HIGH-SPEED DATA INTERFACE, MODEL 9130

HIGH-SPEED DATA INTERFACE II, MODEL 9131

HIGH-SPEED DATA INTERFACE, MODEL 9132

HIGH-SPEED INTER-BUS LINK II, MODEL 9135

HIGH-SPEED INTER-BUS LINK, MODEL 9136

Technical Manual

March 1983

Publication Order Number: 303-000270-200

GOULD
Electronics

This manual is supplied without representation or warranty of any kind. Gould Inc., S.E.L. Computer Systems Division therefore assumes no responsibility and shall have no liability of any kind arising from the supply or use of this publication or any material contained herein.

PREFACE

This manual contains or references information concerning the maintenance and installation of the Model 9131 High-speed Data Interface II (HSDII), Model 9132 High-speed Data Interface (HSD), Model 9135 High-speed Inter-bus Link (IBLII), and Model 9136 High-speed Inter-bus Link (IBL) manufactured by Systems Engineering Laboratories, Incorporated, Fort Lauderdale, Florida.

The information is presented as follows:

Section I - General Description
Section II - Programming
Section III - Theory of Operation
Section IV - High-speed Inter-bus Link

The drawings manual provides kit drawings, assembly drawings, circuit card drawings, and logic diagrams. The drawings manual, publication number 304-329131, comprises the following: High-speed Data Interface II, Model 9131; High-speed Data Interface, Model 9132; High-speed Inter-bus Link II, Model 9135; and High-speed Inter-bus Link, Model 9136.

The firmware manual contains the microprogram listings for the input/output microprogrammable processor (IOM) which controls the computer system's peripheral devices. The firmware manual, publication number 305-329131, comprises the following: High-speed Data Interface II, Model 9131; High-speed Data Interface, Model 9132; High-speed Inter-bus Link II, Model 9135; and High-speed Inter-bus Link, Model 9136.

Included in the front matter of this manual is a list of related publications which provides a reference to the recommended supporting documentation for the high-speed data interfaces (HSD) and the high-speed inter-bus links (IBL).

CONTENTS

Section Page

Section Page

CONTENTS (CONT.)

Section Page

Appendixes

ILLUSTRATIONS

TABLES

PREFACE

This manual contains or references information concerning the maintenance and installation of the Model 9130 High-speed Data Interface)HSD(Model 9131 High-speed Data Interface II (HSDII), Model 9132 High-speed Data Interface (HSD), Model 9135 High-speed Inter-bus Link (IBLII), and Model 9136 High-speed Inter-bus Link (IBL) manufactured by Gould Inc., S.E.L. Computer Systems Division, Fort Lauderdale, Florida.

The information is presented as follows:

Section I	-	General Description
Section II	-	Programming
Section III	-	Theory of Operation
Section IV	-	High-speed Inter-bus Link

The drawings manual provides kit drawings, assembly drawings, circuit card drawings, and logic diagrams. The drawings manual, publication number 304-000270, comprises the following: High-speed Data Interface, Model 9130, High-speed Data Interface II, Model 9131; High-speed Data Interface, Model 9132; High-speed Inter-bus Link II, Model 9135; and High-speed Inter-bus Link, Model 9136.

SECTION I

GENERAL DESCRIPTION

1.1 Introduction

The high-speed data interface (HSD) is an optional feature for the Gould Computers. It provides a full 32-bit parallel interface to a customer-designed device at rates up to 834K transfers per second.

The high-speed data interface (HSD) includes a special SelBUS interface which allows overlapped CPU and memory communications to and from the HSD; a simple 32-bit bidirectional data bus and the appropriate handshake control signals to interface the HSD to a customer designed device interface; and appropriate internal storage registers and sequential control logic for controlling the data flow internal to the HSD, to and from the SelBUS, and to and from the customer handshake interface.

The high-speed inter-bus link (IBL) comprises two interconnected HSD boards configured in the IBL mode.

1.2 Physical Description

The Model 9132 High-speed Data Interface (Figure 1-1) and the Model 9136 High-speed Inter-bus Link (a Model 9132 HSD configured in the IBL mode) are on a 15-inch wide by 18-inch deep wire-wrap circuit board.

The Model 9131 High-speed Data Interface II and the Model 9135 High-speed Inter-bus Link II (a Model 9131 HSD II configured in the IBL mode) are on a 15-inch wide by 18-inch deep multi-wire circuit board.

The Model 9130 High-speed Data Interface High-speed Interbus Link (a Model 9130 HSD configured in IBL mode) are on a 15-inch wide by 18-inch deep multi-layer copper circuit board.

A row of connector pins, which runs the full width of the board, provides the electrical interface to the SelBUS and to the external device(s). These pins are segmented into three groups: 184 pins in the middle of the connector for SelBUS, and two groups of 50 pins each toward the sides for the external device connections. Four smaller sets of connector pins on the opposite end of the board are provided for test purposes.

A small toggle switch on the HSD board, which is accessible to operations and maintenance personnel, provides a means for logically disconnecting the HSD from the SelBUS without removing the board from the chassis. Three sets of jumpers set the HSD's SelBUS priority and address. Jumpers are also available for configuring the HSD board for the IPL mode.

1.3 Functional Description

The functional description of all models of the high-speed data interface boards is identical. Additional information relating to the high-speed inter-bus link mode can be found in Section IV.

The high-speed data interface (HSD) is designed to provide a full 32-bit parallel interface to a customer-designed device at rates up to 834K transfer per second. The HSD provides the salient characteristics as follows.

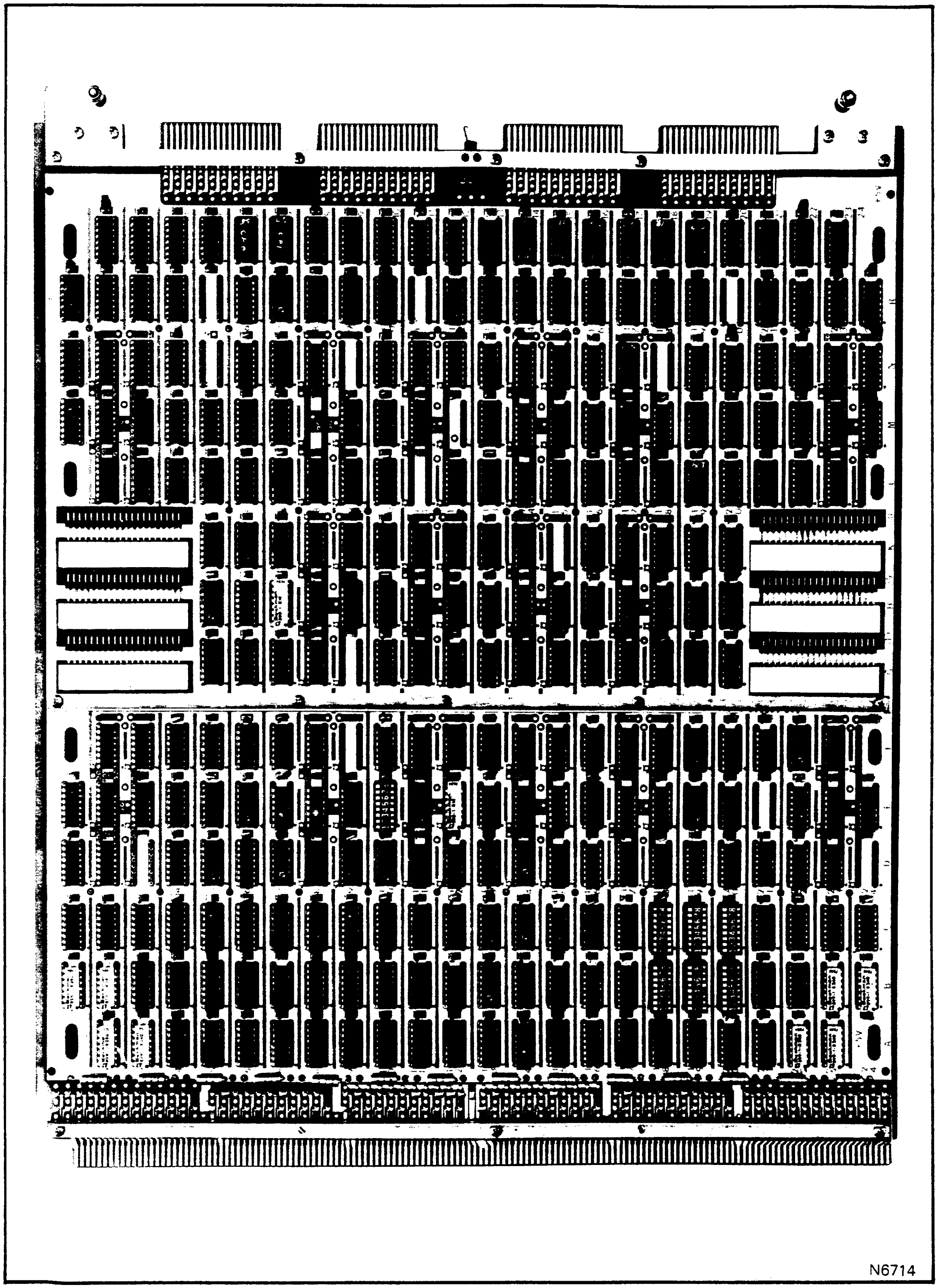


Figure 1-1. Model 9132, High-speed Data Interface

1. High-speed data transfers (up to 834K words per second; maximum rate of 1.2 microseconds per transfer)

2. Up to 64K transfers per block

3. Simple handshake protocol between HSD and customer-designed equipment

4. Maximum data transfer rates for cable lengths up to 50 feet; slower rates for cable lengths up to 250 feet

5. External mode capability which allows the customer device to provide both memory address and data for each transfer

6. Command chaining, data chaining, and transfer-in-channel

7. Automatic status posting

8. Intercomputer link capability

The HSD is essentially divided into three functional parts (see Figure 1-2). The first part consists of a special SelBUS interface which allows overlapped CPU and memory communication to and from the HSD. The second part consists of a simple 32-bit bidirectional data bus and the appropriate handshake control signals to interface the HSD to a customer-designed device interface. The third part consists of the appropriate internal storage registers and sequential control logic for controlling the data flow internal to the HSD, to and from the SelBUS, and to and from the customer handshake interface.

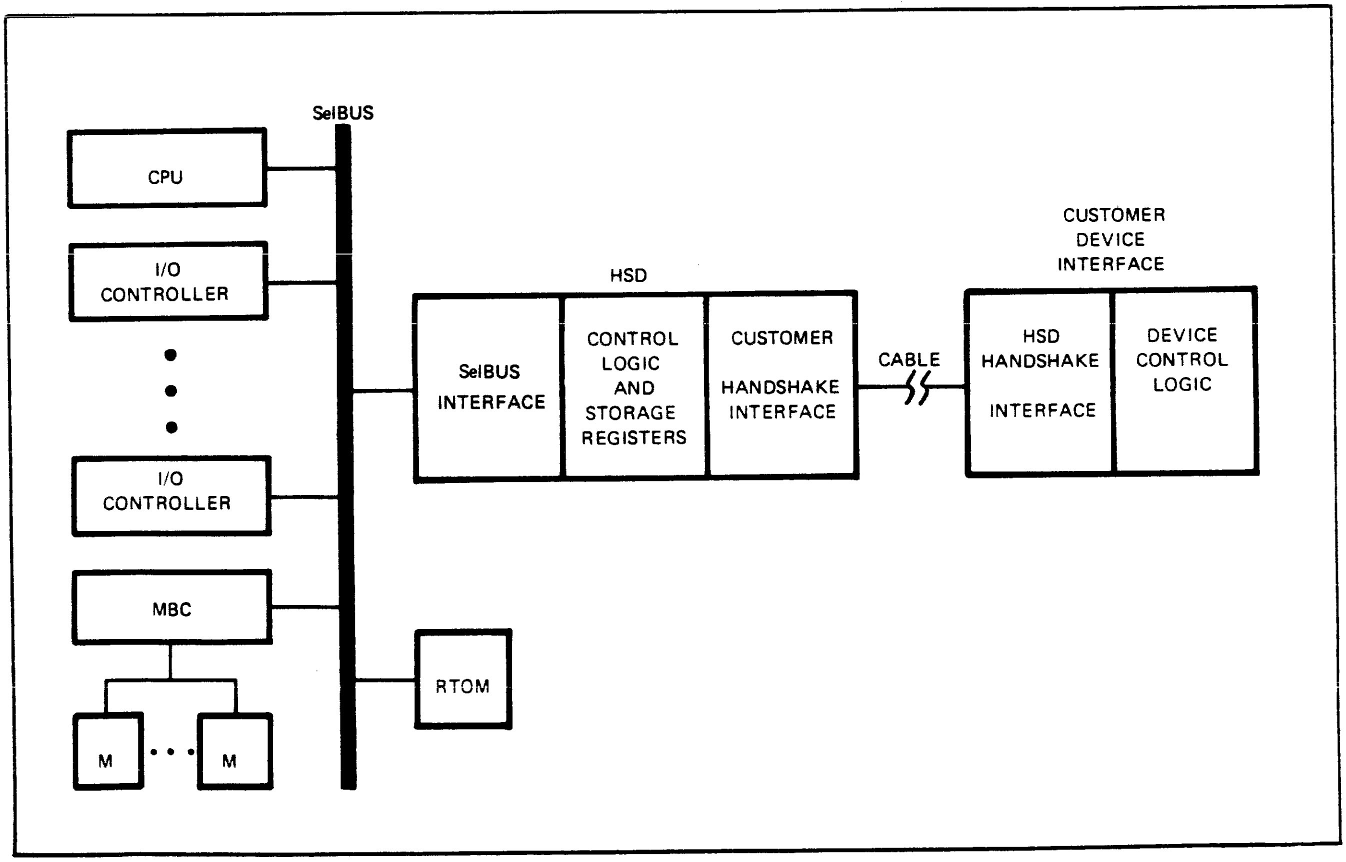


Figure 1-2. Typical 32 SERIES Computer with an HSD Interface

Typical block transfers are initiated by setting up an input/output command list (IOCL), which consists of one or more input/output command blocks (IOCB), and then issuing a command device (CD) start I/O (SIO) instruction. The above functions cause the HSD to fetch the contents of the first IOCB by using a pointer in the transfer interrupt (TI) dedicated location. Each IOCB contains the operation code (which is also sent to the device), the transfer count, the memory buffer address, and space to post error or device status when required.

The HSD controls all data transfers until the transfer count reaches zero, or an error has occurred. The service interrupt (SI) status is then automatically posted in the TI dedicated location, and an SI is generated. An error will cause the error status to be automatically posted in the current IOCB before normal SI status posting.

Automatic status posting eliminates the need for the test device (TD) instruction, except to determine whether the HSD is present in the system. When the HSD is placed in the external mode, each data transfer must be preceded by the transfer to the appropriate memory address from the customer device to the HSD. The customer device has complete control of the HSD operation while the HSD is in the external mode.

1.4 Specifications

The specifications for the high-speed data interface are listed in Table 1-1.

Table 1-1
HSD Specifications

Characteristic	Specification	
Physical dimensions	15 inches wide by 18 inches deep	
Word size		
Data	32 bits	
Microinstruction word		
Memory interface control sequencer	24 bits	
CPU interface control sequencer	8 bits	
Control memories		
Memory interface control sequencer	512 words (24 bits)	
CPU interface control sequencer	512 words (8 bits)	
Order outputs	Latched	Pulsed
Memory interface control sequencer	16	12
CPU interface control sequencer	4	10
Test inputs		
Memory interface control sequencer	38	
CPU interface control sequencer	18	

SECTION II

OPERATION AND PROGRAMMING

2.1 Introduction

This section of the manual contains the operating and programming instructions for the high-speed data interface (HSD).

2.2 Controls and Indicators

2.2.1 Controls

2.2.1.1 Off-Line Switch

The Off-line switch is on the front edge of the circuit card. It must be placed in the ON-LINE position for the HSD to communicate with the SelBUS and the CPU. The ON-LINE position is the right-hand position of the switch when the circuit card is installed correctly in the logic chassis.

2.2.1.2 HSD Physical Address Jumpers

The HSD circuit card has a set of jumpers that selects the physical address (SelBUS address) of the HSD. The address selected by these jumpers must correspond to the HSD physical address configured during the 32 CPU initial program load (IPL) of the initial configuration list.

The physical address jumpers are shown on logic drawings 130-103071, 130-103364, and 130-103554 sheet 7, which is in the drawings manual. The physical address jumpers are referenced by logic call-outs E10-1 through E10-8, and must be set to reflect the low true physical address of the HSD.

2.2.1.3 SelBUS Priority Recognition Jumpers

The HSD circuit card has a set of 21 priority recognition jumpers used to assign priorities to all system modules that have a SelBUS transfer priority higher than this HSD. The HSD's request for transfer is inhibited if any controller with a higher priority wishes to transfer at the same time. The priority recognition jumpers are shown on logic drawings 130-103071, 130-103364, and 130-103554 sheet 6, and are referenced by logic call-outs C18-2 through C18-8, C19-1 through C19-8, and C20-1 through C20-6. To assign devices with higher tranfer priority levels, the jumpers corresponding to the specific higher priority levels must be placed in the closed (ON) position; the jumpers corresponding to the priority level assigned to this HSD and all devices with lower priority levels must be placed in the open (OFF) position.

2.2.1.4 SelBUS Priority Generation Jumpers

The HSD circuit card has a set of 22 priority generation jumpers used to assign the SelBUS transfer priority of this HSD. The priority generation jumpers are shown on logic drawings

130-103071, 130-103364, and 130-103554 sheet 6, and are referenced by logic call-outs B18-1 through B18-8, B19-1 through B19-8, and B20-1 through B20-6. To assign a SelBUS transfer priority to this HSD, the jumper controlling the priority level must be placed in the closed (ON) position, and all remaining jumpers must be placed in the open (OFF) position. The priority level chosen by this procedure must correspond to the priority level used for this HSD in the priority recognition jumpers.

2.2.1.5 SelBUS Priority Enable Jumpers

The SelBUS terminator circuit card which contains a set of 22 priority enable jumpers. The priority enable jumpers must be set so that jumpers connected to priority levels assigned to modules on the SelBUS are in the open (OFF) position; jumpers connected to unassigned priority levels are in the closed (ON) position. The CPU technical manual provides a system level discussion of all jumpers that must be set to communicate on the SelBUS.

2.2.1.6 Inter-bus Link Mode Jumpers

The IBL mode is enabled by configuring the jumpers at location B20 as follows:

IBL mode	Pin 8 to 9 (IN) Pin 7 to 10 (IN)
HSD FIFO Nonoverflow Mode	Pin 8 to 9 (OUT) Pin 7 to 10 (IN)
HSD FIFO Overflow Mode	Pin 8 to 9 (OUT) Pin 7 to 10 (OUT)

The above jumper configuration is effective for HSD/IBL controllers at or above firmware level 531-322577-002.

This must be accomplished on two HSD boards and the appropriate C cable installed between the two boards before the IBL function may be performed. In addition, one of the two boards must have the high priority jumper installed between C20-08 and C20-09 (logic sheet 15). This allows priority to be resolved if both computers request to transmit data simultaneously. Otherwise, priority is established on a first come, first serve basis.

NOTE

> Once the boards are configured and cabled as an IBL, they cannot be operated as an HSD.

2.2.1.7 Function 2 Selection Jumper

HSD function 2 operation is selected by inserting a jumper at location E10 between pins 8 and 9.

2.2.2 Indicators

The HSD circuit card has no indicators that monitor HSD or system operation.

2.3 Power On/Off Procedures

DC power is supplied to the HSD circuit card by the logic chassis in which the HSD circuit card is installed. The power on/off procedures for the logic chassis are covered in the CPU power on/off procedures which are described in the CPU Technical Manual.

2.4 Operating Procedures

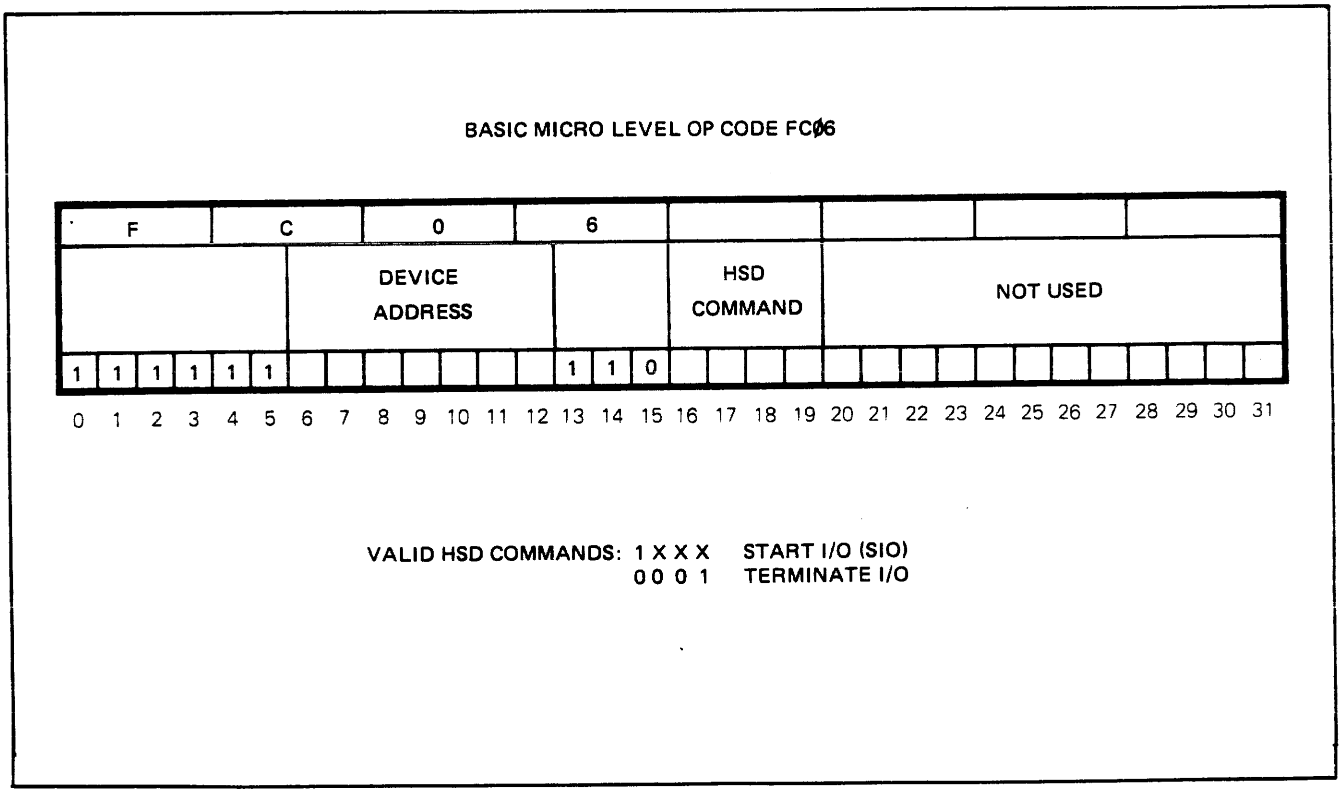
There are no special operating procedures for the HSD; however, the technical manual for the I/O devices attached to the HSD should be consulted before the HSD is operated.

2.5 Programming

The following discussion provides a brief description of the macrolevel instructions used to control and obtain status from the HSD.

2.5.1 Command Device (CD) Instruction

The command device (CD) instruction is the basic macrolevel instruction used to initiate and terminate HSD operations (see Figure 2-1).



```
VALID HSD COMMANDS: 1 X X X   START I/O (SIO)
                    0 0 0 1   TERMINATE I/O
```

Figure 2-1. Command Device (CD) Instruction Format

All combinations of the command device instruction require status to be returned to the CPU by an advance read status (ARSTX), read status (RSTX) bus sequence. The HSD will always return an all-zeros status word and will essentially ignore all illegal command combinations.

The all-zeros status will also be returned for the ARSTX, RSTX sequence preceding the start input/output (SIO) or terminate write data or order transfer (WDOT) transfers.

The command for transfer current addres (TCA) (bits 16 through 19=0010) is invalid and will always return zeros to the transfer interrupt (TI) dedicated location.

2.5.1.1 Transfer Address Word (TAW)

The transfer address word (TAW) is a 32-bit word used with the command device initialize data transfer instruction. The TAW provides the memory address of the first input/output command block (IOCB) as illustrated in Figure 2-2. The TAW is provided in a dedicated memory location for the HSD Interface. Table 2-1 lists the dedicated memory locations (transfer interrupt addresses) used by the CPU.

The controlling software system must diagram the TAW with the IOCB address before the command device initialize data transfer instruction is initiated.

