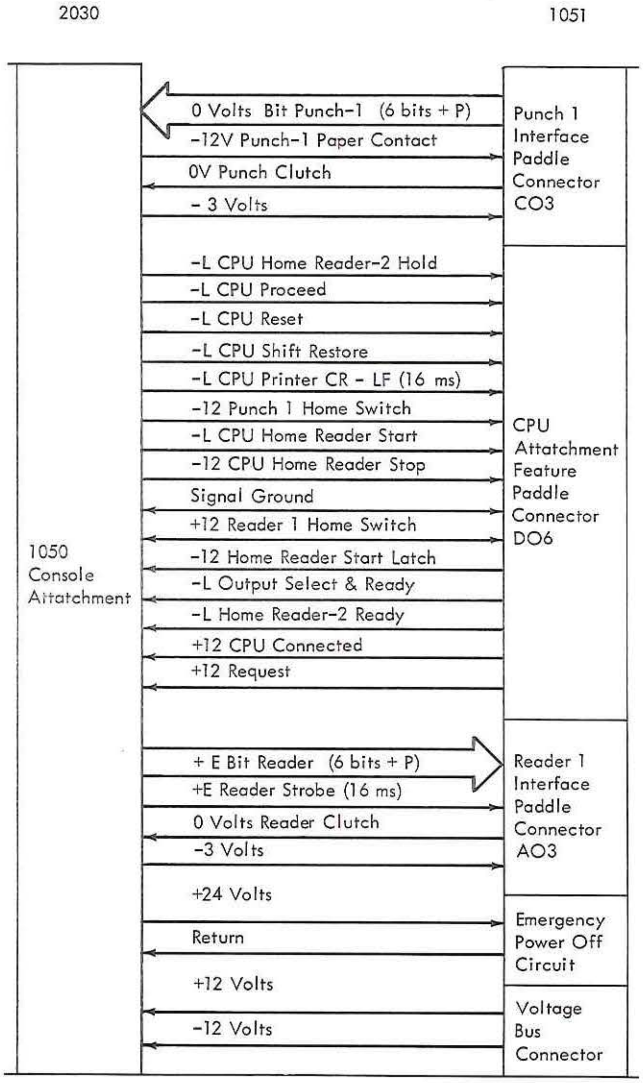


Figure 5-25. 2030 To 1050 Interface

The cable connecting the 1051 to the CPU will be referred to as the <u>interface</u>. This cable can be broken down into 4 basic sections (Figure 5-25):

1. Data to the CPU (Punch-1)
2. Data from the CPU (Reader-1)
3. Controls from 1051 to CPU, controls from CPU to 1051,
4. Voltage interchange and EPO.

Let's take each of the basic sections and explain the lines and their purpose.

READ (PUNCH 1) INTERFACE, FIGURE 5-26

This section of the interface controls the data from the 1051 to the CPU. Input in the 1051 can be either a keyboard or a reader.

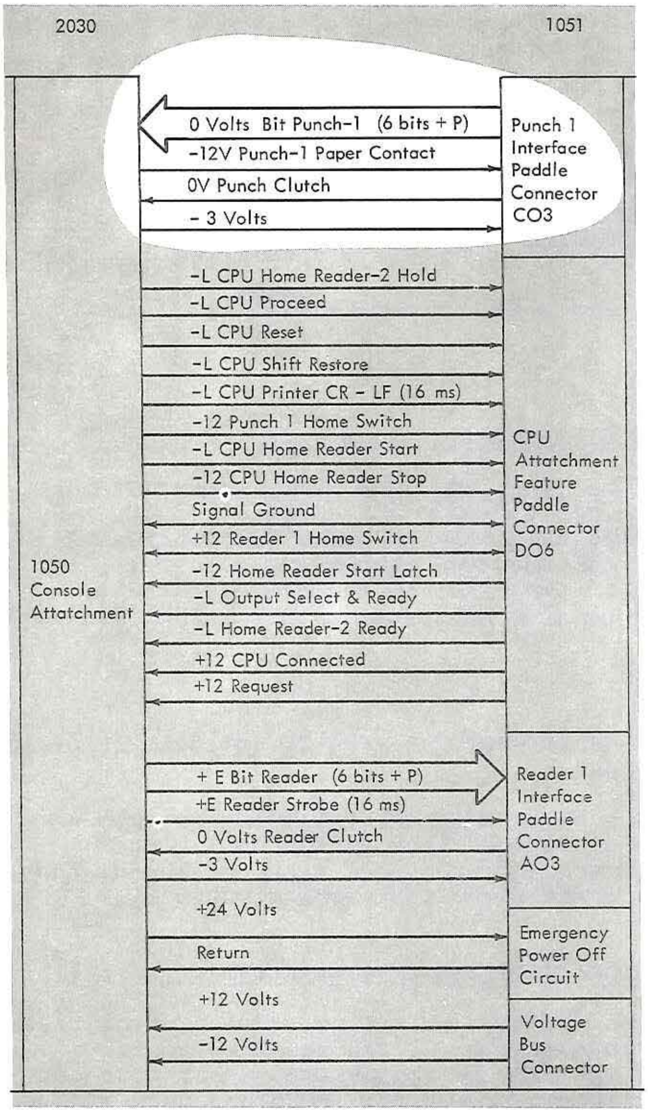


Figure 5-26. Read (PCH-1) Interface

0 Volts Bit Punch-1 (6 bits plus P)

These lines are reed-relay controlled in the 1051. If the character being sent has an A bit, the reed-relay controlling the A-bit line will transfer, indicating an A-bit on the interface. The common of all the bit relays is fed by -3 volts from the 2030 to the 1051 (Figure 5-27). The actual level of the interface signal is -3 volts for a bit and +6 volts for a no-bit condition.

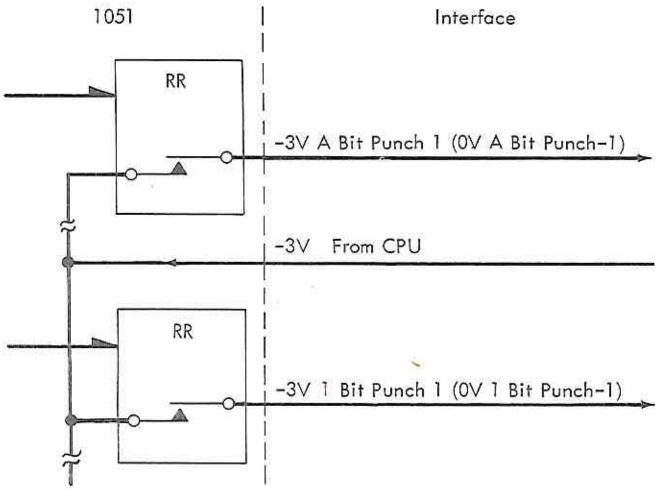

1051
Interface
RR
-3V A Bit Punch 1 (0V A Bit Punch-1)
-3V From CPU
RR
-3V 1 Bit Punch 1 (0V 1 Bit Punch-1)

Figure 5-27. Read Relay Control of PCH Bit Lines

-12V Punch-1 Paper Contact

This line is merely a return of -12 v to the 1051 through the interface. It satisfies the punch-clutch circuits and also indicates that the cable at the 1051 is properly installed.

0 Volts Punch Clutch

This line is reed-relay controlled in the same manner as the data-bit lines. It serves as a strobe signal to the CPU that a character is on the interface. The voltage levels on the interface are -3v on and +6 v off.

WRITE (READER 1) INTERFACE, FIGURE 5-28

Data sent from the CPU during a write operation will enter the 1051 home loop on this section of the interface. Once the data is on the home loop, any output device selected and ready will print or punch the information.

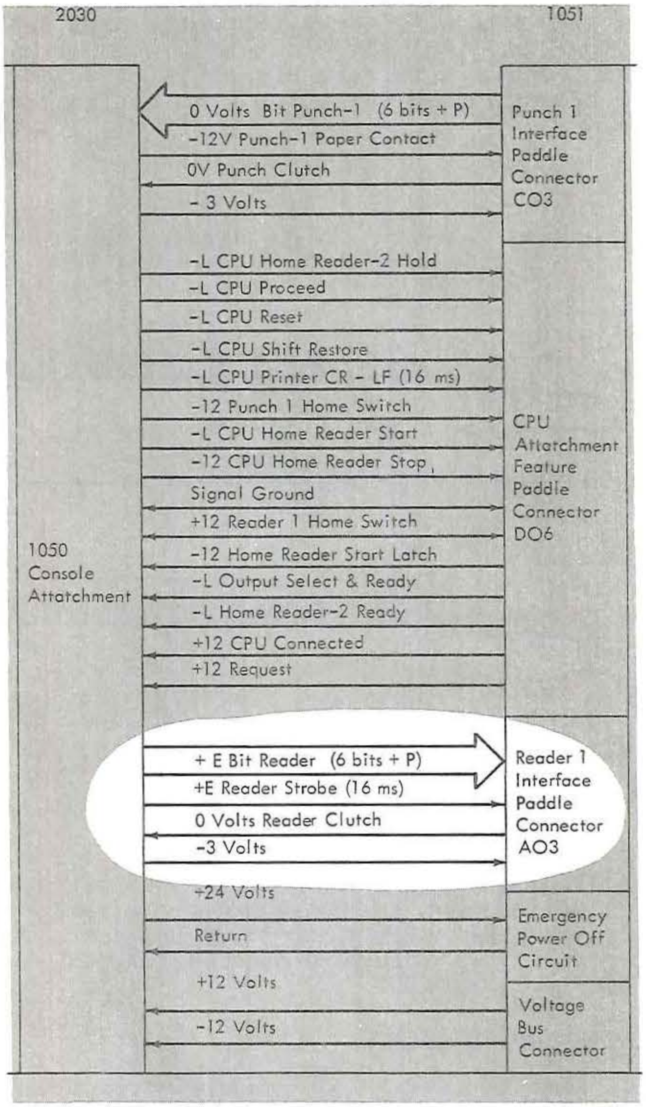


Figure 5-28. Write (RDR-1) Interface

0 Volts Reader Clutch

This line is a signal to the CPU that the reader-control circuits in the 1051 are satisfied. If the 1051 home loop is performing a function, such as CR-LF, reader clutch will be blocked, and the CPU will wait until the 1051 activates the read-clutch line again. The level of this line is -3v on and +6v off, and is controlled by a reed-relay the same as punch clutch.

+E Reader Strobe (Figure 5-29)

The CPU activates this line when it has a character to send to the 1051. The home timer in the 1051 is started by this timed pulse to initiate a cycle in the 1051. A reed-relay in the CPU switches +12v from the 1051 back to the 1051 as a +E for on and -E for off.

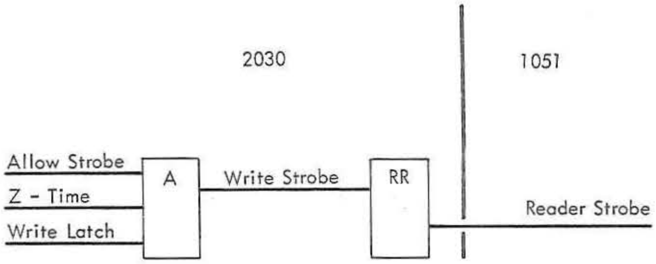


Figure 5-29. Reader Strobe

+E Bit Reader (6 bits + P)

These lines are reed-relay controlled in the CPU the same as reader strobe. The duration of the pulse on the interface is longer than for reader strobe. Once the 1051 home timer is started, the character is handled as a normal 1050 reader cycle.

CONTROLS FROM THE 2030 TO THE 1051 (FIGURE 5-30)

This section of the interface is used to send control lines to the 1051. The operation of the home loop on-line depends on these lines functioning properly.

-L CPU Home Reader-2 Hold (Figure 5-31)

When on (-L), this line prevents reader-2 from reading. It can be gated to the 1050 if the 1051 is operational and the reader-1 home line is active. Reader-2 hold will be on whenever the 1050 adapter in the CPU is in write mode. With the adapter in read mode, the line will be off until read share-request turns on. At this time the adapter has the character stored in the R/W register, waiting until the CPU can accept the data, and the reader-2 hold line will turn on to stop the reader at the 1050.

At the time the CPU writes the character in storage, reader-2 hold is turned off and the reader is allowed to send the next character to the CPU. (Figure 5-32) Using the reader-2 hold in this manner prevents the 1050 from running ahead of the CPU (sometimes referred to as overrun).

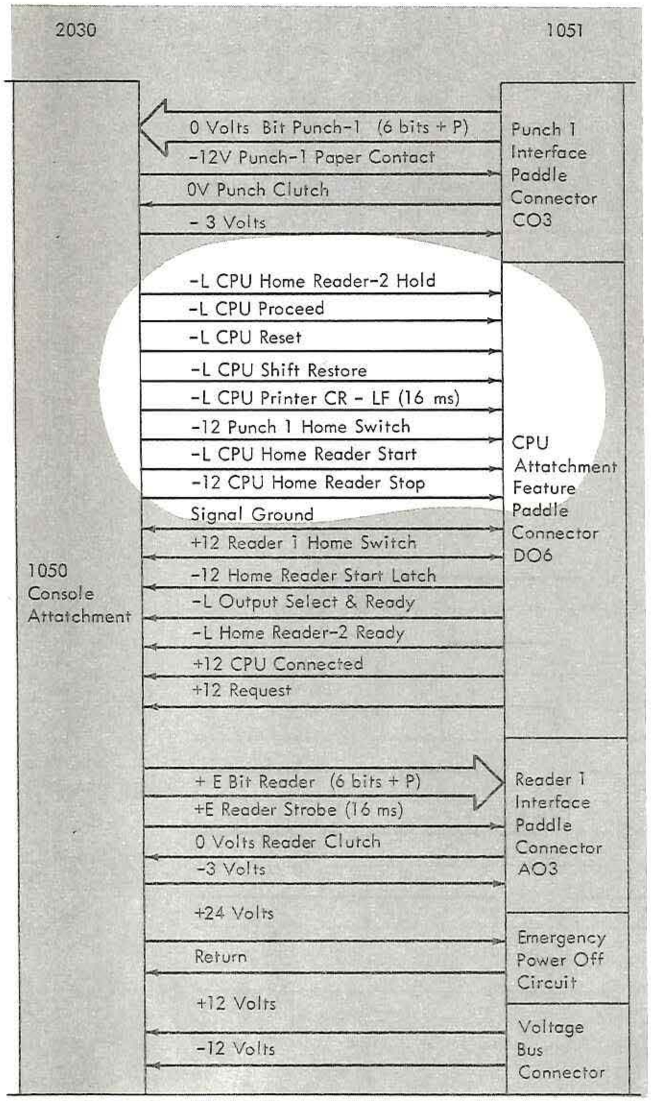


Figure 5-30. Controls from CPU to 1051

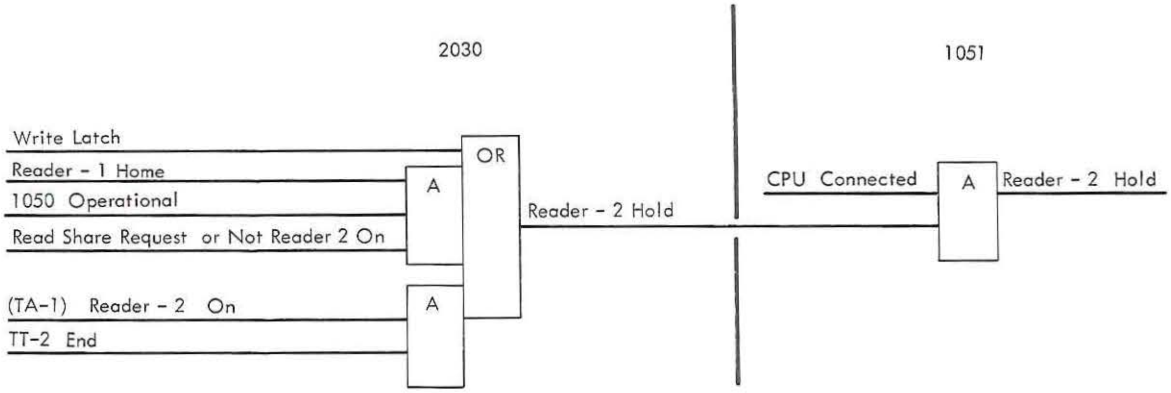
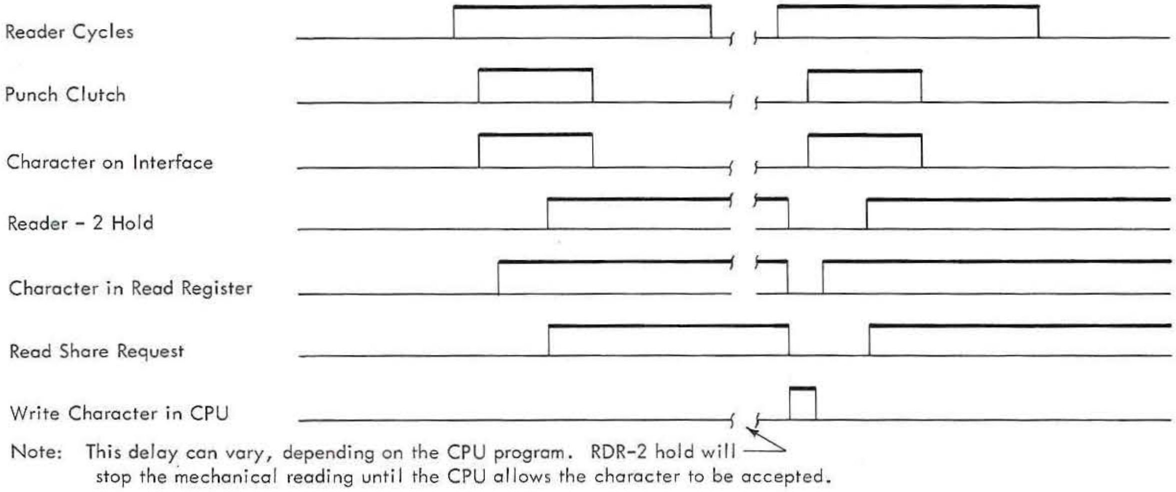


Figure 5-31. Reader 2 Hold



Note: This delay can vary, depending on the CPU program. RDR-2 hold will stop the mechanical reading until the CPU allows the character to be accepted.

Figure 5-32. Reader-2 Hold Timing Chart

-L CPU Proceed (Figure 5-33)

When off, this line causes the keyboard on the 1052 to become locked if it is selected on the home loop, if the 1050 is on-line and the home loop is selected. The 1052 keyboard unlocks and the proceed light comes on when the proceed line is on (-L). The CPU proceed line is controlled during read-inquiry mode by read share-request to provide data overrun protection when entering from the keyboard.

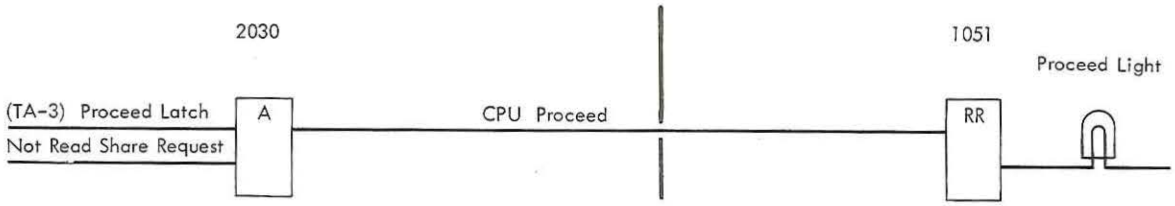


Figure 5-33. CPU Proceed

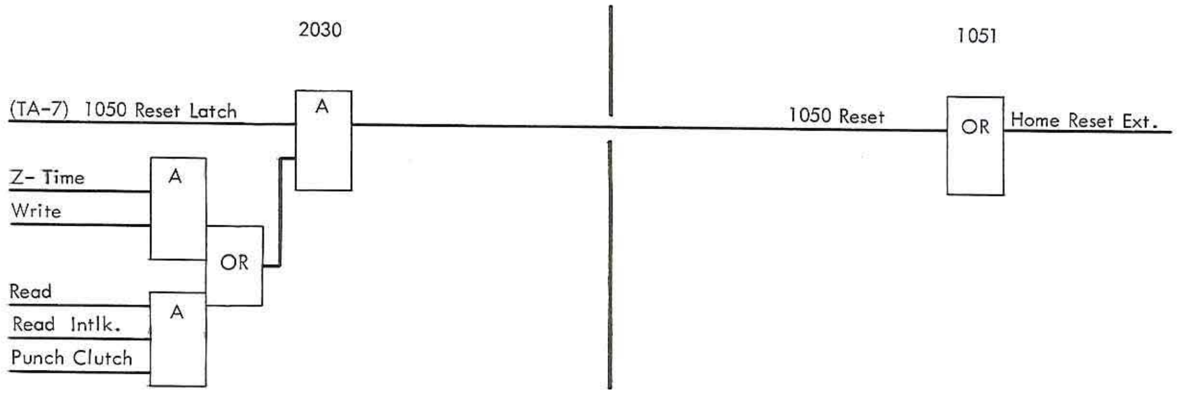


Figure 5-34. 1050 Reset

-L CPU Reset (Figure 5-34)

When on, this line resets the home loop on the N1:

1. At Z-time of a clock cycle in write mode.
2. At not-punch clutch time and read-clock interlock during read mode.

The preceding conditions force up the home-reset line in the 1051.

If the home-component recognition feature is installed in the 1051 and the 1050 home loop is on line to the 2030, CPU reset:

1. Resets Punch-2, Printer-1, and Printer-2, home- component-recognition latches.
2. Does Not Reset Reader 1, Punch 1, and Reader 2. The home-component-recognition latches are always force selected by the 1051 for on-line operation.

-L Restore (Figure 5-35)

When on, this line forces all 1050 printers to shift to lower case if the printer(s) are switched to the home loop. Another function of this line is to force the printer-1 home-component-recognition latch on if the 1051 has the home-component-recognition feature.

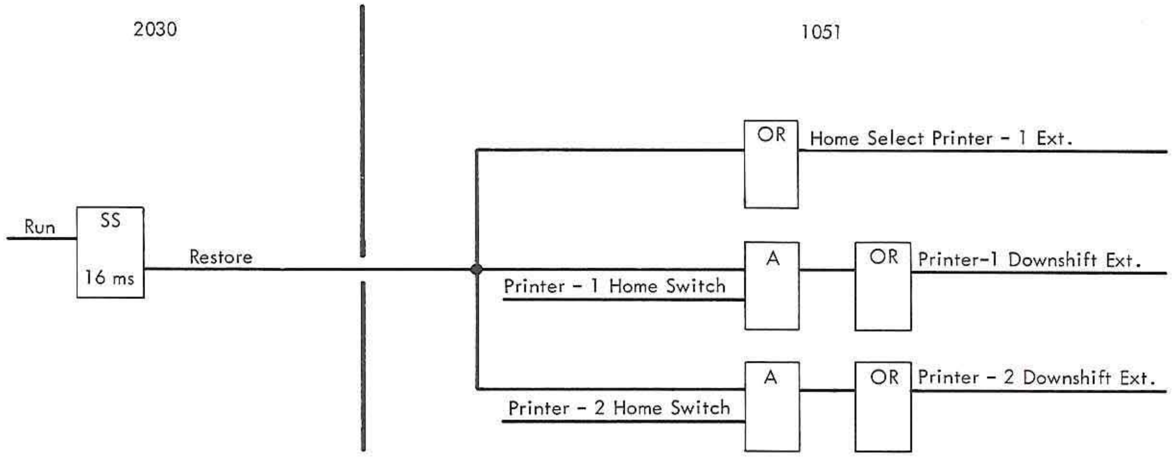


Figure 5-35. Restore

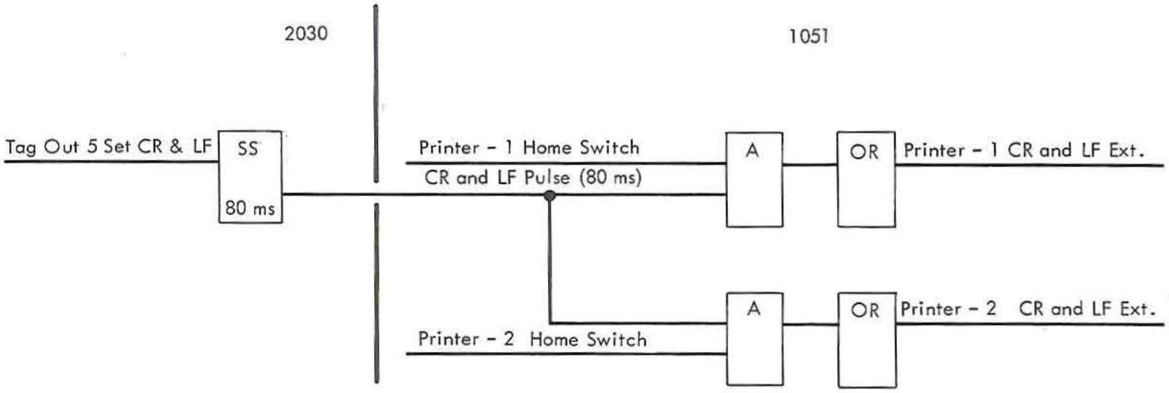


Figure 5-36. Carrier Return and Line Feed

-L CR and LF (80 ms Pulse) Figure 5-36

When on, this line forces a printer carrier-return and line-feed function to all 1050 printers if the printer(s) are switched to the home loop. Internal 1051 circuitry will drop the reader-clutch line to the CPU while performing a CR-LF function.

-12 Punch 1 Home Sw (Figure 5-37)

-12 volts from the 1051 is switched in the Mod 30 adapter via a reed-relay point. The reed-relay will be picked when the adapter is in a read operation. With the punch-1 switch utilized for CPU operation, this line activates the punch-1 home switch line to allow home-loop operation of punch-1 circuits.

-L CPU Home Reader Start (Figure 5-38)

When on, this line holds the home-reader-start latch in the 1051 in the On status. When the CPU is in write mode to the 1050, this line turns on the home-reader-start latch automatically.

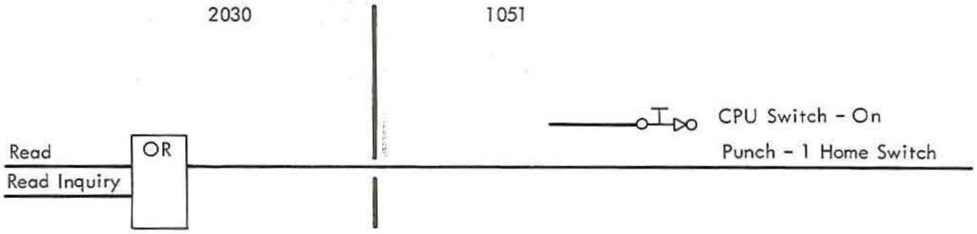


Figure 5-37. PCH-1 Home Switch

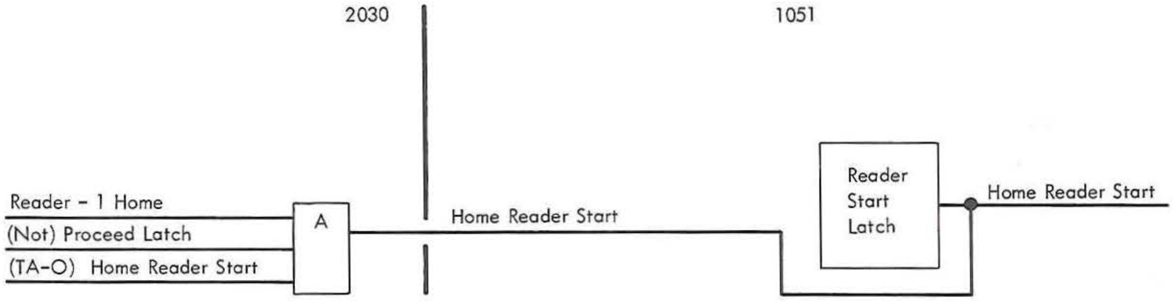


Figure 5-38. Home Reader Start

-12 CPU Home Reader Stop (Figure 5-39)

-12 volts from the 1051 is switched via a reed-relay in the CPU. This line turns the 1051 home-reader-start latch off if the reader-1 switch is set to the Home position and any time the attachment is not in run mode.

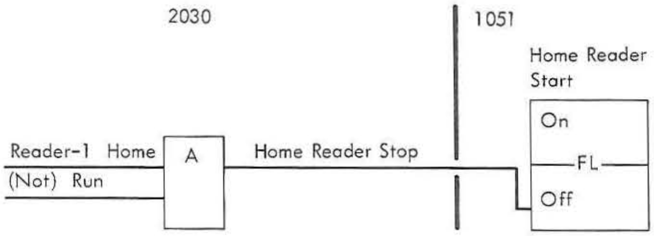


Figure 5-39. Home Reader Stop

CONTROLS FROM THE 1051 TO THE 2030 (FIGURE 5-40)

These lines are used to send controls to the CPU and are generated by the 1051.

+12 Reader 1 Home

The reader-1 switch is removed from the 1050 when the CPU attachment is installed. Therefore, the CPU connect switch activates this line when on line to the CPU.

+12 CPU Connected (Figure 5-41)

This line is controlled by a reed-relay at the 1051. When on, this line indicates:

1. The CPU connect switch is in the On position.

2. The +12, -12, and +48 voltages are available in the 1051.

3. The attend-unattend switch is in the attend position (1050 Mod 1 only).

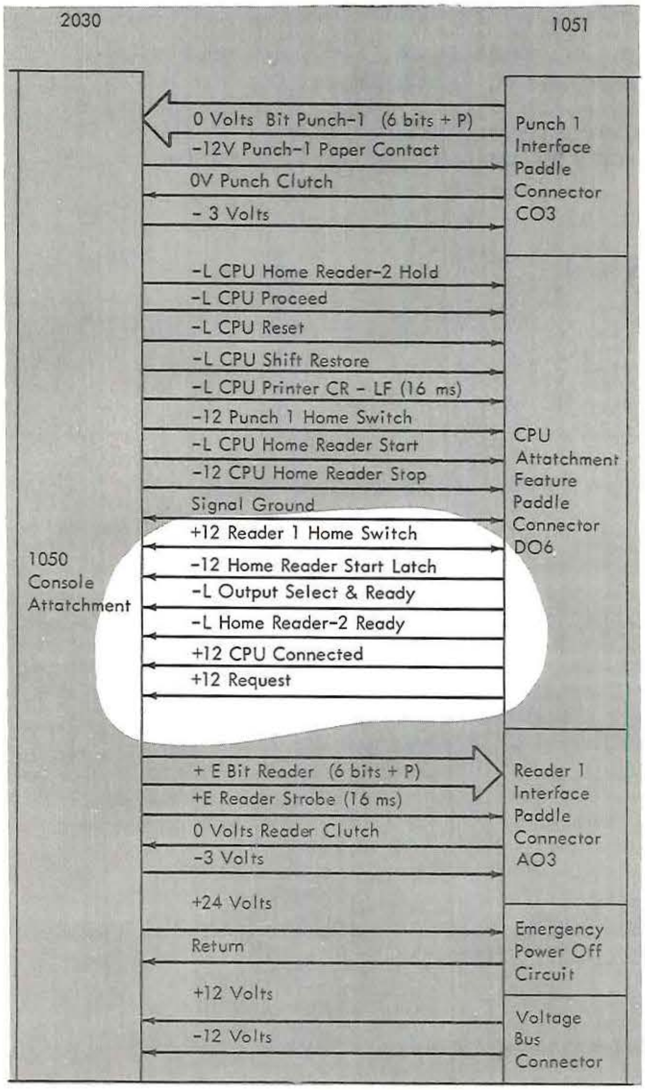


Figure 5-40. Controls from 1051 to CPU

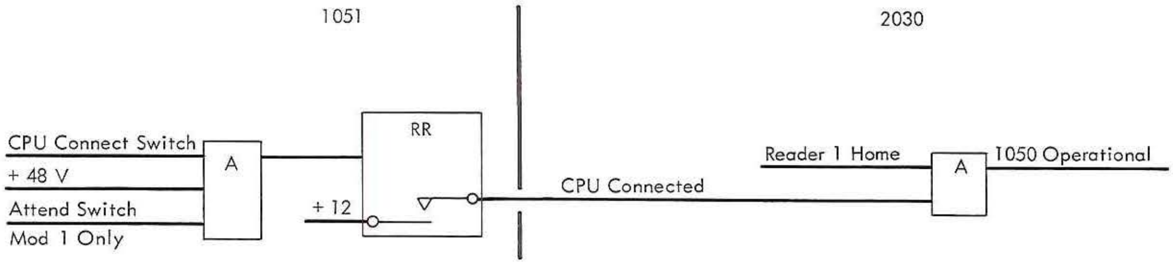


Figure 5-41. CPU Connected

With these conditions, the attachment uses reader-1 home line with this line to indicate the 1050 is operational.