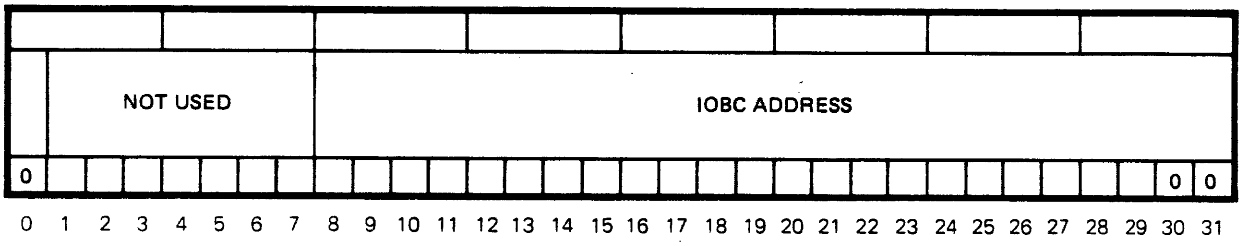


Figure 2-2. Transfer Address Word Format

Table 2-1
Transfer Interrupt (TAW) Dedicated Memory Locations

Memory Dedicated Address (H)	Typical CD or TD Address (Hex) (Note 1)	Function	I/O Controller Service Interrupt Level (Hex)
100	00	Input/Output Controller 0	14
104	04	Input/Output Controller 1	15
108	08	Input/Output Controller 2	16
10C	0C	Input/Output Controller 3	17
110	10	Input/Output Controller 4	18
114	18	Input/Output Controller 5	19
118	20	Input/Output Controller 6	1A
11C	30	Input/Output Controller 7	1B
120	40	Input/Output Controller 8	1C
124	50	Input/Output Controller 9	1D
128	60	Input/Output Controller 10	1E
12C	70	Input/Output Controller 11	1F
130	78	Input/Output Controller 12	20
134	7A	Input/Output Controller 13	21
138	7C	Input/Output Controller 14	22
13C	7E	Input/Output Controller 15	23

Notes:

1. Typical CD and TD device addresses refer to the address configurations most commonly used to address the respective I/O controller. The typical CD and TD addresses are CD or TD instructions, bits 06 through 12, which are configured as follows:

   a. Hex digit 1 represents instruction bits 06 through 08.
   b. Hex digit 1 represents instruction bits 09 through 12.

2. The TI memory location is used to hold the transfer control word (TCW) for the corresponding I/O controller, or the transfer address word (TAW) for the HSD.

2.5.1.2 Input/Output Command Block (IOCB)

The input/output command block (IOCB) consists of four 32-bit words generated by the software system. The four words of the IOCB contain the operation code, transfer count, memory buffer address or device-dependent command, and the error/device status. Figure 2-3 illustrates the format of the IOCB. Table 2-2 defines the operation code in word 0 of the IOCB.

2.5.1.2.1 HSD Operation Code

The HSD sequence for each bit in the operation code field of IOCB word 0 is described as follows:

NOTE

> All operations defined assume that a valid command device (CD) start I/O (SIO) instruction has been executed, and a valid IOCB has been fetched by the HSD.

2.5.1.2.1.1 Bit 0 - I/O Transfer Direction

When this bit is set to a zero (output data transfer), the HSD performs the following operations:

1. The HSD initiates a memory read transfer (MRT) using the contents of the memory address register as the address.
2. The HSD increments the memory address register and decrements the transfer counter.
3. The HSD accepts the data return transfer (DRT) from memory, places the data in the data buffer, and modifies the buffer pointers.
4. The HSD generates an output data ready (ODR) signal to the device and places the appropriate data on the device data bus.
5. The device stores the data and generates an output acknowledge (OA) to the HSD.
6. The HSD repeats steps 1 through 5 until the data block has been transferred, an error has occurred, or an external terminate (EXT) is generated by the device.

When a valid command device instruction to the HSD is executed, the HSD will always fetch word 0 of the IOCB, place it on the device data bus, generate an external function (EF) signal, and wait for an external function acknowledge (EFA) signal to be generated by the device. This procedure provides a means for the device to examine the HSD operation code (word 0, bits 0 through 7) and the device-dependent bits (word 0, bits 8 through 15) to determine what it should do and what will be expected by the HSD. (Refer to paragraph 3.2.4 for more detail).

When bit 0 is set to a one (input data transfer), the HSD performs the following operations:

1. The device generates an input data ready (IDR) signal to the HSD and places data on the device data bus.
2. The HSD loads the data into its data buffer, modifies the buffer pointers, and generates an input acknowledge (IA) signal to the device.

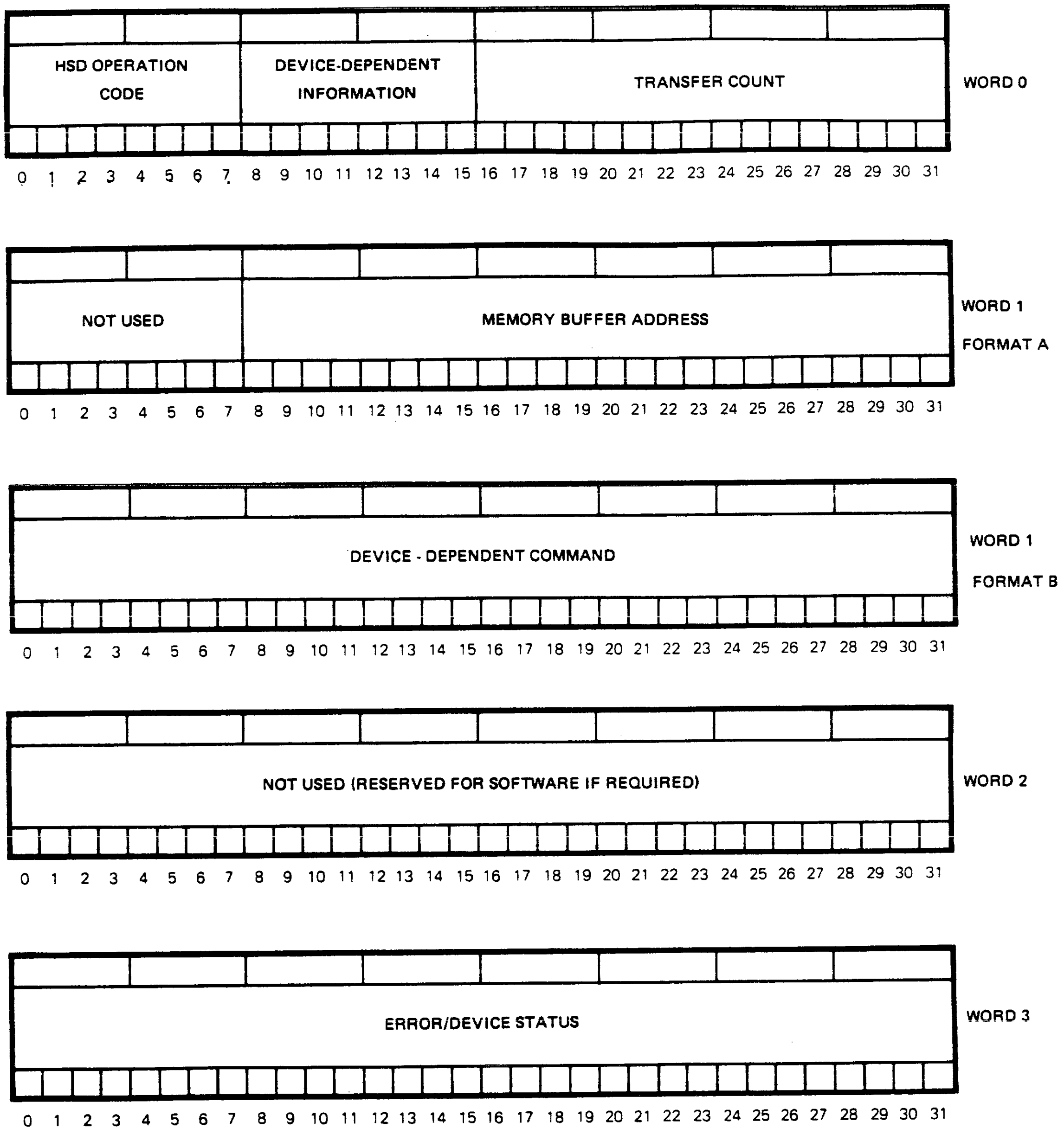


Figure 2-3. Input/Output Command Block (IOCB) Format

Table 2-2
Operation Code Definition - IOCB Word 0

Bit Number	Operation
0	1 = input data transfer, 0 = output data transfer
1	Command transfer
2	Device status request
3	Continue on error
4	Interrupt on EOB (i.e, end of current IOCB function)
5	Transfer in channel (TIC)
6	Command chain
7	Data chain
8-15	Customer device-dependent (may be device commands or a combination of commands and subaddresses)

3. The HSD initiates a memory write transfer (MWT) using the contents of the memory address register as the address.

4. The HSD increments the memory address register and decrements the transfer counter.

5. Steps 1 through 4 are repeated until the data block has been transferred, an error has occurred, or an external terminate is generated by the device.

These sequences may be altered depending on the state of op code bits 6 and 7.

2.5.1.2.1.2 Bit 1 - Command Transfer

When this bit is set to a one, the HSD performs the following functions:

1. The HSD fetches word zero and word one of the IOCB and places them in the FIFO.

2. The HSD places the IOCB first word (in this case, device command and/or address information) on the device data bus and generates an external function signal to the device.

3. The device loads the device-dependent information into its storage register and returns the external function acknowledge signal to the HSD.

4. The HSD places the second word of IOCB on the device bus and generates an external function signal to the device. The second word is also device-dependent information.

5. The device loads the device-dependent information into its storage register and returns the external function acknowledge signal to the HSD.

Normally, the use of this operation requires a second IOCB to be fetched to initiate a data block transfer since the memory address field contained device-dependent information (i.e., command chaining is usually necessary). This sequence may be altered depending on the state of op code bits 4 and 6.

2.5.1.2.1.3 Bit 2 - Device Status Request

When this bit is set to a one, the HSD performs the following functions:

1. The device generates an input status ready (ISR) signal to the HSD since the device detected the device status request bit set in the operation code that was received during the CD-SIO sequence.
2. The HSD stores the device status in the data buffer and returns an input status acknowledge (ISA) to the device.
3. The HSD requests priority to poll the SelBUS for a memory cycle, and the data buffer is loaded into the memory data register.
4. After priority has been generated, the HSD polls the SelBUS for a memory cycle. At the same time, the IOCB address is loaded into the memory destination register.
5. When the SelBUS poll is won, the bus transfer occurs and the HSD performs the next operation.

This sequence may be altered depending on the state of op code bits 4 and 6.

2.5.1.2.1.4 Bit 3 - Continue on Error

When this bit is set to a one, either command or data chaining is specified, and a parity or nonpresent memory error is encountered during data transfer, the HSD performs the following functions.

1. Posts error status in the fourth word of the current IOCB (refer to Data Transfer Termination).
2. Posts SI status in the TI dedicated location.
3. Generates an SI request.
4. Fetches the next IOCB as specified by the command or data chain op code bit.

NOTE

Continue on error does not function when an external terminate (EXT) condition takes place.

This sequence may be altered depending on the state of op code bit 4. Since op code bits 0 through 3 are not loaded when an IOCB is fetched as a result of data chaining, the continue-on-error bit will be ignored if it is specified in the IOCB following any IOCB which specifies data chaining. Conversely, the continue-on-error bit will remain in effect through all successive IOCBs specifying data chaining if it has been called out in the first IOCB which specifies data chaining.

2.5.1.2.1.5 Bit 4 - Interrupt on End-of-Block (EOB)

When this bit is set to a one and the current IOCB has been completed, the HSD performs the following functions:

1. Posts error status in the fourth IOCB word.
2. Posts SI status in the TI dedicated location (refer to Data Transfer Termination).
3. Generates an SI request.

2.5.1.2.1.6 Bit 5 - Transfer-in-Channel (TIC)

When this bit is set to a one, the HSD uses the memory buffer address in the second word of the current IOCB to fetch the next IOCB from memory. This sequence may be altered depending on the state of op code bit 4 in the current IOCB.

2.5.1.2.1.7 Bit 6 - Command Chain

When this bit is set`to a one, the HSD performs the following functions:

1. Increments the current IOCB address to the next IOCB in the sequence (i.e., increments the IOCB address by one word).

2. The HSD fetches the next IOCB from memory.

This sequence may be altered depending on the state of operation code bits 1, 2, 3, and 4.

2.5.1.2.1.8 Bit 7 - Data Chain

When this bit is set to a one, and a normal EOB occurs, the HSD performs the following functions:

1. Increments the IOCB address to the next IOCB in the sequence.

2. The HSD fetches the next IOCB from memory, but the HSD will discard the operation code information in the newly fetched IOCB except for bits 4 through 7. The new op code is not sent to the device when data chaining is specified. This sequence may be altered depending on the state of op code bits 3 and 4 in the current IOCB.

Valid combinations involving HSD operation codes are listed in Table 2-3. The definition of the basic HSD operation codes in the preceding paragraphs describes the operation codes designated with an asterisk in Table 2-3. Table 2-4 covers all other valid HSD operation code descriptions in relation to their variation from the basic operation codes previously described.

2.5.2 Test Device (TD) Instruction

The test device (TD) instruction (see Figure 2-4) is used only to determine whether the HSD is connected to the system. If the HSD if off-line or is not plugged into the system, execution of a TD 8000 will set all four condition code bits to one. If the HSD is plugged in and on-line, the execution of any TD will result in condition code bits of all zeros. Therefore, the TD should only be used to determine whether the HSD is present in the system.

Table 2-3
HSD Operation Code Combinations

Function	I/O	CT	DSR	CE	I	TIC	CC	DC	Hexadecimal
Op Code Bit	0	1	2	2	4	5	6	7	Equivalent
	1	0	0	0	0	0	0	0	80*
	1	0	0	0	1	0	0	0	88*
	0	0	0	0	0	0	0	0	00*
	0	0	0	0	1	0	0	0	08*
	1	1	0	0	0	0	0	0	C0*
	1	1	0	0	1	0	0	0	C8
	0	1	0	0	0	0	0	0	40*
	0	1	0	0	1	0	0	0	48
	1	1	0	0	0	0	1	0	C2
	1	1	0	0	1	0	1	0	CA
	0	1	0	0	0	0	1	0	42
	0	1	0	0	1	0	1	0	4A
	1	0	1	0	0	0	0	0	A0*
	1	0	1	0	1	0	0	0	A8
	1	0	1	0	0	0	1	0	A2
	1	0	1	0	1	0	1	0	AA
	0	0	1	0	0	0	0	0	20*
	0	0	1	0	1	0	0	0	28
	0	0	1	0	0	0	1	0	22
	0	0	1	0	1	0	1	0	2A
	1	0	0	0	0	1	0	0	84*
	1	0	0	0	1	1	0	0	8C
	0	0	0	0	0	1	0	0	04*
	0	0	0	0	1	1	0	0	0C
	0	0	0	0	0	1	0	1	05**
	0	0	0	0	1	1	0	1	OD**
	1	0	0	0	0	1	0	1	85**
	1	0	0	0	1	1	0	1	8D**
	1	0	0	0	0	0	1	0	82*
	1	0	0	0	1	0	1	0	8A
	0	0	0	0	0	0	1	0	02*
	0	0	0	0	1	0	1	0	OA
	1	0	0	0	0	0	0	1	81*
	1	0	0	0	1	0	0	1	89
	0	0	0	0	0	0	0	1	01*
	0	0	0	0	1	0	0	1	09
	1	0	0	1	0	0	1	0	92
	1	0	0	1	1	0	1	0	9A
	0	0	0	1	0	0	1	0	12
	0	0	0	1	1	0	1	0	1A
	1	0	0	1	0	0	0	1	91
	1	0	0	1	1	0	0	1	99
	0	0	0	1	0	0	0	1	11
	0	0	0	1	1	0	0	1	19

Table 2-4
HSD Operation Code Variations

Op Code	Description
C8,48	The combination of the command transfer bit with the interrupt on EOB bit causes the defined command transfer to occur. Upon completion of the transfer, the HSD stores SI status in the TI dedicated location and generates and SI request. (Bit 0 of the op code is a "don't care" in this case.)
C2,CA, 42,4A	The combination of the command transfer bit with the command chain bit causes the defined command transfer to occur. Upon completion of the transfer, the HSD fetches the next IOCB and continues execution. If the interrupt on EOB bit is set, the SI status is stored in the TI dedicated location, and an SI request is generated before fetching the next IOCB. (Bit 0 of the op code is a "don't care" in this case.)
28,A8	The combination of the device status request bit with the interrupt on EOB bit causes the device status function to occur. Upon completion of the device status function, the HSD stores SI status in the TI dedicated location and generates an SI request. (Bit 0 of the op code is a "don't care" in this case.)
22,2A A2,AA	The combination of the device status request bit with the command chain bit causes the device status function to occur. Upon completion of the device status function, the HSD fetches the next IOCB and continues execution. If the interrupt on EOB bit is set, the SI status is stored in the TI dedicated location, and an SI request is generated before fetching the next IOCB. (Bit 0 of the op code is a "don't care" in this case.)
8C,OC	The combination of the TIC bit with the interrupt on EOB bit causes the HSD to store SI status in the TI dedicated location, generate an SI request, and then fetch the next IOCB according to the TIC address. (Bit 0 of the op code is a "don't care" in this case.)
8A,OA	The combination of the command chain bit with the interrupt on EOB bit causes the HSD, upon completion of the current IOCB, to store SI status in the TI dedicated location, generate an SI request, and then fetch the next IOCB according to the command chain sequence.
89,09	The combination of the data chain bit with the interrupt on EOB bit causes the HSD, upon completion of the current IOCB, to store SI status in the TI dedicated location, generate an SI request, and then fetch the next IOCB according to the data chain sequence.
92,12 91,11	The combination of the command or data chain bit with the continue-on-error bit causes the HSD, upon detecting an error during a data transfer sequence, to terminate the data transfer sequence, store the error status in the current IOCB, and fetch the next IOCB according to the command or data chain sequence.
9A,1A 99,19	The combination of the command or data chain bit with the continue-on-error bit, and the interrupt on EOB bit causes the HSD, upon detecting an error during a data transfer sequence, to terminate the data transfer sequence, store the error status in the TI dedicated location, generate an SI request, and then fetch the next IOCB according to the command or data chain sequence.

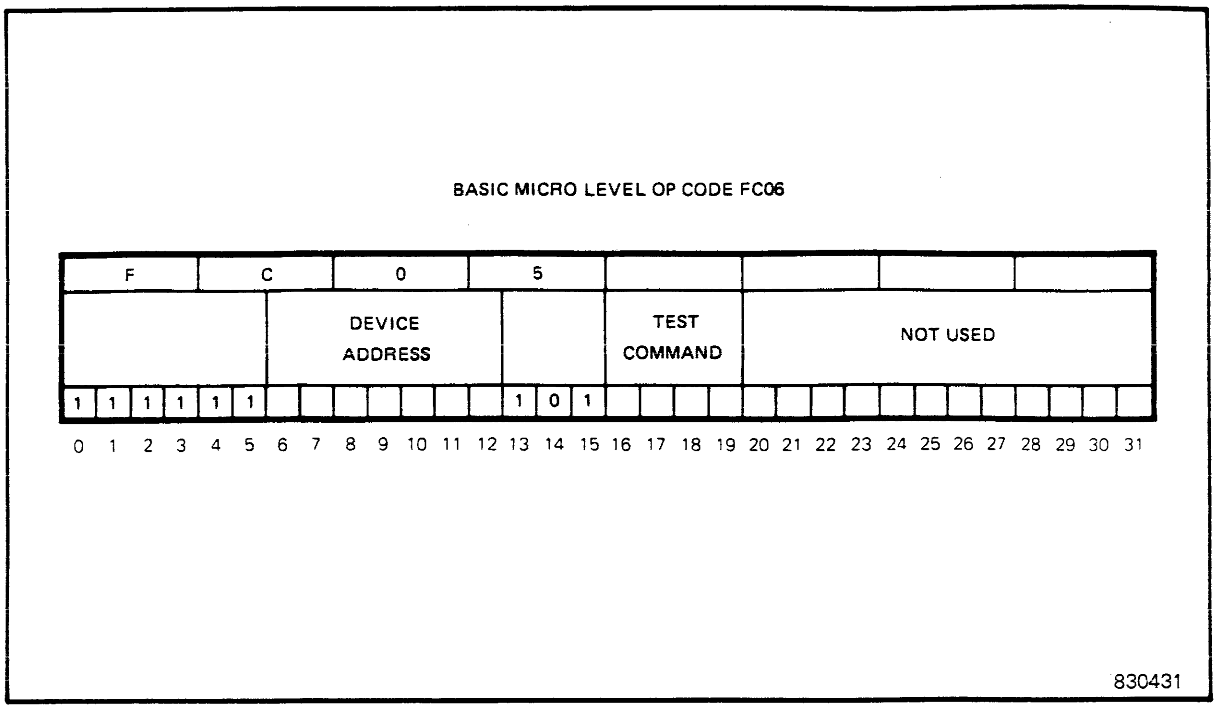


Figure 2-4. Test Device (TD) Instruction Format

2.5.3 Interrupt Control Instructions

The interrupt control instructions are used to control the priority interrupt level of the HSD. Figure 2-5 illustrates the format for the interrupt control instructions and lists the interrupt control functions commanded by the instruction. Each of the 128 priority interrupts in the CPU has a dedicated memory location that contains the address of a software program to handle the interrupt. This memory location is accessed whenever an interrupt occurs from the corresponding interrupt level. If the interrupt level is enabled and has the highest priority, the program addressed by the dedicated memory location is initiated. Table 2-5 lists the priority interrupt levels available to the CPU and the dedicated memory location (SI address) for each level.

NOTE

> Only levels 14 hexadecimal through 23 hexadecimal are applicable to I/O controllers. Interrupt levels 02 hexadecimal are dedicated to transfer interrupt.

The branch and reset interrupt (BRI) instruction is used by the software to clear the active condition of the HSD interrupt level. The format and the functions of the BRI instruction are described in the CPU reference manual.

2.5.4 Status Posting

Typical block transfers are initiated by setting up an I/O command list (IOCL), which consists of one or more I/O command blocks (IOCBs), and then issuing a command device (CD) start

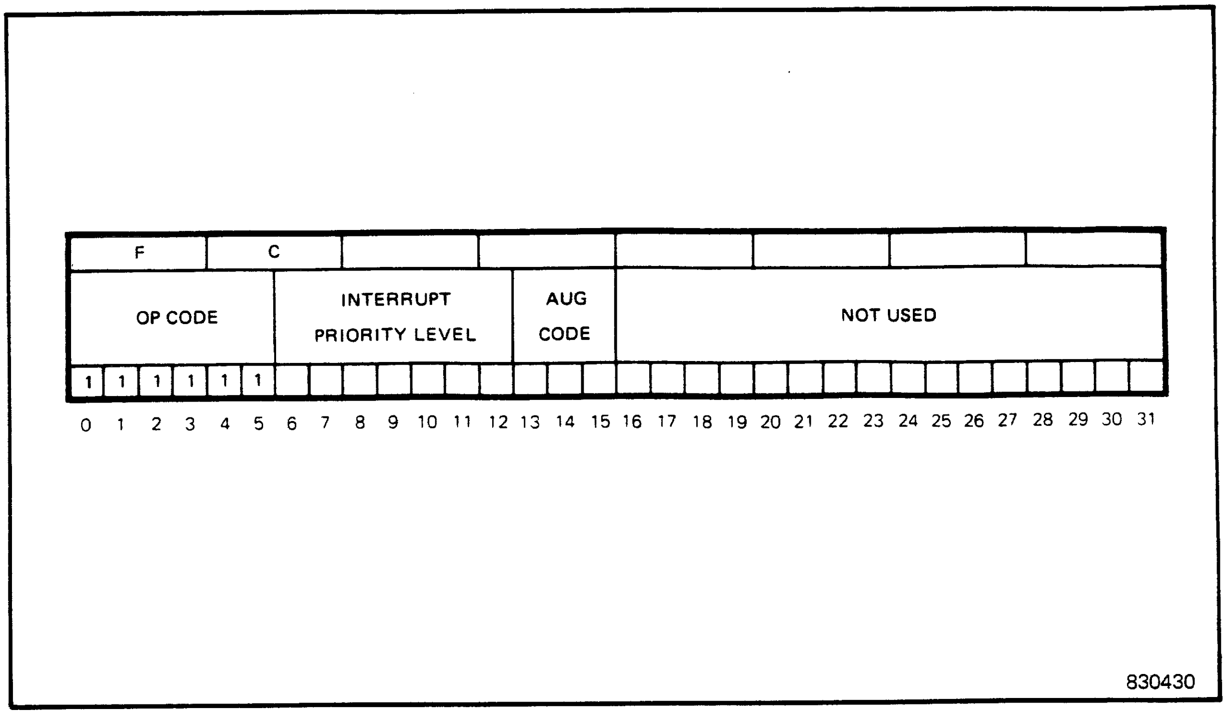


Figure 2-5. Interrupt Control Instruction Format

input/output (SIO) instruction. This procedure causes the HSD to fetch the contents of the first IOCB using a pointer in the TI dedicated location (TAW). Each IOCB contains the operation code (which is also sent to the device), the transfer count, the memory buffer address, and space to post error or device status. Figure 2-6 illustrates the error status format, and Table 2-6 defines the error status bits. Figure 2-7 illustrates and defines the device status format.

The HSD controls all data transfers until the transfer count reaches zero, a device end-of-block (DEB) is received, or an error has occurred. The service interrupt (SI) status is then automatically posted in the TI dedicated location, and a service interrupt (SI) is generated. Error status and the residual transfer count are automatically posted in the fourth word of the IOCB before normal service interrupt (SI) status is posted. Figure 2-8 illustrates the format for the service interrupt (SI) status. Table 2-7 defines the service interrupt (SI) status bits.

Automatic status posting eliminates the need for the test device instruction except to determine whether the HSD is present in the system. When the HSD is placed in the external mode, each data transfer must be preceded by the transfer of the appropriate memory address from the customer device to the HSD. The customer device has complete control of the HSD operation while the HSD is in the external mode.

2.5.5 Software Considerations

No standard software is associated with the HSD controller. Certain software considerations which must be recognized by the user are listed below:

1. A request interrupt (RI) issued to an HSD while it is busy (i.e., transferring data) will immediately cause an interrupt to occur, since an RI does not require SI status posting.

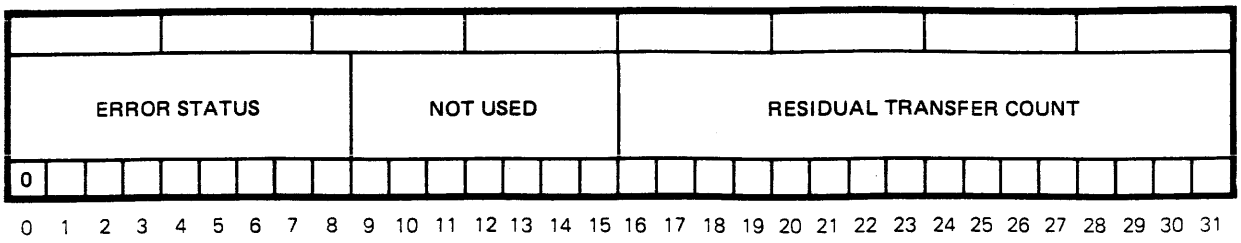


Figure 2-6. Error Status Format

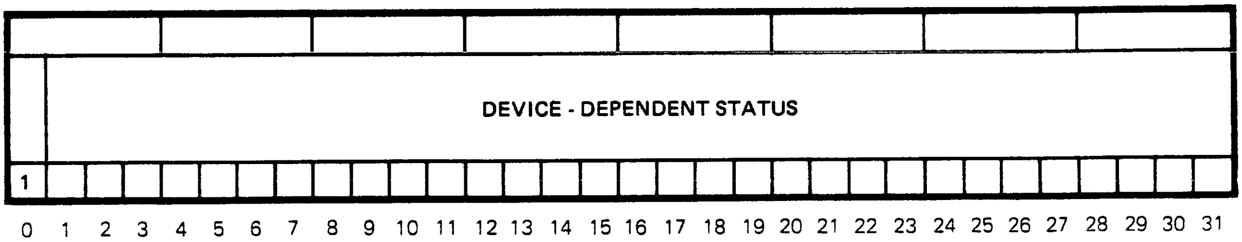


Figure 2-7. Device Status Format

Table 2-5
Priority Interrupt Dedicated Memory Locations

Priority Level (Hex)	Memory Dedicated Address (Hex)	Function
00*	0F0	Power fail safe - auto start interrupt
00*	0F4	Power fail safe - auto start trap
01*	0F8	System override interrupt
01*	0FC	System override trap
02**	100	Input/output controller 0 transfer interrupt
03**	104	Input/output controller 1 transfer interrupt
04**	108	Input/output controller 2 transfer interrupt
05**	10C	Input/output controller 3 transfer interrupt
06**	110	Input/output controller 4 transfer interrupt
07**	114	Input/output controller 5 transfer interrupt
08**	118	Input/output controller 6 transfer interrupt
09**	11C	Input/output controller 7 transfer interrupt
OA**	120	Input/output controller 8 transfer interrupt
OB**	124	Input/output controller 9 transfer interrupt
OC**	128	Input/output controller 10 transfer interrupt
OD**	12C	Input/output controller 11 transfer interrupt
OE**	130	Input/output controller 12 transfer interrupt
OF**	134	Input/output controller 13 transfer interrupt
10**	138	Input/output controller 14 transfer interrupt
11**	13C	Input/output controller 15 transfer interrupt
12*	0E8	Memory parity trap .
13*	0EC	Console interrupt (turnkey panel attention)
14	140	Input/output controller 0 service interrupt
15	144	Input/output controller 1 service interrupt
16	148	Input/output controller 2 service interrupt
17	14C	Input/output controller 3 service interrupt
18	150	Input/output controller 4 service interrupt
19	154	Input/output controller 5 service interrupt
1A	158	Input/output controller 6 service interrupt
1B	15C	Input/output controller 7 service interrupt
1C	160	Input/output controller 8 service interrupt
1D	164	Input/output controller 9 service interrupt
1E	168	Input/output controller 10 service interrupt
1F	16C	Input/output controller 11 service interrupt
20	170	Input/output controller 12 service interrupt
21	174	Input/output controller 13 service interrupt
22	178	Input/output controller 14 service interrupt
23	17C	Input/output controller 15 service interrupt
24*	190	Nonpresent memory trap
25*	194	Undefined instruction trap
26*	198	Privilege violation trap
27*	19C	Call monitor interrupt
28*	1A0	Realtime clock interrupt
29*	1A4	Arithmetic exception interrupt
2A*	1A8	External interrupt

Table 2-5
Priority Interrupt Dedicated Memory Locations (Cont.)

Priority Level (Hex)	Memory Dedicated Address (Hex)	Function
2B*	1AC	External interrupt
2C*	1B0	External interrupt
2D*	1B4	External interrupt
2E*	1B8	External interrupt
2F*	1BC	External interrupt (last PI in standard RTOM)
30	1C0	External interrupt
↓	↓	↓
7F	2FC	External interrupt

* Present in first RTOM.
** These dedicated addresses are reserved for transfer control words and cannot be used by priority interrupt software.

4th WORD of IOCB

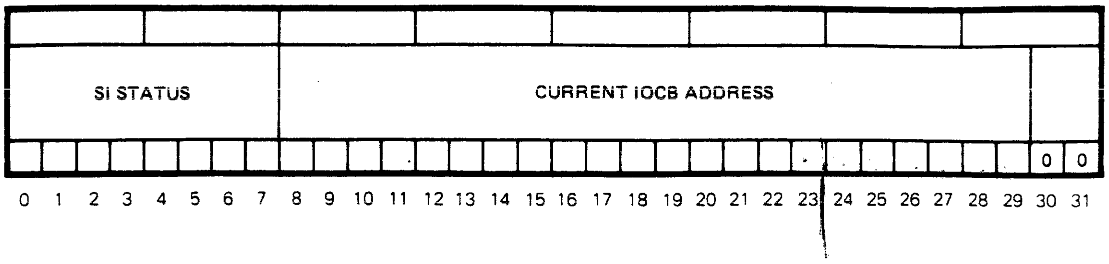
Table 2-6
HSD - Error Status Defined

Bit	Error Status Description
0 24	Indicates that the current content of the fourth word of the IOCB is HSD error status.
1-2 25-26	Not used.
3 27	Indicates that data is remaining in the FIFO when the transfer count = 0 and data chaining was specified with function 2 selected.
4 28	Indicates a parity error has occurred during the current data transfer operation.
5 29	Indicates a nonpresent memory error has occurred during the current data transfer operation.
6 30	Indicates a program violation has occurred while decoding the operation code of the current IOCB (i.e., an invalid operation code was detected).
7 31	Indicates that the customer device is inoperative. This could be power off, or HSD cable C disconnected, or a device-generated inoperative signal.
8	Indicates that the data buffer in the HSD has overflowed, and data from the device has been lost. The data buffer will not overflow when data is being sent to the device.
9-15	Not used.
16-31	16-bit residual transfer count.

Table 2-7
Service Interrupt (SI) Status Bit Definition

Bit	SI Status Description IN TAW Fig 2-8
0 24	Indicates that this status was posted because an interrupt occurred as a result of an external terminate from the customer device. When this type of interrupt occurs, the residual transfer count will be found in bits 16 through 31 of IOCB word 3 (error status word).
1 25	Indicates that this status was posted because an interrupt occurred as a result of an NPM or PE during the fetch of the IOCB address, IOCB word 0, or IOCB word 1 or an NPM during the storage of device status into IOCB word 3.
2 26	Indicates that this status was posted because an interrupt occurred as a result of a device end-of-block from the customer controller. When this type of interrupt occurs, the residual transfer count will be found in bits 16 through 31 of IOCB word 3 (error status word).
3 27	Indicates that this status was posted because an interrupt occurred as a result of an error condition being detected in the HSD.
4 28	Indicates that this status was posted because an interrupt occurred as a result of the completion of the entire I/O command list.
5 29	Indicates that this status was posted because an interrupt occurred as a result of the normal completion of the current IOCB.
6 30	Indicates that this status was posted because an interrupt occurred as a result of the program violation (i.e., a CD (other than terminate) was issued while the HSD was transferring data from a previous CD).
7 31	Unused - this bit should be set to zero.
8-31	Indicates the current value of the IOCB address at the time an interrupt occurred. This value is meaningless if the interrupt is caused by an RI instruction.
Note:	If an interrupt occurs and no SI status is posted (i.e., the TI dedicated location contains what was previously loaded), the software must assume that the interrupt was caused by an RI instruction. Other conditions which cause an interrupt but do not post SI status are discussed under service interrupt generation.

TAW



830429

Figure 2-8. Service Interrupt (SI) Status Format

2. If a data transfer terminates while the SI level is in the active state, no SI status is posted until the active state is cleared. Immediately following the BRI, which resets the SI active state, the SI status is posted, and the end-of-current IOCB interrupt which has been queued will occur.

3. If an external terminate is received while the HSD is busy (i.e., transferring data), and the interrupt level is active, the data transfers are terminated, and the residual transfer count is stored in the fourth word of the IOCB. When the BRI causes the interrupt active state to be reset, the SI status is posted, and the EXT interrupt occurs.

4. The software must issue a device status request command to determine the cause of an EXT interrupt.

5. It is recommended that the software written to interface an HSD to RTM or other customer software include a "lost interrupt" processor. This processor detects lost interrupts by the expiration of a timer which was set before issuing the CD start I/O. When the timer expires, the "lost interrupt" processor gates on the HSD interrupt level, sets the TI dedicated location to zero, issues a CD terminate to the HSD, and deactivates the HSD interrupt level. The HSD responds by posting both error and SI status and then generating an interrupt. The software may inspect the status information to determine the necessary error recovery procedures.

   This same basic detection mechanism can be used to handle situations in which the error made it impossible to post any status.

6. Other considerations regarding a software interface to RTM should come from the RTM reference and technical manuals.

7. Before an external device takes control of the HSD under external mode, the program must issue an order to the device to allow the loading of the HSD's TI register and IOCB address register (CAR).

8. Function 2 Selected. The operation of the HSD, when data chaining has been specified, is modified to allow sustained uninterrupted throughput of up to 834 kilowords per second. This is accomplished by allowing the FIFO to continue to transfer data while the next IOCB is being fetched from memory. This philosophy brings up the potential for more data being loaded into the HSD buffer (FIFO) from the external device than the last IOCB's transfer count is able to handle. This is particularly true when high-speed transfers are in progress. If the transfer count is less than the 17 words for the last IOCB of an IOCL, data may be left in the FIFO when the transfer count is zero. If data is left in the FIFO, an error indication will be posted in the TI location (TAW) and bit 3 of the fourth word of the IOCB will be set to indicate a record length error. In any case, the transfers specified by the transfer counter will be made before an error is posted. Also note that a terminate device (TDV) signal will be sent to the external device to terminate it if necessary.

9. Function 2 Selected. If the HSD is outputting to an external device and data chaining has been specified, it is possible for the external device to terminate the transfer while the HSD is fetching the next IOCB. If this occurs, the number of words left in the FIFO will be reported by adjusting the current IOCB's transfer count by that amount; e.g., the number of words remaining from the previous transfer equals 5 words and the current transfer's count equals 30 words. The residual reported to the software will be 35.

SECTION III

THEORY OF OPERATION

3.1 Introduction

The theory of operation for the high-speed data interface (HSD) is divided into two levels of discussion. The first level is the general theory of operation, which describes the purpose, basic organization, and overall operation of the HSD. The second level is the detailed theory of operation, which describes in detail the operational characteristics of the HSD.

3.2 General Theory

The primary function of the HSD is to provide a full 32-bit parallel interface to a customer-designed device at rates up to 834K transfers per second, after the input/output operation has been initiated by the controlling computer software. Once the controlling computer has initiated the operation, the HSD executes the data transfers between the computer's memory and the addressed customer device independently of the computer's operation.

A data transfer operation can consist of a single transfer or a group of transfers. When the data transfer operation is complete, the HSD automatically transfers status information to the computer and then generates an interrupt. Special commands allow the computer to obtain the customer device status.

3.2.1 HSD Basic Organization

The HSD is divided into three functional parts. The first part consists of a special SelBUS interface which allows overlapped CPU and memory communications to and from the HSD. The second part consists of a simple 32-bit bidirectional data bus and the appropriate handshake control signals to interface the HSD to a customer designed device interface. The third part consists of the appropriate internal storage registers and sequential control logic for controlling the data flow internal to the HSD, to and from the SelBUS, and to and from the customer handshake interface.

3.2.2 Overall Operation

The primary function of the HSD is to control the execution of an input or output operation between the computer/memory and the external device. The input or output operation is executed independently of computer operations after the operation has been initiated by the computer.

The 32 SERIES Computer uses the following software instructions to control or initiate I/O operations in the HSD and external device:

1. The test device (TD) instruction to determine whether the HSD is installed in the system.

2. The command device (CD) instruction to initiate data transfers or control operations within the HSD and the external device.

3. The interrupt control (IC) instruction to condition the HSD interrupt control logic for the following conditions:

   a. Request the interrupt

   b. Enable the interrupt

   c. Disable the interrupt

   d. Activate the interrupt

   e. Deactivate the interrupt

Computer firmware converts the above software instructions into a series of SelBUS transfers (refer to Table 3-1) to the HSD and the external device addressed by the instruction. The SelBUS interface logic of the HSD detects the SelBUS transfer. If the address on the SelBUS matches the address of the HSD, the SelBUS interface logic stores the SelBUS transfer for examination by the HSD CPU/memory input transfer recognition logic.

Typical block transfers are initiated by setting up an I/O command list (IOCL), which consists of one or more I/O command blocks (IOCBs), and issuing a command device (CD) start input/output (SIO) instruction. This procedure causes the HSD to fetch the contents of the IOCB using a pointer in the TI dedicated location. The HSD controls all data transfers until the transfer count reaches zero, a DEB is received, or an error has occurred. The service interrupt (SI) status is then automatically posted in the TI dedicated location, and a service interrupt (SI) is generated.

When the HSD is placed in the external mode, each data transfer must be preceded by the transfer of the appropriate memory address from the external device to the HSD. The external device has complete control of the HSD operation while the HSD is in the external mode.

The significant difference between the HSD and other standard I/O controllers is the dual SelBUS interface implemented on the HSD (see Figure 3-1). The SelBUS interface consists of independent sets of destination and data registers for the CPU and memory. These registers are controlled independently and simultaneously from two separate control sequencers which are the heart of the HSD control logic. This approach allows CPU/HSD transactions to occur simultaneously with HSD/memory transactions. Conflicts between these two types of transactions (i.e., both request the SelBUS at the same time) can last for only one clock cycle (150 nanoseconds).

All interrupt processing is done under hardware control and occurs simultaneously with all other HSD or SelBUS operations.

The two PROM control sequencers provide all internal control functions of the HSD, as well as the SelBUS and customer interfaces, essentially replacing most of the hardwired control logic.

To satisfy high data rate requirements, the HSD requires a first-in-first-out (FIFO) data buffer. This 16-word buffer reduces overflow and underflow situations caused by the CPU obtaining open SelBUS cycles and gaining access to memory before the HSD or other high priority devices do.

Table 3-2 defines the logic contents of each block illustrated on the HSD functional block diagram (Figure 3-1). The functional block number of Table 3-2 corresponds to the number in the lower right-hand corner in each block illustrated on the block diagram.

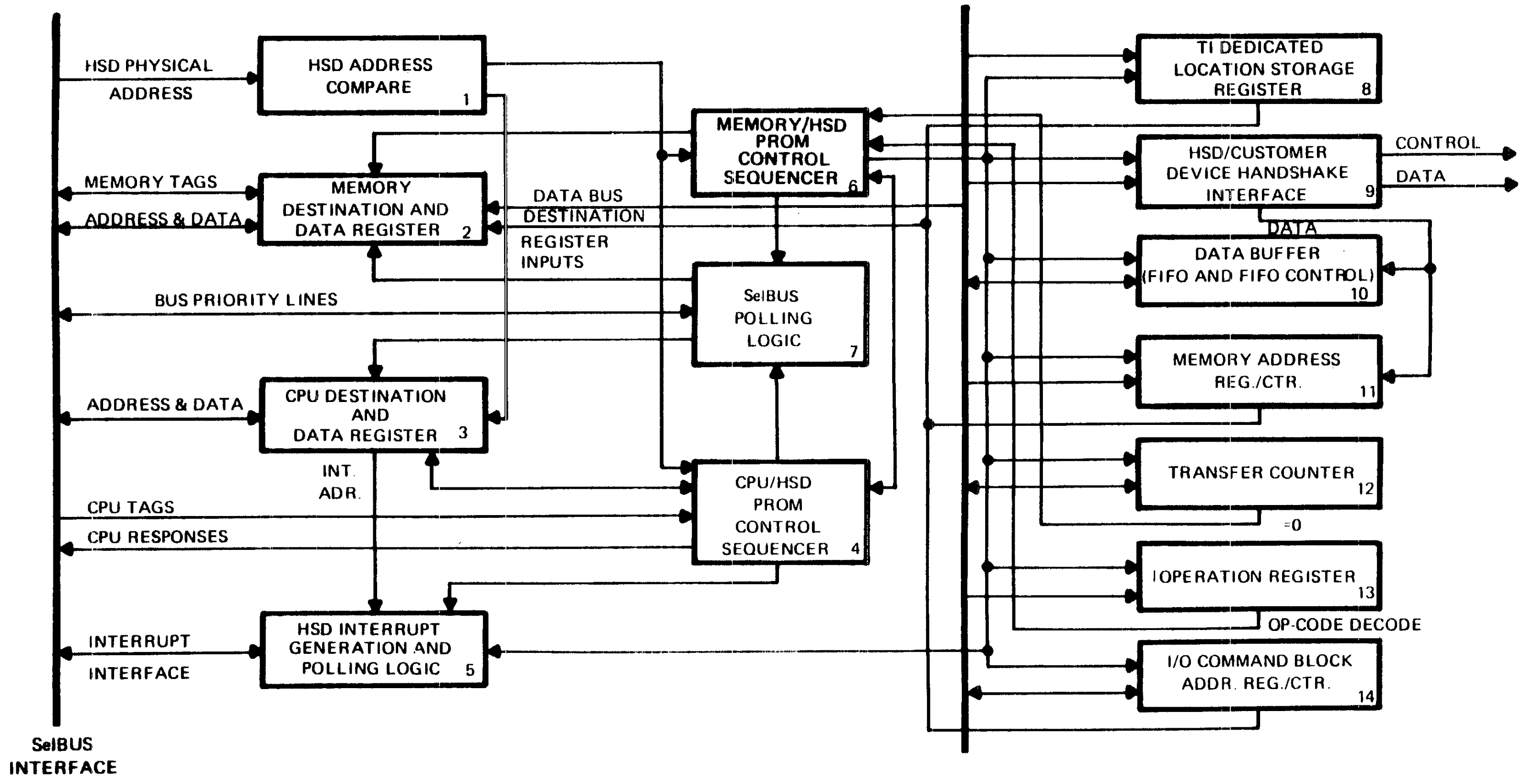

HSD PHYSICAL
ADDRESS
HSD ADDRESS COMPARE 1
MEMORY TAGS
ADDRESS & DATA
MEMORY DESTINATION AND DATA REGISTER 2
DATA BUS DESTINATION REGISTER INPUTS
MEMORY/HSD PROM CONTROL SEQUENCER 6
BUS PRIORITY LINES
SelBUS POLLING LOGIC 7
ADDRESS & DATA
CPU DESTINATION AND DATA REGISTER 3
INT. ADR.
CPU TAGS
CPU RESPONSES
CPU/HSD PROM CONTROL SEQUENCER 4
INTERRUPT
INTERFACE
HSD INTERRUPT GENERATION AND POLLING LOGIC 5
SelBUS INTERFACE
TI DEDICATED LOCATION STORAGE REGISTER 8
HSD/CUSTOMER DEVICE HANDSHAKE INTERFACE 9
CONTROL
DATA
DATA
DATA BUFFER (FIFO AND FIFO CONTROL) 10
MEMORY ADDRESS REG./CTR. 11
TRANSFER COUNTER 12
=0
IOPERATION REGISTER 13
OP-CODE DECODE
I/O COMMAND BLOCK ADDR. REG./CTR. 14

Figure 3-1. HSD Functional Block Diagram

Table 3-1
HSD/SelBUS Transactions

Bus Trans. Number	Macrolevel Instruction/SI & Data Access	Transactions										Destination Bus	Data Bus	Response
		Microlevel Bus Transfer				Bus Tags								
		From	To	Description	Mnemonic	LTX	LMEM	CNT0	CNT1	LRD	LET	L0 L8 L9 L15 L18 L23	L0 L1 L2 L3 L8 L31	
1	CD* (Start I/O)	CPU	HSD	Advance Read Status Transfer	ARSTX	L	H	L	H	H	H	H [Controller] [Device]	(28) Device and Ctrl. Status	Retry/T.A.
2		HSD	CPU	Ready Response To ARSTX	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD	Read Status Transfer	RSTX	L	H	H	L	L	H	H [Controller] [Device]	(28) Device and Ctrl. Status	T.A
4		HSD	CPU		DRT	L	H	L	L	H	H	L [CPU]	All Zeros	Don't Care
5		CPU	HSD	Start I/O	WDOT	L	H	H	H	H	H	H [Controller] [Device]	L H H H [IOCD]	T.A.
6		HSD	MEM	Memory Read (Fetch Second Word of IOCD)	MRT	L	L	H	H	L	H	[IOCD + 1]	(24) (25-31) H [Controller]	Unsuccessful/ T.A.
7		MEM	HSD	Data Return (IOCD32:63)	DRT/ET	L	H	L	L	H	H/L	H [Controller]	IOCD (LSH) [TI Dedicated Location]	Don't Care
8		HSD	MEM	Memory Read (Fetch IOCB Address)	MRT	L	L	H	H	L	H	[TI Dedicated Location]	(24) (25-31) H [Controller]	Unsuccessful/ T.A.
9		MEM	HSD	Data Return (IOCB Address)	DRT/ET	L	H	L	L	H	H/L	H [Controller]	[IOCB]	Don't Care
10		HSD	MEM	Memory Read (Fetch IOCB Word 0)	MRT	L	L	H	H	L	H	[IOCB]	(24) (25-31) H [Controller]	Unsuccessful/ T.A.
11		MEM	HSD	Data Return (IOCB Word 0)	DRT/ET	L	H	L	L	H	H/L	H [Controller]	IOCB Word 0	Don't Care
12		HSD	MEM	Memory Read (Fetch IOCB Word 1)	MRT	L	L	H	H	L	H	[IOCB + 1]	(24) (25-31) H [Controller]	Unsuccessful/ T.A.
13		MEM	HSD	Data Return (IOCB Word 1)	DRT/ET	L	H	L	L	H	H/L	H [Controller]	IOCB Word 1	Don't Care

*Notes: The present CPU firmware executes two ARSTX/RSTX pairs before the WDOT for a CD SIO.
[ ] Denotes "Address"
- Denotes "Not Applicable"

Table 3-1
HSD/SelBUS Transactions (Cont.)

Bus Trans. Number	Macrolevel Instruction/SI & Data Access	Microlevel Bus Transfer				Bus Tags						Destination Bus	Data Bus	Response
		From	To	Description	Mnemonic	LTX	LMEM	CNT0	CNT1	LRD	LET	L 0, L 8, L 9, L 15	L 23, L 0, L 1, L 2, L 3, L 8, L 31	
1	Typical Write Data Transfer	HSD	MEM	Memory Write	MWT	L	L	H	H	H	H	[Data Transfer	] DATA	Unsuccessful/ T.A.
1	Typical Read Data Transfer	HSD	MEM	Memory Read	MRT	L	L	H	H	L	H	[Data Transfer	] (24) (25-31) H [Controller]	Unsuccessful/ T.A.
2		MEM	HSD	Data Return	DRT/ET	L	H	L	L	H	H/L	H [Controller]	DATA	Don't Care
1	Service Interrupt Request	HSD	BUS	Requested interrupt set by HSD. Priority level of interrupt shifted out if highest priority.	HREQINT LIPOL LINTR									
2		CPU	HSD	Advance Read Status Transfer	ARSTX	L	H	L	H	H	H	H [Controller] [Device]	(27) ACK.INT	Retry/T.A.
3		HSD	CPU	Ready Response to ARTSTX	READY	-	-	-	-	-	-	-	-	
4		CPU	HSD	Read Status Transfer	RSTX	L	H	H	L	L	H	H [Controller] [Device]	(27) ACK.INT	T.A.
5		HSD	CPU	Data Return Transfer	DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care
1	TD* 8000/4000	CPU	HSD	Advance Read Status Transfer	ARSTX	L	H	L	H	H	H	H [Controller] [Device]	(28) Device and Ctrl. Status	Retry/T.A.
2		HSD	CPU	Ready Response to ARSTX	READY	-	-	-	-	-	-	-	-	T.A.
3		CPU	HSD	Read Status Transfer	RSTX	L	H	H	L	L	H	H [Controller] [Device]	(28) Device and Ctrl. Status	
4		HSD	CPU	Data Return Transfer	DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care

*Notes: The present CPU firmware executes two ARSTX/RSTX pairs for TD 8000/4000.
The HSD handles them identically so both pairs are not listed above.
[ ]Denotes "Address"
- Denotes "Not Applicable"

Theory of Operation

Table 3-1
HSD/SelBUS Transactions (Cont.)