-L Home Reader Start Latch (Figure 5-42)

When on, this line indicates the home-reader-start latch is on in the 1051. It allows the CPU to identify a keyboard or reader-2 entry during a read inquiry operation by testing TT-4 position.

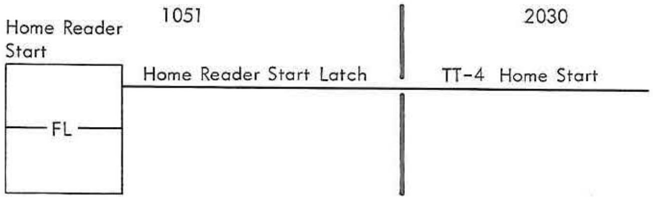


Figure 5-42. Home Reader Start

-L Home Reader-2 Ready (Figure 5-43)

When on, this line indicates that reader-2 has been switched to the home loop and its paper interlock has been satisfied. The TT-1 line is activated by this line.

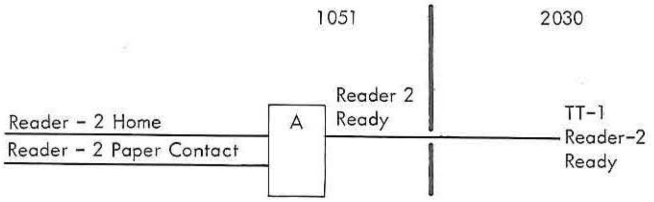


Figure 5-43. Reader-2 Ready

-L Output Select and Ready (Figure 5-44)

This line will be active under these conditions:

1. Without Home Component Recognition Feature. The line will be active if one or more output devices are switched to the home loop and their paper interlocks are satisfied.

2. With Home Component Recognition. The program-dup switch is in the dup position as in 1. The program-dup switch is in the program position. The line will be active if one or more output devices are switched to the home loop, paper interlocks are satisfied, and their home-component-recognition latch is on.

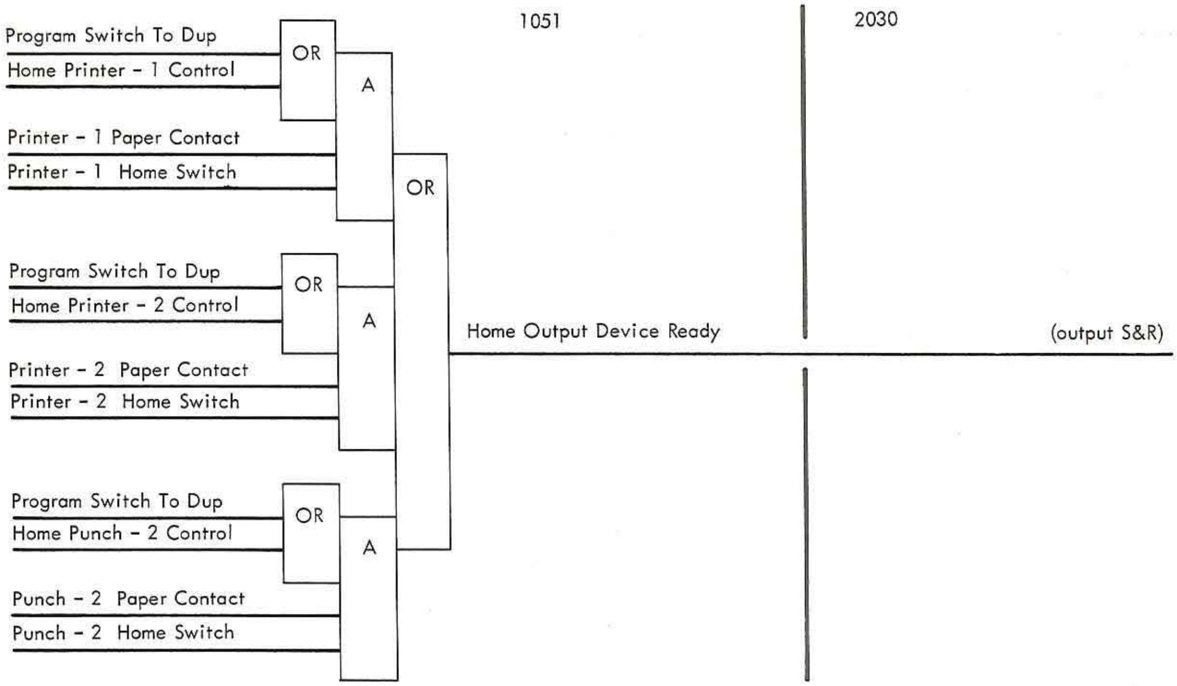


Figure 5-44. Output Select and Ready

+12 Request

When on, this line indicates that the operator has pressed the request key and wishes to send data to the CPU via the 1050. The attention latch will be set by this operation.

Figure 5-45 shows controls that interlock within the 1050 CPU attachment only.

EPO (EMERGENCY POWER OFF) AND DC VOLTAGE EXCHANGE

The EPO interlock provides (Figure 5-46):

1. Normal 1050 off-line operation with normal CPU power-off condition.

2. Controls to prevent the 1050 operation when the CPU is in EPO condition.

3. Operation of the 1050 off line when the CPU is in EPO if a CE is present to make the necessary change (flip the 1050 switch to off-line).

The 24 volts dc, +12, and -12 volts enter the interface via separate connectors and cable.

The EPO cable plugs into the 1051 end panel (Figure 5-47). Also mounted on this panel is the power control switch for on-line/off-line mode, and the HD-1 relay used for EPO.

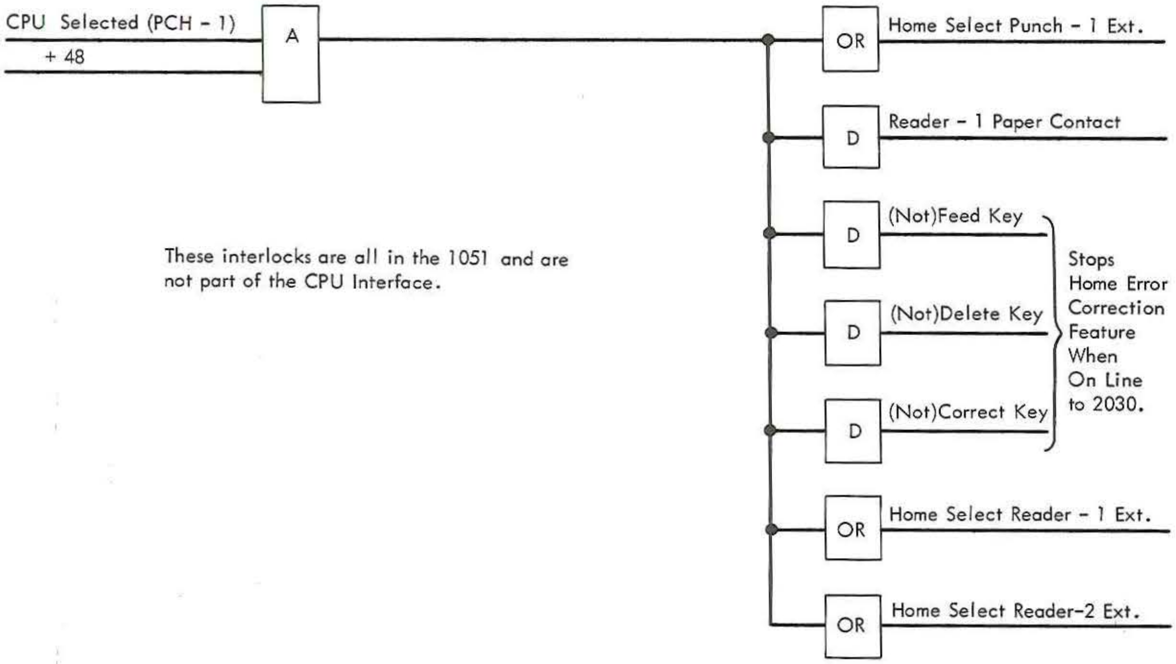
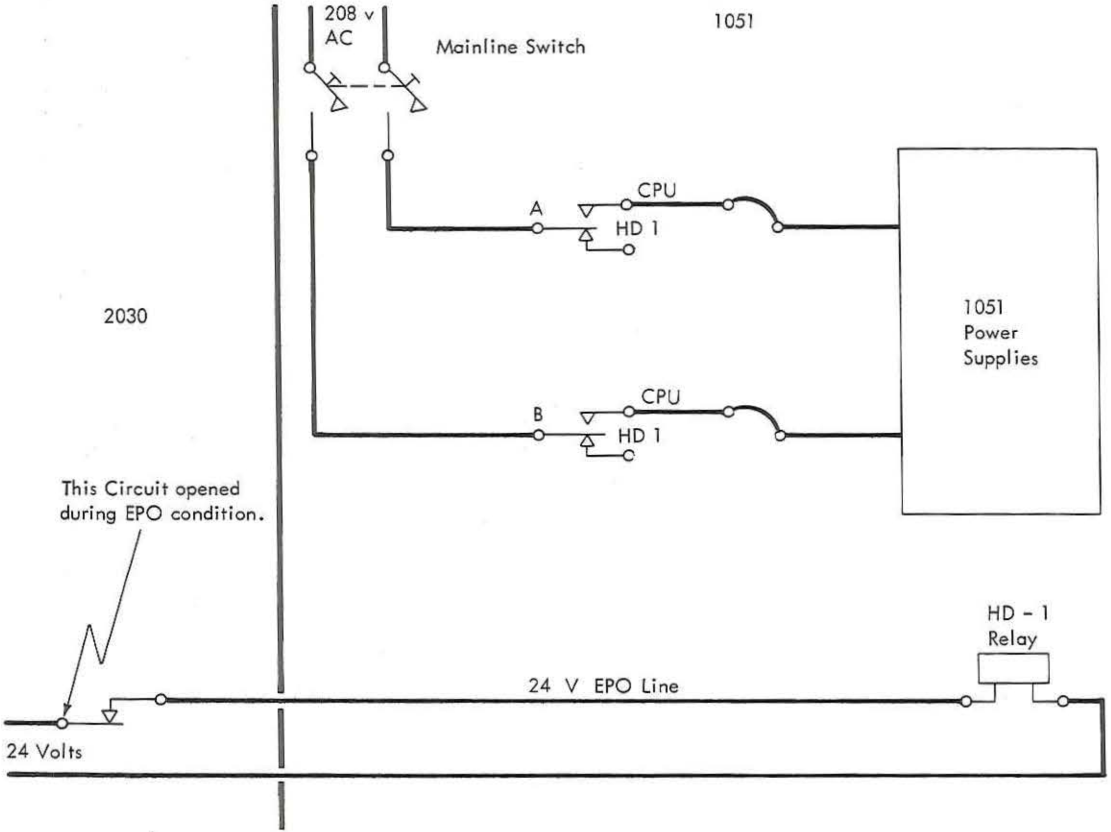
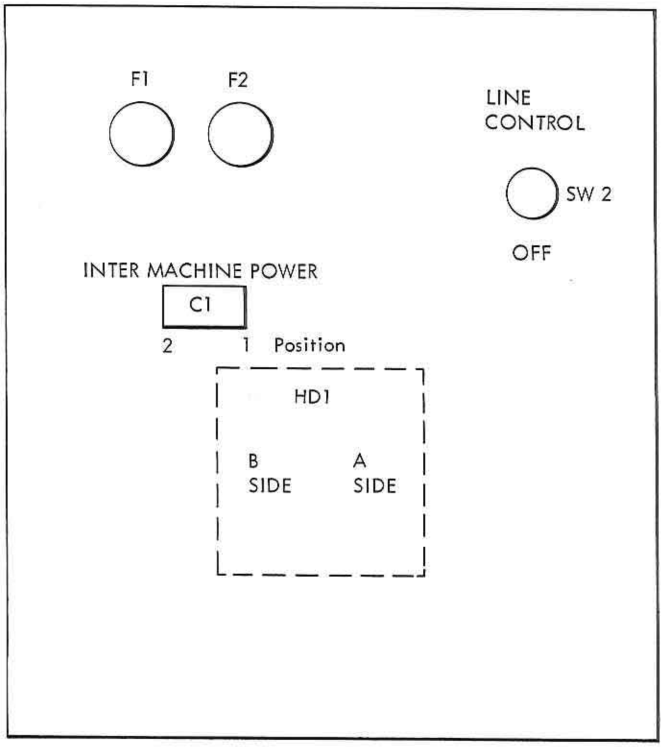

CPU Selected (PCH - 1)
+ 48
A
OR
Home Select Punch - 1 Ext.
D
Reader - 1 Paper Contact
D
(Not)Feed Key
D
(Not)Delete Key
D
(Not)Correct Key
Stops Home Error Correction Feature When On Line to 2030.
OR
Home Select Reader - 1 Ext.
OR
Home Select Reader-2 Ext.
These interlocks are all in the 1051 and are not part of the CPU Interface.

Figure 5-45. CPU Attachment Interlocks



● Figure 5-46. EPO Circuit



● Figure 5-47. 1051 Fuse and EPO Panel

PARITY CHECKING

- Data is parity-checked during read operations only.
- PTTC/8 bits are checked for odd parity.
- Parity checking is blocked during:
  1. Cancel decode
  2. Write latch on
  3. Not run condition.

The PTTC/8 bits are always parity-checked within the data parity-check circuits. In order for the parity-check circuits to activate TT-7 (data check) the bit-count has to be even and the attachment has to be in run mode, not TT-0 (cancel decode), and not write latch on. There is not any latch to remember the data check; therefore it is up to the microprogram to test the TT lines and sample the data- check line each data-transfer cycle.

PARITY BIT GENERATORS

- The PT&T/8 C-bit is generated to maintain odd count.
- The EBCDI P-bit is generated to maintain odd count.

It would be rather difficult and very expensive to maintain the PT&T/8 C-bit and the EBCDI P-bit through normal translation. Therefore, the odd-bit parity needed for both codes is generated by exclusive OR circuits that count the bits in the R/W-reg and insert a parity bit when needed.

MICROPROGRAMS

START I/O MICROPROGRAM

- The 1050 has a fixed multiplexor channel UCW address of 31.
- The 1050 unit status byte is stored in the local storage location K-9 (hexadecimal address 99)
- Commands that can be executed via the start I/O are:

  Read inquiry

  Read Reader-2

  Write

  Write with Auto New Line

  Control No-op

  Sense

  Control Alarm
- The 1050 start I/O shares much of the multiplexor channel microprogramming.

Consider the start I/O operation and its application with the 1050 attachment. Because the 1050 console application is an I/O operation, the CPU should function as a channel when sending to or receiving from the 1050. Although not actually part of the multiplexor channel, the attachment circuits simulate a multiplexor channel. Enough controls are activated to force the CPU to channel mode. The multiplexor channel start I/O microprogram is shared with the 1050. This microprogram is used until it is determined that the unit to be selected is the 1050. The program knows this by the unit address. The 1050 has a fixed address of 31, and once this is determined, the start I/O routine breaks out into the 1050 routine.

When the 1050 routine is finished, the restore CPU and other steps will again be shared with the multiplexor channel programs. One significant fact that should be covered: The 1050 microprogram during start I/O does not make use of the polling circuits to cause selection. The fact that the UCW address is 31 cause as a 1050 break-out. Then the command-start latch coming on prevents any request cycles being generated by the multiplexor channel.

1050 UCW Format

The byte format of the 1050 UCW is the same as the multiplexor channel. However, the bit representation of the channel status byte and the flag and ops byte are different

Channel Status	Op. and Flags	Count	
		High	Low

Data Address		Next CCW Address	
High	Low	High	Low

Bit position of UCW status

0	-	Zero count
1	-	Channel control check
2	-	Not used
3*	-	X
4*	-	Y
5	-	Wrong length record
6	-	Program check
7	-	Protect check

* bits 3 and 4 are used together

X Y
1 0 = Interrupt in unit
1 1 = Interrupt in buffer

Bits 3 and 4 of the UCW status byte are used differently than in a standard start I/O. If bit-3 is set, an interrupt is in the unit. If both are set, an interrupt is in the buffer.

UCW OPS AND FLAGS

Bit Position

0 - Chain data
1 - Command chaining
2 - SLI
3 - Store ACR (Write), Skip (Read)
4 - PCI
5 - Active
6 - { Bits 6 and 7
of the CCW
7 - command byte

The Bit-3 position of the ops and flag byte is used to store an automatic carriage return at the end of a write operation, as well as skip flag during read. The CCW command byte bits 6 and 7 will be stored in bits 6 and 7 of the ops and flags byte to indicate what the operation was when the UCW is used during share cycles.

Unit Status

The unit status byte is stored in the K addressable portion of local storage. Its mnemonic designation is K9, and the actual hexadecimal address is 99. Listed are the bit position, its name, and the purpose of the bit when set to a "1."

0 = Attention. This bit represents a request issued to the CPU from the 1050, and that the read-inquiry operation should be started.

1 = Device End Search. This bit is on when the CPU has reached a zero-count during a read operation, and the 1050 still has not sent an EOB or EOT. This allows the 1050 to keep running until the EOB or EOT is sensed. No data transfers are taken during this time.

2 = Interrupt Stacked. This bit is set when any interrupt condition (device end, channel end, etc.) is sensed and the IB is full.

3 = Busy. Unit is busy.

4 = Channel End. The end conditions caused by the CPU or microprogramming set this bit (zero-count).

5 = Device End. These end conditions by the 1050 set this bit: EOB, EOT or cancel. A ready- cycle will also set device end.

6 - Unit Check. Error in unit.

7 - Unit Exception. Cancel character sensed during a read inquiry operation sets this bit.

NOTE: Bits 1 and 2 are for 1050 microprogram usage only. They will never be presented to the CSW as such.

1050 Commands

The 1050 commands are represented by a single byte. The bit utilization for each command is shown in Figure 5-48. The microprogram branches on these command bits and assembles a byte on the A-bus with the micro order: Gate 1050 tags out. This instruction sets the correct TA register latches to perform the operations dictated by the command.

	CCW Command Byte							
COMMAND	0	1	2	3	4	5	6	7
Read Inquiry	0	0	0	0	1	0	1	0
Read Reader 2	0	0	0	0	0	0	1	0
Write	0	0	0	0	0	0	0	1
Write with ACR	0	0	0	0	1	0	0	1
No Operation	0	0	0	0	0	0	1	1
Sense	0	0	0	0	0	1	0	0
Control Alarm	0	0	0	0	1	0	1	1

Figure 5-48. 1050 Command Byte Configuration

READ INQUIRY: This operation is normally started at the 1050 by the operator pressing the request key which, in turn, initiates an attention interrupt. A read-inquiry command, via a start I/O, initiates a series of events that eventually unlock the keyboard and light the proceed light. Input can be split between the keyboard and reader-2.

READ READER-2: This command attempts to read from reader -2 through programming. No request key operation is necessary.

WRITE: This command can be used to send information to the 1050.

WRITE WITH ACR/LF (AUTOMATIC CARRIER RETURN/LINE FEED): This command is the same as write, except that at the end of the operation an automatic carrier return is sent to all printers selected. The ACR is stored in bit 3 of the UCW ops and flags byte.

SENSE: A command that forces a share-request cycle and tests the status of the sense byte. The sense byte is stored in LOCAL STORAGE at 04.

CONTROL ALARM: A command that sets TA-4 to cause an alarm to sound. This command can follow a write command to alert the operator that a message is waiting for him to read. It could also alert the operator that some device at the 1050 is not ready. The alarm feature can also be used by any other program to alert the operator for any reason.

CONTROL NO-OP: This command performs no usable function with the 1050 but is recognized as a legitimate command by its bit structure.

Read Start I/O-Example

- Branch to 1050 routine on UCW address 31.
- Test for:

  Busy
  Valid command
  1050 Operational
  Reader-2 Select and Ready
  Set TA Register 0 and 1 Positions
  Return to I cycles.

Refer to Figure 5-49. Let us take a read-start I/O operation through the flow chart to get familiar with its use.

The multiplexor channel start I/O will perform all the steps listed in the first large box on the chart. A check is made during these steps to find out if the UCW (subchannel) is active or not. If it is, the 1050 routine will not be entered, and condition code 2 will be set in the PSW. If the UCW is not active, the program will turn on command start and branch on S0 = 0 for a start I/O. The 1050 routine is now entered at 1 on the flow chart.

The following sequence will be used for no errors and 1050 units ready (assume no command chaining):

* Unit status Not Busy

* Command high Order Zeros

* Command Valid

Read or Write Command

* 1050 Is operational

Read operation

Reader-2 operation

* Reader-2 select and ready

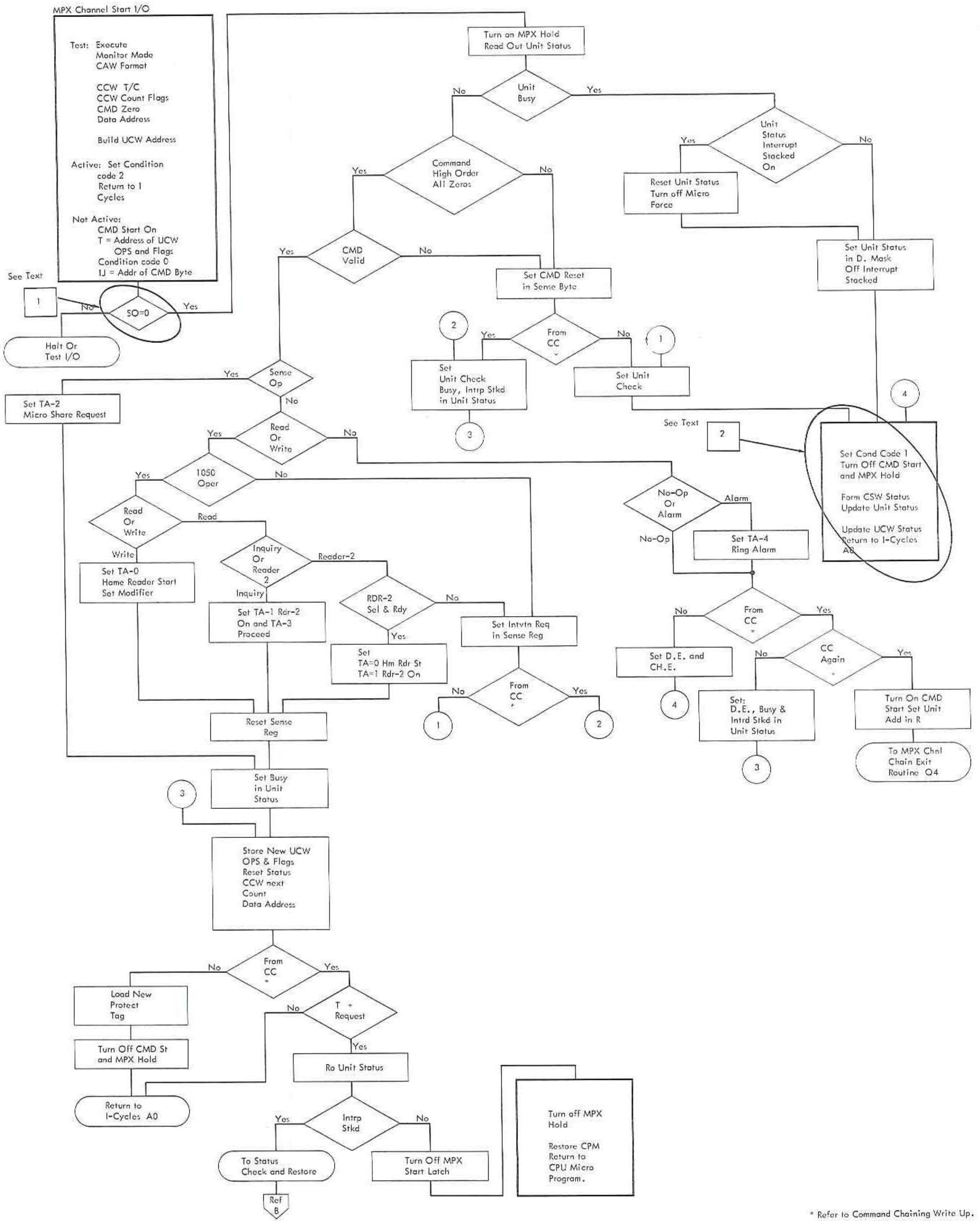
Set TA 0 and TA1.

Turn on busy in unit status

Not from CC

Return to I-cycles.

The branch steps indicated by an asterisk would all result in setting some status and then re-entering the multiplexor channel routine at 2 had there been an error. This would result in setting condition code 1 in the program status word and then return to I-cycles.



* Refer to Command Chaining Write Up.

Figure 5-49. Start I/O

SHARE-REQUEST MICROPROGRAM

- A microprogram routine used to control data transfer and status updating.
- The share-request microprogram will be entered by any one of the following:

  1. Micro share-request
  2. Intervention required
  3. Read share-request
  4. Write share-request
  5. Ready share-request
  6. Attention

After the routines of a microprogram initiating a start I/O operation, the share routine will control all data transfers and check and build status bytes as conditions change during the operation.

The six ways to interrupt the microprogram and activate the share routines follow. Each is used for a different purpose.

Micro Share-Request - TA-2. This interrupt is caused by the previous micro program routine setting TA-2 on. This will force a micro share-request cycle. This operation is used following a sense operation start I/O to get the sense byte on the A-channel and to keep the share-cycles going when a device end is stacked.

Intervention Required. This interrupt causes a share-request cycle to update the status bytes and return to I-cycles.

Read Share-Request. This interrupt is used to send data from the 1051 to the 2030.

Write Share-Request. This interrupt is used to enter the share routine to send data to the 1051.

Ready Share-Request. This interrupt causes a share-cycle when the 1051 is made ready after an intervention required or a power off situation. It forces an end operation and clears out the status to enable the program to start again.

Attention. This interrupt causes a share-cycle to form a CSW with the attention bit on. The attention bit can be interrupted and if the macroprogram wishes, a read inquiry can be initiated though a start I/O.

HALT I/O MICROPROGRAM

- Microprogram routine used to stop an I/O device.
- Halt, active, and interrupt-stacked or halt and available will result in condition code 0 being set.
- Halt and active result in condition Code 1 being set, CD and CC in the UCW flag ops being turned off and zero count in the UCW status turned on.

If for some reason the program wishes to stop an I/O device, the halt I/O instruction can be used. During the microprogram routine, the UCW is read out and the status of the I/O device is determined. The flow chart (Figure 5-50) shows the halt I/O routine. The unit can be in one of three conditions at the time the halt I/O was issued:

1. Available or not active

2. Active and interrupt-stacked

3. Active

The I/O device being available or active and interrupt-stacked results in a condition code 0 being set and a return to I-cycles. Condition code 0 can be set because the 1050 is either doing nothing or has finished and has attempted to interrupt but the interrupt was stacked. A halt I/O under these conditions does nothing because data transfer to the unit was stopped anyhow.

The 1050's being active when the halt I/O was issued results in stopping or halting the I/O operation. The UCW ops and flags byte CD and CC bits are reset, and zero count in the UCW status is turned on. Condition code 1 in the PSW is also set to alert the system that something went wrong with the last command.

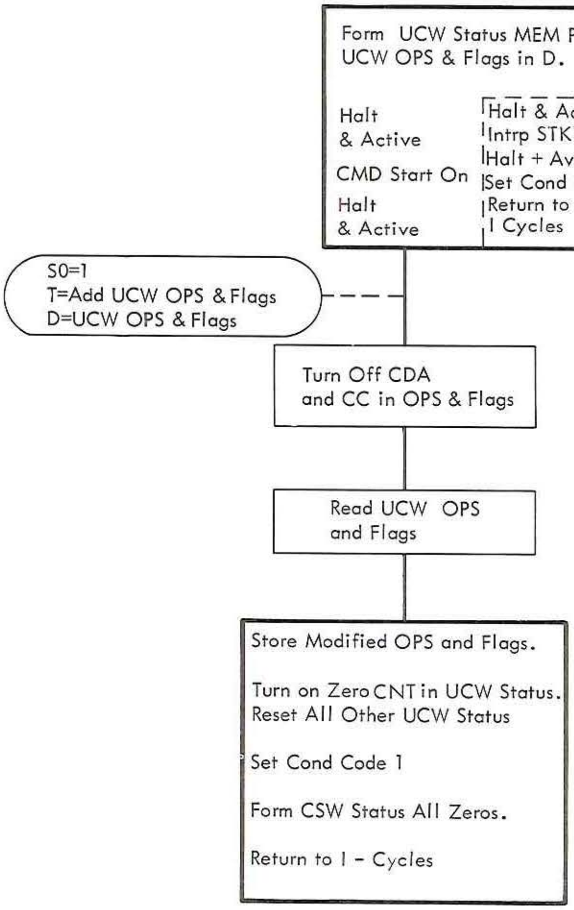


Figure 5-50. Halt I/O

TEST I/O MICROPROGRAM

- A microprogram routine used to test the status of an I/O device.
- Unit status 0 results in condition code 0 being set.
- Unit status non zero results in condition code 1 being set.
- Condition codes stored means:

  0 = Available
  1 = CSW stored
  2 = Channel or subchannel busy.

A test I/O command serves two functions: to test the unit status of any device and to be able to clear an interrupt in the buffer if a device is still active.

Figure 5-51 shows a logic flow of the test I/O to the 1050 attachment. If the attachment is active and no interrupt conditions are present, condition code 2 is set which indicates the unit is busy. Any interrupt condition can be acted upon and condition code 1 set, which indicates that a CSW has been stored.

If the unit is not active, the unit status is checked for a zero condition. If it is zero, a test is made for any pending interrupts, and if there are none, condition code 0 is set and the unit is available. Had the unit status been other than zero or a pending interrupt waiting, condition code 1 would be set indicating that the CSW had been stored.