Bus Trans. Number	Macrolevel Instruction/SI & Data Access	Microlevel Bus Transfer				Bus Tags						Destination Bus	Data Bus	Response
		From	To	Description	Mnemonic	LTX	LMEM	CNT0	CNT1	LRD	LET	L 8, L 9, L 15	L 23, L 0, L 1, L 2, L 3, L 8, L 31	
1	TD ** 2000	CPU	HSD	Advanced Read Status Transfer	ARSTX	L	H	L	H	H	H	H [Controller] [Device]	(29) Device Status	Retry/T.A.
2		HSD	CPU	Ready Response to ARSTX	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD	Read Status Transfer	RSTX	L	H	H	L	L	H	H [Controller] [Device]	(29) Device Status	T.A.
4		HSD	CPU	Data Return Transfer	DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care
1	EI*	CPU	HSD	Advance Interrupt Control Transfer	AICT	L	H	L	H	L	H	H [Controller] [Device]	(31) EI	Retry/T.A.
2		HSD	CPU	Ready Response to AICT	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD	Interrupt Control Transfer	ICT	L	H	H	L	H	H	H [Controller] [Device]	(31) EI	T.A.
4		HSD	CPU	Data Return Transfer	DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care
1	DI*	CPU	HSD		AICT	L	H	L	H	L	H	H [Controller] [Device]	(30) DI	Retry/T.A.
2		HSD	CPU	Ready Response to AICT	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD		ICT	L	H	H	L	H	H	H [Controller] [Device]	(30)	T.A.
4		HSD	CPU		DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care

Notes: * The HSD guarantees that two sync times will occur from the time it receives the ICT until it sends a DRT back to the CPU.
** The present CPU firmware executes two pair of ARTSTX/RSTX pairs for TD 2000. The HSD handles them identically, so both parts are not listed above.
[ ] Denotes "Address"
- Denotes "Not Applicable"

Table 3-1
HSD/SelBUS Transactions (Cont.)

Bus Trans. Number	Macrolevel Instruction/SI & Data Access	Microlevel Bus Transfer				Bus Tags						Destination Bus	Data Bus	Response
		From	To	Description	Mnemonic	LTX	LMEM	CNT0	CNT1	LRD	LET	L L L L 0 8 9 15	L L L L L L L 23 0 1 2 3 8 31	
1	AI*	CPU	HSD		AICT	L	H	L	H	L	H	H [Controller] [Device]	(29) AI	Retry/T.A.
2		HSD	CPU	Ready Response to AICT	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD		ICT	L	H	H	L	H	H	H [Controller] [Device]	(29) AI	T.A.
4		HSD	CPU		DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care
1	DAI*	CPU	HSD		AICT	L	H	L	H	L	H	H [Controller] [Device]	(28) DAI	Retry/T.A.
2		HSD	CPU	Ready Response to AICT	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD		ICT	L	H	H	L	H	H	H [Controller] [Device]	(28) DAI	T.A.
4		HSD	CPU		DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care
1	RI*	CPU	HSD		AICT	L	H	L	H	L	H	H [Controller] [Device]	(27) RI	Retry/T.A.
2		HSD	CPU	Ready Response to AICT	READY	-	-	-	-	-	-	-	-	-
3		CPU	HSD		ICT	L	H	H	L	H	H	H [Controller] [Device]	(27) RI	T.A.
4		HSD	CPU		DRT	L	H	L	L	H	H	L [C P U]	All Zeros	Don't Care

Notes: The HSD guarantees that two sync times will occur from the time it receives the ICT until it sends a DRT back to the CPU.
[ ] Denotes "Address"
- Denotes "Not Applicable"

Table 3-1
HSD/SelBUS Transactions (Cont.)

Bus Trans. Number	Macrolevel Instruction/SI & Data Access	Microlevel Bus Transfer				Bus Tags						Destination Bus	Data Bus							Response
		From	To	Description	Mnemonic	LTX	LMEM	CNT0	CNT1	LRD	LET	L 0 L 8 L 9 L 15 L 18	L 23	L 0	L 1	L 2	L 3	L 8	L 31	
1	CD* (Terminate)	CPU	HSD	Halt I/O	WDOT	L	H	H	H	H	H	H [Controller][Device]		H	H	L	H			T.A.
2		HSD	CPU	Ready Response to HIO	READY	-	-	-	-	-	-	-			-					-
3		CPU	HSD	Halt I/O	WDOT	L	H	H	H	H	H	H [Controller] [Device]		H	H	L	H			T.A.
1	CD (TCA)	CPU	HSD	Status request	ARSTX	L	H	L	H	H	H	H [Controller] [Device]						(30)		Retry T.A. TCA Status
2		HSD	CPU	Ready Response to ARSTX	READY	-	-	-	-	-	-	-			-					-
3	(Invalid Command to HSD)	CPU	HSD		RSTX	L	H	H	L	L	H	H [Controller] [Device]						(30) TCA Status		T.A.
4		HSD	CPU		DRT	L	H	L	L	H	H	L [CPU]						All Zeros		Don't Care
1	86 Emulation Support (IPL Command) (Invalid Command to HSD)	CPU	HSD	IPL	WDOT	L	H	H	H	H	H	H [Controller] [Device]		L	H	H	L	[IOCD]		T.A.
1	86 Emulation Support (Load RAM)	CPU	HSD	Advance Read Status Transfer	ARSTX	L	H	L	H	H	H	H [Controller] [Device]						(28) Device and Ctrl. Status		Retry/T.A.
2		HSD	CPU	Ready Response to ARSTX	READY	-	-	-	-	-	-	-			-					-
3		CPU	HSD	Read Status Transfer	RSTX	L	H	H	L	L	H	H [Controller] [Device]						(28) Device and Ctrl. Status		T.A.
4		HSD	CPU	Data Return Transfer	DRT	L	H	L	L	H	H	L [CPU]						All Zeros		Don't Care
5		CPU	HSD	Load RAM with Physical Control Adr. & Int. Prior. Level	WDOT	L	H	H	H	H	H	(1-7) H [Controller] [Device] Interrupt Priority Level		H	L	H	H	25-31 [Controller]		T.A.

Notes: The HSD must receive the second WDOT - HIO within 12 clock cycles after returning the ready. If not, the HSD will not execute the HIO.
[ ]Denotes "Address"
- Denotes "Not Applicable"

Table 3-2
Logic Function of Block Diagram

Functional Block No.	Functional Logic Within the Block
1	Physical Address Switches Address Compare Logic
2	Memory Destination Staging Register (24 bits) Memory Destination Register Multiplexer Memory Destination Register Output Transmitters (24) Memory Data Staging Register (32 bits) Memory Data Register Multiplexer Memory Data Register Output Transmitters Memory Tag Register, i.e., LMEM, LRD Memory Tag Output Transmitters Memory Response Logic, i.e., TA, US, ET
3	CPU Destination Staging Register (7 bits) CPU Destination Register Output Transmitters (1) CPU Data Staging Register (8 bits)
4	PROM Address Register/Counter PROM SelBUS Tag Input Staging Register and PROM Address Selector PROM Data Output Register CPU Response Logic, i.e., RETRY, TA, READY CPU Bus Tag Generation and Output Staging Register SelBUS Tag Transmitters (3) Sequencer Test Condition Logic Sequencer Order Generation Logic
5	Interrupt Priority Level Register Interrupt Polling Logic Interrupt Request Generation Interrupt Enable/Disable Logic Interrupt Active Logic
6	PROM Address Register/Counter PROM PROM Address Select Logic PROM Data Output Register Sequencer Test Condition Logic Sequencer Order Generation Logic
7	Memory Echo Bit Flip-Flops Memory Inhibit Selector Logic Bus Poll Priority Generation Switches and Transmitters Bus Poll Priority Detection Switches and Logic CPU/Memory Bus Poll Priority Request Logic CPU/Memory Bus Poll Priority Enable Logic

Table 3-2
Logic Function of Block Diagram (Cont.)

Functional Block No.	Functional Logic Within the Block
8	TI Dedicated Location Storage Register (7 bits)
9	Bidirectional Data Bus Drivers/Receivers (32) Handshake Control Logic and Drivers/Receivers Data Staging Register
10	Data Buffer (FIFO) RAM (16 x 32) FIFO Control Counters and Read/Write Logic FIFO Status Logic, i.e., Full, Empty, Overflow, etc.
11	Memory Buffer Starting Address Register/Counter (22 bits) Memory Address Register 2 x 1 Multiplexer
12	Transfer Counter (16 bits) Transfer Counter Control/Detection Logic, i.e., Load, Increment, Count = 0, etc.
13	Operation Register (8 bits) Operation Register Decode Logic
14	Current IOCB Address Register/Counter (22 bits)

3.2.3 Customer Device Control

The control logic of the HSD controls the customer handshake interface logic and the customer device used with the HSD. The HSD sequencer's firmware (microprogram) provides the main elements of control over the customer handshake interface logic by the use of the order structure and test structure logic of the sequencers. The order structure provides the microprogram with the ability to generate individual pulsed signals or level signals at the direction of a decode of a microinstruction. The customer interface is basically enabled or disabled by the firmware and is free-running when enabled.

The sequencers' test structure provides the microprogram with the ability to test preselected individual signals for either a logic high level or low level at the direction of a decoded microinstruction. Since these tests are requested by the firmware, they can be used to determine the current condition of the customer handshake interface logic. The test structure, together with the order structure, provides the firmware with the ability to make a decision based on current conditions within the HSD and customer handshake interface logic, and then to take the appropriate action based on the decision.

The control section of the HSD can output 32 bits of data to the customer handshake interface logic. This data is obtained from the first-in-first-out (FIFO) register file. The output data may represent either data to be recorded by the customer device or command information for the customer device. The microprogram has the ability to define the type of data being outputted by using the order structure logic to generate output control signals to the customer handshake interface in conjunction with the output data transfer.

The microprogram can input data from the customer handshake interface logic in 32-bit words. The input data is stored in the 32-bit FIFO register file of the control section. The input data may represent data read from the customer device, status information, or a memory address, if the device is operating in the external mode. The microprogram has the ability to request the type of data input to the control section by using the order structure logic to generate input control signals to the customer handshake interface in conjunction with the input transfer.

3.2.4 CD Start I/O (SIO)

3.2.4.1 Execution

When the HSD receives a WDOT SelBUS transfer indicating an SIO, the following events occur, assuming the HSD is not busy:

1. The HSD stores the I/O command doubleword (IOCD) address in the memory address register.

2. The HSD increments the memory address register by one and requests priority to poll the SelBUS.

3. Assuming the priority is won, the SelBUS is polled for a memory transfer, and memory address register is transferred to the memory destination register.

4. Assuming the SelBUS poll is won, the memory ready transfer occurs, and the memory returns a DRT, which causes the HSD to load the memory data register with the TI dedicated location address.

5. The HSD transfers the memory data register to the TI dedicated location storage register and requests priority to poll the SelBUS.

6. Assuming the priority is won, the HSD polls the bus, and the TI dedicated location storage register is transferred to the memory destination register.

7. Assuming the SelBUS priority is won, the MRT transfer occurs, and the memory returns a DRT, which causes the HSD to load the memory data register with the IOCB address.

8. The HSD transfers the memory data register to the IOCB address register and requests priority to poll the SelBUS.

9. Assuming the priority is won, the HSD polls the bus, and the IOCB address register is transferred to the memory destination register.

10. Assuming the SelBUS poll is won, the MRT transfer occurs, and the memory returns a DRT, which causes the HSD to load the memory data register with the first word of the IOCB.

11. The HSD transfers bits 0 through 7 of the memory data register to the operation code register and bits 16 through 31 to the transfer counter, increments the IOCB address by one, and requests priority to poll the SelBUS.

12. Assuming the priority is won, the HSD polls the SelBUS and transfers the IOCB address register to the memory destination register.

13. Assuming the SelBUS priority is won, the MRT transfer occurs, and the memory returns a data return transfer, which causes the HSD to load the memory data register with the second word of the IOCB (memory buffer address or device command information depending on bit 1 of the op code).

14. If op code bit 1=0, the HSD transfers the memory data register (memory buffer address) to the memory address register and increments the IOCB address by two.

15. The HSD generates an external function signal to the device and places the first word of the IOCB, containing the op code and transfer count, on the device data bus.

16. The device loads the op code into its storage register and returns an external function acknowledge to the HSD.

3.2.4.2 Error Conditions

If a parity error or nonpresent memory error should occur during the fetch of the TI dedicated location, the HSD will terminate the SIO sequence and generate an SI to the macrolevel software. No status is posted.

If a parity or nonpresent memory error occurs during the fetch of the IOCB address, or the fetch of any of the IOCB words, the HSD will terminate the SIO sequence and cause the following events to occur:

1. Post SI status in the TI dedicated location with the IOCB address error, the error and end of list bits set. No controller error status is posted. The error indication specifies to the user that the error condition was not a lost interrupt.

2. Generate an SI to the macrolevel software.

3.2.5 CD Terminate

3.2.5.1 Execution

When the HSD receives a write data or order transfer (WDOT) SelBUS transfer indicating a halt I/O, the following events occur:

1. If the HSD is transferring data, the present data transfer in progress is completed, but no additional transfers are initiated.

2. The HSD transfers the error status and residual transfer count to the memory data register and requests priority to poll the SelBUS.

3. Assuming the priority is won, the HSD polls the SelBUS, and the IOCB address register is transferred to the memory destination register.

4. Assuming the SelBUS priority is won, a memory write transfer occurs.

5. The HSD transfers the SI status and the current IOCB address register to the memory data register and requests priority to poll the SelBUS.

6. Assuming the priority is won, the HSD polls the SelBUS, and the TI dedicated location register is transferred to the memory destination register.

7. Assuming the SelBUS priority is won, a memory write transfer occurs.

8. The HSD sets the SI request flip-flop, conditions the interrupt logic to begin polling, and resets all critical controller functions to their initial state.

9. Upon reception of the interrupt acknowledge ARSTX, RSTX sequence, the HSD sets the interrupt active flip-flop.

If no data transfer was in progress when the WDOT halt I/O was received, the HSD will generate an SI request to the macrolevel software, and no status will be posted.

3.2.5.2 Error Conditions

During the execution of the CD terminate command, the HSD may encounter nonpresent memory errors while attempting to write the status words. Should this occur, the HSD will ignore such errors, complete the two write operations, and generate the SI to the macrolevel software. The fact that the status was not written into the TI dedicated location will cause the software to proceed under the assumption that an error has occurred.

3.2.6 Data Transfer Termination

3.2.6.1 Normal Termination

A normal data transfer termination occurs when the transfer counter in the HSD reaches zero, signaling the HSD that all the outstanding data transfers for this IOCB have been completed, or a device end-of-block is received. The HSD will then perform the following if the interrupt on EOB op code bit is set.

1. The HSD transfers the SI status and the current IOCB address to the memory data register and requests priority to poll the SelBUS.

2. Assuming the priority is won, the HSD polls the SelBUS, and the TI dedicated location register is transferred to the memory destination register.

3. Assuming the SelBUS priority is won, a memory write transfer occurs.

4. The HSD sets the SI request flip-flop, conditions the interrupt logic to begin polling, and returns to the inactive state to await the next CD.

5. Upon reception of the interrupt acknowledge ARSTX, RSTX sequence, the HSD sets the interrupt active flip-flop.

If the interrupt on EOB bit is not set, no status will be stored, and no interrupt will be generated. The HSD will reset the appropriate control flip-flops and await the next macrolevel CD-SIO.

3.2.6.2 Command Chaining

If command chaining is specified and the transfer counter in the HSD reaches zero, the HSD will perform the following:

1. Increment the IOCB address to the next IOCB in the sequence; i.e., increment the IOCB address by one word, and request priority to poll the SelBUS.

2. Fetch the next IOCB as in steps 9 through 16 of paragraph 3.2.4.1.

3.2.6.3 Data Chaining

3.2.6.3.1 Function 1

If data chaining is specified and the transfer counter in the HSD reaches zero, the HSD will perform the following:

1. Increment the IOCB address to the next IOCB in the sequence and request priority to poll the SelBUS.

2. Fetch the next IOCB as in steps 9 through 14 of paragraph 3.2.4.1., except that in step 11, only bits 4 through 7 of the operation code register are loaded.

3. When operating with function 1 selected (no jumper installed between pins 8 and 9 at (IC) location E10), the data handshaking on the interface will not be allowed when fetching the next IOCB. This arrangement forces the external device to remain in sync with the HSD.

3.2.6.3.2 Function 2

1. Operation is the same as (1) in paragraph 3.2.6.3.1.

2. Operation is the same as (2) in paragraph 3.2.6.3.1.

3. When operating with function 2 selected (jumper installed between pins 8 and 9 at (IC) location E10), data is allowed to continue transferring into and out of the HSD FIFO buffer when fetching the next IOCB.

3.2.6.4 Continue-on-Error

If command or data chaining is specified together with the continue-on-error bit in the operation code of the current IOCB, and a memory access error during data transfer occurs, the HSD performs the functions defined in steps 1 through 4 of paragraph 3.2.5.1 and then fetches the next IOCB as specified by the command or data chain bit.

3.2.6.5 Error Termination

An abnormal data transfer termination occurs when the HSD detects any of the controller or memory access errors specified in Figure 2-6, or when the HSD receives an external terminate signal from the customer device. Detection of any of these conditions causes the HSD to perform the functions defined in steps 1 through 9 of paragraph 3.2.5.1. Bit 3 (error) and bit 4 (EOL) or bit 0 (EXT) of the SI status word will be set when an abnormal data transfer termination occurs.

3.2.7 Service Interrupt Generation

The conditions which generate most of the service interrupts have been described in the SI status definition (Figure 2-8).

The following list includes these conditions and additional situations which cause an SI to be genrated.

1. End-of-block (EOB) if IOCB op code bit 4 is set, and the current IOCB function is complete.
2. End-of-list (EOL) if the command or data chain op code bits are not set in the current IOCB, and the current IOCB is complete.
3. EOL if the command or data chain op code bit is set, but a controller error occurs with the continue-on-error op code bit not set, and the current IOCB is complete.
4. Controller error according to error status definitions in Figure 2-6.
5. External terminate from the device.
6. Device end-of-block condition.
7. Execution of a request interrupt (RI) instruction.
8. Nonpresent memory (NPM) or parity error (PE) while fetching the TI dedicated location.
9. NPM or PE while fetching the IOCB address.
10. NPM or PE while fetching IOCB word 0.
11. NPM or PE while fetching IOCB word 1.

12. Execution of a CD terminate instruction.

Items 7 and 8 cause service interrrupts to occur, but do not post status in the TI dedicated location. Software normally interprets an interrupt with no status in the TI dedicated location to indicate that an RI has occurred. The software may be written to handle condition 8, as if it were an RI, by using the lost interrupt processing technique.

If the HSD is executing CD start I/O, generation of any interrupt is postponed until the contents of the TI location have been fetched by the HSD. This postponement insures that the contents of the TI location are not destroyed by status posting. It also insures that, when an interrupt is received, software is free to modify the contents of the TI location.

3.3 Detailed Theory

The detailed theory of the HSD is divided into the following topics:

1. HSD SelBUS interface
2. Data structure hardware components
3. CPU interface control sequencer
4. Memory interface control sequencer
5. HSD/customer interface
6. External control of the HSD
7. Throughput capabilities

The detailed theory describes the primary function and control signals that control the logic components and link one logic component to another.

The detailed theory includes logic references to locate the logic components on a specific logic sheet. The logic drawing references orient the user to the overall organization of the logic drawings. The logic diagram for the Model 9130 High-speed Data Interface is 130-103554, the logic diagram for the Model 9131 High-speed Data Interface II is 130-103346, and the logic diagram for the Model 9132 High-speed Data Interface is 130-103071. The physical location of components on the board and the list of materials (LM) for the board are on drawings 161-103554, 161-103346, and 161-103071, respectively. The microprogram firmware is installed on the HSD circuit card to form a complete HSD Interface.

The 161-103XXX and 103-103XXX drawings for the HSD are in the Drawings Manual. The conventions required to read the drawings are in the Circuit Registration Manual, publication number 313-000670, which contains a description of integrated circuits used in the 32 SERIES Computer system. Appendix A of this volume of the technical manual provides a Glossary of Terms for the HSD.

3.3.1 HSD SelBUS Interface

The HSD includes a special SelBUS interface to minimize conflicts between memory data transfers and CPU transfers to and from the HSD. The interface contains two sets of data and destination registers. One set is used for CPU/HSD communications and the other for memory/HSD communications. These registers may be loaded by the HSD simultaneously. Additional priority logic determines which set of registers will be placed on the SelBUS after bus polling is completed. The CPU and memory control logic must request priority to poll for the SelBUS before the actual SelBUS polling cycle.

The SelBUS interface logic of the high-speed data interface (HSD) provides the communication path from the HSD to the SelBUS which connects the various modules of the computer system as shown in Figure 3-2. The SelBUS provides a 32-bit bidirectional data bus and a 24-bit bidirectional address bus for communications between the modules of the computer system.

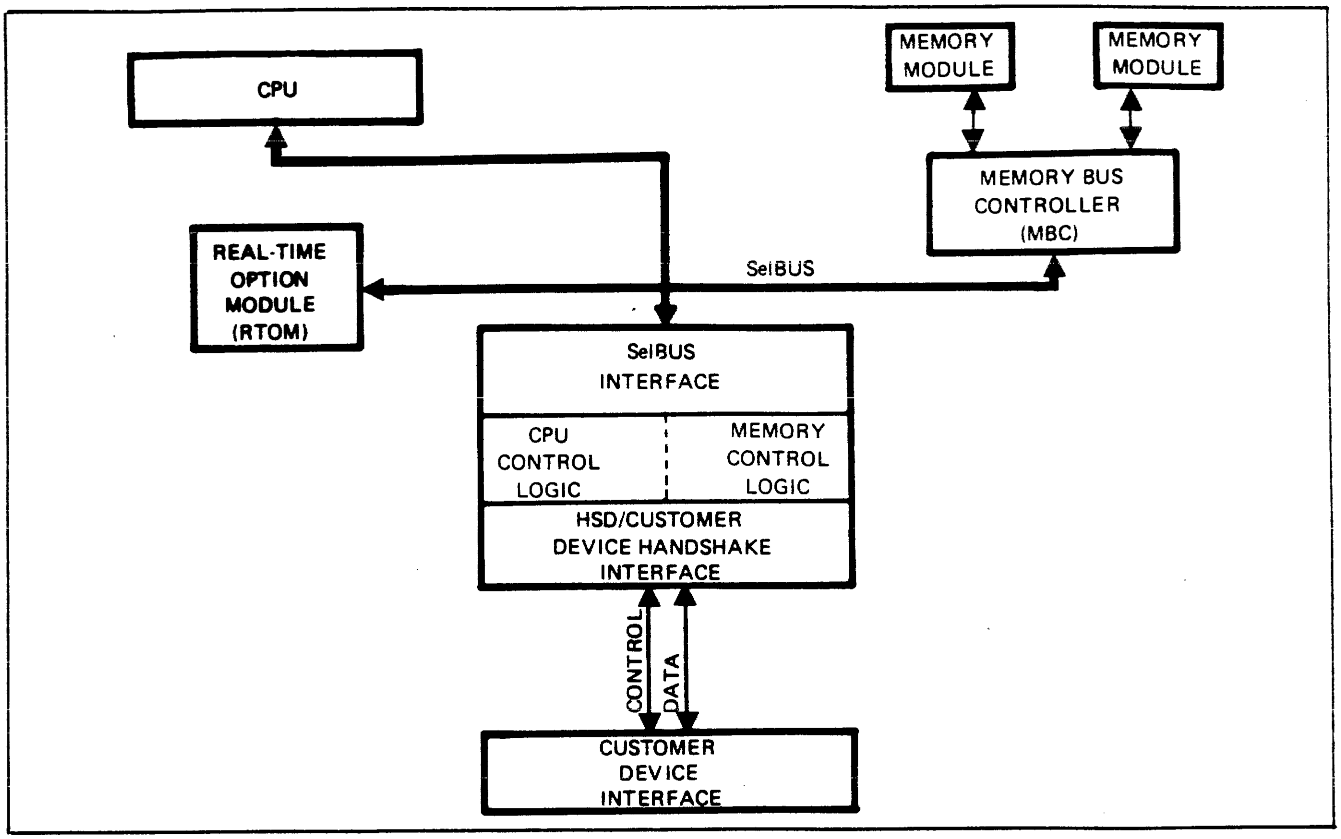

CPU
MEMORY MODULE
MEMORY MODULE
MEMORY BUS CONTROLLER (MBC)
REAL-TIME OPTION MODULE (RTOM)
SelBUS
SelBUS INTERFACE
CPU CONTROL LOGIC
MEMORY CONTROL LOGIC
HSD/CUSTOMER DEVICE HANDSHAKE INTERFACE
CONTROL
DATA
CUSTOMER DEVICE INTERFACE

Figure 3-2. SYSTEMS 32 SERIES Computer Block Diagram

The SelBUS is composed of a number of signal lines as follows:

1. Data, commands, status, device addresses, or memory addresses, are carried on 32-bit bidirectional data lines.

2. Memory addresses, IOM addresses, peripheral device addresses, or priority interrupt levels are carried on 24-bit bidirectional destination lines (address bus).

3. Six tag lines are used to define the type of data and addresses carried on the data and destination buses. The tag lines include transfer, memory, control 0, control 1, read, and error lines. Tag lines define the type of transfer is progress.

4. Five response lines are used to indicate whether a SelBUS transfer was accepted or rejected. Response lines include busy, retry, transfer acknowledge, ready, and unsuccessful memory transfer lines.

5. Twenty-two priority lines are used to control the SelBUS transfer priority scheme.

6. Two memory address echo lines and four inhibit lines determine which memory module(s) are busy.

7. Three priority interrupt lines provide the HSD with priority interrupt capabilities.

8. Four miscellaneous lines include the master clock, stop clock, I/O interrupt inhibit, and I/O reset signals.

The SelBUS Interface logic is divided into three main functions as described in the following paragraphs.

First the SelBUS interface logic receives and stores bus transfers from the SelBUS if the interface logic is not busy. The interface logic rejects the new transfer with a retry response signal when a previous SelBUS transfer being stored in the interface staging register would be overlaid by the new transfer. If the SelBUS interface logic is not busy, the transfer is stored in the staging register, and the interface logic indicates to the control logic that a transfer has been received and generates a transfer acknowledge (TA) to the SelBUS. The interface logic indicates the reception of a transfer using the control logic test structure. The microprogram must examine the transfer and take appropriate action. When the microprogram completes the processing of the transfer, it clears the busy condition in the SelBUS interface logic.

Second, the SelBUS interface logic stores output transfers to the SelBUS while the microprogram is assembling the transfer. The interface logic stores an output transfer in the staging register during assembly, and any input transfers received from the SelBUS must be rejected with a retry signal to prevent the transfer being assembled from being overlaid by the input transfer. Once the output is assembled in the staging register, the interface logic obtains SelBUS transfer priority and completes the output transfer to the SelBUS.

Third, the SelBUS interface logic generates priority interrupts (service interrupts) at the request of the microprogram. When an interrupt has been requested, the interface logic generates that interrupt if no higher priority interrupt is generated by another controller. The SelBUS interface logic generates the interrupt during a time-frame defined by the CPU.

A SelBUS transfer requires 150 nanoseconds. Four SelBUS transfer cycles occur during each 600-nanosecond memory cycle. The leading and trailing edge of a SelBUS transfer cycle is marked by one full cycle of the master clock (LCLK) signal used to define the transfer cycle. The master clock signal is used for timing in the HSD so that both the control instruction cycles and SelBUS transfer cycles require 150 nanseconds and are always synchronized.

During output transfers to the SelBUS, the interface logic drives the output transfer to the SelBUS at the start of the transfer cycle and holds the information on the bus for the duration of the cycle. During an input transfer, the interface logic receives the transfer for the duration of the transfer cycle. However, the information is not strobed into the staging register until the trailing edge of the transfer cycle. The transfer acknowledge (TA) function is always performed during the bus transfer cycle following the input transfer.

3.3.1.1 SelBUS Transfer Classification

All transfers on the SelBUS are classified as input or output transfers with respect to the HSD. Input transfers originate in the CPU or memory bus controller (MBC). The output transfers destination is either the CPU or the MBC.

Each type of transfer is identified by the configuration of the SelBUS tag line signals as follows:

1. The transfer (LTX) signal indicates that the transfer on the SelBUS is valid.

2. The memory (LMEM) signal indicates that the transfer on the SelBUS is destined for memory.

3. The control 0 (LCNT0) signal is one of two signals describing the type of transfer.

4. The control 1 (LCNT1) signal is one of two signals describing the type of transfer.

5. The read (LRD) signal is used in memory transfers to indicate a memory read operation. It also describes the type of transfer in nonmemory transfers.

6. The error (LERROR) signal indicates a parity error has occurred during a memory read access.

The specific tag line configurations defining the types of SelBUS transfers are listed in Table 3-1.

3.3.1.1.1 Write Data or Order Transfer (WDOT)

The write data or order transfer (WDOT) originates in the CPU to initiate or halt operations within the HSD. The HSD is addressed by destination bus bits 09 through 15, which contain the physical controller address. Destination bus bits 18 through 23 specify the individual device to be controlled by the HSD (refer to Table 3-1).

The type of information in the 32-bit data bus depends on the type of WDOT transfer. When data bus bit 00 is equal to one, the WDOT is a start order that causes the HSD to initiate an I/O operation. During the start operation, data bus bits 08 through 31 contain the memory address of the input/output command doubleword (IOCD). Word 2 of the IOCD contains the address of the TI dedicated location.

When data bus bit 01 is equal to one, the WDOT is a load RAM order that indicates data bus bits 25 through 31 contain the HSD's physical address, and destination bus bits 01 through 07 contain the interrupt priority level of the HSD. The load RAM occurs only when the CPU is executing its initial program load firmware sequence or following a system reset and provides each HSD with its interrupt priority level.

When data bus bit 02 is equal to one, the WDOT is a halt order, which causes the microprogram to halt all I/O operations currently in progress. During a halt order, the data bus bits 03 through 31 are not used.

3.3.1.1.2 Advance Interrupt Control Transfer (AICT)

The advance interrupt control transfer (AICT) originates in the CPU and preconditions the microprogram for an interrupt control operation.

The AICT preconditions the microprogram for interrupt control actions and causes the microprogram to perform the procedures required before executing the interrupt control action. When the microprogram is ready to execute the interrupt control action, it generates a ready signal to the computer by the SelBUS. The ready signal (a SelBUS response) is generated by the order structure logic of the microprogram. The ready signal causes the computer firmware to issue an interrupt control transfer (ICT) to the SelBUS. The ICT causes the microprogram to execute the interrupt control operation. In the CPU, the AICT is issued as a result of one of the software interrupt control instructions.

3.3.1.1.3 Interrupt Control Transfer (ICT)

The interrupt control transfer (ICT) originates in the CPU and is used in conjunction with the advance interrupt control transfer (AICT). The specific functions of the ICT were defined under AICT.

The ICT is the second and last transfer of an AICT or ICT pair used to control the interrupt logic of the SelBUS interface and microprogram. The CPU responds to the ready signal (AICT response) by generating an ICT to the HSD causing the microprogram to execute the interrupt control and generate a data return transfer (DRT) to the CPU. The DRT sent to the CPU is issued to acknowledge the ICT. The contents of the DRT are insignificant to the CPU.

In actual operation the CPU may not respond immediately to the ready signal with the ICT. If the ICT is received within 9 bus cycles following the ready signal, the microprogram must wait for 18 bus cycles (for RI and AI instructions only) before generating the DRT to the CPU. The 18 cycles must be timed from the point at which execution of the ICT is complete. The 18 bus cycles can be timed by using the test structure to test the interrupt sync flip-flop. The HSD guarantees that two sync times will occur from the time it receives the ICT until it sends a DRT back to the CPU.

3.3.1.1.4 Advance Read Status Transfer (ARSTX)

The advance ready status transfer (ARSTX) originates in the CPU, preconditions the microprogram for a status transfer to the CPU, and causes the microprogram to assemble the requested status.

NOTE

> The HSD always immediately returns ready and always sends zeros for status as a result of an ARSTX and RSTX.

When the microprogram has the requested status assembled and ready for transfer to the CPU, the microprogram issues a ready signal to the CPU by the SelBUS. The ready signal is generated by the HSD CPU interface control sequencer order structure. The CPU responds to the ready signal by generating a read status transfer (RSTX) to the HSD. The RSTX causes the microprogram to transfer the assembled status (all zeros) to the CPU in a data return transfer (DRT) on the SelBUS.

The acknowledge interrupt function of the ARSTX/RSTX is generated by the CPU firmware in response to a service interrupt that was generated by the HSD. The acknowledge interrupt causes the HSD microprogram to set the HSD interrupt level active and inhibit all lower priority interrupts in the computer system. The HSD interrupt remains active until a deactivate interrupt AICT/ICT is received from the CPU. The CPU firmware generates the deactivate interrupt AICT/ICT as a result of the firmware sequence of the software deactivate interrupt, interrupt control instruction. The content of the DRT returned to the CPU in response to the acknowledge interrupt ARSTX/RSTX is neither used nor significant.

The device and controller status function of the ARSTX/RSTX is generated by the CPU firmware as part of the sequence for the software test device instruction. The test device instruction is used only to determine whether the HSD is configured in the computer system.

3.3.1.1.5 Read Status Transfer (RSTX)

The read status transfer (RSTX) originates in the CPU and is used in conjunction with the advance read status transfer (ARSTX). The RSTX is the second and last transfer of an ARSTX and RSTX pair used to request a status transfer from the HSD. The CPU responds to the ready signal (caused by ARSTX) by generating an RSTX to the HSD, causing the microprogram to transfer the requested status (all zeros) to the CPU in a data return transfer (DRT).

The microprogram assumes that the CPU was interrupted by another device, and that the ARSTX will be reissued. If the RSTX is received within nine bus cycles following the ready signal, the microprogram responds with the DRT containing the requested status.

3.3.1.1.6 Data Return Transfer (DRT)

The data return transfer (DRT) may be originated by either the HSD or the memory bus controller (MBC). When the DRT is originated by the HSD, its destination is the CPU, and the

DRT contains status (always zeros) that was requested by a CPU generated AICT/ICT or ARSTX/RSTX transfer pair. When the DRT is originated by the MBC, its destination is the HSD, and the DRT contains data read from memory at the request of a memory read transfer (MRT).

3.3.1.1.7 Error Transfer (ET)

The error transfer (ET) is a single tag line transfer originating in the memory and consisting of the error tag line.

3.3.1.1.8 Memory Write Transfer (MWT)

The memory write transfer (MWT) originates in the HSD, and its destination is the memory connected to the SelBUS. The destination of the MWT is indicated by the true condition of the memory (LMEM) signal tag line. The MWT is indicated as a memory write by the false condition of the read (LRD) signal tag line.

During an MWT, the destination bus contains the memory address, and the data bus contains the data (32 bits) to be stored at the memory address. The SelBUS interface logic monitors the transfer acknowledge (LTA) signal line during the SelBUS cycle following the MWT. If transfer acknowledge is not received, the MWT is assumed to have addressed nonpresent memory and the SelBUS logic notifies the microprogram of the error condition. The microprogram terminates the I/O operation and generates a priority (service) interrupt to the computer.

3.3.1.1.9 Memory Read Transfer (MRT)

The memory read transfer (MRT) originates in the HSD, and its destination is the memory connected to the SelBUS. The destination of the MRT is indicated by the true condition of the memory (LMEM) signal tag line. The MRT is indicated as a memory read by the true condition of the read (LRD) signal tag line.

During the MRT, the destination bus contains the memory address to be read, and the data bus contains the HSD physical address in bits 24 through 31. When the location in memory specified by the memory address has been read, a data return transfer (DRT) is returned to the HSD specified by the physical address in the memory read transfer (MRT). The data return transfer contains the data read from memory. If an error occurs during the memory read operation, the memory generates an error transfer (ET) on the next cycle following the data return transfer. The error transfer indicates that a parity error had occurred during the memory read.

The SelBUS interface logic monitors the transfer acknowledge (LTA) signal line for the LTA signal on the next cycle following the MRT. If the LTA signal is not present, the MRT is assumed to have addressed nonpresent memory, and the microprogram terminates the I/O operation and generates a priority (service) interrupt to the computer.

3.3.1.2 SelBUS Interface Hardware Components

The Selbus interface logic on the HSD circuit card provides the means of inputting data from the SelBUS to the HSD storage registers or outputting data from the HSD storage registers to the SelBUS. Figure 3-3 illustrates the basic layout of the component side of the HSD circuit card.

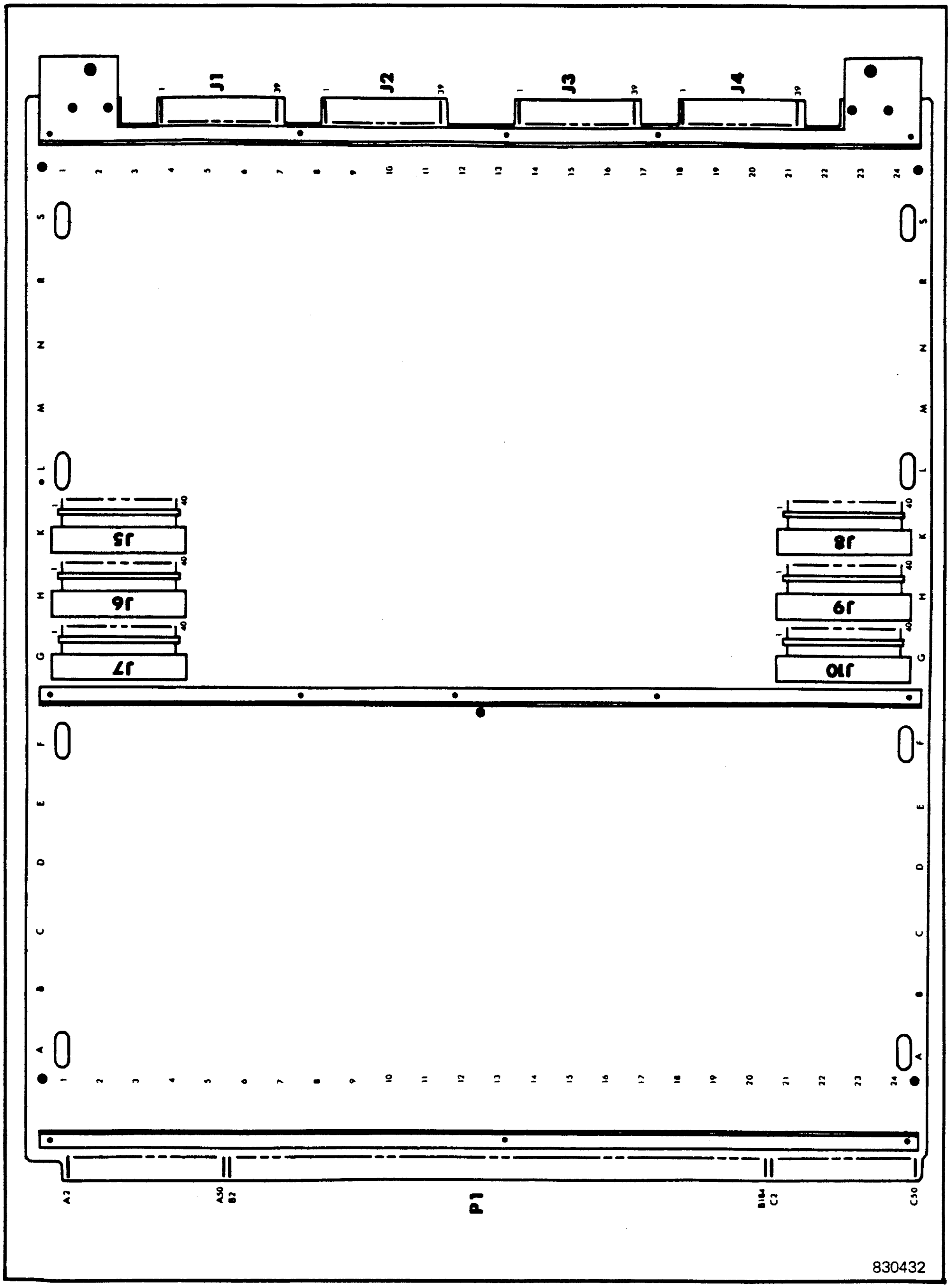


Figure 3-3. High-speed Data Interface Circuit Card Layout (Typical)

The large connector end of the circuit card is the P1 connector, which plugs into the computer logic chassis. The P1 connector is divided into threee subconnectors: P1A, P1B, and P1C. The P1A and P1C connectors are 50-pin connectors that provide connections from the customer interface. The P1B connector, a 184-pin connector, connects the SelBUS interface to the SelBUS. Table 3-3 lists the signal pin assignments for the connector and SelBUS.

The following discussions of the SelBUS interface hardware components contain specific references to the HSD logic diagrams. Logic references call out logic sheet numbers. Figure 3-4 provides a general and functional block diagram of the SelBUS interface.

3.3.1.2.1 SelBUS Transmitters

All signals that pass along the SelBUS are bidirectional, open collector signals. The pull-up resistors for each signal line are provided by the SelBUS. The transmitters used for each signal line driven by the HSD are 75453B integrated circuits (IC) and referenced by a logic callout of 75453 on the logic drawings. All signals on the SelBUS, except HPR01-22, are low true signals.

3.3.1.2.2 SelBUS Receivers

Specific receiver circuits are not used in the HSD to receive SelBUS signals. Each hardware component of the SelBUS interface logic receives its signals from the SelBUS by the functional circuit that uses that signal.

3.3.1.2.3 Address Comparator

The address comparator logic of the SelBUS interface determines when a transfer on the SelBUS is destined for the HSD. The determination is based on a comparison of destination bus bits 09 through 15 with the physical address of the HSD. If the two addresses compare, the memory tag signal and CPU address bits are false, and the SelBUS interface stores the contents of the transfer at the end of the transfer cycle.

The address comparator logic, composed of two comparator ICs and eight jumpers, is shown on logic sheet 7. The eight jumpers must be set at installation to reflect the HSD physical address. The jumper address is compared by the first comparator circuit to destination bus bits 09 through 14 (LDT09 through LDT14) from the SelBUS.

The second comparator tests for the following: (1) a comparison between the least significant bit of the address switch and destination bus bit 15 (LDT15); (2) a logic low level on the transfer (LTX) tag signal line; and (3) logic high levels on the memory (LMEM) tag signal line and CPU address bit (LDT08). A comparison for a logic high level is made by comparing the signals to a +V and a logic low level to ground. The true (low) condition of the transfer tag signal line indicates that a valid transfer is on the SelBUS, and the false (high) condition of the memory tag signal and CPU address bit (destination bus bit 08) indicates that the transfer is not intended for the CPU or memory.

If all comparisons are true, the address comparator assumes that the SelBUS tranfer is for the HSD and enables the input transfer recognition logic. The address comparison occurs during the middle of a SelBUS transfer cycle, and the comparison must be true by the end of the transfer cycle.

3.3.1.2.4 Data Bus Logic

The SelBUS interface data bus logic performs two functions, as follows.

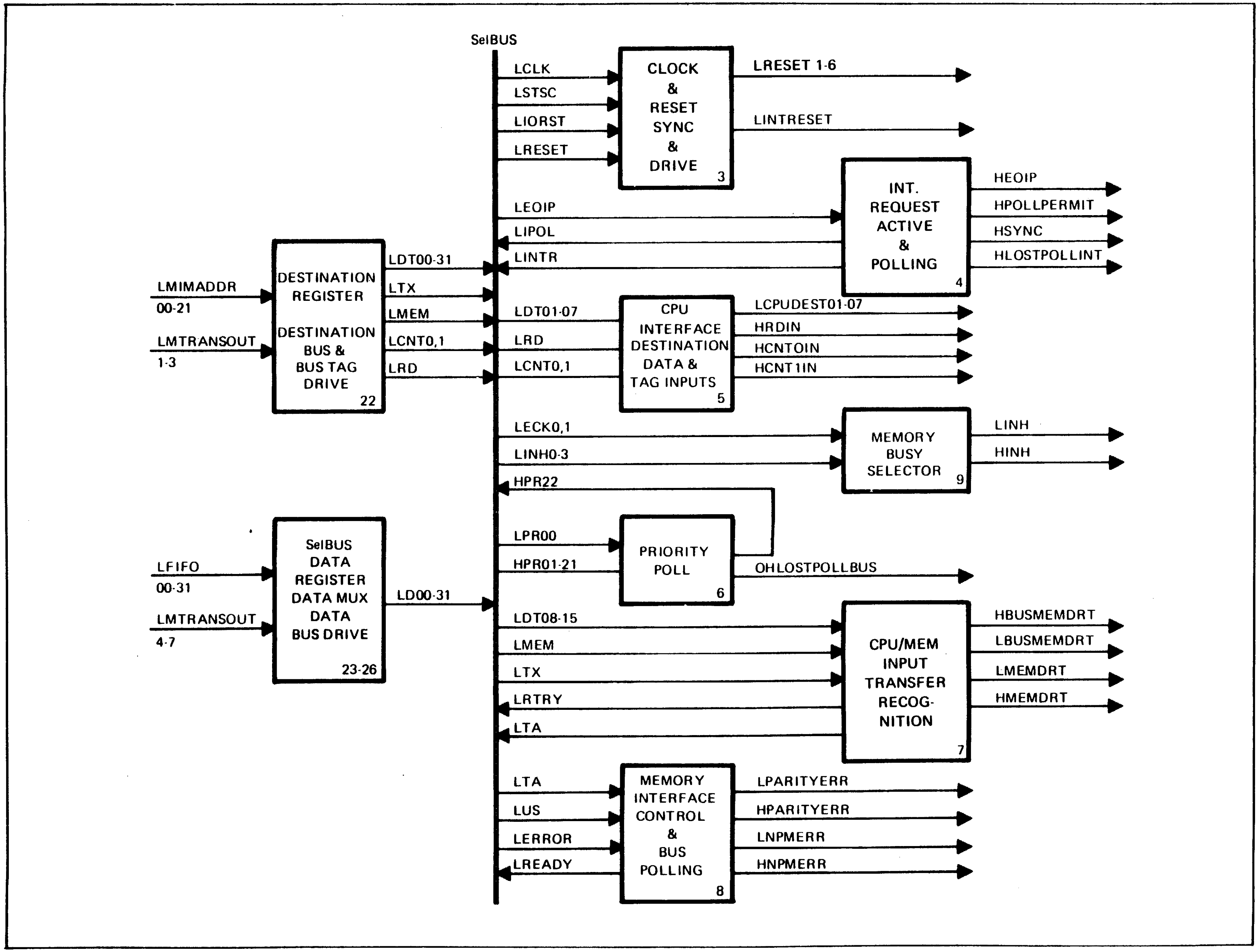


Figure 3-4. SelBUS Interface Block Diagram

Table 3-3
SelBUS Connector (P1B) Assignments

Pin	Signal	Pin	Signal
1	GND BUS	52	LDT02
2	GND BUS	53	LDT04
3	+5V BUS	54	GND
4	+5V BUS	55	LDT06
5	LD01	56	LDT05
6	LD00	57	LDT08
7	LD03	58	LDT07
8	LD02	59	LDT10
9	LD04	60	LDT09
10	GND	61	LDT12
11	LD06	62	LDT11
12	LD05	63	GND
13	SPARE	64	LDT13
14	LD07	65	LDT15
15	LD09	66	LDT14
16	LD08	67	LDT17
17	LD11	68	LDT16
18	LD10	69	LDT19
19	GND	70	LDT18
20	LD12	71	LDT21
21	LD14	72	LDT20
22	LD13	73	LDT22
23	SPARE	74	SPARE
24	LD15	75	LDTF
25	LD17	76	LDT23
26	LD16	77	LREADY
27	LD19	78	GND
28	LD18	79	GND
29	LD20	80	MIUR
30	GND	81	LCLKE
31	LD22	82	GND
32	LD21	83	GND
33	SPARE	84	LCLK
34	LD23	85	LCLKL
35	LD25	86	GND
36	LD24	87	GND
37	LD27	88	LSTSC
38	LD26	89	+12V BUS
39	GND	90	+12V BUS
40	LD28	91	GND BUS
41	LD30	92	GND BUS
42	LD29	93	GND BUS
43	SPARE	94	GND BUS
44	LD31	95	-12V BUS
45	GND BUS	96	-12V BUS
46	GND BUS	97	LCPUSC
47	+5V BUS	98	GND
48	+5V BUS	99	GND
49	LDT01	100	MIUL
50	LDT00	101	LINH00
51	LDT03	102	GND

Table 3-3
SelBUS Connector (P1B) Assignments (Cont.)

Pin	Signal	Pin	Signal
103	LINH01	144	LMEM
104	LRESET	145	HPR02
105	LINH02	146	GND
106	LCLK0V	147	HPR03
107	LINH03	148	LUS
108	LRTC	149	HPR04
109	SPARE	150	LPFM
110	SPARE	151	HPR05
111	GND	152	LMLK
112	LPF	153	HPR06
113	SPARE	154	LIPOL
114	SPARE	155	GRD
115	SPARE	156	LEOIP
116	LMUNLK	157	HPR07
117	SPARE	158	LINTR
118	LERROR	159	HPR08
119	LECK0	160	LIOIN
120	LRTRY	161	HPR09
121	LECK1	162	LEXIN
122	GND	163	HPR10
123	LD32/P0	164	SPARE
124	LCHBSY	165	HPR11
125	LD33/P1	166	GND
126	LTX	167	HPR12
127	LD34/P2	168	SPARE
128	SPARE	169	HPR13
129	LPD0/P3	170	LIORST
130	LTA	171	HPR15
131	GND	172	HPR14
132	LCNT0	173	HPR17
133	LSCPTTN	174	HPR16
134	LCNT1	175	GND
135	LREFRESH	176	HPR18
136	LCPU	177	HPR20
137	+5V BUS	178	HPR19
138	+5V BUS	179	HPR22
139	GND BUS	180	HPR21
140	GND BUS	181	+5V BUS
141	LPR00	182	+5V BUS
142	LRD	183	GND BUS
143	HPR01	184	GND BUS

1. It receives and stores 32 bits from the SelBUS during the input transfers at the command of the SelBUS interface control logic.