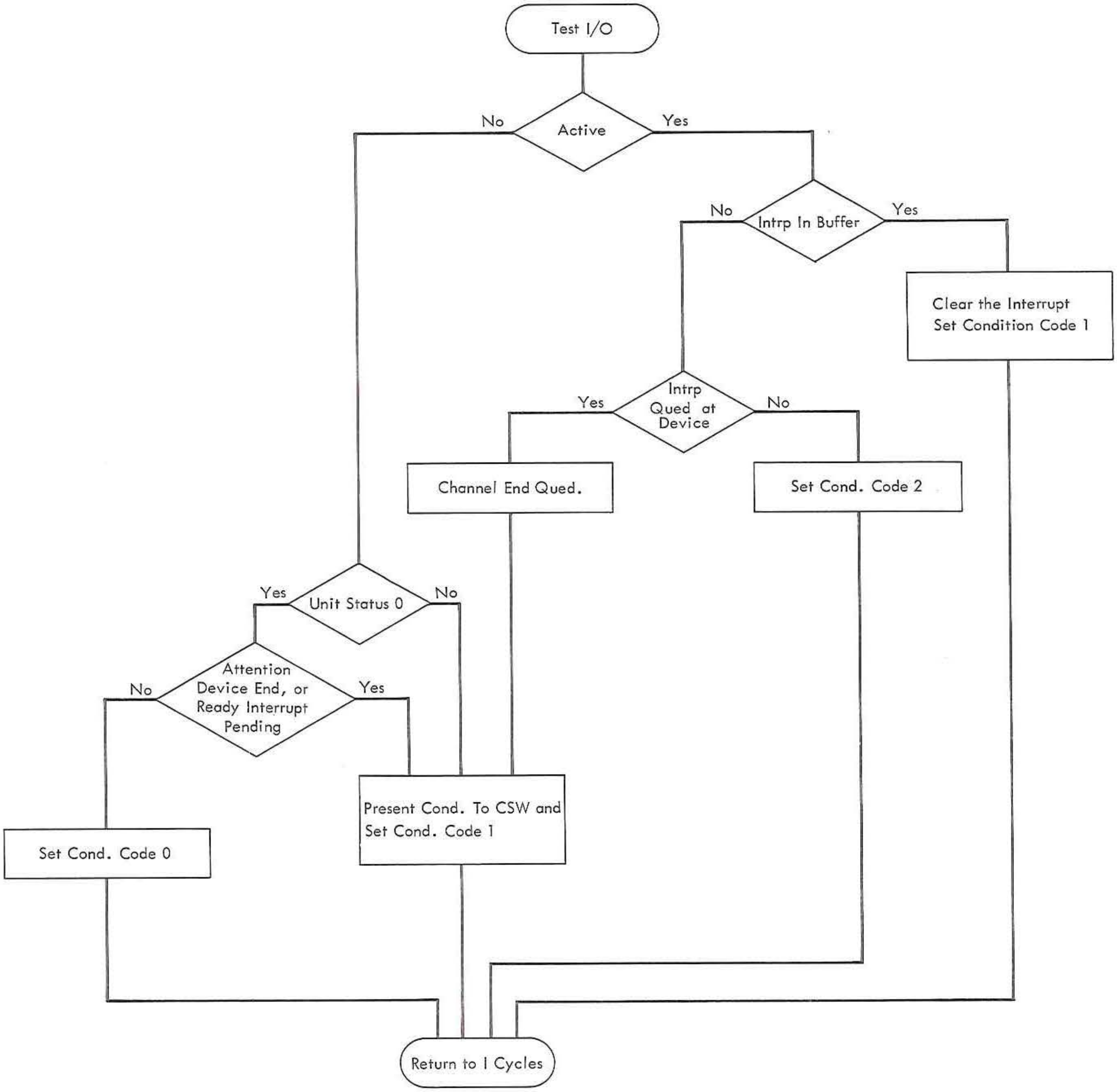


Figure 5-51. Test I/O

I/O INTERRUPT MICRO PROGRAM

- Interrupt conditions are checked at the end of E-phase of any instruction.
- Routine is used to update the CSW, to store the current PSW in the old I/O PSW location, and load new I/O PSW into current PSW.

The I/O interrupt microprogram reads out the unit status byte, updates the CSW, and stores the PSW for that operation as the old PSW. As soon as these operations are performed, a return to I-cycles is taken. See Figure 5-52 for a flow chart.

The I/O interrupt routine is entered at the end of any E-phase if the interrupt latch is on. The 1050 turns the interrupt latch on as a result of ending the previous operation or an attention or ready operation.

WRITE OPERATION

WRITE MICROPROGRAMS

- Conditions checked during a write-start I/O are:

  1. Unit Not Busy
  2. Valid Command
  3. 1050 Operational
  4. Write with ACR/LF Operation

- Conditions checked during write share-request cycles are:

  1. Unit Busy
  2. Intervention Required
  3. PCI
  4. Write operation
  5. Zero Count

- Status routine is entered by one of the following to end the operation:

  1. Any Error Sensed
  2. Zero Count

Write Start I/O

The primary objective of the write-start I/O is to set the attachment circuits to write mode and run condition. Once this is accomplished, the start I/O routine can be considered finished.

The flowchart in Figure 5-53 shows the path that a normal start I/O follows to initiate a write operation. Once it has been determined that a write operation is to be started, TA-0 is set, which is HM RDR start. This sets the attachment to write mode and starts the clock. The attachment operation is covered in Write Operation To 1051.

The next microprogram now handles the data transfer and is called the Share-Request Microprogram.

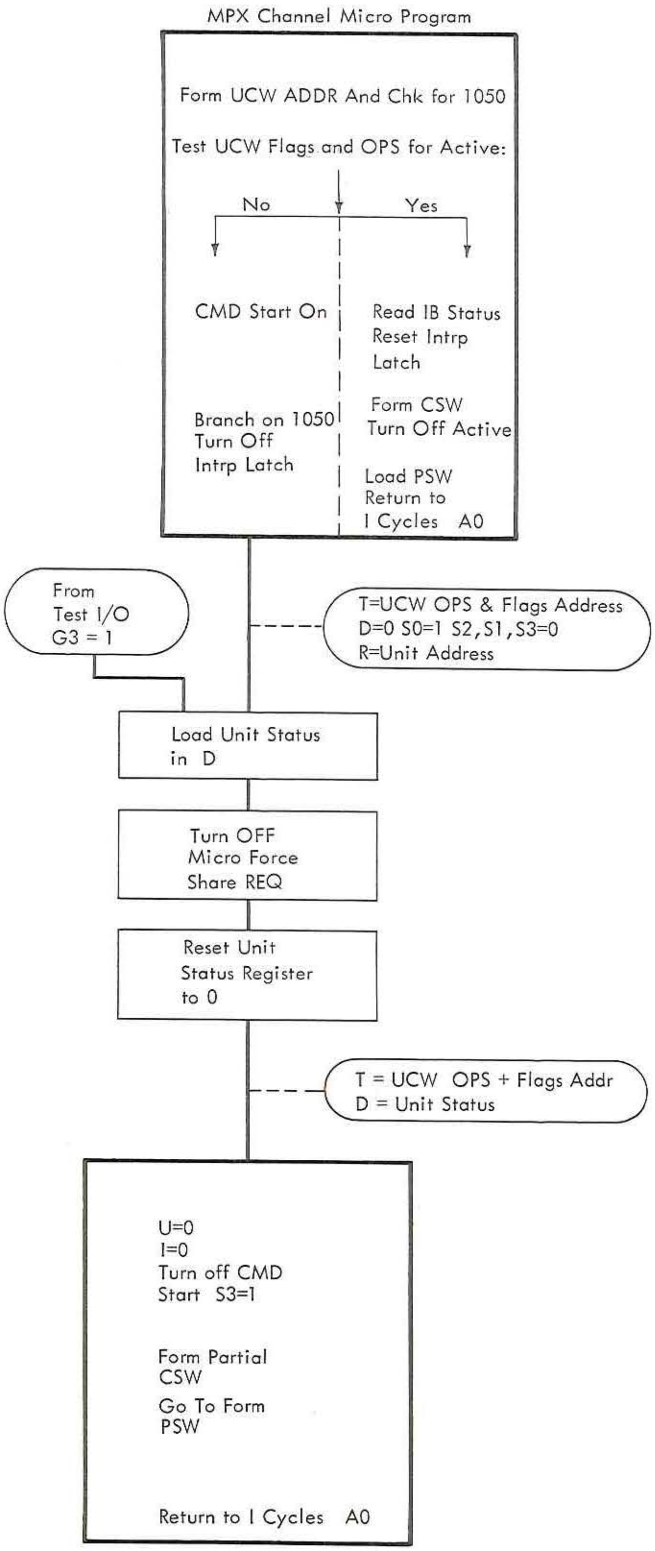


Figure 5-52. I/O Interrupt

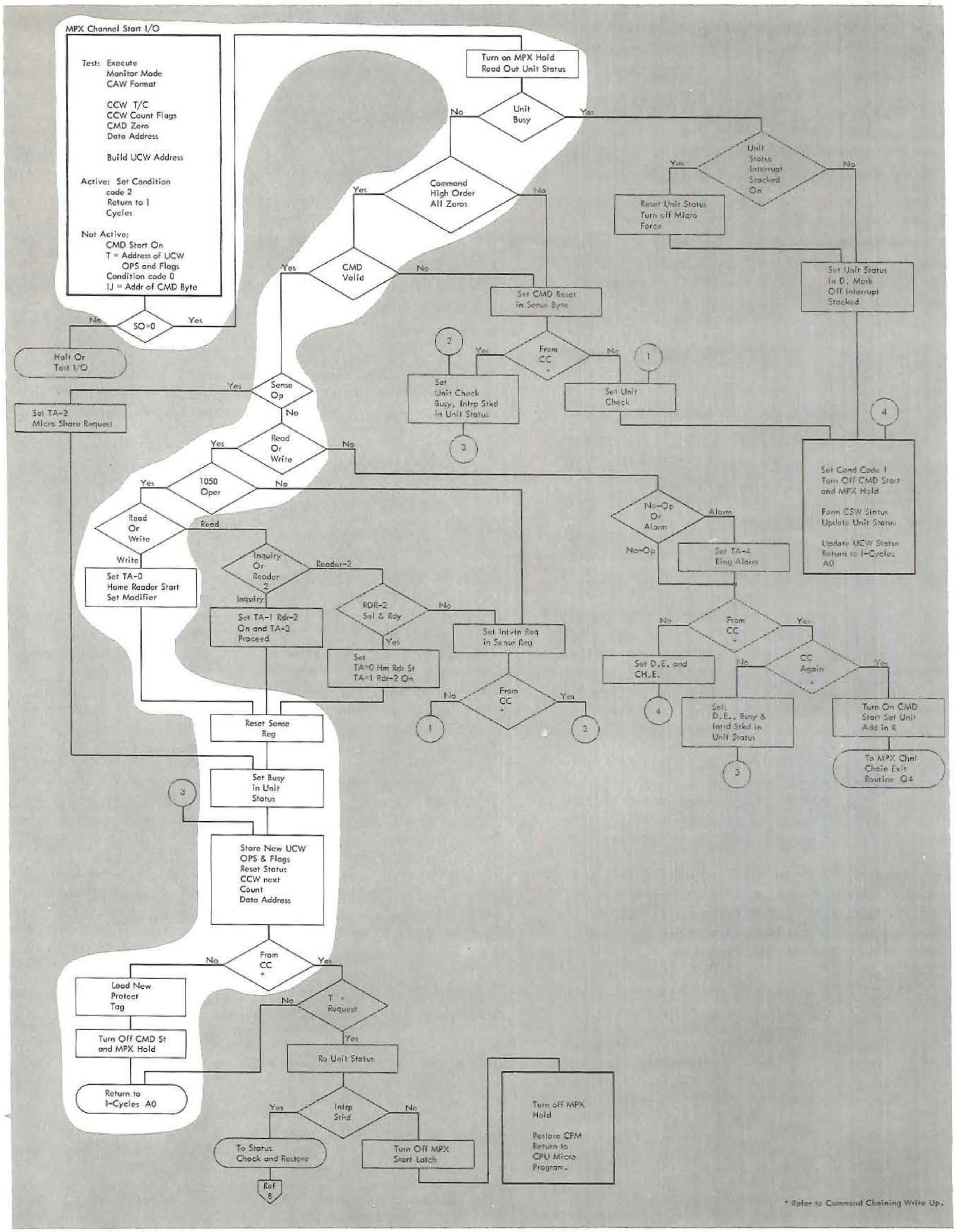


Figure 5-53. Write Start I/O

Write Share-Request

- Started by attachment sending in 1050 Request In
- Each data cycle increases the data-address and decrease the data-count.
- Normal end operation started by the data-count reaching zero.

The 1050 request-in line is recognized by the multiplexor channel controls. When the trap is allowed, the 1050 share-request routine will be entered. Refer to Figure 5-54 for the following sequence of events.

Let's assume that the following data is sent to the 1051:

ABc* *indicates CR character.
de

When entering the share-request routine, the following conditions are set:

Count in LD will be 8
Data address in UV

FIRST SHARE-REQUEST CYCLE:

1. Unit status busy (Arrow 1). This bit in the unit status indicates that the operation is a read, write, or sense.
2. Intervention required (Arrow 2). This test is made early in the program to determine if the 1050 and the attachment are still operational. The test is made on TT-5.
3. Device end search (Arrow 3). If for some reason this bit is on in the unit status, the read share-request routine will be entered. Normally, this bit should not come on for a write operation.
4. Zero count (Arrow 4). This test will be made each time the share-request routine is executed during a write operation. This cycle, the no-path will be taken, because it is a test on the data count.
5. The character is now written out to the attachment with a micro order: gate 1050 bus out. This is when the character is gated to the attachment read-write register.
6. The next two blocks will update the data address and decrease the count by 1. The count will now be 7 in LD.

SECOND THROUGH EIGHTH SHARE-REQUEST CYCLES: These will be the same as the first. At the end of the eight cycle, however, the zero-count bit in the UCW status is set (Bit 0).

NINTH SHARE-REQUEST CYCLE: This is the same until the branch at arrow 4 is reached. Here, the zero-count bit is found on, and the yes path is taken. We know that all of out data has been transferred, so our next objective is to stop the attachment clock and go out of write mode. The ROS order: gate 1050 tags out and the bus containing 0000 0001 turn off Home reader start latch and reset the 1050.

Also, a CR/LF operation would be initiated if bit 3 of UCW ops and flags is on. This will stop the attachment and reset the write latch. The microprogram now enters the status routine or end operation. The conditions set going into the status routine are device end (DE) and channel end (CH E) which are set as a result of the zero-count branch.

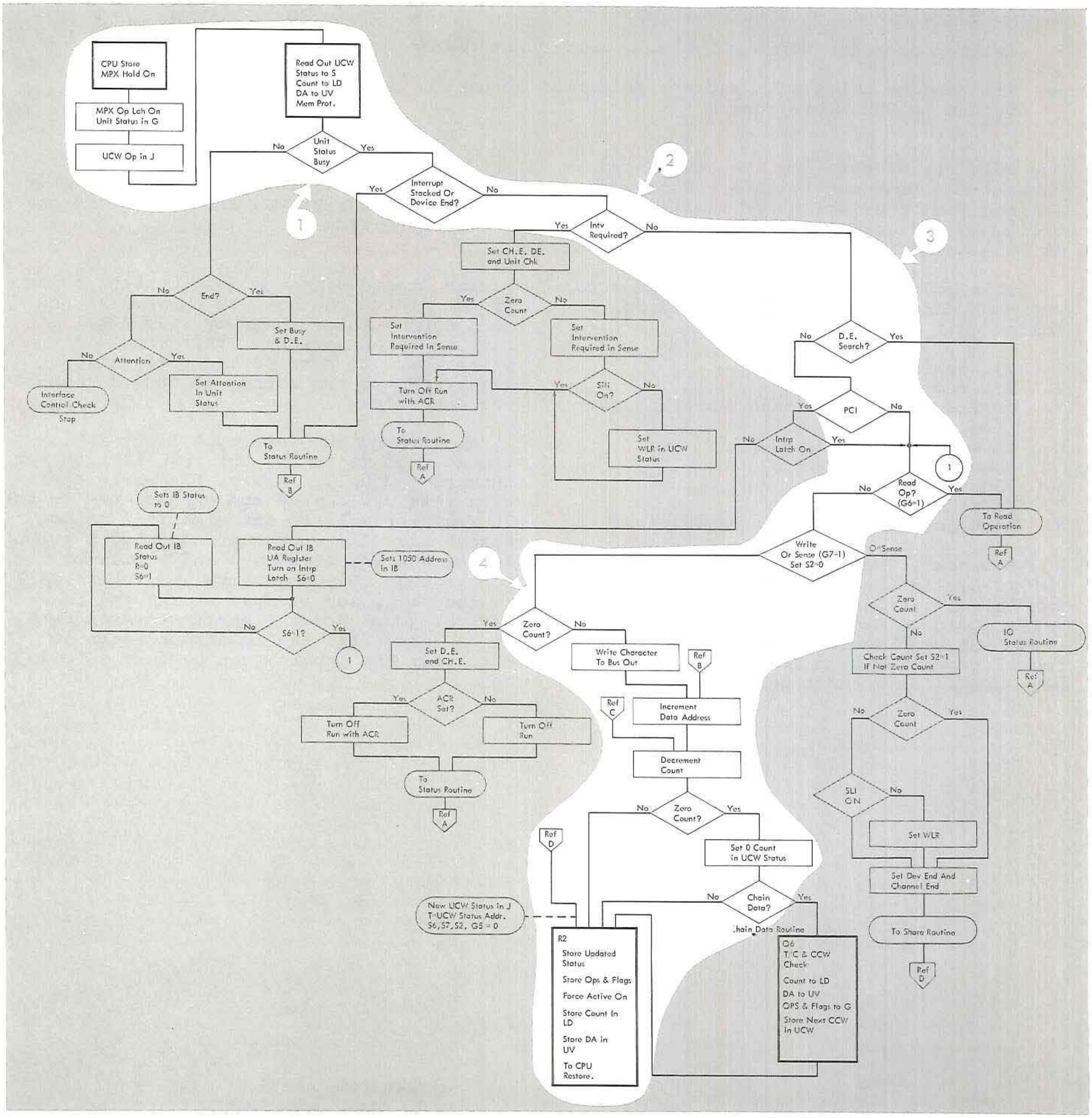


Figure 5-54. Write Share-Request

Write Status Check and Restore (End Operation)

- Entered from the write routine by zero-count conditions.
- Will update all status and set-interrupt conditions and return to CPU microprogram.

The unit status byte has the device-end and channel-end bits set (bits 4 and 5) when exiting from the last write share-request cycle. See Figure 5-55 for the flow chart of this operation.

Let's go through with the end operation and then discuss other ways of ending. Follow this sequence:

1. Not chained data
2. Not command chaining
3. Device-end was set upon entering routine.
4. Interrupt latch is off
5. Channel-end was set
6. Update status
7. Branch to MPX routine
8. Test for another share-request waiting

At arrow 1 the CC branch is tested. Had the CC-bit in the UCW ops and flags been set (bit 1), this branch would be taken. Tests would be made for not-error, not-attention, and device-end. This would put the microprogram in a CC-loop to start the next operation without waiting for the auto-interrupt routine and the normal I-cycle start I/O. The CC-operation simply reads out the next CCW and enters the start I-O routine from CC.

The branch at arrow 2 is testing to see if the status routine was entered as the result of a channel-end by itself by testing for device-end, unit check, or attention. If none of these conditions exist, assume the entry was from a channel-end operation.

At arrow 3 the interrupt latch is tested. If it is on, the next test is for channel-end only. Again, the write operation sets device-end and channel-end so the TA register will be set with TA-2 (micro force) on. The interrupt conditions set up in the UCW status bits 3 and 4 will be 10. Recall that this signifies the interrupt is in the unit, and because device-end is active, keeps forcing share-requests until it is found that the interrupt latch is off. Now the program can perform a normal end.

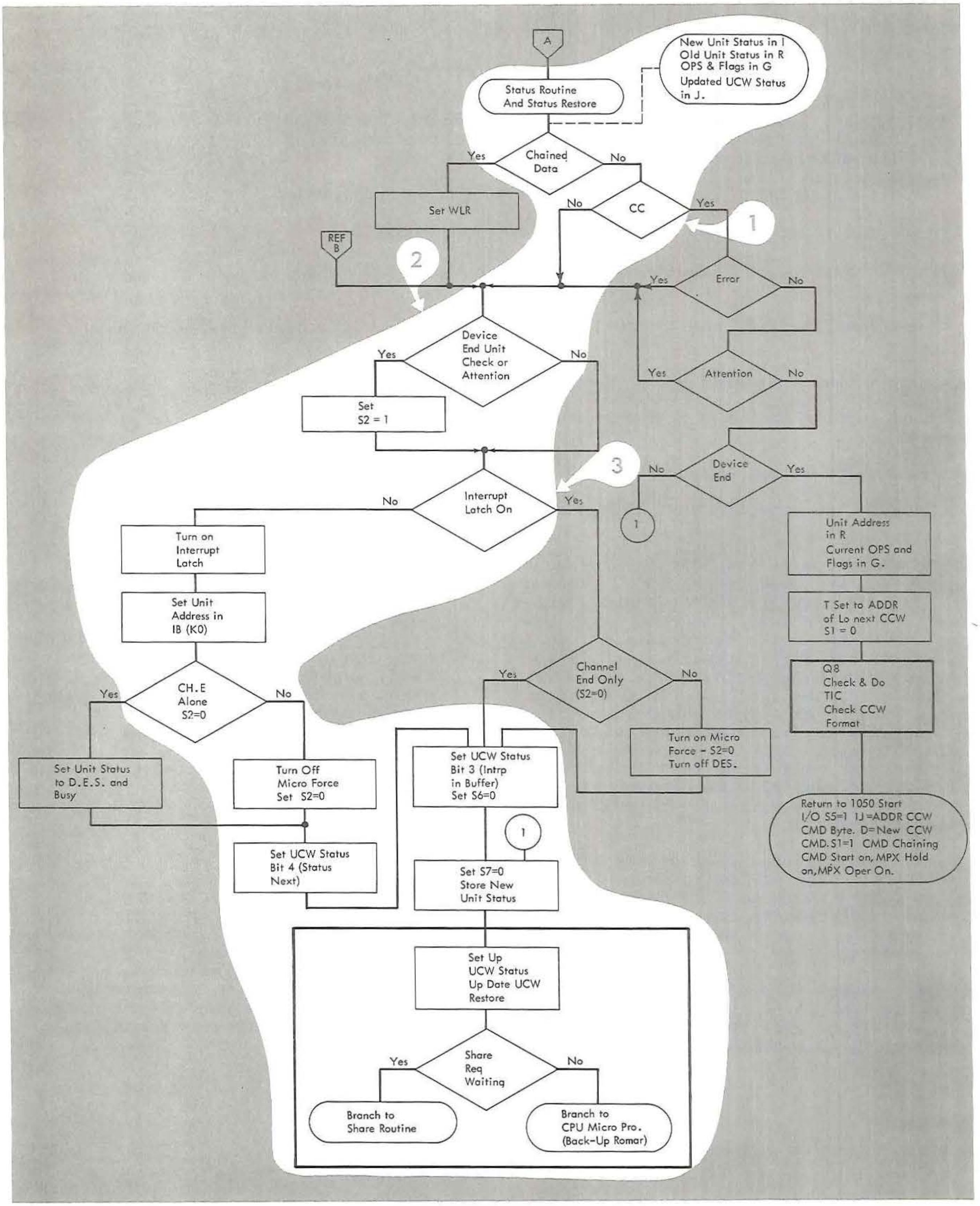


Figure 5-55. Write Status Check and Restore

DESCRIPTION

- The 1050 must have these conditions established:

  1. CPU Connect Switch (PCH-1) to CPU On.
  2. Printer 1 set to home loop and paper contacts satisfied.
  3. Program switch to dup if the program-control feature is installed (for this example only).
  4. Power on.
  5. Attend-unattend switch to attend position on Mod 1 only.

- Start I/O routine sets Home-Reader-Start TA-0 position.
- Write condition and Reader-1 clutch starts the attachment clock.
- Clock runs at 68 ms per cycle.
- Clock W-time initiates write share-request cycles.
- Share routine sets R-W register with the character to be written.
- TE latch on allows strobe to 1051.
- Character is accepted and printed at the 1050.

A write operation to the 1050 is a transfer of data from the 2030 to the 1051. Recall that the 2030 uses the reader-1 and punch-1 interfaces at the 1051. In this case, the objective of the write operation will be to simulate to the 1051 circuits that a read operation should be performed on reader-1, and then allow the 1051 home loop to transfer the data to an output device, in this case, printer 1. The attachment circuits in the 2030 must activate lines on the interface that will cause this operation of reader-1, and also hold the character repetition rate to the speed of the 1051 (14.8 characters/second).

Let's consider the setup of the 1050 system first. The following switch settings and conditions must be present in the 1051 and on the interface to perform a write operation. Both models of the 1050 are considered.

1. PCH-1 Switch (CPU-connect Sw). This switch must be in the CPU-On position. It activates the interface line CPU Connected in both the Mod I and NI and also generates the interface line reader-1 home switch. The line 1050 operational (TT-3) is developed in the 2030 attachment (Figure 5-41).
2. Printer-1 Switch. Must be set to the home loop and ready (paper in the forms feed, paper-control mode). This activates the interface line Home-output device ready.
3. Program Switch (with program control only). For this example, set to the dup position, because there will not be any prefix characters used.
4. Attend-Unattend Switch (Mod 1 only). Must be in the attend position.

When the three microprograms were explained, it became apparent that any operation can be broken into three distinct operations: starting, data transfers, and ending. The start I/O microprogram sets up conditions to start; the share-request microprogram controls the data transfers, and the status microprogram controls the ending operation.

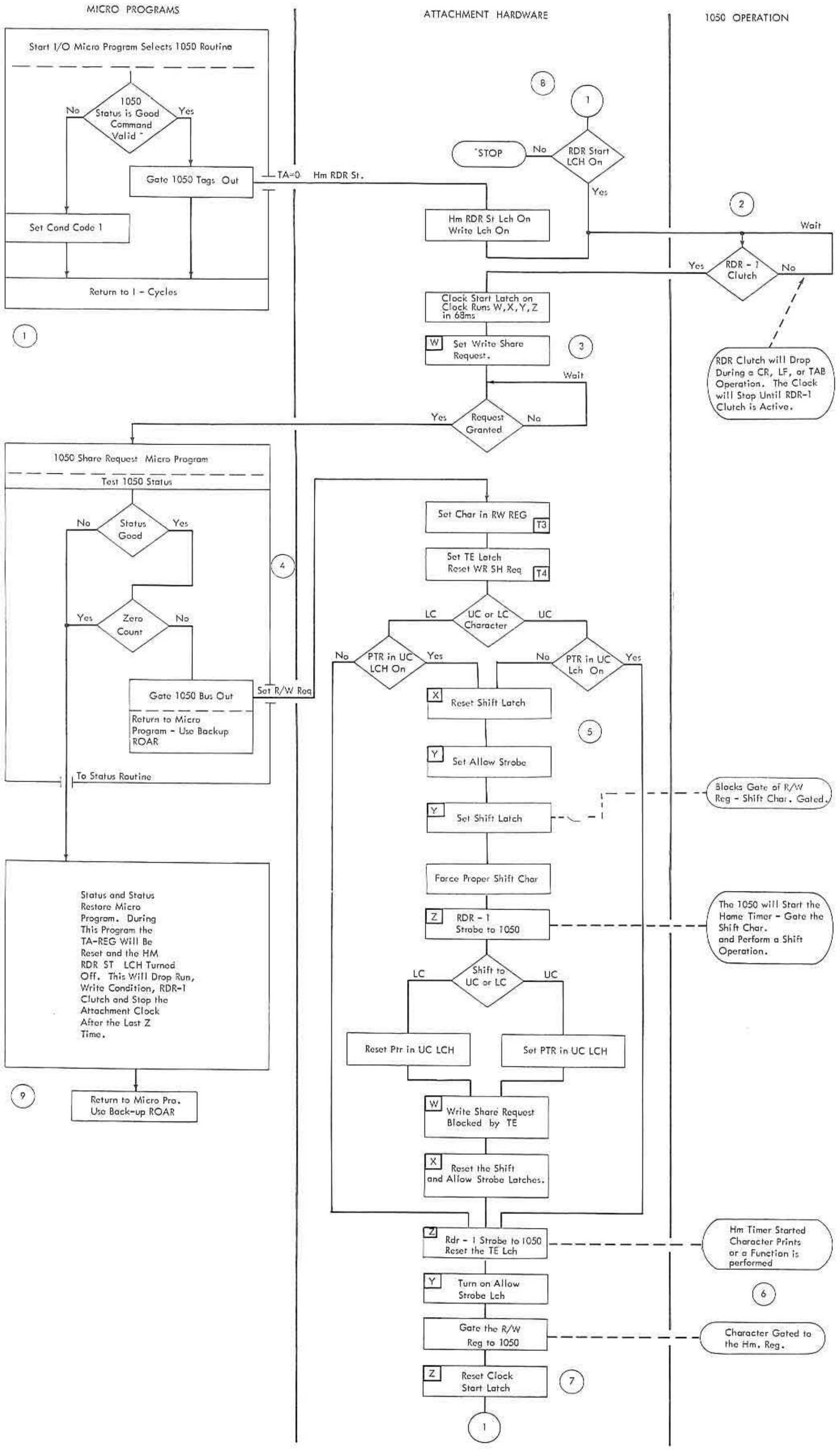

MICRO PROGRAMS
ATTACHMENT HARDWARE
1050 OPERATION
Start I/O Micro Program Selects 1050 Routine
1050 Status is Good Command Valid
No
Yes
Gate 1050 Tags Out
TA=0 Hm RDR St.
Set Cond Code 1
Return to I - Cycles
1
8
1
RDR Start LCH On
No
STOP
Yes
Hm RDR St Lch On Write Lch On
2
Wait
RDR - 1 Clutch
Yes
No
Clock Start Latch on Clock Runs W,X,Y,Z in 68ms
RDR Clutch will Drop During a CR, LF, or TAB Operation. The Clock will Stop Until RDR-1 Clutch is Active.
W Set Write Share Request.
3
Wait
Request Granted
Yes
No
1050 Share Request Micro Program
Test 1050 Status
Status Good
No
Yes
Zero Count
Yes
No
4
Gate 1050 Bus Out
Return to Micro Program - Use Backup ROAR
Set R/W Req
To Status Routine
Set Char in RW REG T3
Set TE Latch Reset WR SH Req T4
UC or LC Character
LC
UC
PTR in UC LCH On
No
Yes
PTR in UC Lch On
No
Yes
X Reset Shift Latch
5
Y Set Allow Strobe
Y Set Shift Latch
Blocks Gate of R/W Reg - Shift Char. Gated.
Force Proper Shift Char
Z RDR - 1 Strobe to 1050
The 1050 will Start the Home Timer - Gate the Shift Char. and Perform a Shift Operation.
Shift to UC or LC
LC
UC
Reset Ptr in UC LCH
Set PTR in UC LCH
W Write Share Request Blocked by TE
X Reset the Shift and Allow Strobe Latches.
Status and Status Restore Micro Program. During This Program the TA-REG Will Be Reset and the HM RDR ST LCH Turned Off. This Will Drop Run, Write Condition, RDR-1 Clutch and Stop the Attachment Clock After the Last Z Time.
9
Return to Micro Pro. Use Back-up ROAR
Z Rdr - 1 Strobe to 1050 Reset the TE Lch
Hm Timer Started Character Prints or a Function is performed
Y Turn on Allow Strobe Lch
6
Gate the R/W Reg to 1050
Character Gated to the Hm. Reg.
Z Reset Clock Start Latch
7
1

Figure 5-56. Write Operation

Figure 5-56 shows the relationship of these three microprograms, the attachment circuits, and 1051 operation. Let's go through the figure once to see how the logic of the write operation works. The circled numbers on the figure refer to the following numbered steps:

1. Start the I/O microprogram. The only real tie here between the program and the attachment hardware is the micro order: gate 1050 tags out. The rest of the microprogram routine is explained in the write-start I/O microprogram. For the write operation, the TA register 0 position is set with the command: gate 1050 tags out. This, in turn, turns on the write latch.
2. The home-reader start latch activates the reader-1 clutch line in the 1051 and returns this signal to the attachment.
3. The reader-1 clutch line and write latch starts the attachment clock in write mode. At W-time, a write share-request is initiated. The attachment share-request controls, and select-out pulse, activates 1050 op-in and captures polling.
4. Share-request microprogram again tests the 1050 and attachment for status and end conditions. If status is good and not-end condition, a micro order: gate 1050 bus out, initiates the actual data transfer cycle. The character is set in the R/W register, the TE latch is turned on, and write share-request is turned off.
5. The character is first analyzed to see if a shift cycle is necessary, and, if so, the R/W register is not gated to the 1051, but instead, the shift character is sent. At clock Z-time, the reader-1 strobe line is activated, and the shift character is translated and sent to the 1051 home register. The 1050 operation of shifting the printer follows. The turn-off of write share-request at W-time is blocked by TE latch still on.
6. The next Z-time the R/W register is gated to the 1051, reader-1 strobe sets the character to be printed in the 1051 home register, and the normal 1051 print cycle to the 1052 follows.
7. An attempt is made to turn off clock start at Z-time. However, if the set conditions are still available, the clock continues to run.
8. If the home-reader start latch had been turned off, this block would branch to a stop which means that the attachment clock has stopped, and all operations end until the clock is instructed to start again.
9. If, during the share-request microprogram, the status of the 1050 was found to be bad, or zero-count had been reached on the previous transfer, this status routine would have been entered and, besides updating status byte 1, the micro order: gate 1050 tags out would have reset home-reader start and stopped the attachment.