2. It receives and stores 32 bits for the data bus from the first-in-first-out (FIFO) storage registers while an output transfer is being assembled and then transmits the 32 bits to the SelBUS at the command of the SelBUS interface control logic.

The data bus logic shown on logic drawing sheets 23 through 26 consists of 12 four-line-to-one-line multiplexers; 4 two-line-to-one-line multiplexers; 6 hex D flip-flops; and an output transmitter for each of the 32 bits. The storage flip-flops are also referred to as the staging register because they provide staging for the data bus during input transfer and during output transfer assembly.

The multiplexer uses the D multiplexer select (DMUXSELA and DMUXSELB) signals to select the inputs for storage in the staging register. The D multiplexer select signals are generated by the memory interface control sequencer execute gating and control logic (sheet 16). The memory data register clock (HMDATAREGCLK) generated by the memory interface clock control (sheet 9) strobes the operand gated through the multiplexer into the staging register when the HMDATAREGCLK signal goes high.

The outputs of the staging register are the LDREG00 through LDREG31 signals. The LDREG00-LDREG31 signals go to the output transmitters for the SelBUS and to the FIFO and FIFO input multiplexer (sheets 27 and 28). The staging register outputs are gated through the output transmitters to the SelBUS when the transmit output (LMTRANSOUT) signal is low. The transmit output is generated by the SelBUS interface control logic when bus transfer priority has been obtained by the control logic as shown on logic drawing sheet 8.

3.3.1.2.5 Destination Bus Logic

The SelBUS interface destination bus logic performs two functions as follows:

1. During input transfers, the SelBUS interface's destination bus logic receives and stores 7 (LDT01 through LDT07) of the 24 destination bus bits at the command of the SelBUS interface control logic. Eight bits (LDT08 through LDT15) of the destination bus are used by the address comparator and are not used by the destination bus logic.

2. During output transfers, the destination bus logic receives and stores 22 bits from the control logic and then transmits the 22 bits to the SelBUS at the command of the SelBUS interface control logic.

The HCPIFREGCLK signal generated by the input transfer recognition logic (sheet 7) is used to strobe the destination bus bits 01 through 07 into the storage flip-flop (staging register).

During the input transfers, destination bus bits 08 through 15 provide the HSD physical address, which is used only by the address comparator.

During output transfers, the destination bus bits 00 through 21 are assembled in the staging register (logic sheet 22). If the transfer is to memory, the output transmitters are turned on by the low condition of the memory transmit output (LMTRANSOUT) signals and gate the contents of the staging register to the SelBUS destination bus. The memory transmit output signal is generated by the SelBUS interface control logic when the bus transfer priority has been obtained by the control logic. If the output transfer is to the CPU, the output transmitters are turned on by the low condition of the CPU transmit output (LCPUTRANSOUT) signal and gate only bit 08 (LDT08) on the SelBUS destination bus.

3.3.1.2.6 Tag Bus Logic

The SelBUS interface tag bus logic receives the tag signals during input transfers and transmits the tag signals during output transfers. Tag bus logic handles the following signals:

1. The transfer (LTX) signal
2. The memory (LMEM) signal
3. The control 0 (LCNT0) signal
4. The control 1 (LCNT1) signal
5. The read (LRD) signal
6. The error (LERROR) signal

3.3.1.2.6.1 Input Transfer Tag Bus Logic

The input transfer tag bus logic receives the tag signals and stores them in a staging register for examination by the microprogram test structure logic. However, all input tag signals are not staged through the staging register. The transfer signal (which indicates that the transfer on the SelBUS is valid) and the memory signal (which indicates that a transfer is addressed to memory) are used directly by the address comparator (sheet 7) to determine whether the transfer is for this HSD.

The tag bus read, control 0, and control 1 signals are received by the SelBUS control logic (sheet 5) and are used to determine the interface response to the SelBUS transfer. The interface can generate a transfer acknowledge or retry response to the transfer. However, if the transfer is accepted, the interface must generate the transfer acknowledge response signal. The logic that receives the three tag bus signals generates the HRD, HCNT0, and HCNT1 signals, which are presented to the staging register (sheet 5). At the end of the cycle, the interface control logic generates the HCPIFREGCLK signal that strobes the tag signals into the staging register. The outputs of the staging register are presented to the microprogram test structure.

The tag bus error (LERROR) signal is received from the SelBUS by the staging register flip-flop (sheet 8). The error signal is clocked into the flip-flop by the HTRANSIN+1 signal at the end of a transfer cycle. The content of the error flip-flop is valid only at the end of the transfer cycle following a data return transfer (DRT) from memory to the HSD. When set, the error flip-flop generates the parity error (HPARITYERR) signal to the microprogram test structure. The set condition of the error flip-flop indicates that the data return transfer received from memory on the previous transfer cycle contained a parity error. The error flip-flop is tested by the microprogram following every memory read access to insure that the data received from memory had good parity.

3.3.1.2.6.2 Output Transfer Tag Bus Logic

The output transfer tag bus logic generates the tag bus signals as directed by the microprogram during output transfer to the SelBUS. The output transfer tag bus logic (sheet 22) consists of a staging register shared with destination bus bits 18 through 21 (LDT18 through LDT21) and a set of output transmitters.

The read (LRD) tag bus signal is loaded into the staging register by the memory destination register clock (HMDESTREGCLK) at the same time the destination bus staging register is presented to the output transmitter to generate the tag bus signals. The tag bus signals LRD, LMEM, and LTX are placed on the SelBUS by the memory transmit out (LMTRANSOUT) signal during a memory transfer. The LCNT1, LCNT0, and LTX signals are placed on the SelBUS by the CPU transmit out (LCPUTRANSOUT) signal when the transfer is to the CPU.

3.3.1.2.7 Response Bus Logic

The SelBUS interface response logic receives or generates response signals to/from the SelBUS. The interface response logic receives signals, generated by the memory bus controller (MBC). Two types of response signals are received from the MBC as follows:

1. The transfer acknowledge (LTA) signal
2. The unsuccessful (LUS) signal

The SelBUS interface logic generates any of the following three response signals after a WDOT, ARSTX, RSTX, AICT, or ICT transfer from the CPU:

1. The ready (LREADY) signal
2. The transfer acknowledge (LTA) signal
3. The retry (LRTRY) signal

3.3.1.2.7.1 Transfer Acknowledge Signal (Received)

The transfer acknowledge signal (sheet 8) received by the HSD is generated by the MBC to acknowledge receipt of a memory read transfer or a memory write transfer on the previous bus transfer cycle. The HTA signal is ANDed with the write tranfer out+1 (HWRTROUT+1) signal and the false condition of the unsuccessful (LUS) signal to generate the write successful (LWRITESUCC and HWRITESUCC) signals. The LWRITESUCC signal is used by the memory interface control sequencer to indicate that the memory write operation has been successfully completed.

If the transfer acknowledge signal is not received from the MBC on the cycle following the memory transfer (MWT or MRT), the response logic assumes that the memory location addressed is not present in the computer system. The false condition of the transfer acknowledge (LTA) is ANDed with the transmit out+1 (HTRANSOUT+1) signal to generate set steering to the nonpresent memory flip-flop. The flip-flop is triggered set at end of the transfer cycle. The nonpresent memory error (LNPMERR) signal is presented to the memory interface control sequencer test structure to identify the error condition. In this error condition, the nonpresent memory flip-flop remains set until the microprogram generates the reset status (HRESSTATUS) signal which clears the nonpresent memory flip-flop.

3.3.1.2.7.2 Unsuccessful Signal

The unsuccessful (LUS) signal is generated by the MBC to indicate a busy condition. The LUS (sheet 8) signal received by the HSD is used to inhibit the setting of the write successful flip-flop. The HUS signal is ANDed with the HTRANSOUT+1 signal to initiate another SelBUS polling cycle for the memory transfer.

3.3.1.2.7.3 Ready Signal

The ready signal (sheet 8) is generated by the SelBUS interface logic at the command of the microprogram to indicate to the CPU that the HSD is ready to execute a status transfer or an interrupt control operation.

3.3.1.2.7.4 Transfer Acknowledge Signal (Generated)

The transfer acknowledge (LTA) signal (sheet 7) is generated by the SelBUS interface logic whenever the interface logic is not busy, and the SelBUS transfer can be accepted.

3.3.1.2.7.5 Retry Signal

The retry (LRTRY) signal (sheet 7) is generated by the SelBUS interface control logic in response to a WDOT, ARSTX, RSTX, AICT, or ICT transfer from the CPU. The retry signal indicates that the previous transfer has been rejected due to a busy condition of the SelBUS interface or microprogram, and the CPU should reissue the SelBUS transfer. The retry signal implies that the HSD will not be busy for an extended period of time, such as the period of time required for a complete I/O block transfer.

3.3.1.2.8 SelBUS Transfer Priority Polling Logic

The SelBUS interface bus transfer priority polling logic (sheet 6) consists of a set of priority recognition jumpers and a set of priority generation jumpers. Refer to the appropriate CPU technical manual for system jumper configuration.

3.3.2 Data Structure Hardware Components

The HSD data structure block diagram is illustrated on logic diagram 130-103071, sheet 2. The following discussions of the data structure hardware components contain specific references to the HSD logic diagram. Logic references call out logic sheet numbers.

3.3.2.1 Input/output Operation Code Register

The load operand one (HLOADOPR1 and load transfer count and operand two (LLDTCANDOPR2) generated by the memory interface control sequencer gating and control logic (sheet 16) load operand codes one and two into the input/output operand code registers one and two (sheet 36), respectively.

The HLOADOPR1 signal clocks the operand bits (LDREG00) through (LDREG03) from the SelBUS data register (sheet 24) into the operand one register, which is composed of hex D-type flip-flops.

The LLDTCANDOPR2 signal loads the operand two register with operand bits (LDREG04) through (LDREG07) from the SelBUS data register. The operand two register is a 4-bit synchronous counter that is parallel loaded.

The outputs of the operand registers are applied to a programmable ROM (sheet 36), which monitors the operand registers to determine whether the operand code is valid. If the operand code is not valid, the 4 x 512 programmable ROM outputs a bad operand code (LBADOPCODE) signal and applies it to the memory interface control sequencer test structure (sheet 15).

3.3.2.2 Transfer Counter

The transfer counter (sheet 36) consists of four 4-bit synchronous counters which are loaded with the transfer count (LDREG16 through LDREG31) from the SelBUS data register (sheets 25 and 26) by the load transfer count and operand two (LLDTCANDOPR2) signal at H3CLK9 time.

The transfer counter is decremented by the decrement transfer count (HDECXFRCNT) signal generated by the memory interface control sequencer gating and control logic (sheet 16) when HMSCREG21 is present in the microinstruction, and one of the following conditions exists:

1. Execute
2. HRDEXCOND o HMEMDRT/LNPMERP

3. HWRITESUCC • HWREXCOND

3.3.2.3 Command Address Register

The command address register (sheets 34 and 35) consists of five 73169 four-bit synchronous counters and one 73163 four-bit counter.

The command address register is loaded with the memory buffer address (word address) from the SelBUS data register output (LDREG08 through LDREG29) by the load command address register (LLOADCAR) signal, which is generated by the memory interface control sequencer gating and control logic (sheet 16).

The command address register is incremented by the increment command address register (LINCCAR) signal generated by the memory interface control sequencer order structure (sheet 14).

3.3.2.4 Transfer Interrupt Address Register

The transfer interrupt address register (sheet 35) consists of two four-bit D registers with tri-state outputs. The TI address register is loaded with the transfer interrupt address from the output of the SelBUS data register (LDREG18 through LDREG29) by the load transer interrupt address (LLOADTIADR) signal generated by the memory interface control sequencer gating and control logic (sheet 16).

The contents of the TI address register, memory address bit 14 through memory address bit 21 (LMEMADDR14 through LMEMADDR21), is gated out by the enable multiplexer out (HENAMUXOUT) signal generated by the memory interface control sequencer gating and control logic (sheet 16).

3.3.2.5 Memory Address Register Multiplexer

The memory address register multiplexer (sheet 33) consists of six 2-line-to-1-line multiplexers.

Inputs to the memory address register multiplexer (sheets 25 and 26) are taken from the SelBUS data register bits 08 through 29 (LDREG08 through LDREG29) or from the customer input register (sheets 29 through 32) bits 00 through 29 (LCIR00 through LCIR29).

The input to the memory address register multiplexer is selected by the true or false condition of the memory address multiplexer select customer input register (HMAMUXSELCIR) signal generated by the memory interface control sequencer gating and control logic (sheet 16). The high true condition of the HMAMUXSELCIR signal selects the customer input register outputs as the input to the memory address register multiplexer. The false condition of the HMAMUXSELCIR signal selects the outputs of the SelBUS data register as the inputs to the memory address register multiplexer.

3.3.2.6 Memory Address Register

The memory address register (MAR) consists of six four-bit binary counters (sheet 33). The MAR is loaded from the output of the memory address register multiplexer by the load memory address register (LLOADMAR) signal generated by the memory interface control sequencer gating and control logic (sheet 16). The memory address in the MAR can be incremented by the increment memory address register (LINCMAR) signal generated by the memory interface control sequencer gating and control logic (sheet 16).

3.3.2.7 Address Multiplexer

The address multiplexer (sheets 34 and 35) consists of six quad 2-line-to-1-line data selector/multiplexers with tri-state outputs.

The inputs to the address multiplexer are taken from the outputs of the memory address register (sheet 33) or the output of the command address register. The inputs are determined by the state of the address multiplexer select command address register (HAMUXSELCAR) signal generated by the memory interface control sequencer gating and control logic.

The high true state of HAMUXSELCAR signal selects the command address register bits 00 through 21 (LCAR00 through LCAR21). The false condition of the HAMUXSELCAR signal selects the memory address register bits 00 through 21 (LMAR00 through LMAR21) as the input. The output of the address multiplexer is enabled by the enable multiplexer out (LENAMUXOUT) signal generated by the memory interface control sequencer gating and control logic (sheet 16).

3.3.2.8 Destination Register

The destination register (sheet 22) consists of four hex D flip-flops, which are loaded from the output of the address multiplexer by the memory destination register clock (HMDESTREGCLK) signal generated by the memory interface clock control logic (sheet 9). The output of the destination register is applied to the SelBUS drivers (sheet 22) and gated onto the SelBUS by the memory transmit out (LMTRANSOUT1, 2, and 3) signals generated by the memory interface control logic (sheet 8).

3.3.2.9 Customer Interface Transceivers

The customer interface transceivers (sheets 29 through 32) consist of eight 75138 quad bus transceivers.

The receiver portion of the transceivers receives the input from the customer device (LDAT00 through LDAT31) and applies them to the customer input register. The transmit portion of the transceivers places the outputs of the customer output register on the customer interface bus (LDAT00 through LDAT31) when the gating signal, gate data out (LGATEDATOUT), generated by the customer device interface control logic (sheet 20) is applied to the transceivers.

3.3.2.10 Customer Input Register

The customer input register (sheets 29 through 32) consists of eight quad D flip-flops and receives its inputs from the customer interface bus transceivers.

The customer input register is loaded by the load customer input (LLOADCUSIN) signal generated by the customer device interface input control logic (sheet 19).

3.3.2.11 Customer Output Register

The customer output register (sheets 29 through 32) consists of eight quad D flip-flops. The customer output register receives its inputs from the first-in-first-out (FIFO) register (LFIFO00 through LFIFO31) and is loaded by the load customer output (LLOADCUSOUT) signal generated by the customer device interface output control logic (sheet 20).

3.3.2.12 First-In-First-Out Multiplexer (F-Mux)

The first-in-first-out multiplexer (F-MUX) (sheets 27 and 28) consists of eight 2-line-to-1-line multiplexers.

The inputs to the F-MUX are taken from the outputs to the customer input register (sheets 29 through 32) or from the outputs of the SelBUS data register (sheets 23 through 26).

The inputs to the F-MUX are selected by the memory sequence write first in (HMSEQWRITE1) signal generated by the memory interface control sequencer gating and control logic (sheet 16). The high true condition of the HMSEQWRITE1 signal selects the data register inputs (LDREG00 through LDREG31). The false condition of the HMSEQWRITEF1 signal selects the customer input register inputs (LCIR00 through LCIR31).

3.3.2.13 First-In-First-Out (FIFO) Buffer

The first-in-first-out (FIFO) buffer consists of eight 64-bit (16 x 4) random access memories (RAMs) with tri-state outputs. The RAMs are arranged to form a buffer that will store sixteen 32-bit words on a first-in-first-out basis.

The inputs to the FIFO buffer are taken from the outputs of the FIFO mux. During the write cycle, the complement of the information at the data input is written into the selected location when both the chip-enable input and the read/write input are low. During the read cycle, the stored information (complement of the information applied at the data inputs during the write cycle) is available at the outputs when the read/write input is high and the chip-enable input is low.

3.3.2.14 First-In-First-Out (FIFO) Address Mulitplexer

The first-in-first-out (FIFO) address multiplexer (sheet 17) is a quad 2-line-to-1-line multiplexer that selects either the FIFO write address or the FIFO read address generated by the 4-bit synchronous counters (FIFO write address or FIFO read address).

The read or write address is selected by the state of the file select write read address (HFSELWRADR) signal. The high true state of the HFSELWRADR signal selects the FIFO write address, and the false condition of the HFSELWRADR signal selects the FIFO read address.

3.3.2.15 Data Multiplexer

The data multiplexer (D-MUX) consists of 12 dual 4-line-to-1-line and 4 quad 2-line-to-1-line multiplexers (sheets 23 through 26).

The inputs to the D-MUX may be the outputs of the command address register (LCAR00 through LCAR21), service interrupt (SI) status, HSD error status, the FIFO buffer file (LFIFO00 through LFIFO31), or SelBUS data bits 00 through 31 (LD00 through LD31). The inputs to the D MUX are selected by the data multiplexer select A1 and B1 (HDMUXSELA1 and HDMUXSELB1) signals generated by the memory interface control sequencer gating and control logic (sheet 16).

3.3.2.16 SelBUS Data Register

The SelBUS data register consists of hex D flip-flops (sheets 23 through 26) and stores the data selected by the D-MUX. The selected data is strobed into the data register by the memory data register clock (HMDATAREGCLK) signal generated by the memory interface clock control logic (sheet 9).

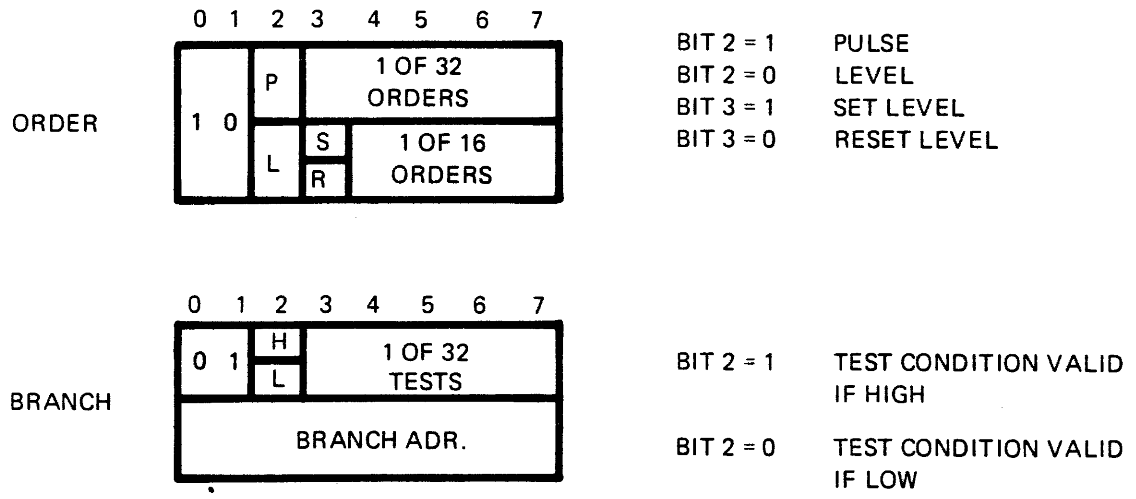
The output of the data register is placed on the SelBUS by the memory transmit output (LMTRANSOUT) signal generated by the memory interface control logic (sheet 8).

3.3.3 CPU Interface Control Sequence

The CPU interface control sequencer provides the primary control of the CPU transfers for the SelBUS interface logic on the HSD circuit card. In operation, it is the microprogram (firmware) that provides primary control of the HSD by commanding various portions of functional logic in the CPU interface control sequencer. The functional logic of the CPU interface control sequencer then generates the control signals to the SelBUS interface as commanded by the firmware. The firmware primary control can be broken into several areas of control which are the basis of organization of the firmware microprogram.

3.3.3.1 Microinstruction Format

The microinstruction is in an 8-bit word stored in a read-only memory (ROM) called the control memory (CROM). When a microinstruction is accessed by the program counter, it is read from the CROM into the control register (CREG). The content of CREG is used to control the basic operation of the microinstruction in the CPU interface control sequencer. Figure 3-5 illustrates the basic formats for the CPU interface control sequencer microword.



BIT 2 = 1	PULSE
BIT 2 = 0	LEVEL
BIT 3 = 1	SET LEVEL
BIT 3 = 0	RESET LEVEL

BIT 2 = 1	TEST CONDITION VALID IF HIGH
BIT 2 = 0	TEST CONDITION VALID IF LOW

Figure 3-5. CPU Interface Control Sequencer Microword Format

3.3.3.2 Program Counter

The program counter (PC) is an eight-bit counter that contains the current read-only memory address (microinstruction address). The program counter (sheets 10 and 11) is incremented consecutively for sequential microinstruction execution or loaded in parallel with a new value for branch microinstructions. When the program counter is loaded in parallel, it is loaded from a decode of the branch microinstruction with a value from the control read-only memory. The most-significant bit of the CROM address is the bank select flip-flop, selecting the upper or lower 256 words. The result of the test specified by the branch microinstruction must be true before the branch is executed (parallel loaded).

3.3.3.3 Control Memory

The CROM (sheets 10 and 11) is an eight-bit wide read-only memory (ROM) which contains up to 512 locations. The control memory addressing is provided by the program counter and the bank select bit for the firmware microprogram instruction location.

3.3.3.4 Control Register

The control register (sheets 10 and 11) is the data or instruction register for the control memory. The primary functions of any microinstruction, except test information for branch instructions, are decoded from the outputs of the control register.

3.3.3.5 Order Structure Logic

The order structure logic (sheets 10 and 11) generates either pulsed (refer to Table 3-4) or level (refer to Table 3-5) order signals by a decode of the microinstruction. The pulsed order signals are all negative-going pulses. The level order signals are all high true levels. The order signals generated by the order structure are used as logic control signals.

3.3.3.6 Test Structure Logic

The test structure logic (sheets 10 and 11) selects an input signal (refer to Table 3-6), tests the signal, and generates either a test true or a test false condition. In a test true condition, a branch instruction causes a branch in the microprogram execution; in a test false condition, the branch is not taken, and the next sequential microinstruction is executed.

The input signal to the test structure that is tested and the polarity for which the selected signal is tested are determined by decoding the branch microinstruction. One microinstruction may specify one input signal.

3.3.3.7 CPU Interface Control Sequencer Flowcharts

Table 3-7 lists the figure number and the title of the flowcharts applicable to the CPU interface control sequencer.