For a more relative timing relationship, Figure 5-57 is a representative timing chart of the attachment hardware and clock cycles. The line names followed by an asterisk are microprogram-controlled.

One more consideration to the write operation: interlocking the CPU while the printer is performing a carrier return, line feed, or tabulate operation. These three functions can take longer than one clock cycle in the attachment, so the reader 1 clutch line will be controlled by interlocks from the printer.

For example, using Figure 5-56, assume the last character sent to the 1051 was a carrier return. The block at reference 7 resets the clock-start latch and goes to reference 8. The reader start latch is still on, but at reference 2, the reader clutch line will be down until the function is complete; therefore, the attachment clock stops until this line becomes active again and turns on the clock start latch.

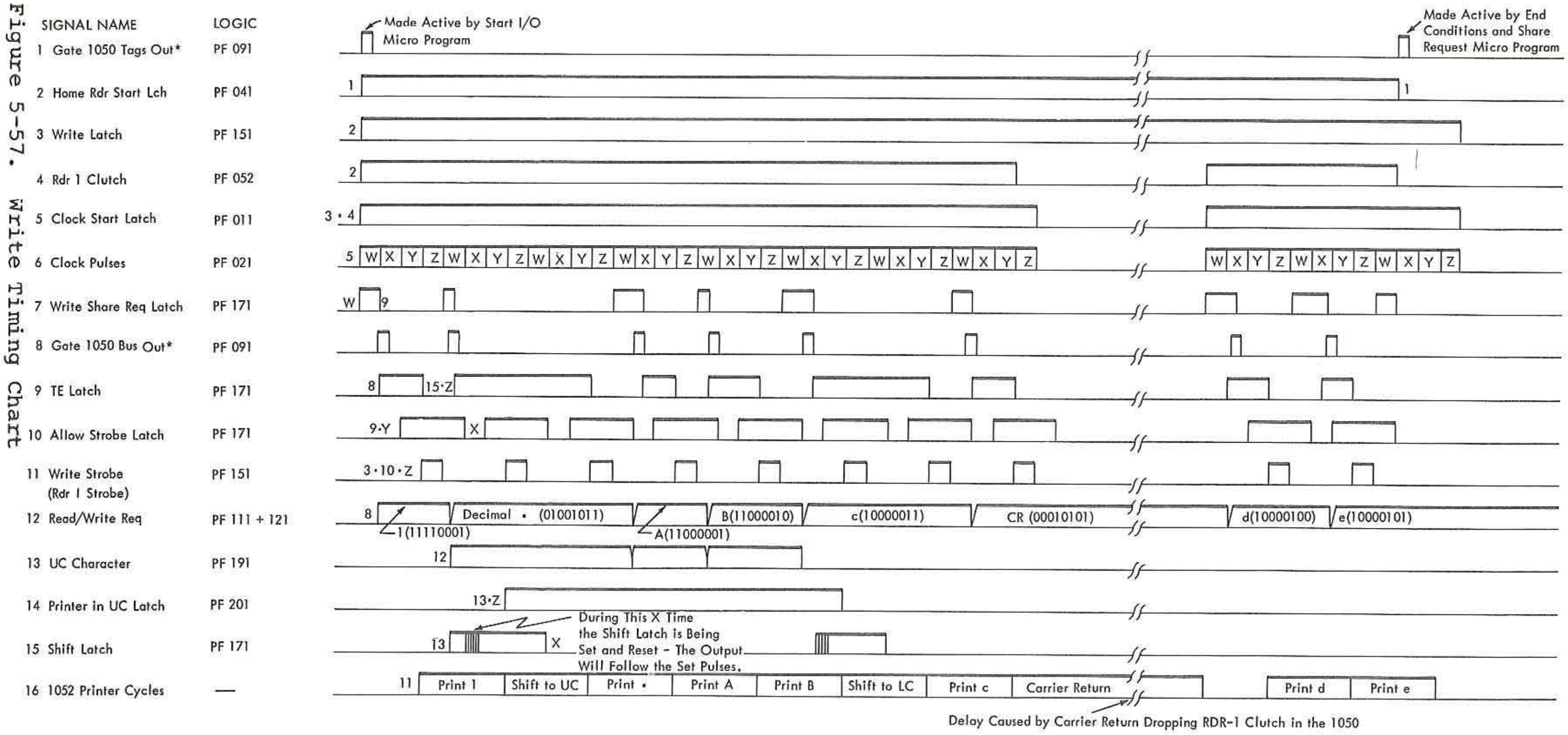
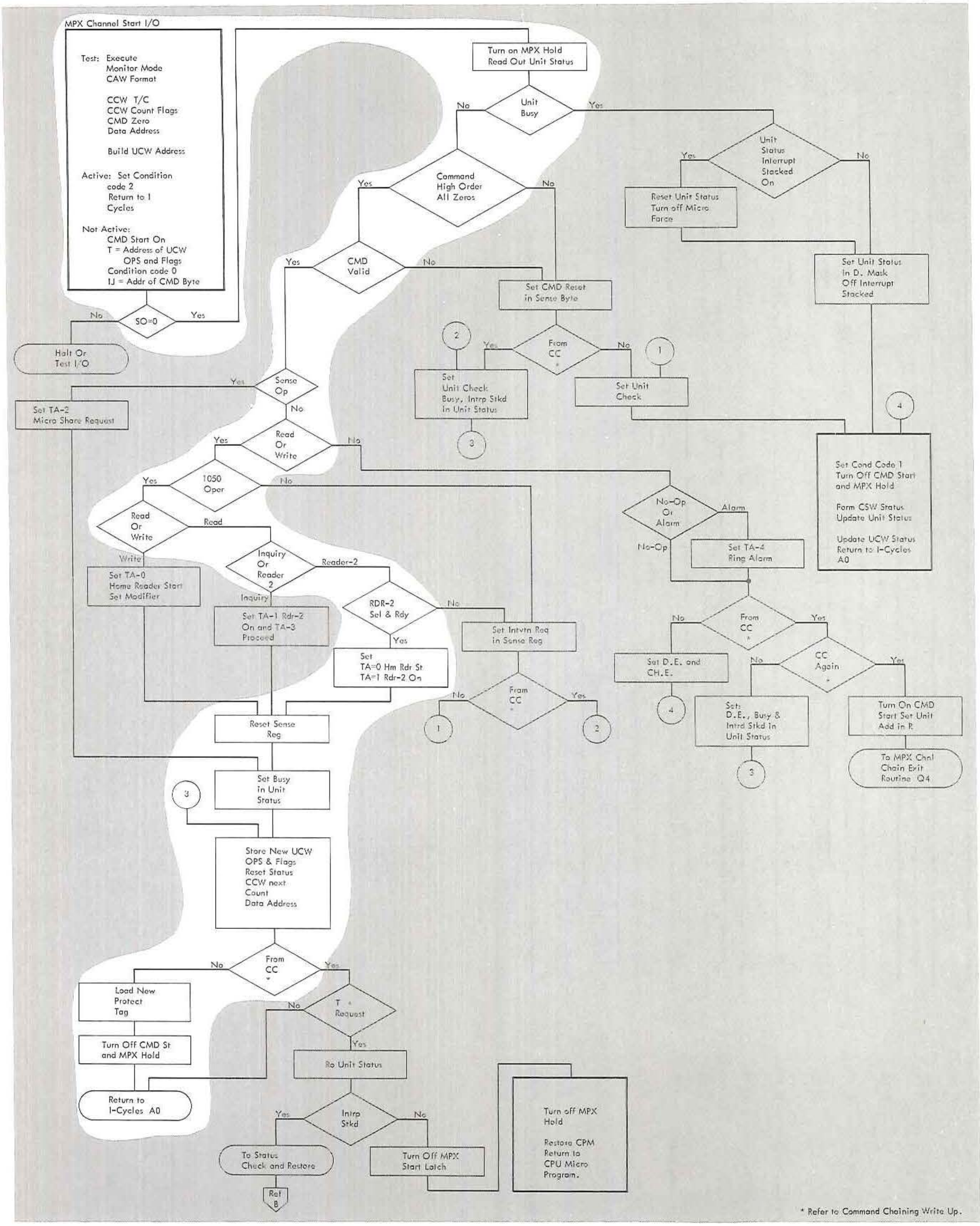


Figure 5-57. Write Timing Chart



* Refer to Command Chaining Write Up.

Figure 5-58. Read RDR-2 Start I/O

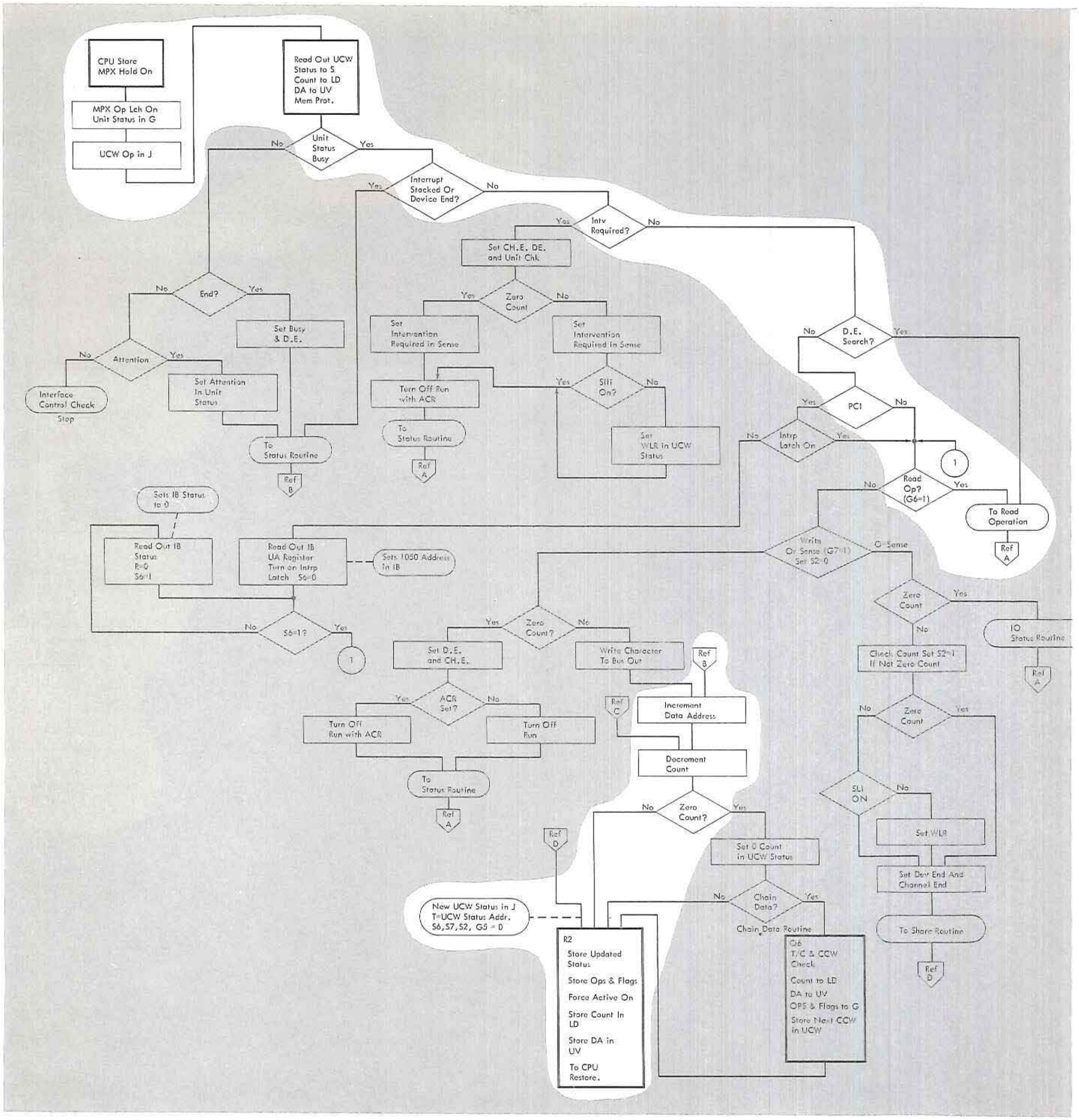


Figure 5-59. Read RDR-2 Share-Request

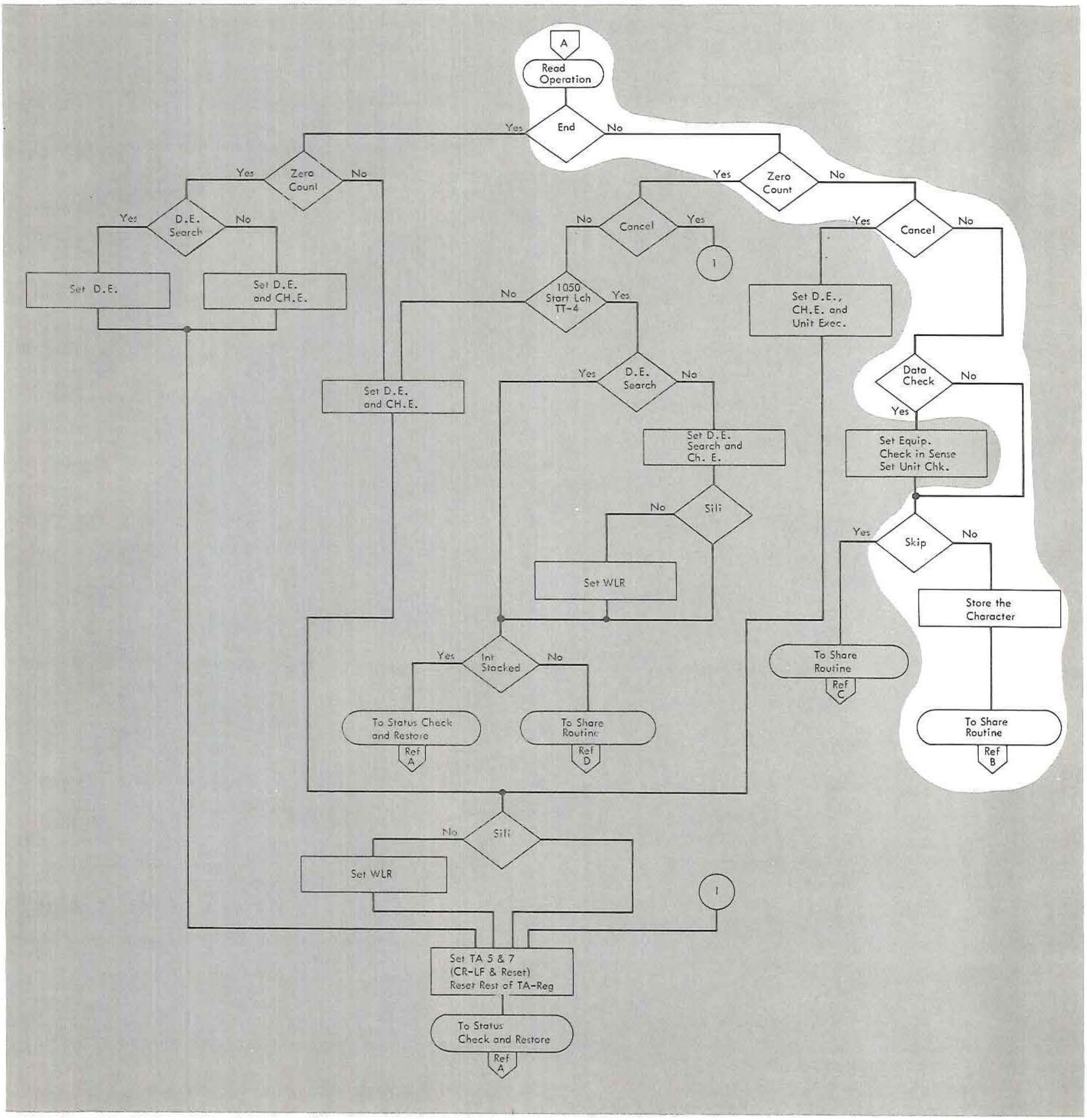


Figure 5-60. Read RDR-2 Status Check

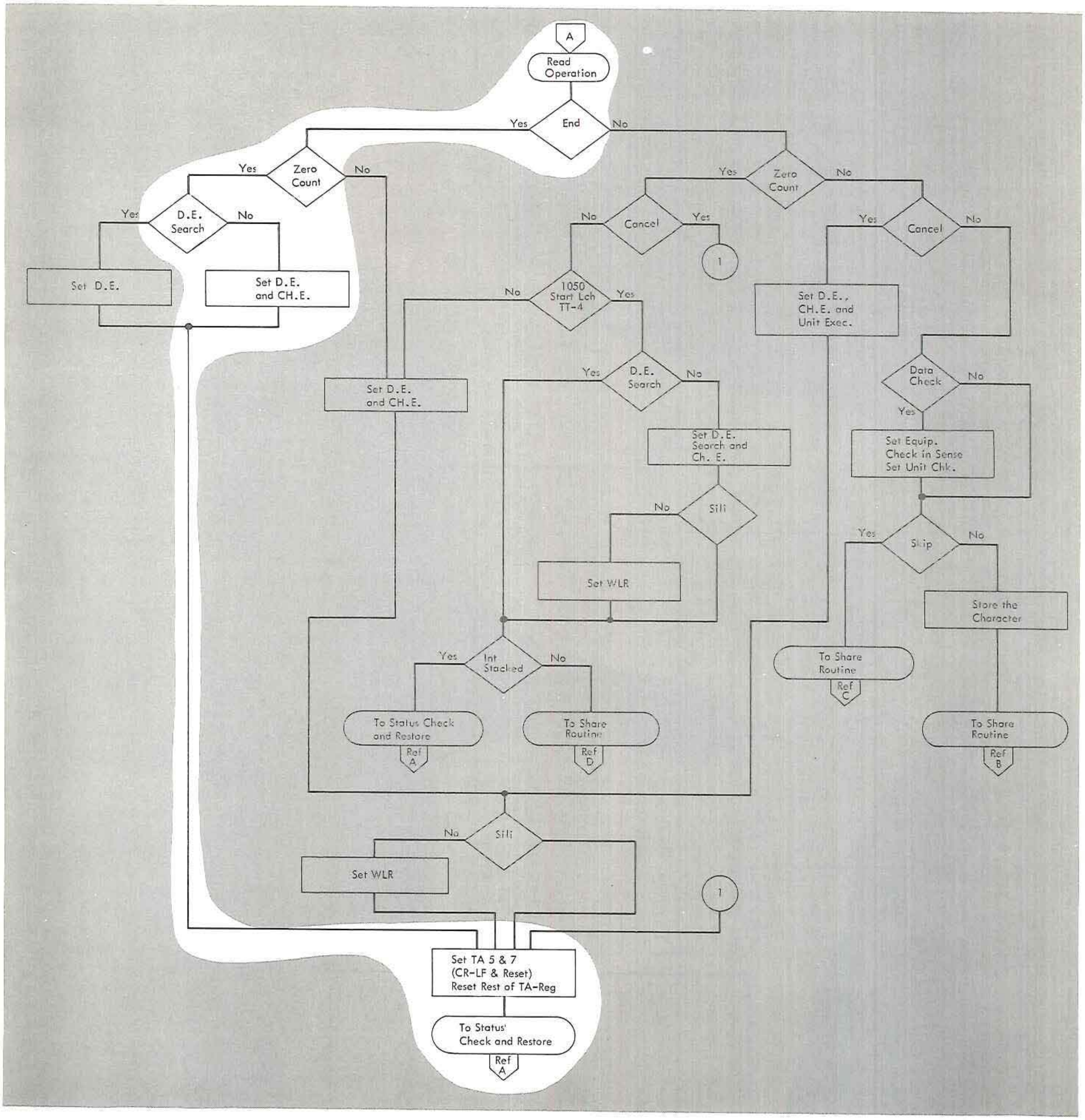


Figure 5-61. End Conditions Read RDR-2

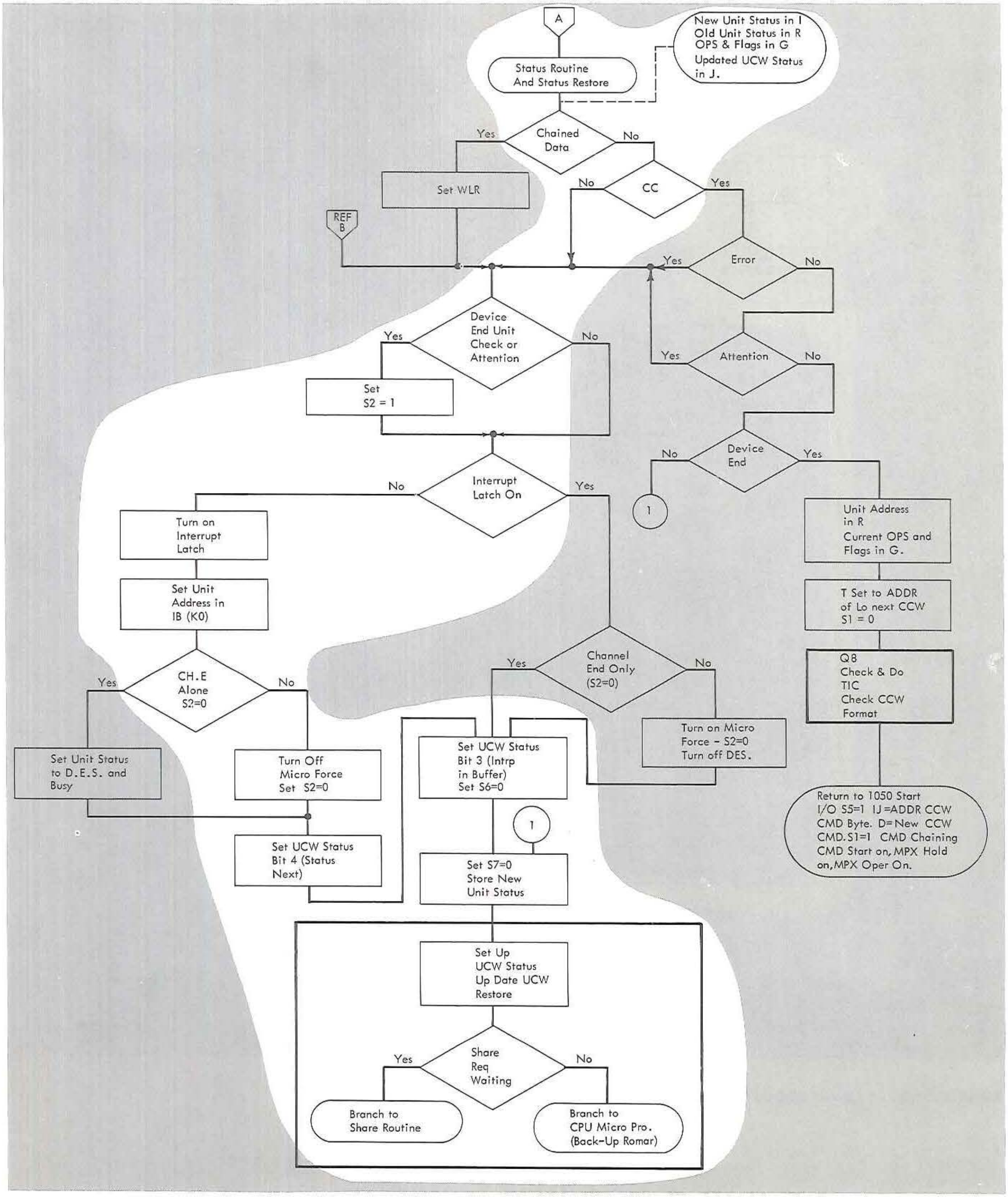


Figure 5-62. Read RDR-2 Status Check and Restore

PREFIX OPERATION TO 1051

- Prefix character blocks sending UC shift or LC shift characters as the component-select character.

With the home-component-recognition feature installed in the 1051, the prefix character could be used in the data stream to activate component-selection circuits in the 1051. An example of this would be:

ABC*2*5DEF
* = Prefix character

This message started in upper case. When the first prefix character is sent, the printer is still in upper case. The prefix character is sent, and then the component select numeral (2). The 2 is a lower-case character and would normally try to force a shift character first, before sending the 2. Recall that when the 1050 recognizes a prefix operation by receiving the prefix character, the next character is the component-select.

Had the forced-shift character been sent before sending the 2, the component-select circuits would not function properly. Therefore, anytime the prefix character is sent to the 1050, the next character will always be sent, regardless of UC or LC, and any automatic-generation shift characters will be blocked.

READ READER-2 OPERATION

READ READER-2 MICROPROGRAMS

- Sequence of microprogram are:

1. Start I/O (Read reader-2)
2. Read share-requests
3. Status check and restore

- Conditions checked for during read start I/O are:

1. Unit not busy
2. Valid Command
3. 1050 operational
4. Read operation (RDR-2)
5. RDR-2 select and ready

- Conditions checked for during read share-request cycles are:

1. Unit busy
2. Intervention required
3. Device-end search
4. PCI
5. Read operation
6. End
7. Zero count
8. Cancel
9. Data check
10. Skip

- Status check and restore cycle will be entered by:

1. Intervention required
2. EOB or EOT character (End)
3. Zero-Count conditions

Start I/O

The start I/O microprogram makes the standard checks of unit busy, valid command, 1050 operational, read reader-2 and reader 2 select and ready. The start I/O sets the reader-2 On latch and home-reader start latch in the TA-register (TA-1 and TA-0) This activates the run controls in the attachment and signals to the 1051 to start the reader. The attachment clock is set up to run in read mode, or 34 ms per w, x, y, and z-cycle. See Figure 5-58 for the flow chart of this operation.

Read Share-Request

Each time a character is read at the reader, a read share-request cycle is needed to transfer the data from the 1051 to the 2030 storage unit. Each data transfer, the microprogram is checking for status and conditions as shown in Figures 5-59 and 5-60. The decode of EOB or EOT characters initiates the end operation and forces a branch to the status check and restore microprogram.

Status Check and Restore

This microprogram is entered any time the operation is to be terminated. This entry can be from many of the decision blocks on the flow charts. Many variations are possible. The best way to follow any combination is to use the flow charts, keeping track of status and other conditions as decision blocks are reached. The representative flow chart is shown in Figures 5-61 and 5-62.

DESCRIPTION

- Initiated by start I/O microprogram.
- Reader 2 at 1050 sends the data to the CPU.
- Operation normally ended by the reader sending EOB or EOT characters.
- 1050 switch settings to perform a read reader-2 are:

  1. CPU connect On
  2. Reader-2 switch to Home.
  3. Program-dup switch is set to dup if the program-control feature is installed.
  4. Attend-unattend switch to attend (Mod 1 only).

With the proper switch settings at the 1051, the microprogram can start the operation with a start I/O. During this microprogram, the command is sent to the attachment with the order: gate 1050 tags out.

To initiate the read reader-2 command, the TA register positions 0 and 1 are set. The sequence of events that follow is shown in Figure 5-63.

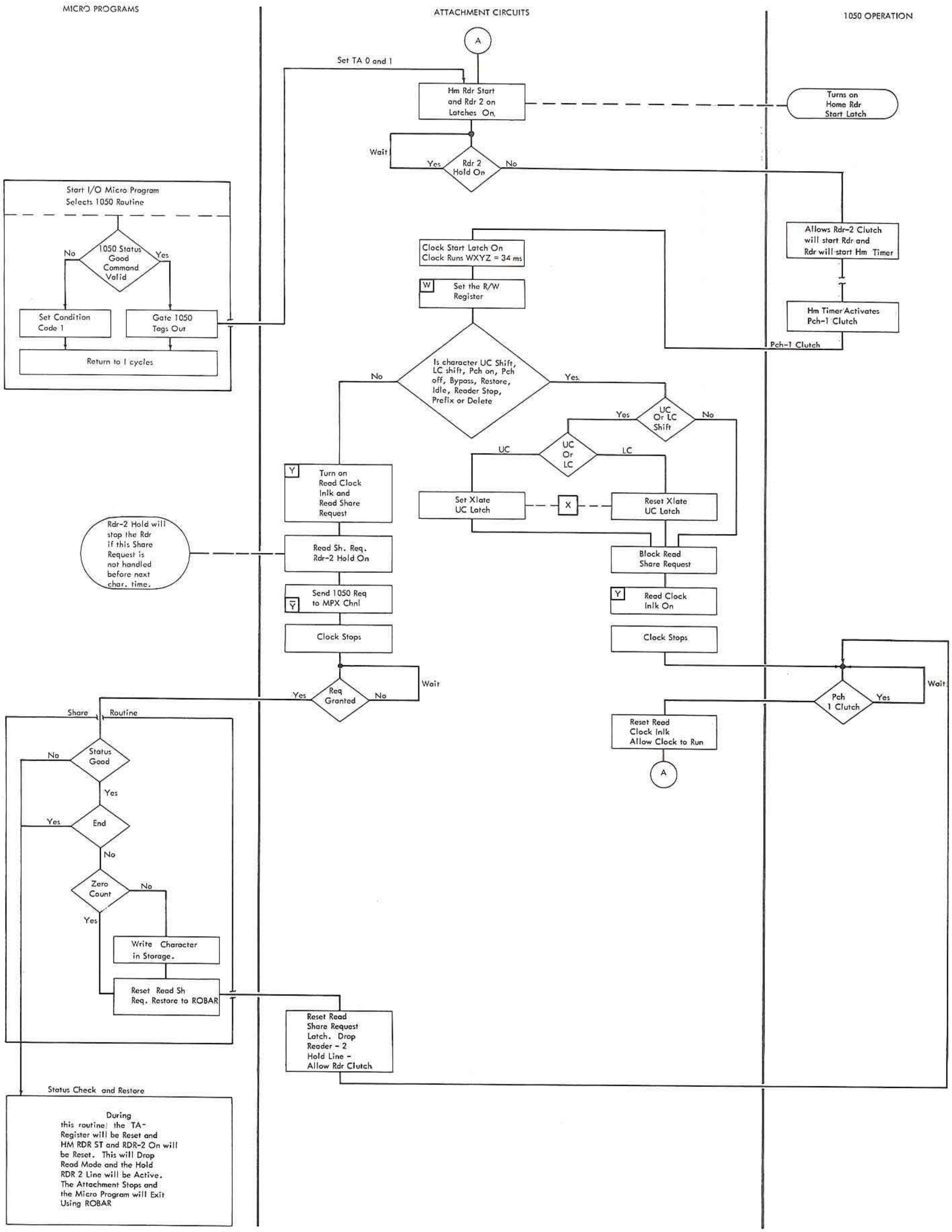

MICRO PROGRAMS
ATTACHMENT CIRCUITS
1050 OPERATION
A
Set TA 0 and 1
Hm Rdr Start and Rdr 2 on Latches On.
Turns on Home Rdr Start Latch
Wait
Yes
Rdr 2 Hold On
No
Start I/O Micro Program Selects 1050 Routine
1050 Status Good Command Valid
No
Yes
Set Condition Code 1
Gate 1050 Tags Out
Return to I cycles
Clock Start Latch On Clock Runs WXYZ = 34 ms
W
Set the R/W Register
Allows Rdr-2 Clutch will start Rdr and Rdr will start Hm Timer
Hm Timer Activates Pch-1 Clutch
Pch-1 Clutch
Is character UC Shift, LC shift, Pch on, Pch off, Bypass, Restore, Idle, Reader Stop, Prefix or Delete
No
Yes
UC Or LC Shift
Yes
No
UC Or LC
UC
LC
Y
Turn on Read Clock Inlk and Read Share Request
Set Xlate UC Latch
X
Reset Xlate UC Latch
Rdr-2 Hold will stop the Rdr if this Share Request is not handled before next char. time.
Read Sh. Req. Rdr-2 Hold On
Block Read Share Request
Send 1050 Req to MPX Chnl
Y
Read Clock Inlk On
Clock Stops
Clock Stops
Wait
Req Granted
Yes
No
Pch 1 Clutch
Yes
Wait
Reset Read Clock Inlk Allow Clock to Run
A
Share
Routine
Status Good
No
Yes
End
Yes
No
Zero Count
No
Yes
Write Character in Storage.
Reset Read Sh Req. Restore to ROBAR
Reset Read Share Request Latch. Drop Reader - 2 Hold Line - Allow Rdr Clutch
Status Check and Restore
During this routine the TA-Register will be Reset and HM RDR ST and RDR-2 On will be Reset. This will Drop Read Mode and the Hold RDR 2 Line will be Active. The Attachment Stops and the Micro Program will Exit Using ROBAR

Figure 5-63. Read RDR-2 Operation

1. TA register reader-2 on and home reader-start latches are turned on. This activates the interface lines to start reader-2 at the 1050.
2. Reader strobe starts the 1051 home timer, which develops the interface line punch-1 clutch, and gates the punch-1 bus to the 2030.
3. The attachment clock will start. At W-time the R/W register is set with the PT&T/8 character and be tested to see if the character should be stored. If not, a further check is made to see if it is a UC or LC shift character. If it is, the translate UC latch is set or reset accordingly.
4. At Y clock time, the read clock-interlock latch is turned on to stop the clock. Also, if the character decode is such that a store cycle is needed, the read share-request latch is turned on, and reader-2 hold made active.
5. At not-Y clock time, the 1050 request-in line is made active. When the channel controls can allow the trap, Sel-0 is sent out. The attachment share-request controls capture the poll, send operational-in, and T-request in. The 1050 share microprogram routine is now entered.

The share microprogram checks status and end conditions, and with no end conditions causes the character to be stored. When the exit from this program is performed, the read share-request latch is reset, allowing the reader-2 hold line to drop and keep the reader running. The reader senses the next character, starts the 1051 home timer, and the sequence is repeated. Eventually, the data-count reaches zero, and the next cycle should be the EOB or EOT character from the reader. This causes a normal end.

Figure 5-64 shows a timing chart of the reader-1 operation giving relative timings.

End Operation (Zero-Count and EOB or EOT)

With zero-count bit in the UCW status byte and a character decode causing END (TT-2) condition, the status check and restore microprogram is entered to end the operation. The TA-register is reset, and the attachment drops run mode. The microprogram uses ROBAR and returns to the program that was running when the share-request trap was taken.

End Operation (Zero-Count Only)

When reading from the reader, each block of data is separated by the EOB or EOT charac-

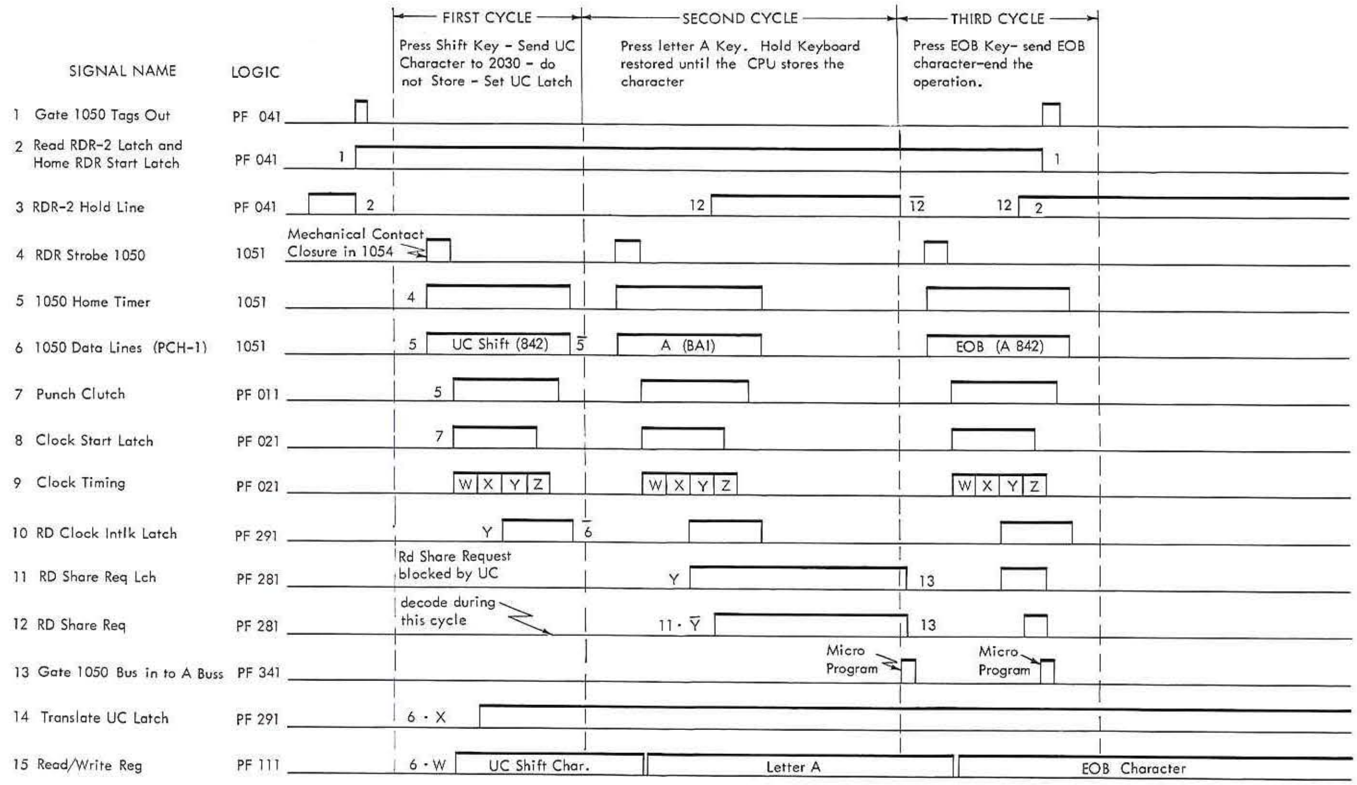


Figure 5-64. Read RDR-2 Timing Chart

ters. Paper-tape transmission can be of any number of characters, dictated only by when the operator punches in the EOB or EOT characters.

Suppose the record we are transmitting is fifty characters long and the 51st character is EOB. The data-count set up by the CCW is 45 characters. Therefore, the zero-count condition will be sensed after storing the 45th character, because the count is reduced by one each time a character is stored. It would not work too well if we stopped the reader now, because 5 characters are left until the EOB is read. This would make the next record very short (5 characters); then End would be sensed and the reader stopped. This is incorrect operation.

Some means must be provided to keep the reader running until EOB or EOT is read. When zero-count and not-End is sensed, channel-end and device-end search are set in the unit status. Channel-end indicates the CPU should not store any more characters, and device-end search keeps the reader running looking for EOB or EOT. When the EOB or EOT character is read, End along with zero-count, ends the operation in the normal manner.

READ INQUIRY OPERATION

READ INQUIRY MICROPROGRAMS

- Sequence of microprograms is:

  1. Share-request (attention)
  2. Start I/O (Read Inquiry)
  3. Read share-requests (data transfers)
  4. Status restore (end operation)

- Conditions checked for during read inquiry start I/O are :

  1. Unit not busy
  2. Valid command
  3. 1050 operational
  4. Read Inquiry operation

- Conditions checked for during read-inquiry share-request cycles are:

  1. Unit busy
  2. Intervention required
  3. Device end search
  4. PCI
  5. Read operation
  6. End
  7. Zero Count
  8. Cancel
  9. Data Check
  10. Skip

- Status restore routine is entered by

  1. Intervention required
  2. EOB or EOT character (End)
  3. Cancel character
  4. Zero-count conditions

Attention Share-Request

The primary objective of the attention share-request cycle is to alert the 2030 programs that a request has been initiated from the keyboard. The CSW will be updated with the attention bit set on. Also, during the status restore, the I/O interrupt condition will be set on, which will normally cause a check of the CSW and initiate the start I/O that is associated with the read inquiry. Figures 5-65 and 5-66 show the share-request and status-restore flow charts and the path this operation will take.

Read Inquiry Start I/O

The start I/O microprogram makes the standard checks of unit busy, valid command, 1050 operational, and the actual command checking--in this case, a read inquiry. Again, the start I-O sets a latch in the TA-register. The latch or latches set activate the attachment circuits.

During read inquiry, the primary objective at this time is to signal the operator at the 1050 that the request has been honored. This is done by setting TA-3 (Proceed) and TA-1 (RDR-2 on). This lights the proceed light in the 1050 and sets the attachment to read-inquiry mode. This ends the start I/O as far as the attachment circuits are concerned. See Figure 5-67 for the flow path of the read-inquiry start I-O.

Read Inquiry Share Requests

Each time a character is sent from the 1050, a read-inquiry share-request is needed to transfer the data to the 2030. For each character transferred, the program is checking the status and conditions as shown in Figures 5-68 and 5-69. The decode of EOB, EOT, or cancel characters can initiate the status-restore or end operation. Of course, if the 1050 should lose its operational status, the operation ends also.

Read Inquiry Status Check and Restore

This microprogram is entered any time time the operation is to be terminated. This can be from many of the decision blocks on the flow charts. The possible variation of conditions are many. The best way to follow any combinations is to keep track of the conditions and use the flow charts for decisions. The representative flow chart is shown in Figures 5-62 and 5-70.

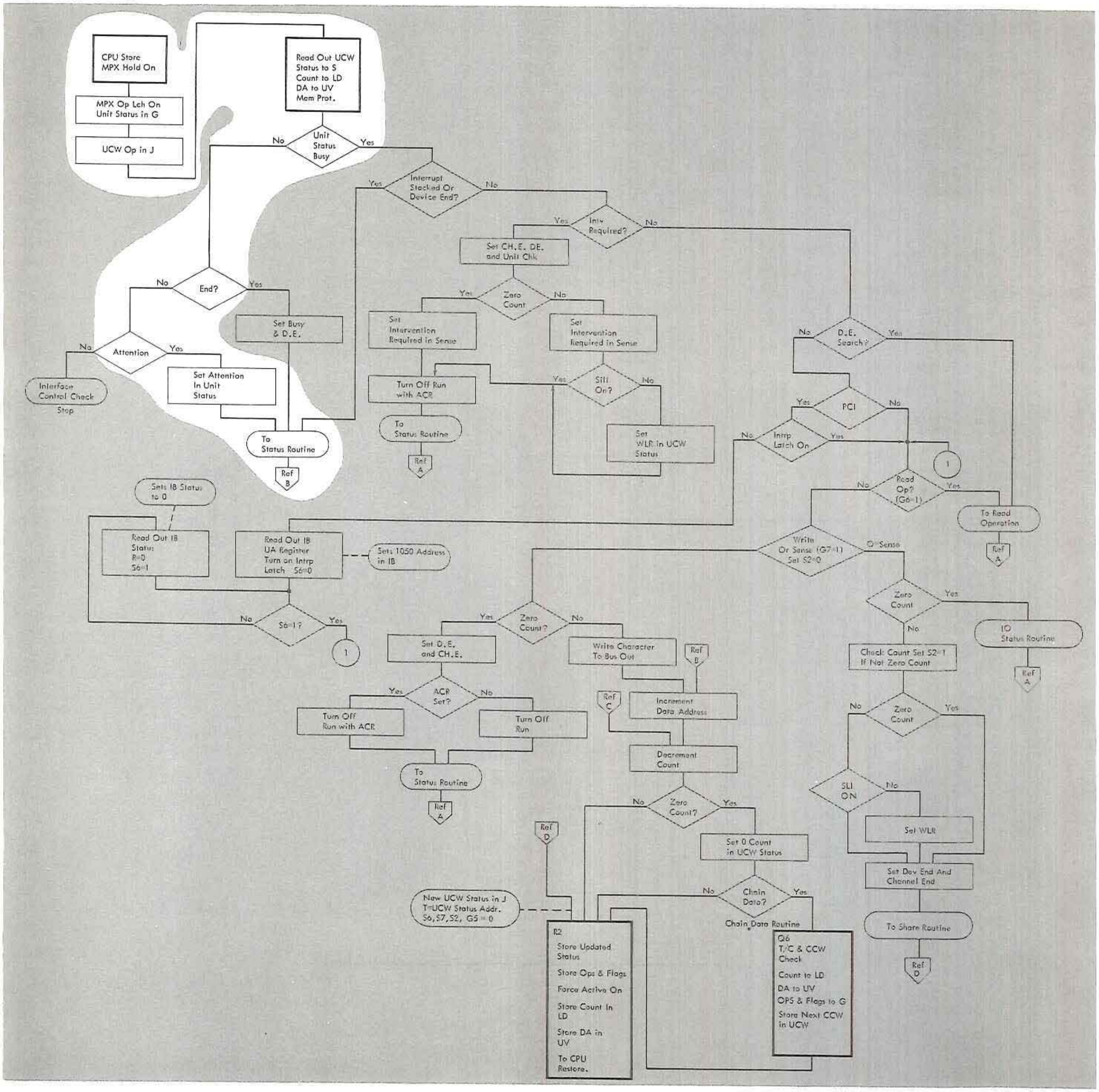


Figure 5-65. Attention Share-Request

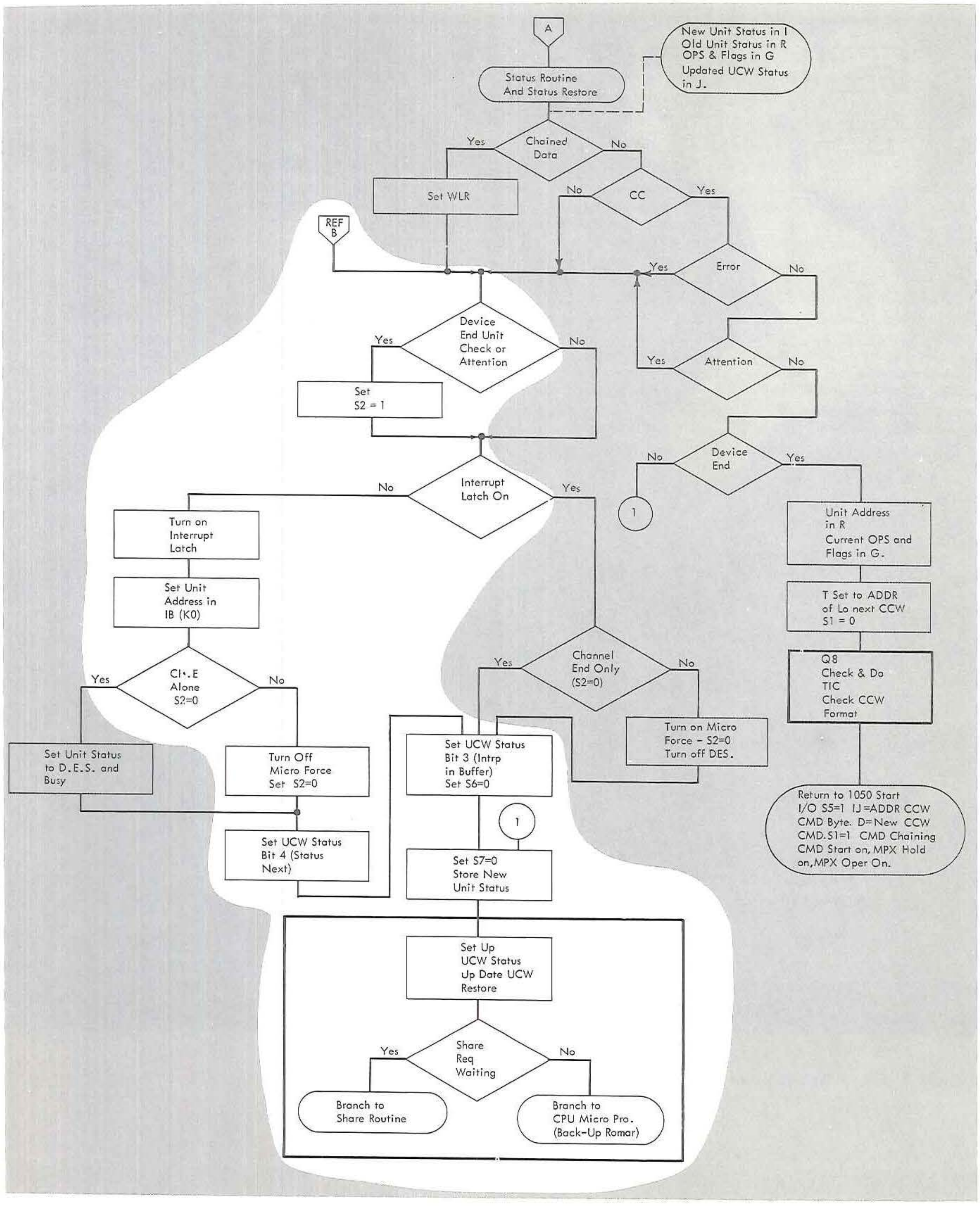
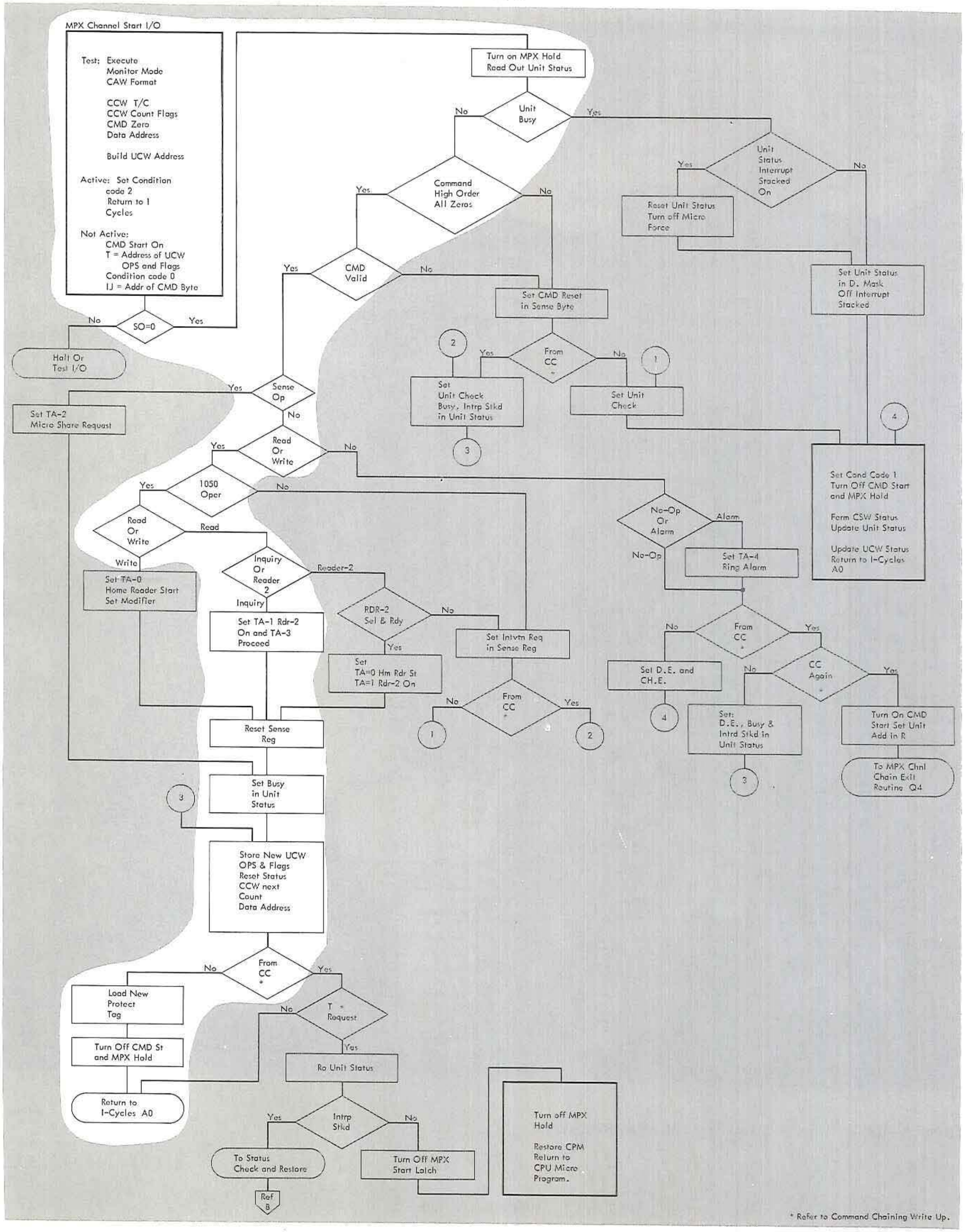

A
Status Routine And Status Restore
New Unit Status in I Old Unit Status in R OPS & Flags in G Updated UCW Status in J.
Chained Data
Yes
No
Set WLR
CC
No
Yes
REF B
Error
Yes
No
Device End Unit Check or Attention
Yes
No
Set S2 = 1
Attention
Yes
No
Device End
No
Yes
1
Interrupt Latch On
No
Yes
Turn on Interrupt Latch
Set Unit Address in IB (K0)
Unit Address in R Current OPS and Flags in G.
T Set to ADDR of Lo next CCW S1 = 0
Q8 Check & Do TIC Check CCW Format
Cl·.E Alone S2=0
Yes
No
Channel End Only (S2=0)
Yes
No
Set Unit Status to D.E.S. and Busy
Turn Off Micro Force Set S2=0
Set UCW Status Bit 3 (Intrp in Buffer) Set S6=0
Turn on Micro Force - S2=0 Turn off DES.
Return to 1050 Start I/O S5=1 IJ =ADDR CCW CMD Byte. D=New CCW CMD.S1=1 CMD Chaining CMD Start on, MPX Hold on, MPX Oper On.
Set UCW Status Bit 4 (Status Next)
1
Set S7=0 Store New Unit Status
Set Up UCW Status Up Date UCW Restore
Share Req Waiting
Yes
No
Branch to Share Routine
Branch to CPU Micro Pro. (Back-Up Romar)

Figure 5-66. Attention Status Check and Restore



* Refer to Command Chaining Write Up.

Figure 5-67. Read Inquiry Start I/O

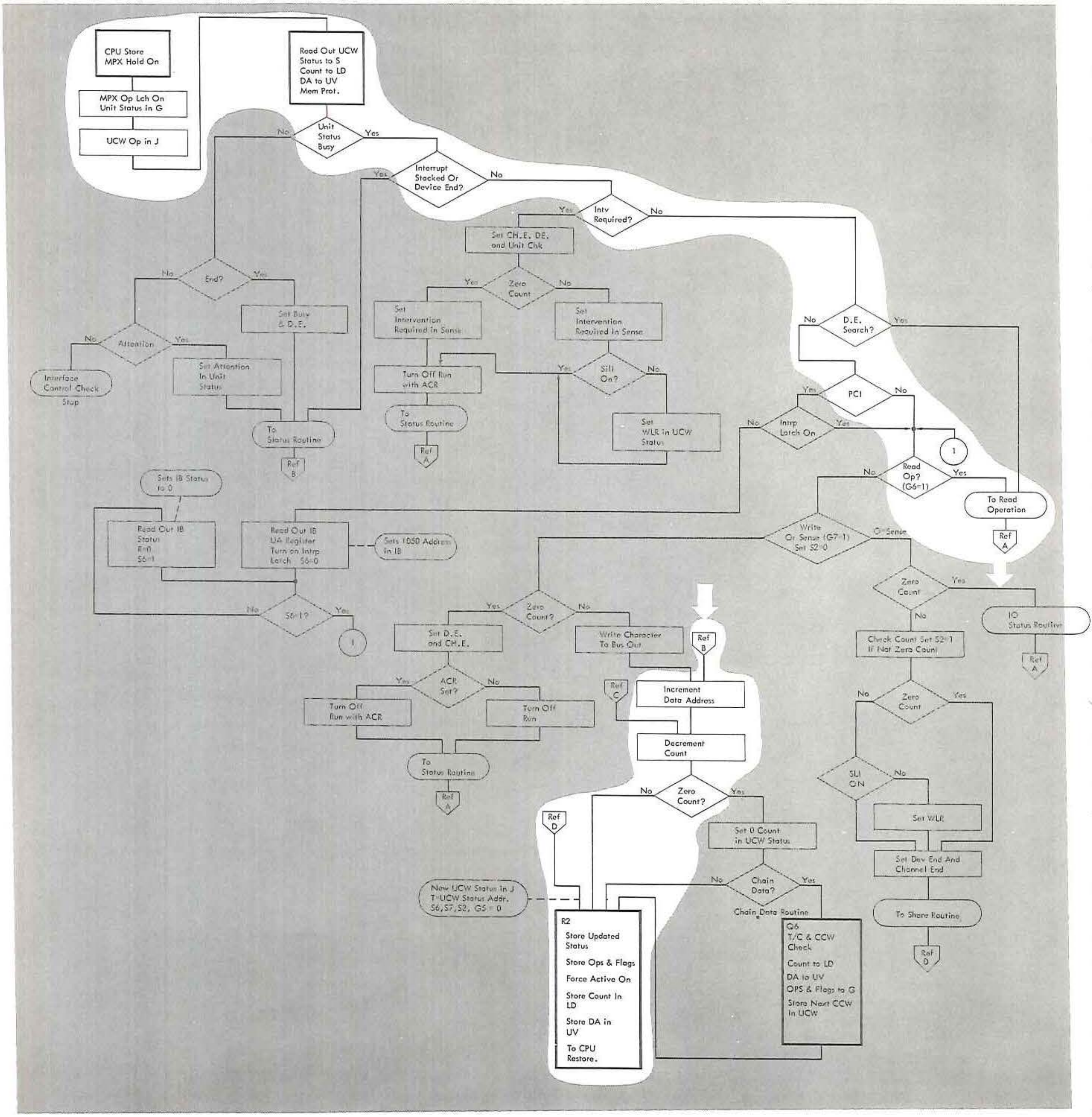


Figure 5-68. Read Inquiry Share-Request

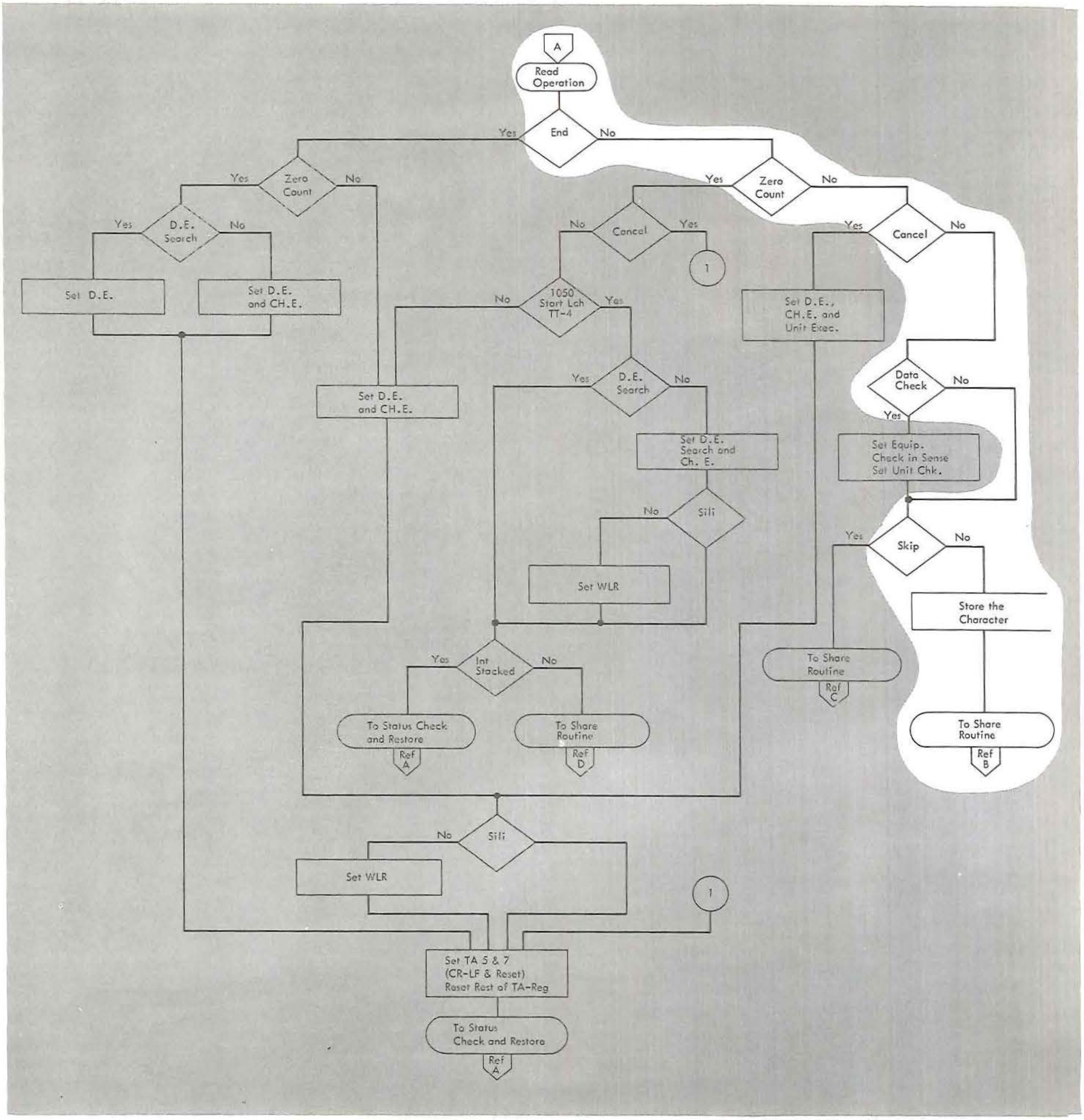


Figure 5-69. Read Inquiry

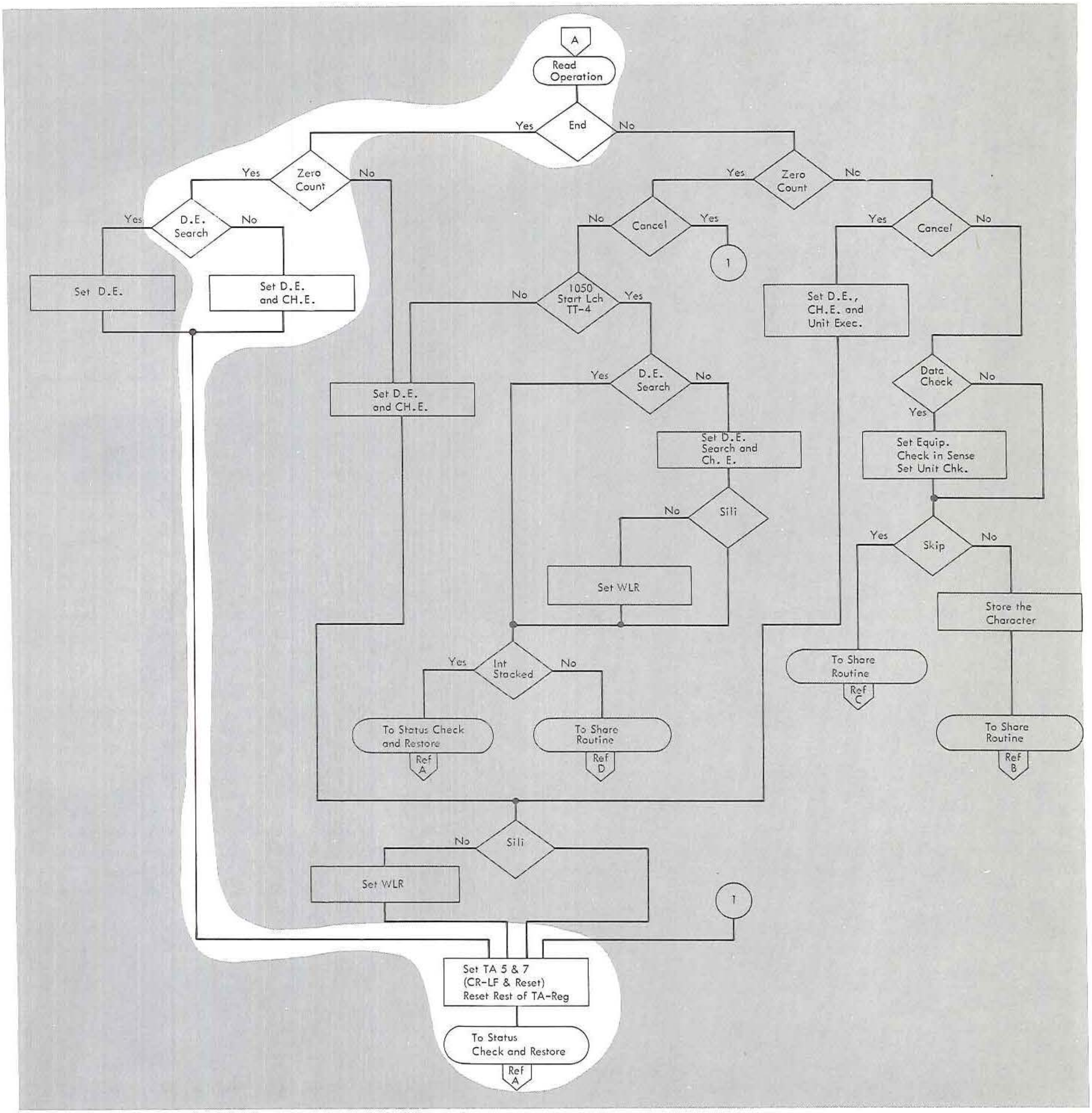


Figure 5-70. Read Inquiry End Operation

Description

- Initiated by pressing the request key at the 1050.
- Acknowledge when the proceed light turns on at the 1050.
- Data entered by normal keyboard operation at the 1050.
- Operation normally ended by sending EOB, EOT, or cancel character from 1050.