3.3.4 Memory Interface Control Sequencer

The memory interface control sequencer provides the primary control of memory transfers (MRT/MWT) for the SelBUS interface and HSD customer device interface logic on the HSD circuit card. In actual operation, it is the microprogram (firmware) that provides the primary

Table 3-4
CPU Interface Control Sequencer Pulse Orders

HCPSCREG03-07 Binary Address	Signal Name
0 0 0 0 0	LRESCPUDATAH
0 0 0 0 1	LRISETREQ
0 0 0 1 0	LRESREQ
0 0 0 1 1	LSETACTIVE
0 0 1 0 0	LRESACTIVE
0 0 1 0 1	LLDINTADR
0 0 1 1 0	LSETSIO
0 0 1 1 1	LSETHIOFLG
0 1 0 0 0	LSETREADY
0 1 0 0 1	LSETPROGVIOL

Table 3-5
CPU Interface Control Level Orders

HCPSCREG04-07 Binary Address	Signal Name
0 0 0 0	HSELBANK
0 0 0 1	HINTENABLE
0 0 1 0	HHIODELAYFLG
0 0 1 1	HSETCPUDRT

Table 3-6
CPU Interface Control Sequencer Test Conditions

HCPSCREG03-07 Binary Address	Signal Name
0 0 0 0 0	+V
0 0 0 0 1	HCPUDATAHERE
0 0 0 1 0	HRDIN
0 0 0 1 1	HCNTOIN
0 0 1 0 0	HCNT1IN
0 0 1 0 1	LCPDATA00
0 0 1 1 0	LCPDATA02
0 0 1 1 1	LCPDATA03
0 1 0 0 0	LCPDATA27
0 1 0 0 1	LCPDATA28
0 1 0 1 0	LCPDATA29
0 1 0 1 1	LCPDATA30
0 1 1 0 0	LCPDATA31
0 1 1 0 1	HEOIP
0 1 1 1 0	HHSDBUSY
0 1 1 1 1	HSIO
1 0 0 0 0	HHIODELAYFLG
1 0 0 0 1	HHIOFLAG

Table 3-7
CPU Interface Control Sequencer Flowcharts

Figure Number	Title
3-6	Start I/O (CPU)
3-7	Start (CPU)
3-8	Write Data or Order Transfer (CPU)
3-9	Load RAM (CPU)
3-10	Interrupt Control Transfer (CPU)
3-11	Enable Interrupt (CPU)
3-12	Request Interrupt (CPU)
3-13	Disable Interrupt (CPU)
3-14	Activate Interrupt (CPU)
3-15	Deactivate Interrupt (CPU)
3-16	Request Status Transfer (CPU)
3-17	Master Clear (CPU)
3-18	HALT I/O (CPU)

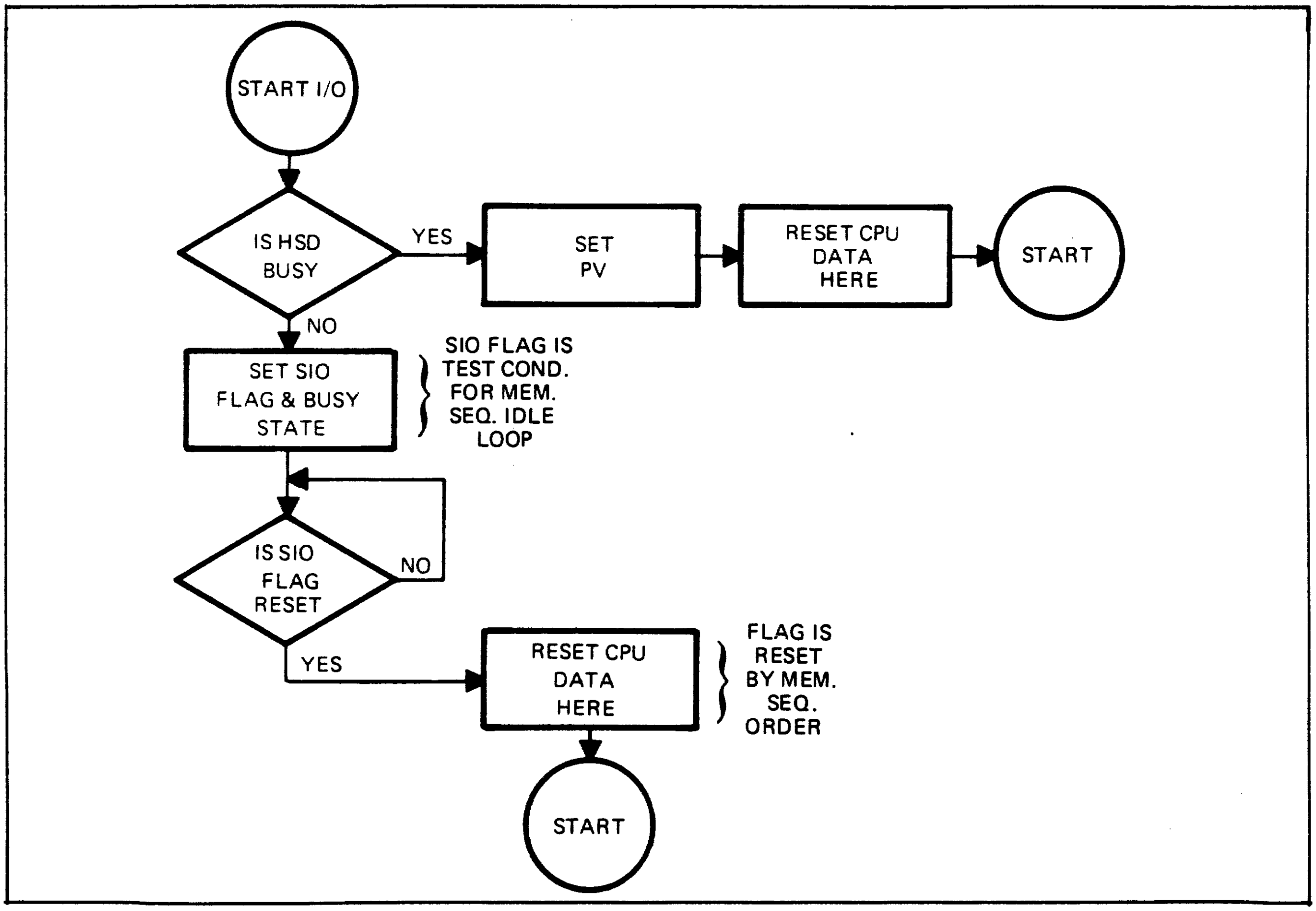

START I/O
IS HSD BUSY
YES
SET PV
RESET CPU DATA HERE
START
NO
SET SIO FLAG & BUSY STATE
SIO FLAG IS TEST COND. FOR MEM. SEQ. IDLE LOOP
IS SIO FLAG RESET
NO
YES
RESET CPU DATA HERE
FLAG IS RESET BY MEM. SEQ. ORDER
START

Figure 3-6. Flowchart - Start I/O (CPU)

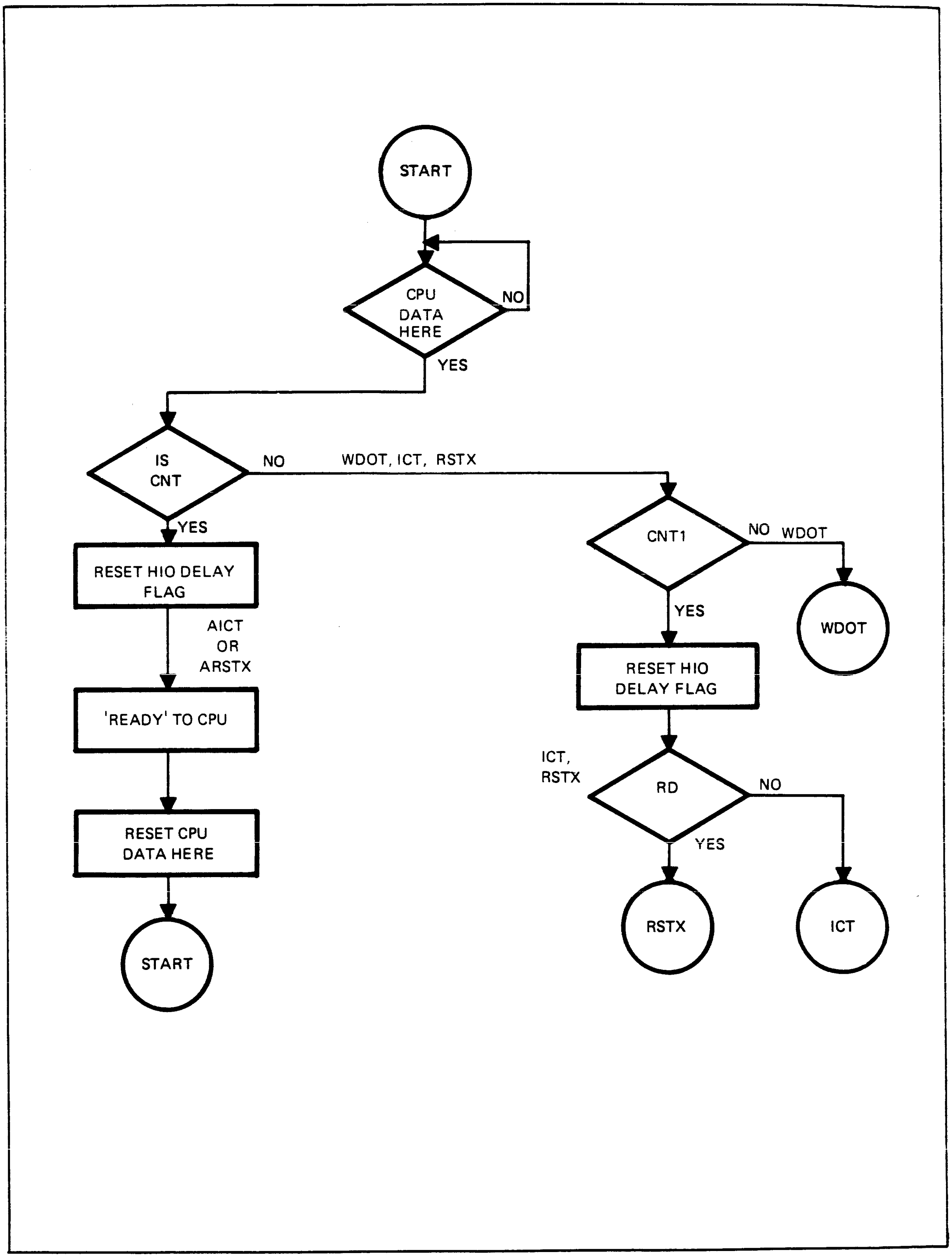


Figure 3-7. Flowchart - Start (CPU)

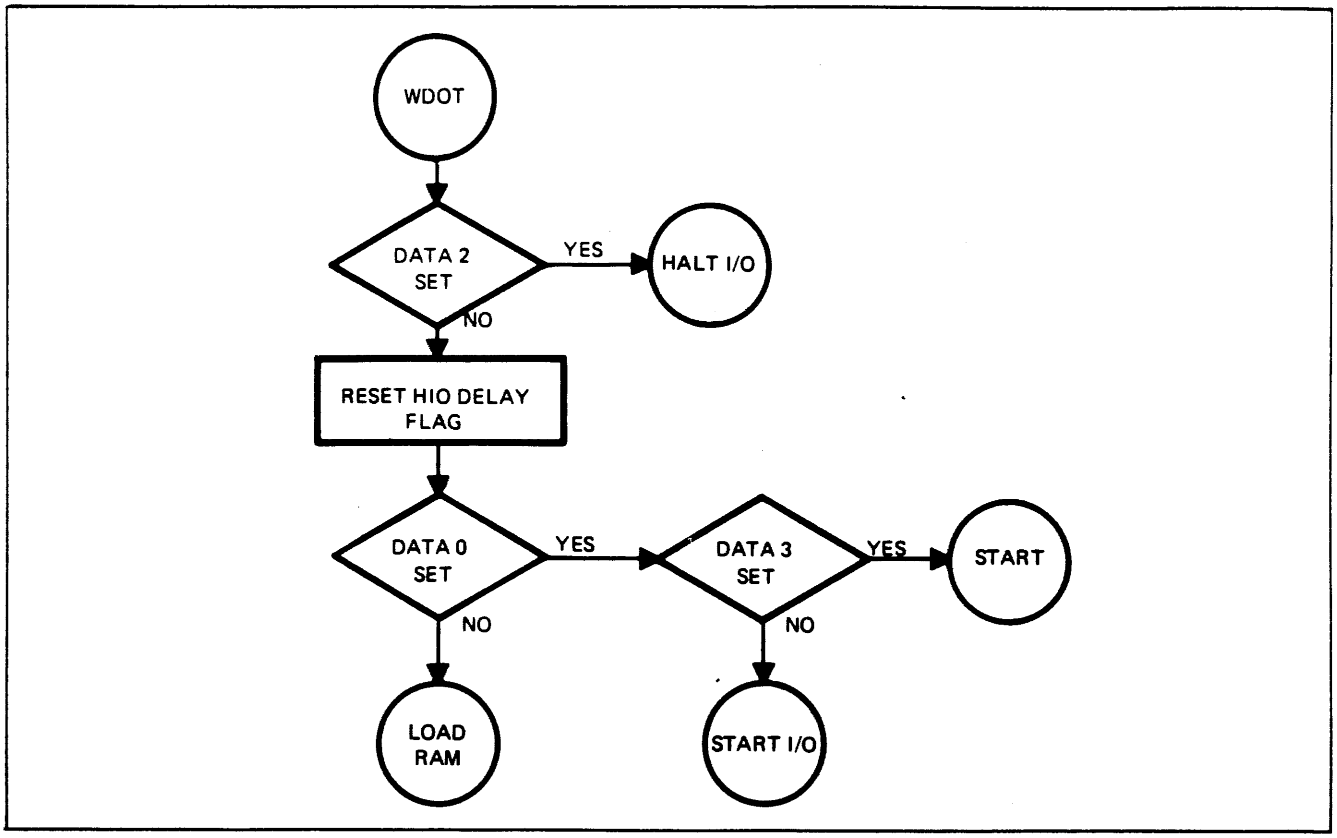


Figure 3-8. Flowchart - Write Data or Order Transfer (CPU)

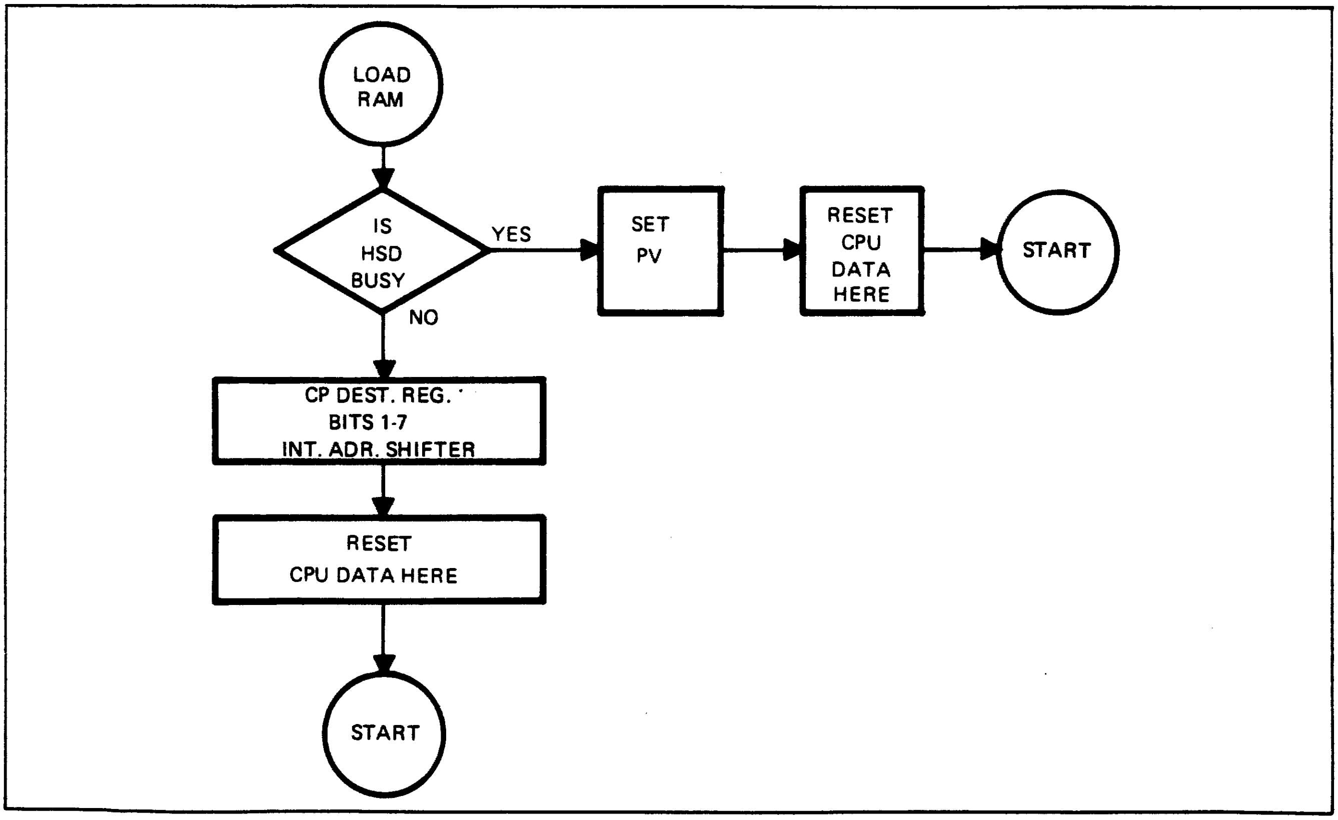


Figure 3-9. Flowchart - Load RAM (CPU)

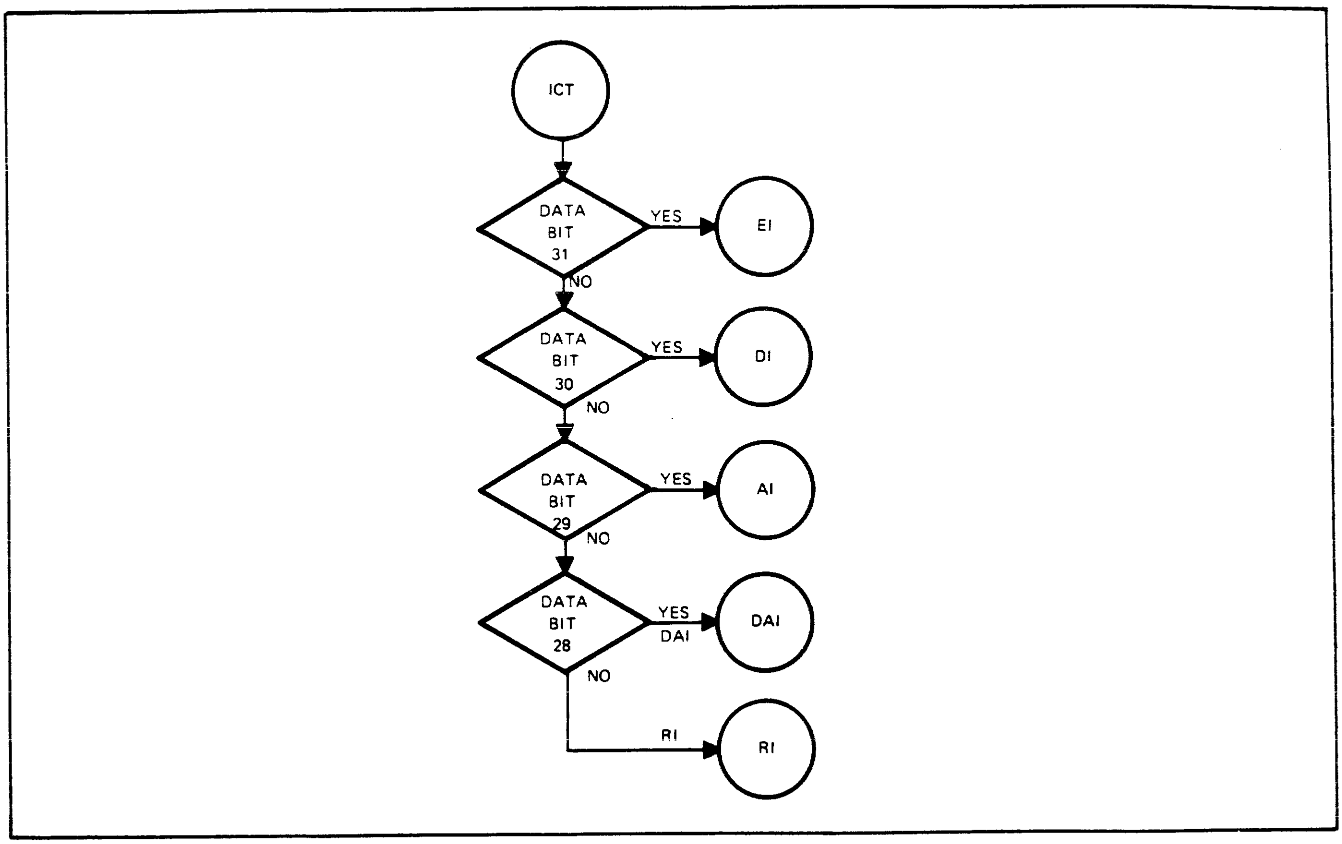


Figure 3-10. Flowchart - Interrupt Control Transfer (CPU)

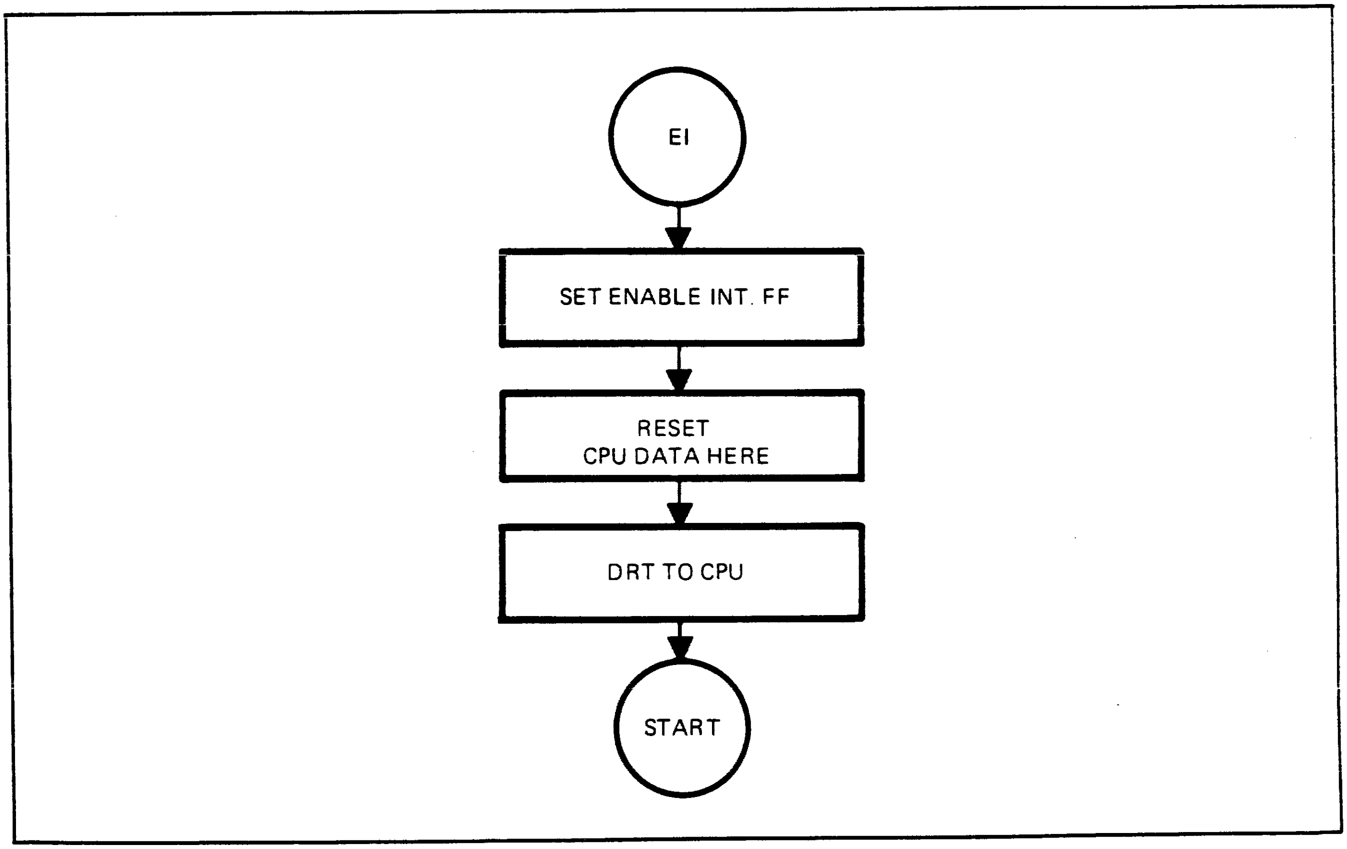


Figure 3-11. Flowchart - Enable Interrupt (CPU)

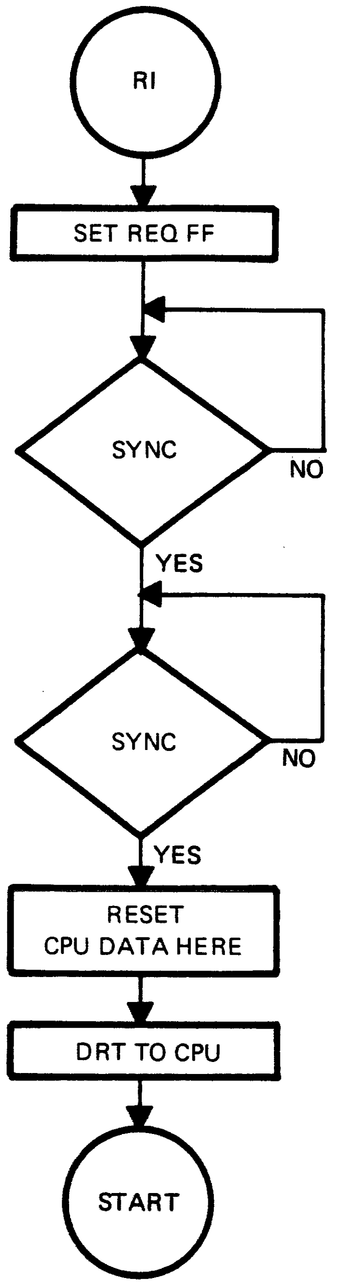


Figure 3-12. Flowchart - Request Interrupt (CPU)

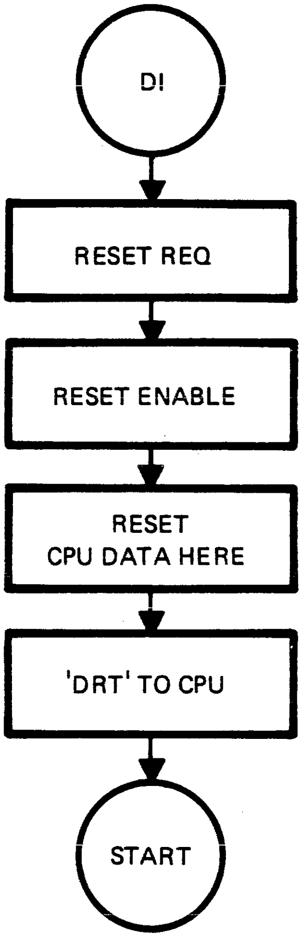


Figure 3-13. Flowchart - Disable Interrupt (CPU)