Initiating the Operation

- 1050 switch settings to perform a read inquiry are:

  1. CPU Connect-On
  2. Keyboard switch to Home.
  3. Program-dup switch to dup if program-control feature is installed.
  4. Power on.
  5. Attend/unattend switch to attend (Mod 1 only).

With the proper switch settings at the 1051, the operator can press the request key. This starts a series of events as shown in Figure 5-71. The sequence is:

1. Set attention latch.
2. Send 1050 share request to channel.
3. Request granted-reset attention
4. Start I/O with test status and send 1050 tags out: TA-3 and TA-6.
5. Proceed line to 1050 is active.
6. Proceed light on indicates to the operator that the request has been granted.

Data Transfer (Figure 5-71)

When the proceed light indicates the request has been granted, the operator can start sending data via the keyboard on the 1052. Each time a key is pressed, a character is sent to the attachment via the interface. This data transfer uses the punch-1 interface and the operation looks like a keyboard to punch operation at the 1050. The punch-1 clutch line to the attachment signals that a character is available on the interface and starts the clock. The sequence of the operation once the clock starts is:

1. At W-clock time, set the read-write register.
2. Test to see if the character is one that will be stored or ignored.
3. If the character is not to be stored, a check for LC or UC shift character is made. A shift character sets or resets the translate upper case latch at X-clock time. This latch is used in the read translator for proper EBCDI translation.
4. At Y-clock time, the read-clock-interlock latch is turned on. This stops the clock after one cycle. Also at Y-time, if the character is to be stored, the read share-request latch is turned on.
5. Read share-request drops the proceed line to the 1051, and when Y-time falls, it sends in the line 1050 request-in.
6. When the 1050 request is granted, the MPX share-request trap is taken in the microprogram. Sel-O is sent to the 1050 attachment and operational-in is sent back from the attachment. During the share routine, status and end conditions (EOB, EOT, or Cancel) are tested. The next test is for zero-count, if zero-count has been reached, the character is not stored.
7. Read share-request is reset in the attachment by microprogramming (FB = KXX,0) and proceed is allowed to be active to the 1051 again. If punch-1 clutch has ended, the clock is able to run as read-clock-interlock is reset. The next character can now be sent.

Figure 5-72 shows a timing chart of read-inquiry data cycles. The rest of the data cycles are a repetition of the sequence just explained. Lets go on and see how the operator can end the read-inquiry operation.

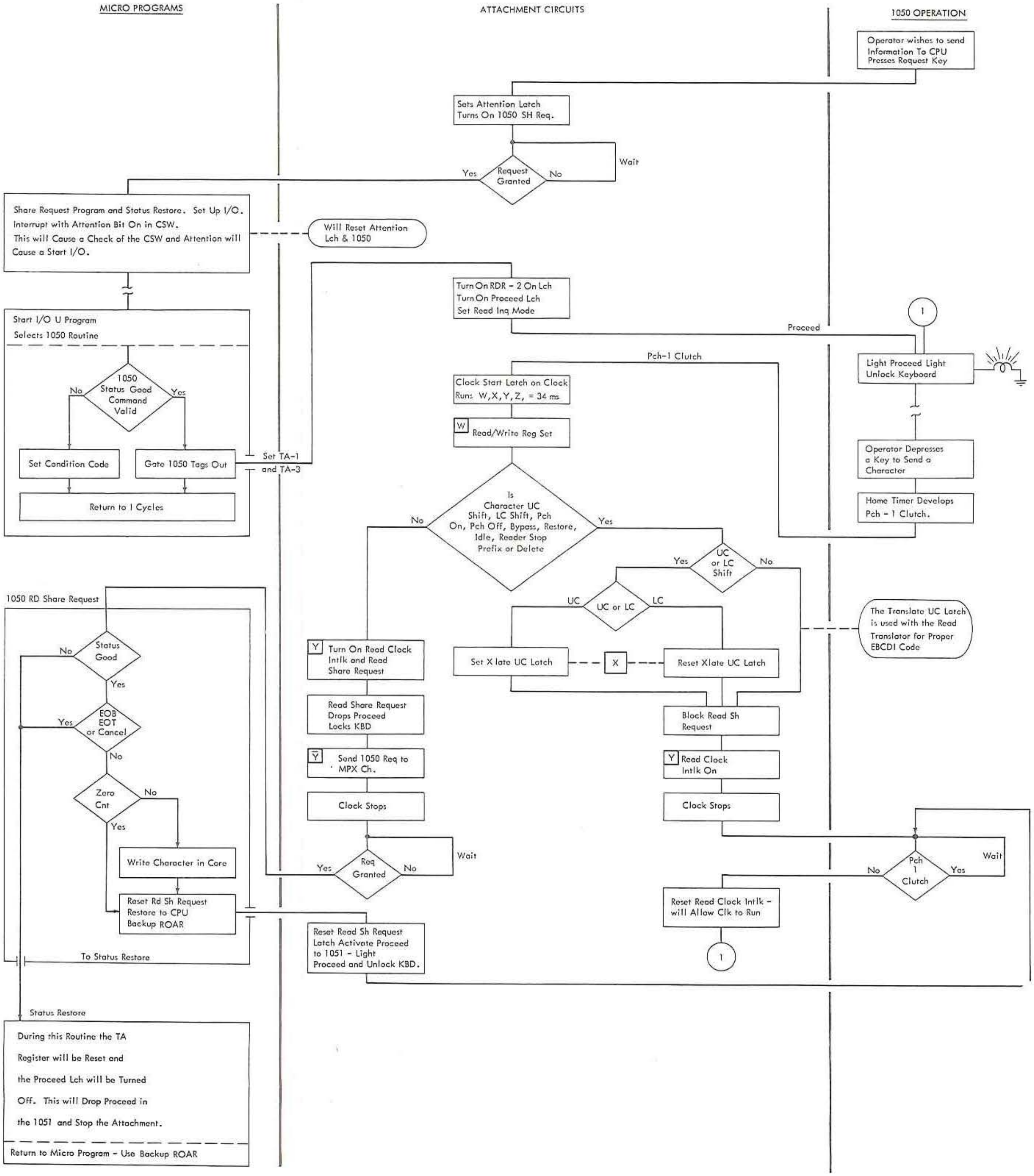

MICRO PROGRAMS
ATTACHMENT CIRCUITS
1050 OPERATION
Operator wishes to send Information To CPU Presses Request Key
Sets Attention Latch Turns On 1050 SH Req.
Request Granted
Yes
No
Wait
Share Request Program and Status Restore. Set Up I/O. Interrupt with Attention Bit On in CSW. This will Cause a Check of the CSW and Attention will Cause a Start I/O.
Will Reset Attention Lch & 1050
Turn On RDR - 2 On Lch Turn On Proceed Lch Set Read Inq Mode
Proceed
1
Start I/O U Program
Selects 1050 Routine
1050 Status Good Command Valid
No
Yes
Set Condition Code
Gate 1050 Tags Out
Set TA-1 and TA-3
Return to I Cycles
Pch-1 Clutch
Clock Start Latch on Clock Runs W,X,Y,Z, = 34 ms
W Read/Write Reg Set
Light Proceed Light Unlock Keyboard
Operator Depresses a Key to Send a Character
Home Timer Develops Pch - 1 Clutch.
Is Character UC Shift, LC Shift, Pch On, Pch Off, Bypass, Restore, Idle, Reader Stop Prefix or Delete
No
Yes
UC or LC Shift
Yes
No
UC or LC
UC
LC
The Translate UC Latch is used with the Read Translator for Proper EBCDI Code
Set X late UC Latch
X
Reset Xlate UC Latch
1050 RD Share Request
Status Good
No
Yes
EOB EOT or Cancel
Yes
No
Zero Cnt
No
Yes
Write Character in Core
Reset Rd Sh Request Restore to CPU Backup ROAR
To Status Restore
Y Turn On Read Clock Intlk and Read Share Request
Read Share Request Drops Proceed Locks KBD
Y Send 1050 Req to MPX Ch.
Clock Stops
Req Granted
Yes
No
Wait
Reset Read Sh Request Latch Activate Proceed to 1051 - Light Proceed and Unlock KBD.
Block Read Sh Request
Y Read Clock Intlk On
Clock Stops
Pch 1 Clutch
No
Yes
Wait
Reset Read Clock Intlk - will Allow Clk to Run
1
Status Restore
During this Routine the TA Register will be Reset and the Proceed Lch will be Turned Off. This will Drop Proceed in the 1051 and Stop the Attachment.
Return to Micro Program - Use Backup ROAR

Figure 5-71. Read Inquiry Operation

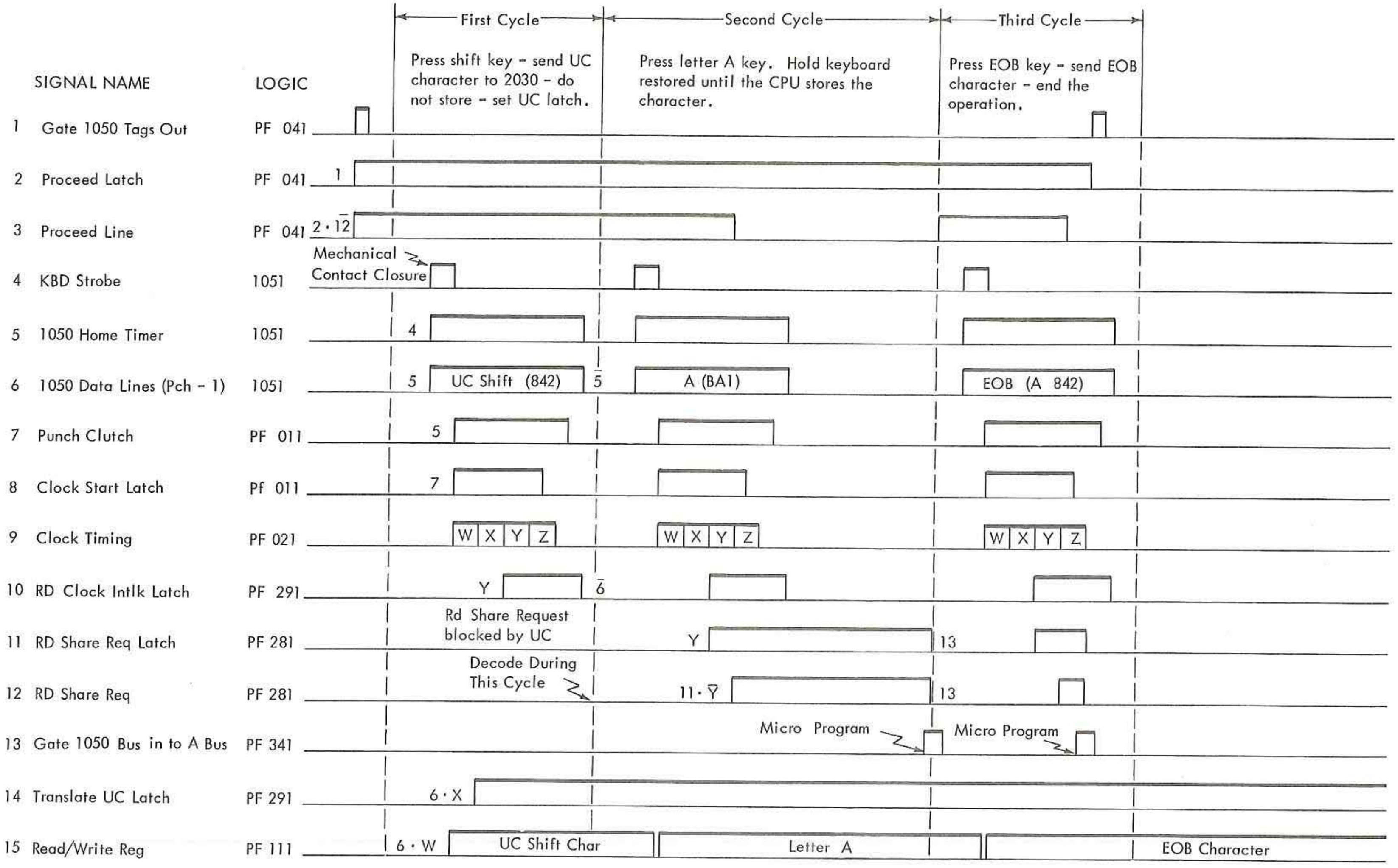


Figure 5-72. Read Inquiry Timing Chart

End Conditions

The operator can press the alternate coding key and the 5 (EOB), 6 (EOT) or 0 (Cancel) keys, and the character sent to the attachment causes an end operation.

1. EOB or EOT. These characters cause a normal end and activate the TT-2 line (End). The microprogram now branches to the end routine or status check and restore. See Figure 5-63 for the beginning of the routine. The proper operation would result in EOB or EOT being sent with zero-count in the UCW status byte. The status restore flow chart, Figure 5-62, sets the end conditions in status bytes and returns to the back-up ROAR for return to the previous microprogram.

2. CANCEL. This character causes an end operation with the TT-0 position active. The result of this end operation is to cause device- end, channel-end, and unit-exception to be set in the unit-status byte and an I/O interrupt condition established.

Split Entry

It is possible to split the entry on a read inquiry between the keyboard and reader-2 at the 1050. Once the entry from keyboard is finished, the operator can switch the reader-2 to home loop. Press the reader-start key, and the rest of the inquiry simulates a normal read reader-2 operation until reader stop code is sensed or the reader-stop key is pressed. At this time the control returns to the keyboard.

READY SHARE-REQUEST OPERATION

- Any time the 1050 is made operational, a share-request is initiated.
- Channel-end and device-end conditions are set during the share routine.
- The CPU is alerted to the fact the 1050 is now operational.

  The 1050 operational line is testing for two things:

  1. The CPU Connect Switch is on and the 1051 power is on.
  2. RDR-1 home line is active.

With these conditions, the 1050 is considered to be operational and the TT-3 line will so indicate this. If, for some reason, the 1050 were to become not operational for a period of time, it would be very nice if we could somehow alert the CPU that the 1050 were operational again. This is the purpose for the ready share-request operation.

Figure 5-73 is a flow chart of the ready share cycle. As long as the 1050 is operational, the sequence is never entered and no action occurs. However, if the TT-3 line were to indicate not operational, the latch called not operational is turned on. Next, a test is made. Was the attachment in read or write mode?. If it were in read or write mode the intervention required routine would be satisfied before the ready share-request routine is entered.

Finally, the 1050 is made operational again, and this turns on the ready share latch. If the Intervention required indicator were on, it would be reset, and during the share-request routine, end conditions TT-2 will be activated. This causes this share routine to appear as a normal-end operation. Status is set to cause channel-end, device-end, and I/O interrupt conditions. This, in turn, allows checking and updating of the CSW.

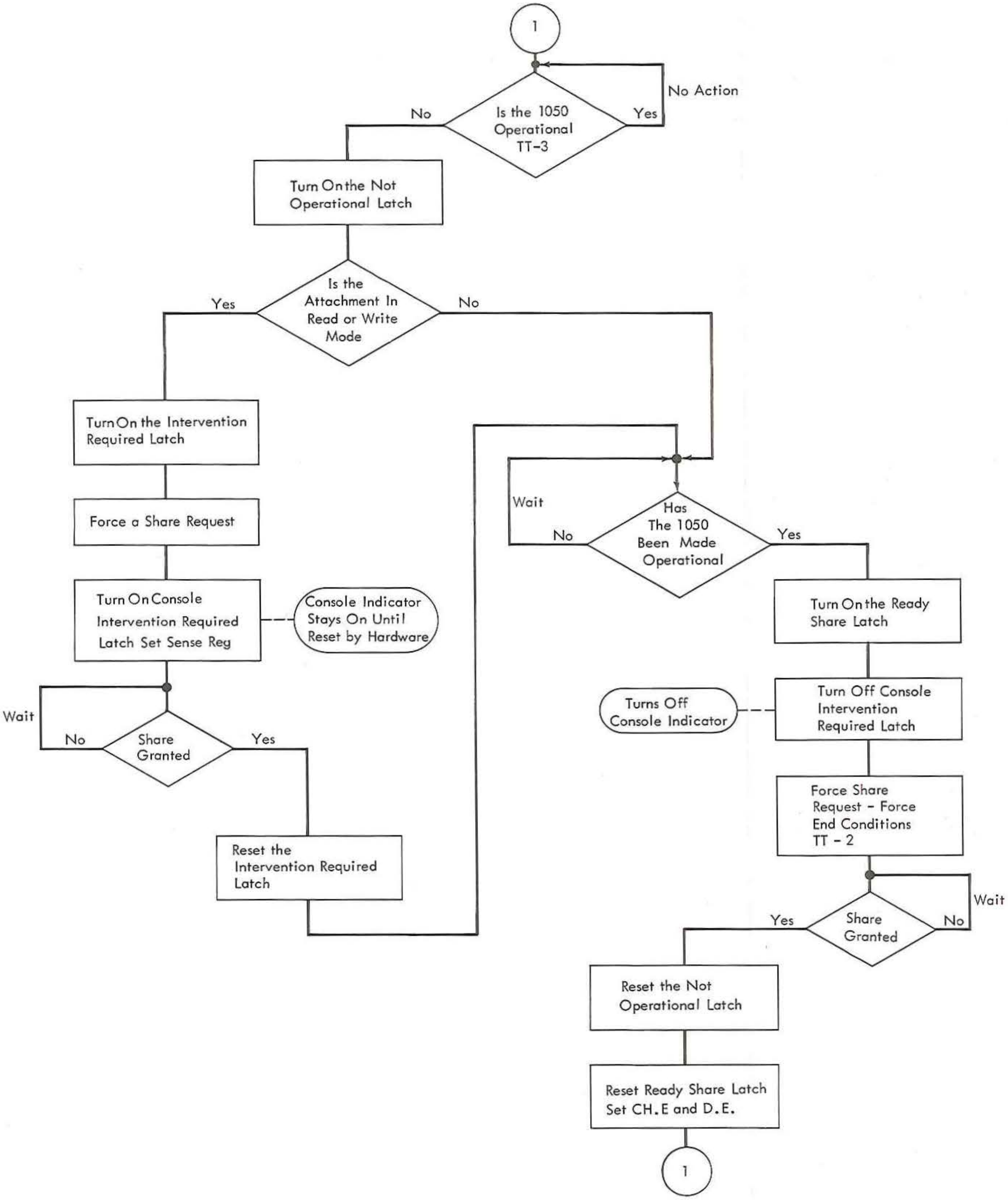

1
Is the 1050 Operational TT-3
No
Yes
No Action
Turn On the Not Operational Latch
Is the Attachment In Read or Write Mode
Yes
No
Turn On the Intervention Required Latch
Force a Share Request
Turn On Console Intervention Required Latch Set Sense Reg
Console Indicator Stays On Until Reset by Hardware
Wait
No
Share Granted
Yes
Reset the Intervention Required Latch
Wait
No
Has The 1050 Been Made Operational
Yes
Turn On the Ready Share Latch
Turns Off Console Indicator
Turn Off Console Intervention Required Latch
Force Share Request - Force End Conditions TT - 2
Yes
Share Granted
No
Wait
Reset the Not Operational Latch
Reset Ready Share Latch Set CH.E and D.E.
1

Figure 5-73. Ready Share-Request

SENSE OPERATION

- Start I/O forces a micro share-request.
- Share-request cycle reads out the sense-register conditions and stores in the area designated by data address.
- Attachment circuits are not activated.
- Operation is terminated with channel-end, device-end and interrupt conditions.

A sense operation is generally used following a command that could not be executed or following some unusual conditions that occurred. Most I/O devices will assemble sense information on the bus-in lines like data cycles. The 1050 attachment does not have the facility to assemble data bytes as sense information. The cycle is used to interrogate the sense byte stored in the CPU local storage position 04 (hexadecimal). The sense byte is read out of local storage and stored in the location of main storage specified by the data address of the CCW. This byte of information is now available in main storage; so further macroprogram interrogation can be performed.

The sense byte bit positions and their uses are:

Bit 0 Command Reject: An invalid command was directed to the 1050 console during the previous start I/O command or attempt to command chain.

Bit 1 - Intervention Required: The last operation could not be executed or completed because of a condition requiring some type of intervention at the 1050. These conditions are:

1. 1050 power off.
2. CPU Connect Switch off.
3. Reader 1 Switch off. (1051 Mod 1 only)
4. Attachment in CE mode.
5. Output not Select and Ready (write command).
6. Reader 2 not Select and Ready (read reader-2 command)
7. Attend/unattend switch in Unattend position (1051 Model 1 only).

Bit 2 Bus out check: Device or control unit has sensed invalid parity character. This position does not apply to 1050 because bus-out checking is not done in the attachment or 1050 home loop.

Bit 3 - Equipment Check: Detection of equipment malfunction during last operation. Again, this does not apply to the 1050, because all the equipment malfunctions are recognized by intervention required.

Bit 4 - Data Check: Data error detected during a read operation exclusive of bus-out check errors. The 1050 sets this position with a read operation and TT-7 position active which is data-check.

Bit 5 - Overrun: Does not apply to the 1050 because the interface functions with overrun protection.

Bits 6-7; Not used.

CHAINED DATA OPERATION

- Allows transfer of units of data from or to different areas of storage.
- Ops and flags in new CCW are ignored except for transfer in channel.

The chained data operation with the 1050 is the same as any other I/O device. It allows the utilization of different areas of main storage for the data transfers using different CCW's for each area of main storage to be used. The ops and flags for each new CCW are ignored unless transfer in channel is set. Figure 5-74 shows an example of a chained data-write operation. Note that the output record was assembled from many different locations in main storage.

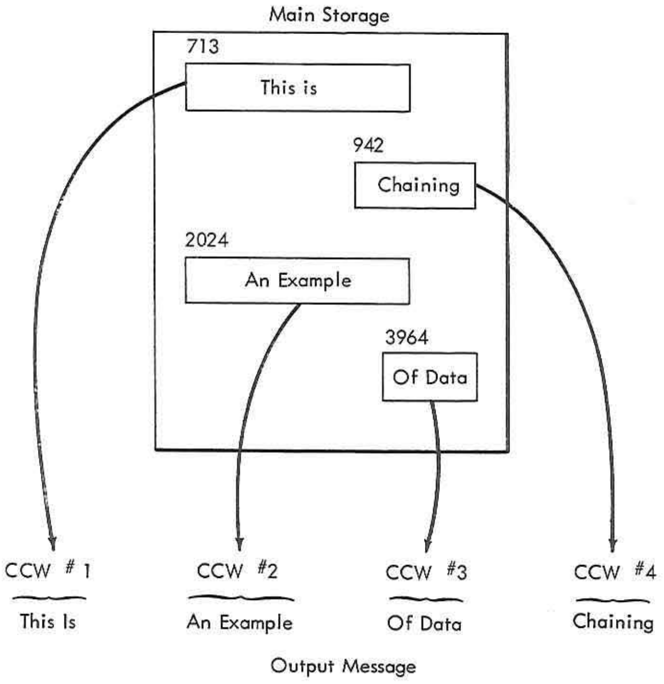


Figure 5-74. Data Chaining

The chained data routine is entered during the share routines and only when zero-count is reached. If the chained data-bit is set in the current CCW, the chained data routine is entered, (a new CCW fetched with a new data address and count number). This is repeated until the chained data flag is absent in the CCW. Now a normal end occurs along with the I/O interrupt conditions.

COMMAND CHAINING OPERATION

- Command can be chained without waiting for I/O interrupt.
- Allows command chaining of operations such as read, write, and sense without an I/O Interrupt.

Command chaining with the 1050 is the same as any other I/O device. It allows the changing of the actual command without ending and going through the I/O interrupt routine. The loading of the new CCW is treated much in the same manner as a share routine, and the sequence of specific commands continues.

The sequence of command chaining can be broken by conditions such as unit check, unit exception, or incorrect length. In this case, the operation will be terminated by I/O interrupt conditions, and the next CCW will not be fetched. An example of the use of command chaining could be:

1. Read data from reader 2.
2. Write new data to printer 2.

These two commands can be executed from one start I/O operation.

COMMAND CHAINING WITH COMMAND IMMEDIATE

- Used to initiate subsequent commands without waiting for an interrupt status and macro programming.

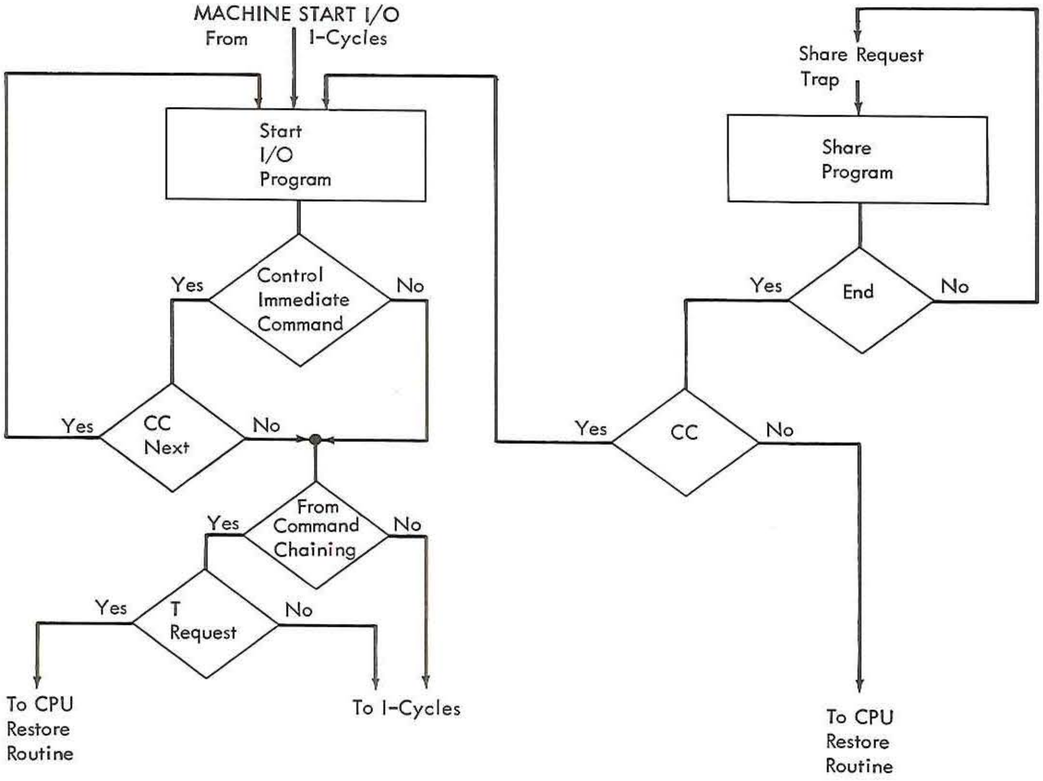


Figure 5-75. Command Chaining--Command Immediate

This function, as it applies to the 1050 microprograms, is rather unique. The problem is with the command-immediate type operations. Refer to Figure 5-49. All decision blocks marked with an asterisk refer to a command-chain operation. Figure 5-75 is a flow chart of the start I/O and share-request microprograms.

Three conditions to consider when command-chaining are:

1. Control Alarm (command immediate)
   Read/Write (normal command)

2. Read/Write
   Control Alarm

3. Read/Write
   Control Alarm
   Read/Write

The problem with a command- operation and command-immediate operations is that the microprogram must know where to return to, I-cycles or CPU restore routine (ROBAR).

Example 1:

Control Alarm
Read Operation

The control alarm is a command-immediate operation, and to perform it, the share-request routine is not entered. Upon exiting from start I/O, the read operation is chained. Therefore, the start I/O is re-entered, and the read operation set up. This entry to start I/O is treated like a share-request and, when exiting from the start I/O this time, a return to I-cycles is performed.

Example 2:

Read Operation
Control Alarm

In this example, the command immediate follows the read operation. When the read operation start I/O is exited, the return will be to I-cycles. During the data transfers the share-request operation is performed, with T-request being set each time. Finally device-end and channel-end are sensed, and the status restore is entered. Because the command-immediate is chained, the restore routine causes an entry to 1050 start I/O with T-request set. The command-immediate is executed. Now the return should be to ROBAR microprogram instead of to I-cycles because the T-request is active.

POWER SUPPLY AND INTERLOCKS

- With normal CPU power off, the 1050 powэr-on condition is under control of the 1050 mainline switch.
- With the CPU in EPO condition, the 1050 can power up only if the on-line/off-line power switch is in the off-line position (Figure 5-46).

Because the 1050 can operate off-line without the CPU, a normal power-off condition at the CPU does not affect the 1050. Power can be turned off or on, independent of the CPU, by the mainline switch at the 1050. The sequencing of the 1050 is not under control of the CPU.

If something causes the CPU to activate an EPO (Emergency Power Off) condition, power drops at the 1050. The EPO line drops the EPO relay (HD1) in the 1051, and now manual intervention is required to bring power up at the 1050. The on-line/off-line power switch in the 1051 must be turned to the off-line position. However if the EPO condition at the CPU is corrected, and the EPO relay again picks, power drops at the 1051, and the switch must be returned to the on-line position.

These are the only power-supply sequencing interlocks between the 1050 and the 2030.

CONTROL ALARM OPERATION

- A normal start I/O command that sets TA-4 causing an alarm signal.
- Channel-end and device-end conditions are set along with condition code 1 and return directly to I-cycles.
- Audible alarm in the 2030 rings for 2 seconds for each execution of this command.

This special feature gives the system a method of alerting the operator that something is wrong somewhere in the system. Its purpose is primarily to give an audible alarm to alert the operator. No other action occurs, and the start I/O is terminated with channel-end device-end, and condition code 1 set with a return to I-cycles.

SHARED I/O PANEL

- The shared portable CE Panel is used for CE mode of operation.
- 1050 commands and functions can be initiated by setting TA Register.
- Condition of 1050 and attachment is indicated by displaying TT.
- Write data and control characters can be set and are displayed both before and after translation.
- Read data and control characters are displayed both before and after translation.
- In run mode, the CE panel provides dynamic display of controls and data.

See Figure 5-76 for the overlay mask used with the shared CE panel for 1050 usage.

SWITCHES

DATA ENTRY: Used with the rotary switch to control Tags Out and Bus Out.

DATA ENTER: Used with data entry to gate Tags Out and act as a single cycle bus out in write mode. In read mode, it simulates single cycle handling of a share request for the character that is read.

LAMP TEST: Test that all indicators are operational.

CE RESET: OR'ed with machine reset to become Reset Attachment. Resets Clock Start latch and sets the clock triggers so they are conditioned for a start. Also resets other latches in the attachment circuitry.

START AND STOP: Have no function when shared CE panel is used with 1050 attachment.

ROTARY SWITCH: Must be off when connecting or disconnecting the shared CE panel. The three active positions are:

1. Gate Tags Out. Simulates microprogram control of outbound tags. Used in conjunction with the Data Entry and Data Enter switches to set up a read or write run condition and perform other Tag functions.
2. Gate Bus Out. In this position during write share requests. Used with data-entry and either data enter for single cycle, or run for continuous operation.
3. Gate Bus-In. In this position during read share-requests. Used with either data-enter for single cycle, or run for continuous operation.

CE SELECT OUT SWITCH: With any pending attachment share request will bring up 1050 Op In.

CE EXIT MPX SHARE: Simulates the microprogram exit from a share routine and drops 1050 Op In.

Data Enter Mode Switches

CE SWITCH: With the CE switch on, the microprogram control of the 1050 can be simulated independent of CPU. CPU can be operating and the attachment will propagate select-out and ignore polling. With the CE switch off, the panel monitors 1050 console operation under CPU control in run or single-cycle mode. Also, by setting up run and micro share-request, the multiplexor channel polls the attachment.

LINE SWITCH: Has no function with basic home loop 1050 console.

RUN SWITCH: Simulates continuous share-for read or write modes. This switch must be off when the rotary switch is in gate-tags-out position.

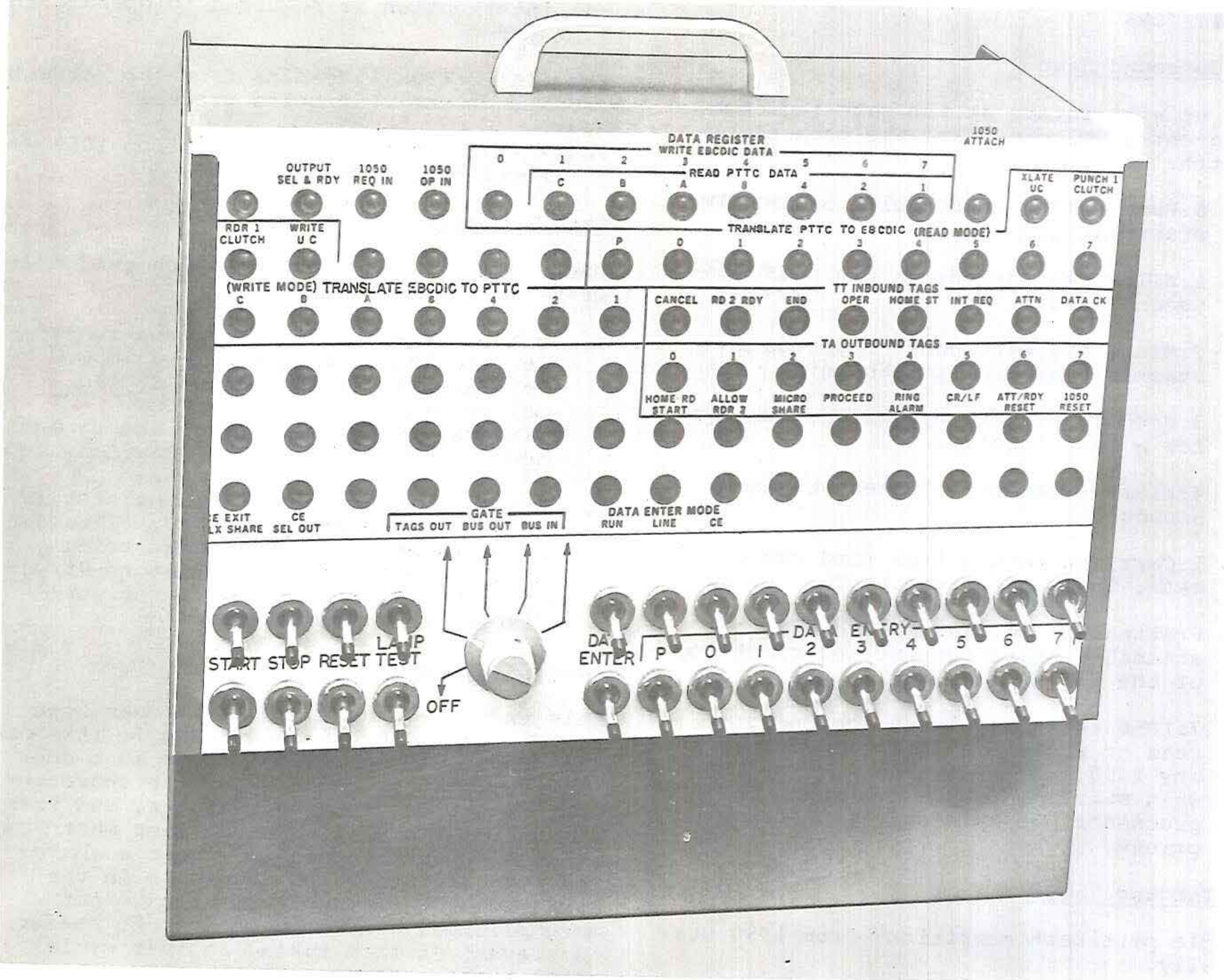


Figure 5-76. Shared CE Panel Mask and Switches

INDICATORS

TA Outbound Tags

Set up with rotary at the gate-tags-out, data-entry switches, and the data-enter switch.

Pos 0:Home reader start pulls on the 1050 start latch.

Pos 1:Allow reader-2 drops the hold on the second reader.

Pos 2:Micro share-request simulates a stacked share-request at unit.

Pos 3:Proceed unlocks typewriter keyboard for a read-inquiry.

Pos 4:Alarm (feature) if present alarm sounds

Pos 5:Carriage return line feed causes CRLF, independent of the EOL contact.

Pos 6:Attention-reset simulates taking attention share-request and resetting of the latch.

Pos 7:1050 reset simulates the end of a read or write operation and deselects any 1050 components that are selected. When starting an I/O operation from the processor, an automatic restore selects printer 1.

TT Inbound Tags

Sample pertinent conditions from 1050 statically.

Pos 0: Cancel indicates the cancel key on keyboard was pressed.

Pos 1:Reader-2 select and-ready indicates the unit is ready to receive a read command.

Pos 2:End can be obtained in several ways:

1. On the keyboard input, pressing the EOT or EOB key.

2. On the card input, an EOB or EOT character punched in a column or the trailing edge of the card.

3. On the paper-tape input, an EOB or EOT punched in the paper tape.

Pos 3 (1050 Operational): - 1050 voltages present and CPU switch on.

Pos 4(Home Reader Start): Indicates the 1050 start latch is on.

Pos 5: Intervention-required: Operator intervention is required to operate the system.

Pos 6(Attention): Results from the request key on the 1052 being pressed.

Pos 7 (Data Check): PTTC data from 1050 was received as even parity.

Data Register

Data register is common for both read and write.

1. Read. Data entering the data register is PTTC data coming from the 1050. This data is translated to EBCDIC before being gated to the CPU. (Indicators directly below the data register display this translation.)

2. Write. Data entering the register is EBCDIC data coming from CPU. This data is translated to PTTC before being gated to the 1050. (Indicators directly to the left of the data register display this translation.)

Punch 1 Clutch

This indicates that a character has been read by the 1050 and is waiting in its home register. The attachment clock must be started at 34 ms cycle rate, the character must be set in the data register, and translated to EBCDIC. The attachment must now attempt to secure polling, cause a share-request, and store the character in the data field of storage. If this is not accomplished, a hold is put on the reader to prevent it from taking another cycle until a share-request has handled the character in the data register.

Translate UC

This indicates that the PTTC to EBCDIC translator should be translating the character in the data register as an upper case character. No share-request is taken when shifting from one case to another.

Read 1 Clutch

This is brought up from 1050 on write operation to start attachment-clock at 67.5 ms cycle rate. Attachment-clock runs continuously during write operation, A share-request is attempted during the cycle. If it is not honored, no strobe will be sent to the 1050, its home timer will not run, and no clutch will be released.

Write Up Case

This indicates the EBCDIC character in the data register is a PTTC upper case character. When the characters change from one case to another, the shift character is strobed to the 1050 during one cycle, and the PTTC character is strobed to the 1050 during the second cycle. No share is requested until the PTTC character is strobed to the 1050.

Output Select and Ready

This indicates an output device on the 1050 is ready to receive data from the 1050.

Reader 1 Home

On a 1050 N1 and M1, CPU-connect should pull on reader-1 home.

1050 OP IN

This indicates the 1050 op-in line is active as a result of being polled with a share-request waiting.

1050 Req In

This indicates that the 1050 request-in line is active in the attachment circuits.

REVIEW QUESTIONS

COMPREHENSIVE INTRODUCTION

1. The 1050 console system will consist of four major units:
   a. ________
   b. ________
   c. ________
   d. ________

2. The 1050 console system operates with the 2030 on the ________ interface.

3. The code transmitted to and from the 1051 is Perforated Tape and Transmission (PT&T). It is translated to and from ________ in the 2030 attachment.

4. The PT&T character composed of the BA 842 bits (will/will not) be translated and stored in the CPU.

5. The EBCDI character represented by the byte 00110101 (will/will not) be translated and sent to the 1051.

6. Draw a data flow of a write operation to the 1052 printer. Start from R-register in the CPU to the printer in the 1050.

7. Draw a data flow of a read operation from the keyboard to the R-register.

8. The 1050 N1 differs from the 1050 Mod 1 in that ____________________.

9. The only 1050 home loop feature that will not function on line to the 2030 is__________.

10. The ________ or cable connecting the CPU and the 1050 can be broken into four parts:
   a. ________
   b. ________
   c. ________
   d. ________

11. Commands to the 1050 can be listed in five categories:

   a.

   b. ________

   c. ________

   d. ________

   e. ________

REVIEW QUESTIONS-1051 ATTACHMENT CLOCK

1. The attachment clock is driven by an oscillator and one clock cycle results in the timed outputs of ______, ______, ______, and______.

2. When in write mode, the clock will run (continuously/intermittently) and each complete cycle is ______ms long.

3. List the On or Off condition of the following clock controls when the clock is stopped:
   a. Oscillator start ______
   b. TR-1 ______
   c. TR-2 ______
   d. Oscillator ______
   e. C2 ______
   f. C4 ______

4. When in read mode, the clock will run (continuously/intermittently) and each complete cycle is ______ms long.

5. When in write mode, the clock can be stopped when the 1052 printer is performing a carrier return by the line ______.

REVIEW QUESTIONS--CONTROL IN-TT

1. The TT lines (controls in) are used in the microprogram to ____________________.

2. A ROS word contains the statement, TT KL->Z. (K = 2). This is checking for __________.

3. The TT-5 line is active. This will indicate to the microprogram that ____________________.

4. The fact that end conditions have been sensed during a read operation is signified by TT __________.

5. TT0 can be set only if read inquiry is active and the character decode is ________ bits.

REVIEW QUESTIONS--CONTROLS OUT-TA

1. The TA lines (controls out) are used in the microprogram to ____________________.

2. A ROS word contains the statement, 0 +

KH->TA. (K = 12). The operation being initiated is ________

3. The attachment is in write mode (TA-0 on). The microprogram gates TA out with this byte on the Z bus-00000101. What three actions will result?

   a.

   b.

   c.

4. TA-6, if on, will perform what function? ____________________.

5. Could the combination of bits, 11010000, be a legitimate command to the 1050? Why? ____________________

REVIEW QUESTIONS--SHARE-REQUEST CONTROLS

1. The 1050 attachment (does/does not) need polling circuits during a start I/O. Why?__________________

2. The 1050 is installed first on the channel poll lines. A start I/O is performed to the MPX channel. The SEL-I pulse must be handled by the 1050 attachment circuits. True/False.

3. Refer to Figure 23. The prepare to share latch will not reset. Will the MPX channel poll lines function properly during a start I/O? Why?

   ____

4. When 1050 request-in is sent to the channel controls, the request is granted immediately. (True/False) Why? ____________________

5. Why does the 1050 have to be installed first or last in the channel poll lines? ____________________.

REVIEW QUESTIONS--1051-2030 INTERFACE

1. The cable connecting the CPU and the 1051 is called the ________.

2. The write portion of the interface will attach to the ________ circuits in the 1051.

3. The read portion of the interface will attach to the ________ circuits in the 1051.

4. Refer to Figure 5-53. When and for how long will the restore line be active? ____________________.

5. What major function does the reader-2 hold line serve during a read operation (see Figure 5-31 and 5-32)?

   ____________________.

6. Why is the punch-1 home-switch line necessary in the interface?

   ____________________.

REVIEW QUESTIONS--MICROPROGRAMS

1. The unit address of the 1050 developed during a start I/O is ________.

2. The hexadecimal address of the UCW status byte for the 1050 is ________ in MPX storage.

3. The 1050 unit status is stored in the K addressable byte in local storage designated as ________.

4. During write mode, bit-three of the UCW ops and flag byte is set on indicating ____________________.

5. Zero-count conditions set bit________ of the UCW status and bit ________ of the unit status bytes.

6. A read inquiry command has to be initiated by a request operation. (True/False) Why?__________________

7. The ________ microprogram will start any operation.

8. Data transfers are handled by the ________ microprogram.

9. End conditions and status setting are handled during the ________, ________, and ________

10. The ________microprogram is used to stop a device by macroprogram instruction.

11. The Test I/O microprogram can be used to test the status of the share request status path or to ____________________.

12. The I/O Interrupt microprogram updates the ________, store the ________, and load the ________.

13. During a start I/O, the unit is found to be busy. What status or conditions will be set (see Figure 49)? Assume no CC. __________.

14. There are six ways to trap and cause a share-request cycle from the 1050 attachment:

   a. ____________________
   b. ____________________
   c. ____________________
   d. ____________________

e. ____________________

f. ____________________

15. During a start I/O, the command byte was 00100010. This will result in (correct operation/error termination). Why? ____________________.

16. A halt I/O command is issued to the 1050. The unit is found to be active with interrupt stacked. What condition code is set? ____________________.

REVIEW QUESTIONS--WRITE OPERATION

1. The start I/O for a write operation sets TA ________ position(s) and the attachment assumes write mode.

2. Each time the write-share request microprogram is executed, the data address will be ________ and the count will be ________.

3. A write operation is normally ended when the ________ reaches ________.

4. During write mode, the attachment clock will be started when ________ and ________ are active.

5. Will the attachment clock stop during a write operation? Explain. ____________________.

6. During a write operation, the following message was sent:

```
                             E
     LOAD TAPE DRIVE 2 O WITH TAPE 29.
                             B
```

The data count was 29 when the operation started. What printed?

____________________.

7. The following message is sent to the 1052 printer: This is a test. The total characters sent to the printer are ____________________.

8. During a write operation, the PT&T/8 BA 21 bits gated to the 1052 and will cause a data check. (True/False)

9. The ________ latch indicates that a share-request has been granted and the character is in the R/W register.

10. The UCW status byte is 10000000. This will cause ________ and ________ to be set in the unit-status byte.

REVIEW QUESTIONS_READ READER-2 OPERATION

1. The start I/O for a read reader-2 operation sets TA ________ position(s) and the attachment assumes read mode.

2. A read operation is normally ended by ________ conditions or ________ count.

3. During read mode, the attachment clock will be started when ________ and ________ are active and will run one cycle.

4. The latch that stops the clock after one cycle for each character in read mode is the ________ latch.

5. During read operations, the 1050 home timer is started by ____________________.

6. During a read operation, the message read was:

```
          E
ABCD*EFGH O          *=Punch on character
          B
```

The message stored in the CPU would be ________.

7. During a read reader-2, the message in the paper tape is:

```
              E
ABCDEFGHIJKLM O
              T
```

The data count when starting is 6.

a. What is read into core? ________

b. What is read from the reader? ________.

```
                      E
8. The record ABCDEF O was read from
   reader-2.          T
```

The parity was bad on the character C (the C-bit was dropped). What was read into storage during the operation? ____________________.

REVIEW QUESTIONS--READ INQUIRY OPERATION

1. During the start I/O microprogram for read inquiry, the TA ________ position(s) will be set and the attachment is set to read-inquiry mode.

2. What indication does the operator receive that indicates a read-inquiry operation has been started? ________.

3. Does the proceed light turn off during a keyboard operation? ________.

4. What function other than lighting the proceed light does the proceed line serve? ________.

5. The cancel character (BC bits) will/will not set data check, ________.

6. What conditions and status will be set with cancel decode? ________.

7. Can a read-inquiry entry be split between keyboard and reader-2 and back to keyboard? ________.

8. Refer to question 7. What would be punched in the tape for reader-2 to allow reading from keyboard again? ____________________.

9. Will the operation end if zero-count is sensed during a keyboard read, or will the program keep looking for end conditions? ____________________.

10. How does the program know that read-inquiry data is coming from reader-2? ____________________.

REVIEW QUESTIONS--READY-SHARE-REQUEST OPERATION.

1. A ready-share cycle will be taken any time the 1050 ____________________.

2. During the ready-share cycle, the TT-________ line is forced on to set ________ and ________.

3. The purpose of the ready-share operation is to ____________________.

REVIEW QUESTIONS--SENSE OPERATION

1. The only function of the start I/O during a sense command is to ____________________.

2. During a sense operation share-request cycle, the sense byte will be read from ________ in ________.

3. The sense byte will be stored in a location specified by the ________ of the UCW.

4. Bit-0 of the sense byte would be set to a 1 if ____________________.

5. Bit-4 of the sense byte is data check. It will be set on by a data error during a ________________ and not by ________________ errors.

REVIEW QUESTIONS--COMMAND CHAINING WITH C.I.

1. When command chaining a read command with a write command, the return to programming after execution of the write command will be to ____________________.

2. A command-immediate is executed via start I/O and is command chained with a read command. The return after execution of the read start I/O will be to ____________________.

3. A read-start I/O is executed. The read command is chained with a C.I. The return after execution of the C.I. start I/O will be to ____________________.

REVIEW QUESTIONS--I/O TESTER

1. The I/O tester (can/cannot) be used for both CE mode and on-line operation.

2. Describe the function of the data-enter switch for:
   a. Read mode_______________.
   b. Write mode_______________.

3. The start and stop switches on the I/O tester will control cycling of the 1050. (True/False)

4. The rotary switch must be on/off when connecting or disconnecting the I/O tester.

5. The CE select-out switch when activated, will activate 1050 op-in if ____________________.

6. With the I/O tester in CE mode, the channel polling is stopped. True/False

REVIEW QUESTION ANSWERS

COMPREHENSIVE INTRODUCTION

1. a. 1050 system b. 1050 to 2030 interface c. 1050 attachment in 2030 d. CPU attachment in 1051.

2. native

3. EBCDI

4. will not

5. Will

6.

7.

8. The 1050 NI is a home-loop only device and has no line-loop capabilities.

9. Home error correction

10. Interface a. read b. write c. control d. EPO

11. a. read b.read inquiry c. write d. sense e. control.

1051 ATTACHMENT CLOCK

1. W, X, Y, Z

2. Continuously, 68

3. a.off b. off c. on d. off e. off f. on

4. 34

5. Reader/clutch

CONTROLS IN-TT

1. Test the status of the 1050 or 1051 attachment through programming

2. Attention

3. Intervention is required at the 1050

4. 2

5. BC

CONTROLS OUT-TA

1. Set run conditions or initiate operations to the 1050 system.

2. Read reader-2.

3. a. Reset of write mode in the attachment.
b. Carrier return and line feed of all printers selected.
c. Reset of the 1050 home loop.

4. Reset the attention latch.

5. No. A read and read inquiry is being attempted at the same time.

SHARE-REQUEST CONTROLS

1. Does not. The unit address is fixed at 1F (31) and the microprogram branches on this fact.

2. False. With the 1050 installed first, the Sel-0 pulse generated by the start-select-out latch will go directly to the channel and, if Sel-I returns, it goes to the start-select-out and resets it.

3. No. The Sel-I pulse turns on the 1050 op-in latch, preventing the Sel-I from returning to the channel circuits.

4. False. If any MPX channel unit has op-in status, the 1050 must wait for the request.

5. It is physically located in the CPU and has to be polled first or last due to cable limitations.

1051-2030 INTERFACE

1. Interface.

2. Reader 1.

3. Punch 1.

4. Any time run mode is made active. For 25 ms.

5. Prevents the 1050 from running faster than the CPU can handle data (overrun).

Activates the punch- home-switch lines in the 1051 to activate the punch-1 home-loop circuits. The CPU connect switch used to be punch-1 and now serves a different function.

MICROPROGRAM

1. 1F (Hexadecimal) or 31.

2. F8

3. K9

4. Auto new line (ACR) at the end of the write operation.

5. 0 and 4

6. False. The request key merely alerts the CPU that attention status is set. The read inquiry is still macroprogram controlled and can be issued any time.

7. Start I/O.

8. Share-request.

9. Status, check, restore.

10. Halt I/O.

11. Clear an interrupt when in supervisor state.

12. CSW, current I/O PSW, new I/O PSW.

13. Condition code 1.

14. a. Micro share-request
    b. Read share-request
    c. Write share-request
    d. Ready share-request
    e. Attention
    f. Intervention required.

15. High order not all zeros. Will set command reject, unit check condition code 1.

16. Condition code zero.

WRITE OPERATION

1. Zero (0)

2. Incremented; decremented.

3. Count, zero.

4) Reader/clutch, write latch.

5. Yes, any time reader/clutch line drops. This can be from a carrier-return, tabulate, or line-feed function.

6. LOAD TAPE DRIVE 2 WITH TAPE. The EOB will not print nor end the operation during write.

7. **T*his is a test. *=Lower case shift character.
**=Upper case shift character.

8. False. Data is not error checked during write mode.

9. T E

10. D.E., CH.E. Zero-count is indicated which will set device-end and channel-end.

READ READER 2 OPERATION

1. Zero, one.

2. End, zero.

3. Punch clutch, read or read inquiry.

4. Read interlock.

5. Mechanical action at the 1050, by the reader running.

6. ABCDEFGH

7. a. ABCDEF

```
                     E
   b. ABCDEFGHIJKLM  O
                     T
```

8. ABCDEF. The data-check condition was sensed and stored in the unit status as unit check and the sense byte was set with equipment check.

READ INQUIRY OPERATION

1. 1, 3

2. Proceed light turns on if the keyboard switch is set at home position.

3. Yes, any time the read-share request or punch clutch line is active and the keyboard select switch is set at the home position.

4. Unlocks the keyboard when active if the keyboard select switch is set at the home position.

5. Will not. Decode blocks data check.

6. Device End, Channel End and Unit exception.

7. Yes.

8. Reader stop character.

9. It will end the operation.

10. Reader-start latch interface line to the CPU. (TT-4)

READY SHARE-REQUEST OPERATION

1. Is made operational after such trouble is power off, intervention required, etc.

2. 2, DE; Ch.E

3. Set conditions to alert the CPU the 1050 is now ready; then set I/O interrupt.

SENSE OPERATION

1. Initiate a micro share-request operation.

2. 04,local storage.

4. A command had been rejected previously.

3. Data address.

5. Read operation, but out (write).

COMMAND CHAINING WITH C.I.

1. Because the write command was shared

with a share-request cycle, the return would be to FWX (ROBAR).

2. I-cycles. The attachment was never activated in the C.I. start I/O, and the start I/O routine is never left.

3. FWX-(ROBAR). The command immediate was entered from share-request; therefore the return is to microprogramming, not I-cycles.

I/O TESTER

1. Can

2. a. single-cycle handling of a share-request
   b. single-cycle bus out.

3. False. They serve no function with the 1050.

4. Off.

5. The attachment has a share-request pending.

6. False.

INTRODUCTION

- The direct control feature allows one byte of information to be transmitted to or read from a direct control interface.
- The direct control feature attaches a CPU to a CPU or a CPU to an I/O device.
- The write direct instruction (84) immediately places the data in storage on the bus-out lines.
- The read direct instruction (85) causes the computer to wait until the data on the bus-in lines is valid before it is entered into storage.
- Both instructions use the SI format.

The Direct-Control feature in System/360 provides a means of communicating between two CPU's, or between a CPU and external devices (8 max.) It is intended primarily for transmission of control information. A CPU communicates by using the external-interruption mechanism and the instruction Write Direct or Read Direct. Information and control signals are exchanged over the direct-control interface lines (Figure 6-1 and 6-2).

The op-code 84 is a direct control write command, and the op-code 85 is a direct control read command. Both op-codes use the SI format, and are priveleged instructions.

WRITE DIRECT

The Write Direct instruction is used to place information on the Dir-Out (Direct Control Bus-Out) lines, and Read Direct is used to take information from the Dir-In (Direct Control Bus-In) lines. The Write Direct instruction causes a byte of information (8 bits) to be placed as static signals on the Dir-Out lines. These signals may be changed by repeating Write Direct instructions or they may be allowed to remain an indefinite period. No parity is presented with these 8 bits of control information. The instruction Write Direct 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 lines (Sig-Out). 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 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 are normally 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 and the Write-Out pulse is used to insure 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 storage, provided the Hold-In signal is absent. Information on the Dir-In lines may not be valid while the Hold-In signal is active. No parity is available with the control information, but a parity bit is generated as the data is placed in storage.

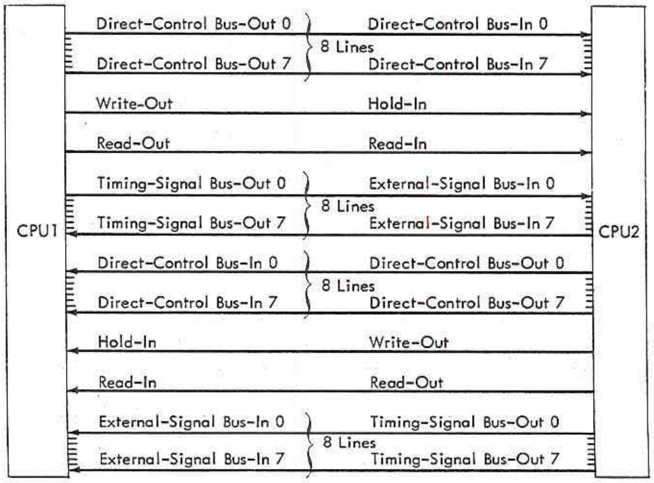


Figure 6-1. Direct Control Interface, CPU to CPU

Prior to accepting the control information, the instruction Read Direct 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 lines (Sig-Out). 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.

The function of the Hold-In signal is to prevent (or hold up) the Read operation while information on the Dir-In lines is changing and therefore invalid. When communicating between CPU's, 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 respond to the CPU's Read Out signal by dropping the Hold-In line. Since Hold-In overlaps the period when information is changing on the Dir-In lines, time is allowed to complete a data sending operation should the external device have one in progress.

The external interruption provides a means by which the CPU responds to signals from another CPU or from external equipment. These signal pulses appear on six External-Signal Bus-In lines--Sig-In-2 through Sig-In-7. When they occur they are stored until honored by the CPU. The source of the signal is identified by the interruption code in bit positions 26-31 of the old PSW (Program Status Word).

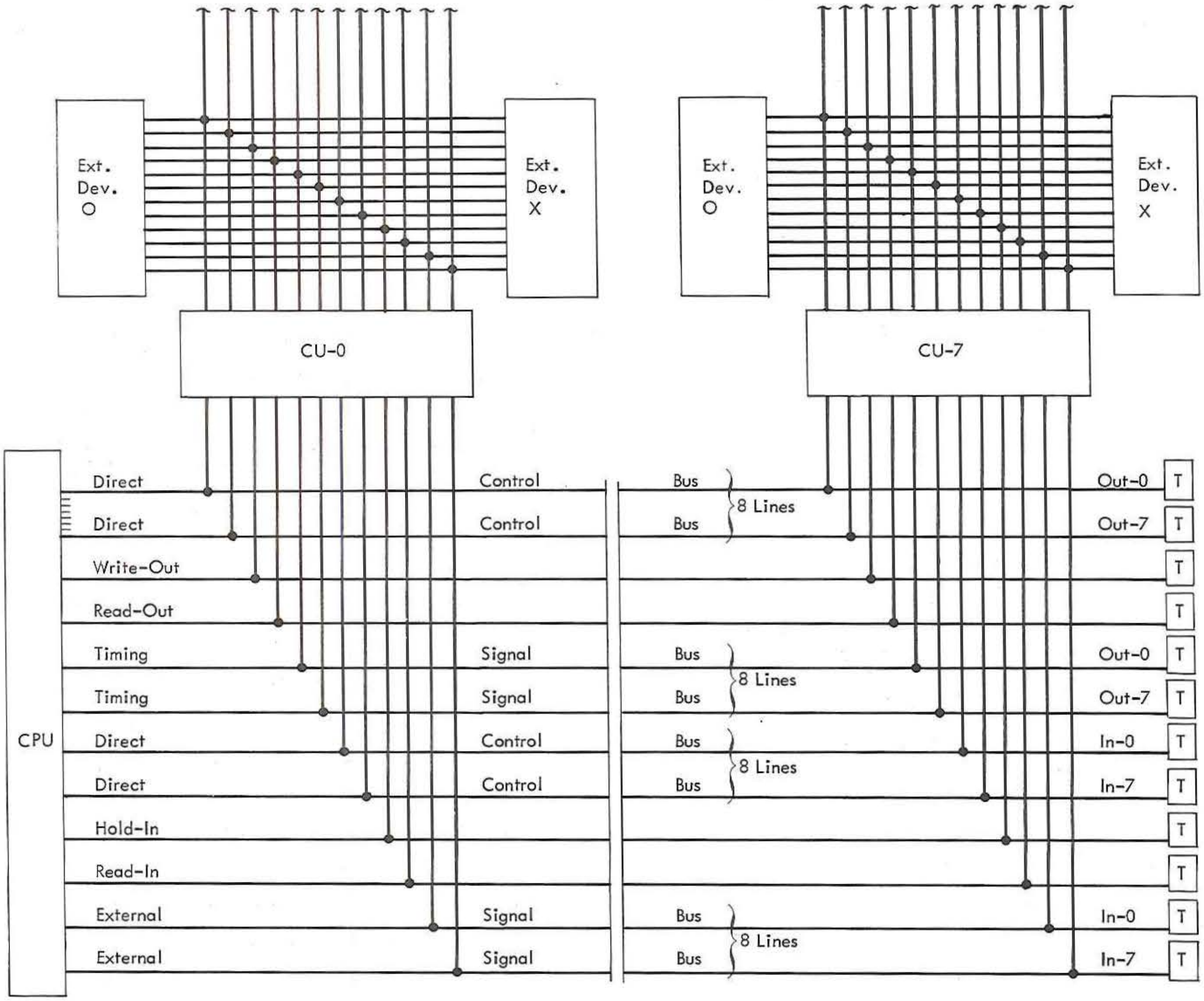


Figure 6-2. CPU to External Device

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 are placed only during the execution of the instruction Write Direct. The data on the lines represent 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 remain valid until intentionally changed, as for example, at the execution of the next Write Direct. The Write-Out pulses overlap changes on the Direct Control Bus-Out by 100 nsec, i.e., data are considered valid for at least 100 nsec before the fall of the Write-Out pulse below its up-level and until at least 100 nsec after the rise of the Write-Out pulse to 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 the data is therefore presently invalid. The down-level of Write-Out indicates the data on the Dir-Out lines in valid.

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.

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, and serve no purpose.

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 these bus lines. The Timing Pulses have a duration of 500nsec to 1000nsec. The leading edge of the timing pulses coincides with the leading edge of 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 position 2 to position 7 cause an external-signal interruption at the receiving CPU.

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 this 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 are stored in the location designated by the operand address of Read Direct instruction. The reading of the Direct Control Bus-In by the CPU is performed only when the Direct-Control Bus-In is valid and after the Read-Out pulse occurs. The data is valid on the bus until Hold-In is active.

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.

The Hold signal is active for at least 100 nsec on either side of any signal change on Direct-Control Bus-In.

After the Read-Out pusle is generated, during the execution of the Read Direct, the CPU tests for not Hold-In condition to read the Direct Control Bus-In.

Since the CPU will "hang-up" waiting for not Hold-In condition, the external equipment maintains the Hold-In inactive for at least 500 nsec after the termination of every Read-Out pulse.

The Hold-In signal can occur at any time; it does not have to be synchronized with the Read-Out pulse.

READ-IN

Read In provides no function except as a termination for the Read Out line.

EXTERNAL-SIGNAL BUS-IN

External-Signal 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 External-Signal Bus-In connects to the Timing-Signal Bus-Out of the other CPU. The Sig-Out-0 and Sig-Out-1 provide termination and serve no other function.

The External-Signal pulses have a minimum active duration of 500 nsec and a maximum active duration of 1000 nsec. The External-Signal has a minimum inactive duration of at least 500 nsec.

The External-Signal pulse may occur at any time and has no relation to the timing of other signals on the Direct Control interface.

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 when system mask bit 7 is a one and after the current instruction is completed. The interruption causes the old PSW to be stored at location 24 and a new PSW to be fetched from location 88.

A total of six signal lines comprise the External-Signal Bus-In. (Sig-In-2 through Sig-In-7). As a result of an external interruption the external signals are placed in bit locations 26 to 31 of the old PSW.

An external signal request may occur at any time. The requests are pending 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.

The Sig-In-0 and Sig-In-1 provide no function except as a termination of the Sig-Out-0 and Sig-Out-1 lines from another CPU.

APPLICATION

- The CPU can initiate a read or write command over the direct control interface.
- One byte of information is transferred for each read or write command.

Two situations below are used to illustrate the sequence of operation for the direct control feature. For illustration purposes, two CPU's are assumed to be attached by the direct control interface. It should be noted at this time that the second CPU could also be a control unit for any machine (IBM or Non-IBM) the customer desires to attach to the direct control interface.

Situation A:

CPU #1 desires to send a byte of information to CPU #2.

1. CPU #1 executes a write direct (84) instruction. The I2 field of the instruction is placed on timing-signal Bus-Out 0-7. The character in the storage location designated by the B1 and D1 field of the instruction is read out and placed on direct-control Bus-Out Out 0-7 JE-reg).

2. CPU #2 receives the timing signals on External-Signal Bus-In 0-7. Receiving the timing signals causes CPU #2 to take an interrupt. CPU #2 tests the interruption code in the old PSW and determines the cause of the interrupt. CPU #2 now executes a read direct (85) instruction and reads into storage the character present on direct control Bus-In. The storage address is determined by the B1+D1 address in the Read Direct (85) instruction. CPU #2 also transmits the I2 field of the read

direct instruction back to CPU #1 on timing signal Bus-Out 0-7.

3. CPU #1 receives the signal on External-Signal Bus-In and an interrupt occurs. CPU #1 tests the interrupt and determines that CPU #2 received the character CPU #1 transmitted. CPU #1 is now able to execute another write direct instruction if necessary.

Situation B

CPU #1 is interrupted by the direct control because CPU #2 desires to send CPU #1 a byte of information.

1. CPU #1 receives a signal on external-signal Bus-In 0-7. This signal causes CPU #1 to take an interrupt. Interrogation of the old PSW determines that a character is being sent to CPU #1 on the direct control interface.

2. CPU #1 executes a read direct (85) instruction stores the character present on direct-control Bus-in into storage. The location in storage for that character is determined by the B1 and D1 field of the instruction. CPU #1 also transmits the I2 field of the instruction out on timing-signal Bus-Out to CPU #2. These timing signals notify CPU #2 that CPU #1 received the character on Direct-Control Bus-in.

Multiple devices can be attached to the direct control interface. When multiple devices are attached to the direct control interface, the timing-signal Bus-out bit configurations will normally be used to select the I/O device desired.

DATA FLOW

- The L REG handles timing-signal Bus-out.
- The JE register handles data.

Both read direct and write direct use the SI format. During I-cycles, the operation of these op-codes is the same. That is, the op-code 84 (Write direct) or 85 (Read Direct) is placed in the G-register, the I2 field is placed in the L-register, and the base and displacement addresses select the main storage position to be used.

The micro program executes both instructions. The first step, for either operation, reads out the storage position selected by the instruction. If the instruction is a write direct, the next step places the data in the D-register and returns it to storage. The following step gates the D-register to the JE-register, gates the L-register to the timing signal out lines, and turns on the write out-signal. At this point, 17 signal lines are controlled to the interface, i.e.

8 direct control bus out lines, representing the data byte in main storage.
8 timing signal bus out lines, representing the data in the I2 portion of the instruction, and

1 write out line.

The data byte is a static signal that remains on the bus out lines until another write instruction is executed. The others are pulses with 500-1000 usec time duration.

The last step in a write direct operation resets the L-register and advances the system to the next I-cycles program.

If the instruction is a read direct, the step following the read out of main storage turns on the read out signal and gates the L-register to the timing signal bus out lines. At the same time, the FT 1-bit is set to zero to ensure the validity of the following test.

The program then goes into a test loop to wait for a signal from the attached control unit or computer that the data on the direct control bus-in lines is ready to be read. The hold-in signal controls this operation. The control unit, on receiving a read-out signal from the computer, places data on the bus-in lines. The control unit then drops the hold-in line to indicate that the read data is ready.

Dropping the hold-in line sets the FT 1-bit to a one. When this is tested during the next test loop, the data on the bus-in lines is entered into storage and the read direct instruction is completed.

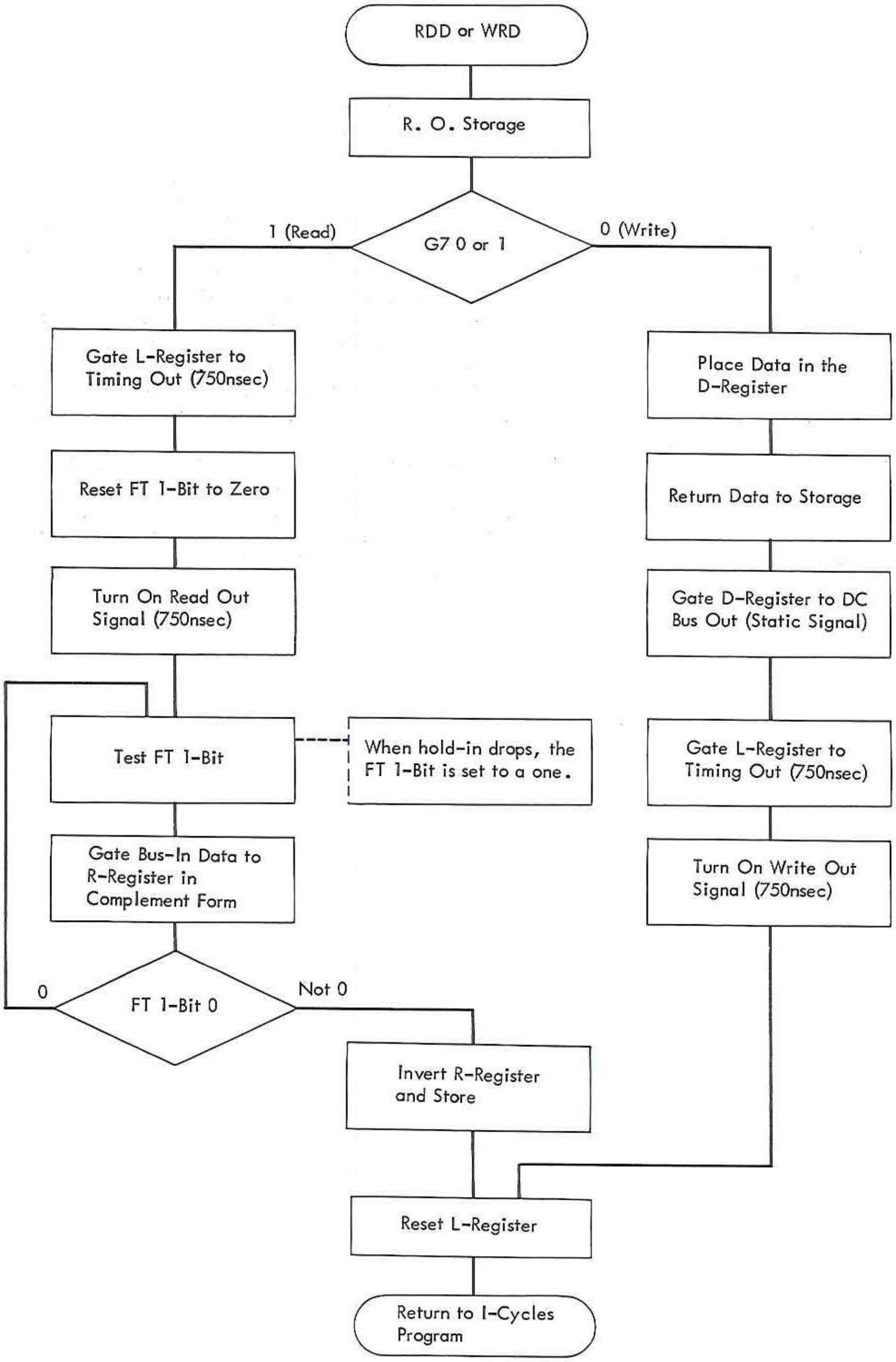

RDD or WRD
R. O. Storage
G7 0 or 1
1 (Read)
0 (Write)
Gate L-Register to Timing Out (750nsec)
Reset FT 1-Bit to Zero
Turn On Read Out Signal (750nsec)
Test FT 1-Bit
When hold-in drops, the FT 1-Bit is set to a one.
Gate Bus-In Data to R-Register in Complement Form
FT 1-Bit 0
0
Not 0
Invert R-Register and Store
Place Data in the D-Register
Return Data to Storage
Gate D-Register to DC Bus Out (Static Signal)
Gate L-Register to Timing Out (750nsec)
Turn On Write Out Signal (750nsec)
Reset L-Register
Return to I-Cycles Program

Figure 6-3. Direct Control Data Flow

CHANNEL MICROPROGRAM CHARACTERISTICS

Figures 7-1 through 7-15 are provided to aid you in following microprograms.

Op (8)	Ignored (8)	B (4)	D (12)

I/O Instruction

Start I/O	9C	10011100
Test I/O	9D	10011101
Halt I/O	9E	10011110
Test Channel	9F	10011111

Figure 7-1. I/O Instructions

MEM. PROT.	(4) 000000000000	CCW Address (24)

Figure 7-4. CAW (Location 72; Address Hex 48)

Unused (8)	Ignored (13)	Channel Address (3)	Device Address (8)

Figure 7-2. Result Of Indexing I/O Instru ction (Contents of B+D)

Command Code (8)	Data Address 00000000 (24)	Flags (5)	000 (3)	Ignored (8)	Count (16)

Command Code	
01234567	Bit
****0000	Invalid
mmmm0100	Sense
****1000	TIC
mmmm1100	Read Backward
mmmmmm01	Write
mmmmmm10	Read
mmmmmm11	Control

*-Bit Ignored
m-Modifier Bit

Flags
0-CD
1-CC
2-SLI
3-SKIP
4-PCI

Figure 7-3. CCW

Mem Prot Tag (4)	0000	Next CCW Address (24)	Status Unit (8)	 Channel (8)	Residual Count (16)

Unit Status	Channel Status
0-Attention	0-PCI
1-Status Modifier	1-Incorrect Length
2-Control Unit End	2-Program Check
3-Busy	3-Protection Check
4-Channel End	4-Channel Data Check
5-Device End	5-Channel Control Check
6-Unit Check	6-Interface Control Check
7-Unit Exception	7-Chaining Check

Figure 7-5. CSW (Location 64; Address Hex 40)

UNIT CONTROL WORD
UCW

000	001	010	011	100	101	110	111
Channel Status	Op. and Flags	Count		Data Address		Next CCW Address	
		High	Low	High	Low	High	Low

Channel Status Byte

0-Not Used (Channel Data Check)
1-Channel Control Check
2-Interface Control Check
3-First Status Received, Coded explanation below.
4-(No Name)
5-Incorrect Length
6-Program Check
7-Protection Check

0-Chain Data Address (CDA)
1-Command Chaining (CC)
2-Suppress Length Indication (SLI)
3-Skip
4-Program Controlled Interrupt (PCI)
5-Active
6-Output (Write)/Input (Read or Sense)
7-Decrement/Increment Data Address

3	4		
0	0	Handling Data; Expecting Data.	Valid Only When Active Bit Is On
0	1	Device Instructed to Stop; Expecting Status.	
1	0	Status Stacked at Control Unit.	
1	1	Status Is in Interrupt Buffer (IB).	

Figure 7-6. Unit Control Word Format

3 7 System Mask	8 11 Key	12 15 AMWP	16 31 Interruption Code	32 33 ILC	34 35 CC	36 39 Program Mask	40 63 Instruction Address

Figure 7-7. PSW

Location	Address	Length	Purpose
0	00000000 (00)	Double Word	Initial program Loading PSW
8	00001000 (08)	Double Word	Initial program Loading CCW1
16	00010000 (10)	Double Word	Initial program Loading CCW2
24	00011000 (18)	Double Word	External old PSW
32	00100000 (20)	Double Word	Supervisor call old PSW
40	00101000 (28)	Double Word	Program old PSW
48	00110000 (30)	Double Word	Machine check old PSW
56	00111000 (38)	Double Word	Input/output old PSW
64	01000000 (40)	Double Word	Channel status word
72	01001000 (48)	Word	Channel address word
76	01011100 (5C)	Word	Unused
80	01010000 (50)	Word	Timer
84	01010100 (54)	Word	Unused
88	01011000 (58)	Double Word	External new PSW
96	01100000 (60)	Double Word	Supervisor call new PSW
104	01101000 (68)	Double Word	Program new PSW
112	01110000 (70)	Double Word	Machine Check
120	01111000 (78)	Double Word	Input/Output new PSW
128	10000000 (80)		CPU Machine Check Register
129-131	10000001-10000011(81-83)		Multiplexor Log-Out Area
133-135	10000101-10000111(85-87)		Selector 1 Log-Out Area
137-139	10001001-10001011(89-8B)		Selector 2 Log-Out Area

Figure 7-8. Permanent Storage Assignments

Value of CK field	Name of latch Set by CK parity bit on Reset by CK parity bit off
5	Operational-out Control
6	MPX Operation Latch
10	Suppress Control Latch
12	MPX Channel Interrupt Latch

Value of CK field	Name of latch
3	PSW System Mask Latches These four latches are set from the contents of the R-register when the CS field specifies FB and the CK-field value is 3. The CK-field parity bit does not affect a set or reset of the latches.
9	XXH, XH and XL Latches These latches are set by the contents of the S-register when the CS field specifies FB and the CK-field value is 9. The CK-field parity bit does not affect a set or reset of the latches.

● Figure 7-9. FB Register (MPX Channel)

CK Field		
Value	Bit	Name of Latch
8	0	Bus-out Control
4	1	Address Out
2	2	Command Out
1	3	Service Out
	P	Command Start

Note: (May appear as a combination of Bits to cause multiple functions)

● Figure 7-10. FA Register (MPX Channel)

Bit	Active Condition Indicated
0	Suppress out (Diagnostic Use Only)
1	Hold-in Latch (Direct Control Feature)
2	MPX Operation Latch
3	MPX Share Request
4	IPL
5	Select in
6	Select out (Diagnostic Use Only)
7	MPX Channel Interrupt Latch

Figure 7-11. FT Bus (MPX Channel)

OPI-Operational in
AI-Address in
SVI-Service in See explanation
STI-Status in below
INT-Interrupt

Status-in Service-in branch

STI	SVI	
1	1	Op in is down.
0	1	Service in and Op in are active.
1	0	Status in and Op in are active.
0	0	None of the above. Usually Op-in is active and neither service in or status in is active.

Figure 7-12. MPX Channel Direct Branching Signals

CK Field		Name of Latch
Value	Bit	
8	0	Bus-out Control
4	1	Address Out
2	2	Command Out
1	3	Service Out

Figure 7-13. GA Register

CK Field Decode	Name
K0	Set Program Check
K1 (0 or 1)*	SX1 or SX2 Selection
K2	Operational-Out Reset
K3	Reset PCI
K4	Set Selector Interrupt Set
K5	Set Channel Control Check
K6	Set GR to Zero
K7	Not Used
K8	Set Count Ready - Not Zero
K9 (0 or 1)*	Channel Reset
K10 (0 or 1)*	Suppress-Out
K11 (0 or 1)*	Poll Control
K12	Reset Select-Out
K13	Set Channel Busy
K14	Set Halt I/O Latch
K15	Set Interface Control Check

* (0 or 1) Refers to the CK Field Parity Bit.

Figure 7-14. GB Controls (Selector Channel)

Direct Input to A Register

Bit	GS Bus	GT Bus
P	None	None
0	GR Full	Select In
1	Chain Detect	Service-in Not Service-Out
2	Not Used	Poll Control or Halt I/O Stop
3	Interrupt Condition	Channel Busy
4	CD	Address In
5	1=Chnl 1, 0=Chnl 2	Status In
6	Not Used	SX1 or SX2 Interrupt Latch
7	Chain Request	Op-In

Input to GJ Assembler

Bit	GE Bus	Diagnostic Controls Bus	Diagnostic Tags Bus
P	None	None	None
0	PCI	Count Ready, Not Zero	Input
1	Incorrect Length	SLI Flag	Suppres Out
2	Program Check	Output	SX1 ROS Request
3	Protection Check	Count Ready, Zero	Address Out
4	Channel Data Check	Not Used	Command Out
5	Channel Control Check	CC Flag	Service Out
6	Interface Control Check	Read Backward	Bus-Out Control
7	Not Used	Skip Flag	Operational Out

Figure 7-15. Selector Channel

0 Volts Bit Punch 1   5-33
0 Volts Punch Clutch   5-33
0 Volts Reader Clutch   5-34
1051-2030 Interface   5-32
2030 to 1051 Interface   5-12
1050 Commands   5-46
1050 End Operation   5-58
1050 Model 1   5-9
1050 Op In   5-95
1050 Req In   5-95
1050 UCW Format   5-45
1051 Attachment Clock   5-18
1051 Attachment in 2030   5-13
1051 Model N1   5-6
+12 CPU Connected   5-39
+12 Reader 1 Home   5-39
+12 Request   5-41
+E Bit Reader   5-34
+E Reader Strobe   5-34
-12 CPU Home Reader Stop   5-39
-12 Punch 1 Home Switch   5-38
-12V Punch -1 Paper Contact   5-33
-L CPU Home Reader-2 Hold   5-34
-L CPU Home Reader Start   5-38
-L CPU Proceed   5-36
-L CPU Reset   5-37
-L CR and LF   5-38
-L Home Reader-2 Ready   5-40
-L Home Reader Start Latch   5-40
-L Output Select and Ready   5-40
-L Restore   5-37

Address-In   2-14
Address-Out   2-10
Address-Out Latch   3-8
AMWP   1-12
Attend-Unattend Switch   5-11, 5-60
Attention   5-49
Attention Share-Request   5-73

Bus-In   2-9
Bus Lines   2-8
Bus-Out   2-9
Bus Out Check   5-86
Bus Out Control   3-8

CAW   1-8
CCW   1-9
CCW Ending Procedure   4-32
CCW Flag Operations   3-33, 4-41
CD Flag Bit   4-42
CE Exit MPX Share Switch   5-92
CE Reset Switch   5-92
CE Select Out Switch   5-92
Chain Command Control   2-20
Chain Data Address   3-34
Chained Data Operation   5-86
Channel Command Word   1-9
Channel Control Checks   3-50
Channel Instructions   1-2
Channels   1-2
Channel Selection   4-19
Channel Share Priority   4-30
Channel Status Word   1-10
Clock Operation During Read Mode   5-21
Clock Operation During Write Mode   5-19
Clock-Out   2-18
Clock (Selector Channel)   4-15
Code Translation   5-4
Command Address Word   1-8
Command Chaining   3-34, 4-45, 4-48
Command Chaining Operation   5-87
Command Chaining with Command
  Immediate   5-88
Command-Out   2-12
Command-Out Latch   3-8
Command Reject   5-86
Command-Start Latch   3-8
Commands to the 1050   5-15
Condition Code   1-12
Control   5-15
Control Alarm   5-47
Control Alarm Operation   5-91
Control No-Op   5-47
Controls   5-12
Controls from the 1051 to the 2030   5-39
Controls from the 2030 to the 1051   5-34
Controls In-TT   5-13, 5-24
Controls Out-TA   5-13, 5-25
Console Channel Operation   5-1
CPU Attachment in 1051   5-11
CPU Switch   5-8, 5-11
CSW   1-10
CSW Store   3-29
Current PSW   1-12

Data Check   5-86
Data Enter Mode Switches   5-92
Data Enter Switch   5-92
Data Entry Switch   5-92
Data Flow   3-3, 5-16, 6-5
Data Register   5-94
Data Transfer   2-22, 5-81
Direct-Branch Signals   3-13
Direct-Control Bus-In   6-3
Direct Control Bus-Out   6-3

EBCDI   5-3
Emergency Power Off   5-41
End Conditions   5-84
Ending Procedure   3-24
End Operation   2-22
End Operation (Zero-Count and EOB
  or EOT)   5-72
End Operation (Zero-Count Only)   5-72
EPO and DC Voltage Exchange   5-41
Equipment Check   5-86
Error Detection   4-59
Errors Which are not Program Checks   3-48
Extended Binary-Coded Decimal
  Interchange   5-3
External Mask Latch   3-10
External-Signal Bus-In   6-4

FA-Register 3-8
FB-Register 3-10
FI-Bus 3-14
First Share-Request Cycle 5-56
FT-Bus 3-13
FWX-Register 3-14

GA Register (Tag Line Control) 4-4
GB Controls 4-5
GC and GD Registers 4-6
GE Register 4-6
GF Register 4-6
GG Register 4-7
GH Controls 4-7
GI Bus 4-8
GJ Assembler 4-9
GJ Bus 4-8
GK Register 4-9
GO Register 4-10
GR, GS, and GT Busses 4-11
GR Register 4-10
GU and GV Registers 4-12
GW, GX Registers 4-12

Halt I/O 3-45, 4-53
Halt I/O Microprogram 5-49
Hold-In 6-3
Hold-Out 2-7

Inbound Tag Lines 2-14
Initial Program Load 3-42, 4-56
Initial Program Load (IPL) PSW 1-13
Initial Selection Sequence 2-21
Initial Selection and CCW Load 4-20
Input/Output Control 1-1
Input/Output Instruction Format 1-7
Input Selector Share Cycle 4-21
Input Translation PTTC/8 to EBCDI 5-4
Instruction Counter 1-12
Instruction Length Code 1-12
Interface Control Checks 3-51
Interface Lines 2-2
Interlock Lines 2-16
Intervention Required 5-49, 5-86
IPL 4-56
I/O and External-Mask Latches 3-10
I/O Commands 3-15
I/O Control Words 1-8
I/O Instructions 1-3
I/O Interface Control 3-7
I/O Interrupt 3-27, 4-33
I/O Interrupt Execution 4-40
I/O Interrupt Microprogram 5-53
I/O New PSW 1-13
I/O Old PSW 1-13

Keys, Lights and Switches (1050, Model 1) 5-10
Keys, Lights and Switches (1050, Model N1) 5-8

Lamp Test Switch 5-92
Line Switch 5-92
Load Latch 3-14
Local Storage 1-14
Logout Procedures 3-49

Memory Protect 4-60
Metering-In 2-19
Metering-Out 2-18
Microprograms 5-45
Micro Share-Request 5-49
Modifier 4-13
MPX Channel Interrupt Latch 3-10
MPX Channel Mask Latch 3-10
MPX Channel Micro Program 3-6
MPX Error Routine 3-48
MPX Errors 3-47
MPX Microprogram Entry 3-6
MPX Microprogram Exits 3-6
MPX Storage 3-5
Multiplexor Channel ROS Request 3-18
Multiplexor Operation Latch 3-10
Multiplexor Share-Cycle 3-16
Multiplexor Trap 3-19

Native Interface 5-1
New PSW Load 3-33
Ninth Share Request Cycle 5-56

Operational-In 2-17
Operational Interlock 3-12
Operational-Out 2-16
Operational-Out Control Latch 3-10
Original Microprogram Resumption 3-23, 4-40
Outbound Tag Lines 2-10
Output Select and Ready 5-95
Output Selector Share Cycle 4-26
Output Translation - EBCDI to PTTC/8 5-5
Overrun 5-86

Parity Bit Generators 5-44
Parity Checking 5-43
Pch-1 Switch 5-60
PCI Flag Bit 4-49
Perforated Tape and Transmission 5-2
Poll Control 3-12, 5-27
Poll Control - 1050 Request Key Operation 5-29
Poll Control - Start I/O - 1050 Installed First 5-27
Poll Control - Start I/O - 1050 Installed Last 5-29
Power Supply and Interlocks 5-90
Prefix Operation to 1051 5-69
Printer-1 Switch 5-60
Proceed Light 5-8
Program Checks 3-52
Program Controlled Interrupt 3-40, 4-49
Program Status Word 1-11
Program Switch 5-60
Protection Checks 3-52
PSW 1-11
PSW Bit 1-12
PSW Store 3-28
PTT Code/8 5-2
Punch-1 Attachment 5-12
Punch 1 Clutch 5-94
Punch 1 Interface 5-33

Read 5-12, 5-15
Read 1 Clutch 5-94

Read and Load Channel Address Word 4-18
Read Direct 6-1
Reader 1 Attachment 5-12
Reader 1 Home 5-95
Reader 1 Interface 5-33
Read-In 6-4
Read Inquiry 5-15, 5-46
Read Inquiry Microprograms 5-73
Read Inquiry Operation 5-73
Read Inquiry Share Requests 5-74
Read Inquiry Start I/O 5-74
Read Inquiry Status Check and Restore 5-74
Read Interface 5-33
Read Out 6-3
Read Reader-2 5-46
Read Reader-2 Microprograms 5-69
Read Reader-2 Operation 5-69
Read Share-Request 5-49, 5-70
Read Start I/O - Example 5-47
Read-Write Register 5-26
Ready Share Request 5-49
Ready Share-Request Operation 5-84
Request-In 2-6
Request Key 5-8
ROBAR 3-14
Rotary Switch 5-92
Run Switch 5-92
R/W Clock 5-13
R/W Controls 5-13
R/W Register 5-13
R/W Translators 5-13

Scan Controls 2-3
Second Through Eighth Share-Request Cycle 5-56
Select-In 2-6
Selective Reset 2-20
Selector Channel 1 Mask Latch 3-10
Selector Channel 2 Mask Latch 3-10
Selector Channel Interrupt ROS Request 4-35
Selector Channel Interrupt Trap Microprogram 4-38

Selector Share Cycle 4-21
Select-Out 2-5
Sense 5-15, 5-47
Sense Operation 5-86
Service-In 2-15
Service-Out 2-13
Service-Out Latch 3-8
Share-Request Controls 5-13, 5-27
Share-Request Microprogram 5-49
Skip 3-39
SKIP Flag Bit 4-48
SLI Flag Bit 4-48
Special Controls 2-18
Split Entry 5-84
Start and Stop Switch 5-92
Start I/O 3-15, 5-70
Start I/O; Commands to Control Units 4-18
Start I/O Microprogram 5-45
Start I/O Transfer-in-Channel Command 3-42
Status Check and Restore 5-70
Status-In 2-15
Storage Protect Key 1-12
Sub Channels 3-3
Subchannel Storage 3-5
Suppress Control Latch 3-10
Suppress Data Transfer 2-20
Suppress Length Indication 3-38
Suppress-Out 2-19
Suppress Status 2-19
System Mask 1-12

TA Outbound Tags 5-94
Test Channel 3-46
Test I/O 3-44, 4-51
Test I/O Microprogram 5-51
TIC 3-42, 4-50
Timing-Signal Bus-Out 6-3
Transfer In Channel 4-50
Translate UC 5-94
TT Inbound Tags 5-94

UCW 1-13, 3-3
UCW Active 3-30, 3-44, 3-45
UCW Not Active 3-31, 3-46
UCW Ops and Flags 5-46
Unit Control Word 1-13, 3-3
Unit Status 5-46

Voltages and EPO 5-12

Write 5-12, 5-15, 5-46
Write Direct 6-1
Write Interface 5-33
Write Microprograms 5-53
Write Operation 5-53
Write Out 6-3
Write Share-Request 5-49, 5-56
Write Start I/O 5-53
Write Status Check and Restore 5-58
Write Up Case 5-94
Write with ACR/LF 5-47

XL Latch 3-10
XH Latch 3-10
XXH Latch 3-10

FE
System
Maintenance
Library

System

cut here

Y24-3362-1

IBM
International Business Machines Corporation
Field Engineering Division
112 East Post Road, White Plains, N. Y. 10601

IBM 2030 Printed in U. S. A. Y24-3362-1