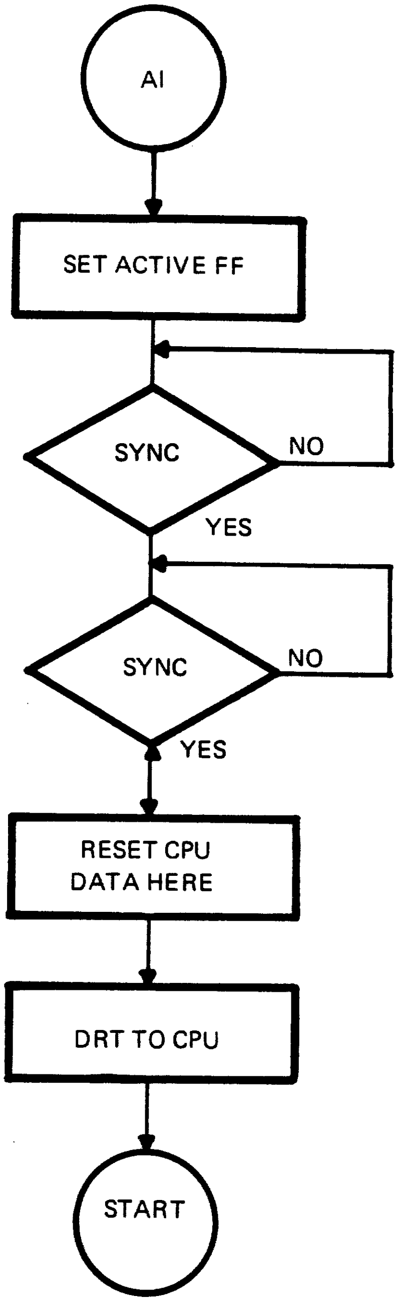


Figure 3-14. Flowchart - Activate Interrupt (CPU)

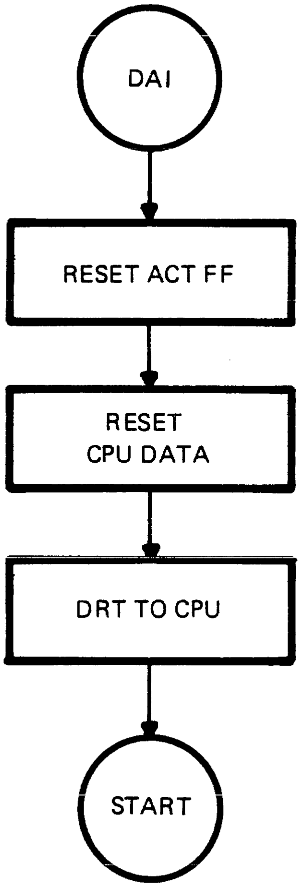


Figure 3-15. Flowchart - Deactivate Interrupt (CPU)

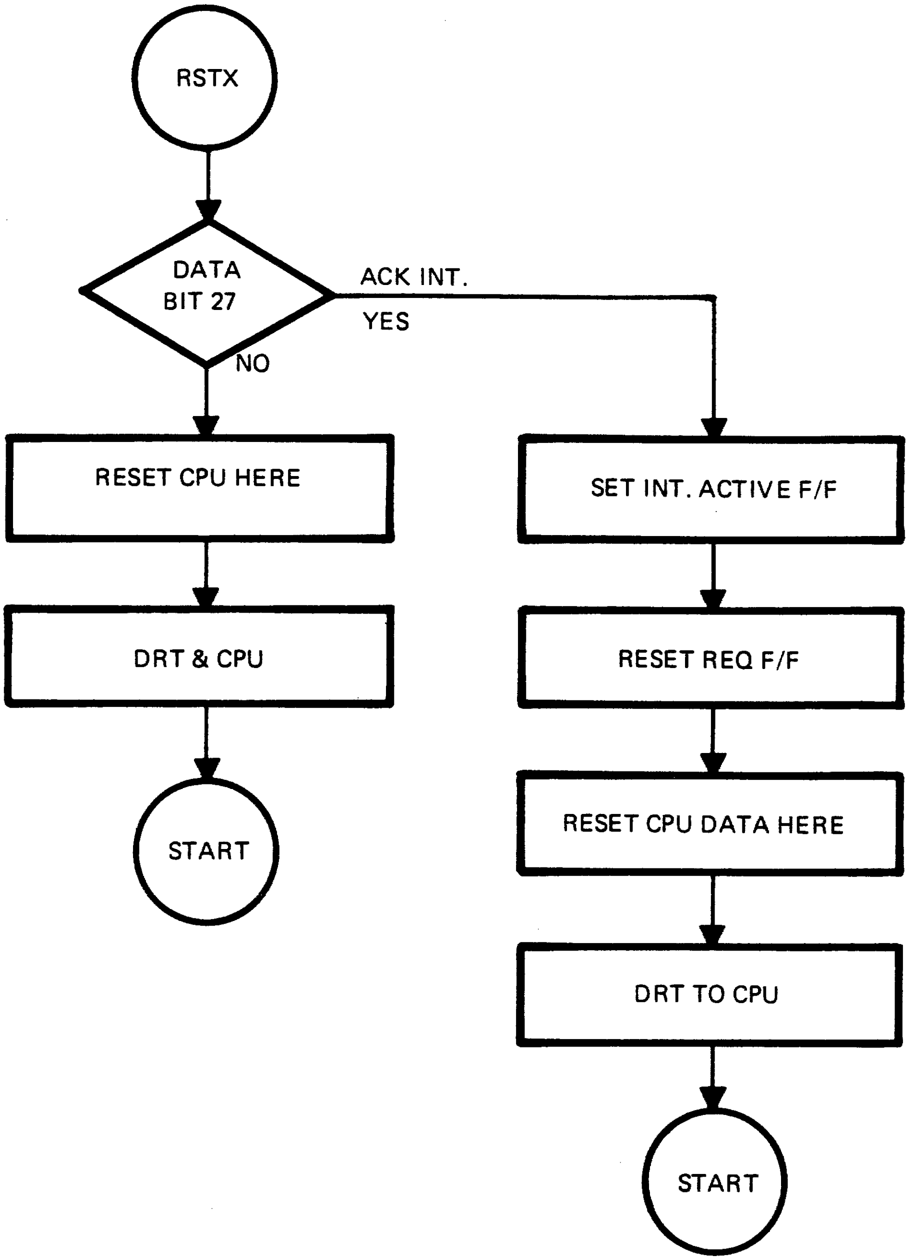


Figure 3-16. Flowchart - Request Status Transfer (CPU)

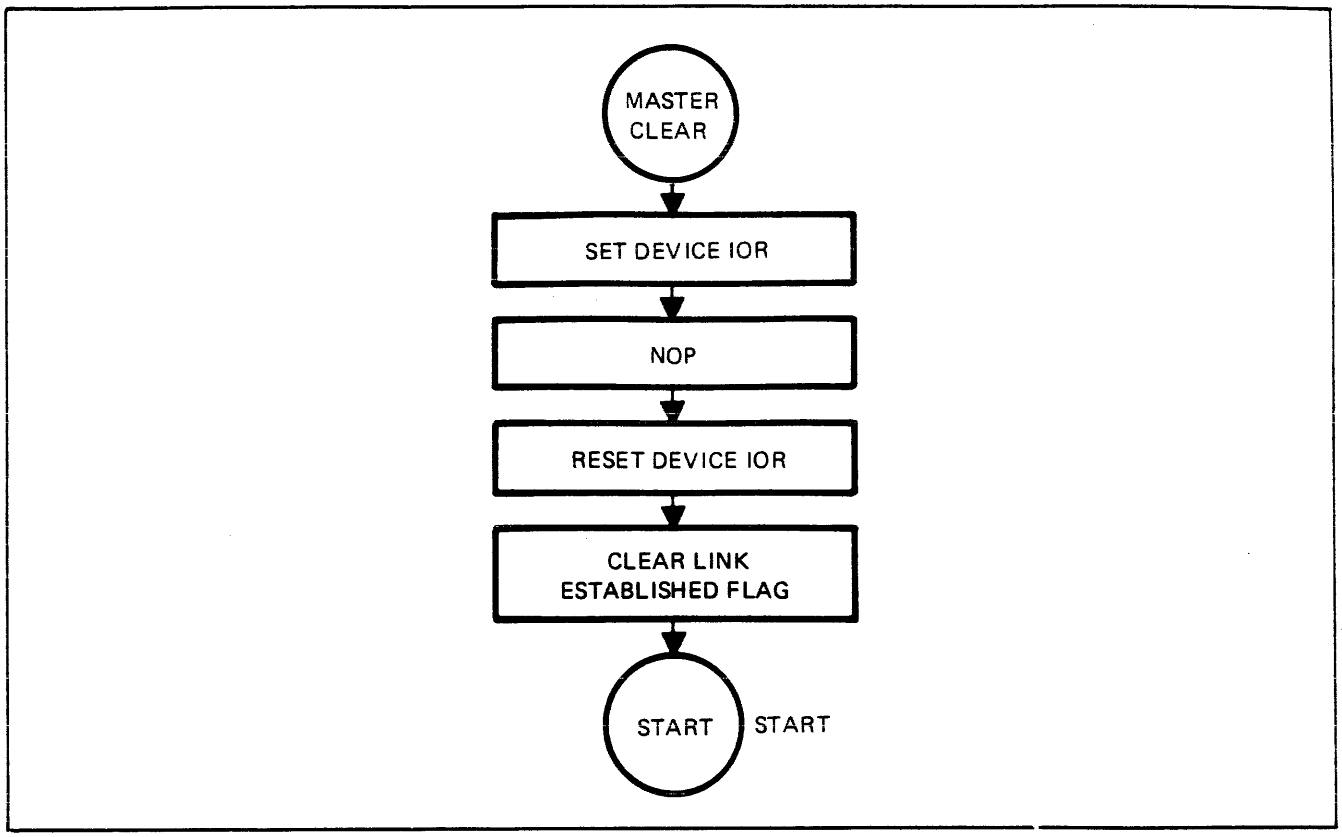


Figure 3-17. Flowchart - Master Clear (CPU)

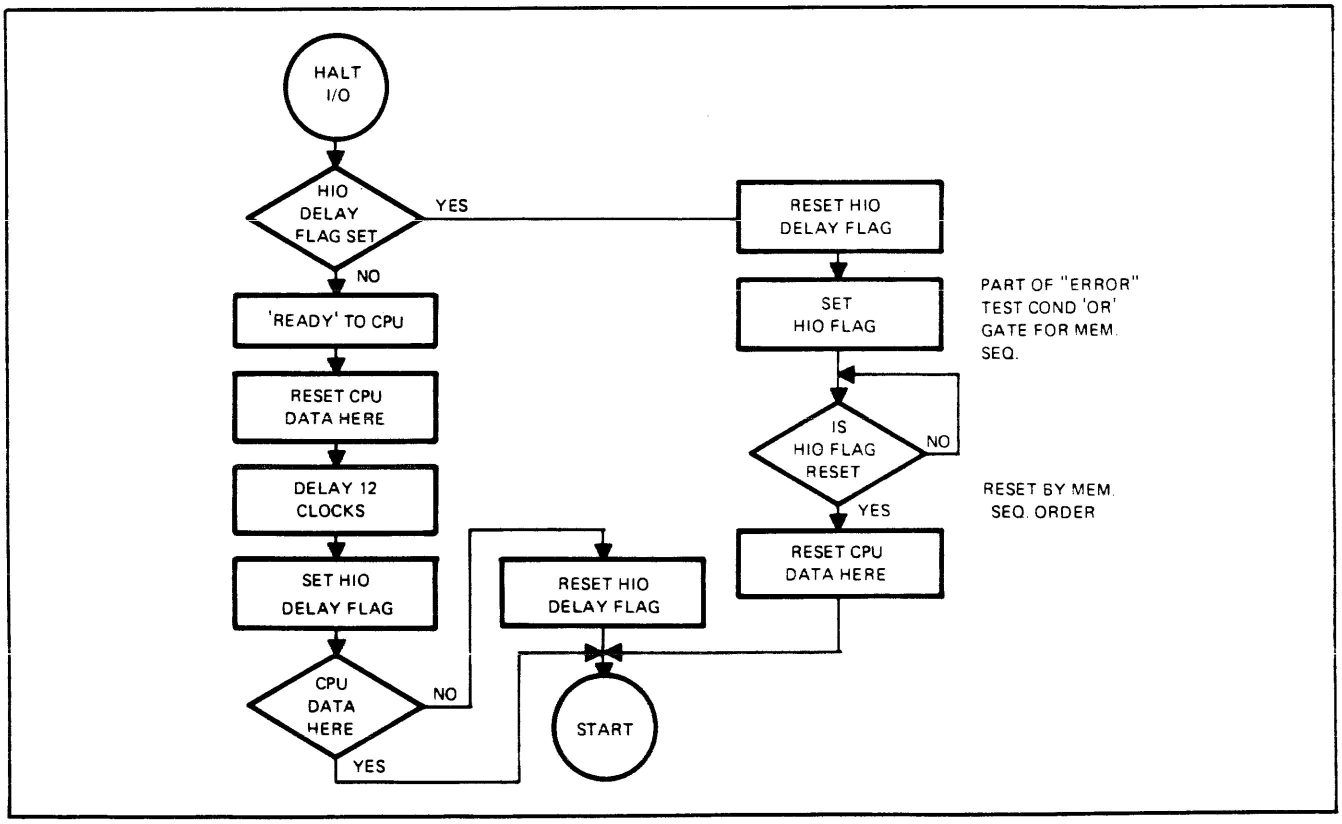


Figure 3-18. Flowchart - Halt I/O (CPU)

control of the HSD by commanding various portions of functional logic in the memory interface control sequencer. The functional logic of the memory interface control sequencer then generates the control signals to the SelBUS interface or the customer device interface as commanded by the firmware. The firmware primary control can be broken into several areas of control which are the basis of organization of the firmware microprogram.

3.3.4.1 Microinstruction Format

The microinstruction is in a 24-bit word stored in a read-only memory called the control memory. When a microinstruction is accessed by the program counter, it is read from the CROM into the control register. The content of CREG is used to control the basic operation of the microinstruction in the memory interface control sequencer. Figure 3-19 illustrates the basic formats for the memory interface control sequencer microword.

3.3.4.2 Execute Gating and Control Logic

The memory interface control sequencer execute gating and control logic (sheet 16) generates most of the gating and control signals required throughout the various logic of the HSD data structure. Sheet 2 is the HSD data structure block diagram which depicts many of the execute gating and control signals. Table 3-8 lists the memory interface control execute instruction enable signals and the memory interface control sequencer control register (HMCSCREG) bit number and defines each of the signals.

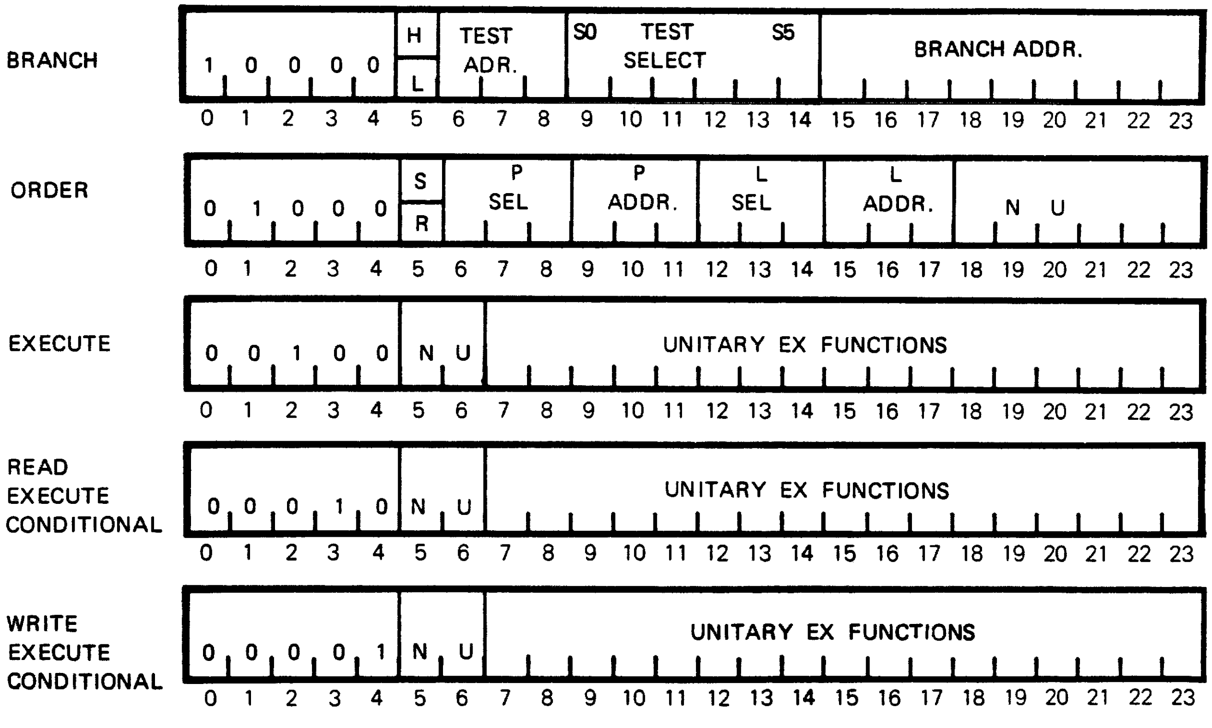


Figure 3-19. Memory Interface Control Sequencer Microword Formats

Table 3-8
Memory Interface Control Sequencer EXECUTE Instruction Enable Signals

HMCSCREG Bit No.	Signal Name	Description
07	HDMUXSELB	D MUX select
08	HDMUXSELA	D MUX select
09	HMEMREAD LMEMREAD	D MUX select (Address Switches) Drives SelBUS line LRD
10	LLOADDATAREG	Loads memory interface data register
11	HLOADDESTREG	Loads memory interface dest. register
12	HMSEQWRITEF LMSEQWRITEF	F MUX select D register Inhibit customer FIFO write
13	LMSEQREADF	Inhibit customer FIFO write
14	LENAMUXOUT HENAMUXOUT	3-state enable A MUX to dest. register 3-state enable TI register to dest. register
15	HAMUXSELCAR	Address MUX select (IOCB address = high, MAR = low)
16	HMAMUXSELCIR	Memory address MUX select (CI register - high, D register = low)
17	LLOADMAR	Load memory address register
18	LINCMAR	Increment memory address register
19	LLDTCANDOPR2	Load op code register 2 and load transfer counter
20	HLOADOPR1	Load op code register 1
21	HDECSFRCNT	Decrement transfer counter
22	LLOADTIADR	Load TI dedicated location register
23	LLOADCAR	Load IOCB address register

3.3.4.3 Program Counter

The program counter (PC) is a nine-bit counter that contains the current read-only memory address (microinstruction address). The program counter (sheets 12 and 13) is incremented consecutively for sequential microinstruction execution or loaded in parallel with a new value for branch microinstructions. When the program counter is loaded in parallel, it is loaded from a decode of the branch microinstruction with a value from the control register (CREG). The results of the test specified by the branch microinstruction must be true before the branch is executed (parallel loaded).

3.3.4.4 Control Memory

The control memory (CROM) (sheets 12 and 13) is a 24-bit wide read-only memory (ROM) which contains up to 512 locations. The control memory addressing is provided by the program counter for the firmware microprogram instruction location.

3.3.4.5 Control Register

The control register (CREG) (sheets 12 and 13) is the data or instruction register for the control memory. The primary functions of any microinstruction, except test information for branch instructions, are decoded from the outputs of the control register.

3.3.4.6 Order Structure Logic

The order structure logic (sheet 14) generates either pulsed or level order signals (refer to Table 3-9) by a decode of the microinstruction. The pulsed order signals are all negative-going pulses. The level order signals may be either logic high or low level signals. The order signals generated by the order structure are used as logic control signals. One microinstruction can generate up to four order signals, including a maximum of two pulsed signals and/or two level signals. The level signals must have the same polarity.

3.3.4.7 Test Structure Logic

The test structure logic (sheet 15) selects an input signal, tests the signal for either a high or low level (refer to Table 3-10), and generates either a test true or a test false condition. In a test true condition, a branch instruction causes a branch in the microprogram execution; in a test false condition, the branch is not taken, and the next sequential microinstruction is executed.

The input signal to the test structure that is tested and the polarity for which the selected signal is tested are determined by decoding the branch microinstruction. One microinstruction may specify up to six input signals. However, the polarity for which all are tested must be the same. When multiple signals are tested by one microinstruction, if any signal is in the state specified by the test, a test true condition results. If all signals are the opposite of the state specified by the test, a false condition results.

3.3.4.8 Memory Interface Control Sequencer Flowcharts

Table 3-11 calls out the figure number and the title of the flowcharts applicable to the memory interface control sequencer.

Table 3-9
Memory Interface Control Sequencer Pulse and Level Orders

HMSCREG09-11 HMSCREG15-17 P/L Adr.	P0 - HMSCREG06	P1 - HMSCREG07	L0 - HMSCREG12	L1 - HMSCREG13
0 0 0	LRESFIFOADR	LRESCUSIFC	HSETOUTENB	HSETINENB
0 0 1	LMSSETHSDBSY	LINCCAR	LERROR	LEOL
0 1 0	LRESSIO	LSENDEF	HIOINPROGRES	LEOB
0 1 1	LSISETREQ	LWRITESTATUS	HTERMDEV	HDEVIOR
1 0 0	LRESHIOFLAG	LRESHSDBUSY	LNPMSTATUS	LIOCBADRERR
1 0 1	LRESSTATUS	LCORRECTTC	HRESSTATL	HRESNPMERR
1 1 0			HLINKREQOUT	HLINKACKIN
1 1 1			LINCXFRCNT	LLINKESTBFLG

Table 3-10
Memory Interface Control Sequencer Test Conditions

HMSCREG 06-08 Test Address	HMSCREG09 S0	HMSCREG10 S1	HMSCREG11 S2	HMSCREG12 S3	HMSCREG13 S4	HMSCREG14 S5
000	GND (Uncond. Branch)	LDATACHAIN (DC)	LCOMNDCHAIN (CC)	LEXTMODE (EM)	LSIO (SIOFLAG)	LXFRINCHNL (TIC)
001	LHIOFLAG +LEXTINTREQ +LDEVINOP	LFIFOOVERFLO (F-OV)	LTCOUNTZERO (TC=0)	LNPMERR (NPM)	LMEMDRT	LFIFOEMPTY (F-EMP)
010	LINTREQENAB	LINTACTIVE (ACT)	LINPUT (IN)	LCOMMANDXFR (CT)	LDEVSTATREQ (DSR)	LOUTACKREC (OA)
011	LHIOFLAG +LEXTINTREQ +LDEVINOP	LPARITYERR (PE)	LNPMERR (NPM)	LTCOUNTZERO (TC=0)	LDEVENDBLK (DEB)	LFIFOEQ15 (F=15)
100	LEXTFUNACK (EFA)	LWRITESUCC (WRTSUC)	LNPMERR (NPM)	LCONTONERR	LFIFOFULL (F-FULL)	LGENINT (1)
101	LXFRINCHNL (TIC)	LSTATUSRDY (ISR)	LIOINPROGRES	LIBLMODE	LEXMODEIN (EM-IN)	LIBLHIGHPRIO
110	LEXTINTREQ (EXT)	LHIOFLAG	LBADOPCODE	LLINKESTBFLG	LFUNC2SEL	
111	LLINKREQIN	LLINKACKOUT				

Table 3-11
Memory Interface Control Sequencer Flowchart Index
(Reference for Figure 3-20)

Subroutine Name	Flowchart Page Location	Subroutine Name	Flowchart Page Location
ACKWAIT	38	NPMTST1	31
ACKCHT	39	NPMTST2	31
CLRFIFO1	42	NPMTST3	31
CLRFIFO0	42	NPMTST4	32
CMDXFR	19	NPMTST5	32
CNTTC	13	NPMTST6	32
COMDCHN	23	OPTEST	5
COMMON	19	OUTCHK	13
COMPRD	11	OUTEMP	12
COMPRD1	12	OUTENBL	10
DATACHN	25	OUTFAST	11
DCFTCH	25	OUTERR	8
DCIN1	26A	OUTFILL	9
DCIN2	26A	OUTSLOW	10
DCTST	27	OUTXFR	9
DEBTST	8	RLERR	8
DOTIC	22	RSTBUSY	39
DVSTAT	21	SETEOB	15
EFAWAIT	10	SETOB1	20
EM1	33	SETERST	17
EMERR	37	SETSTAT	15
EMEXIT	37	SIREQ	29
EMIN	35	STATPOST	22
ERRSTAT	15	START	1
EXTMOD	33	START	2
GENINT	16	TERMDC	26C
HIO1	29	TERMTST	41
HLTIO	28	TESTFIFO	14A
HLTIO1	28	TIC	18
HLTIO2	30	TICDC	25
HLTIO3	29	TICINT	18
HLTIO4	28	TICINT1	18
HLTIO13	30	TICNORM	18
HLTIO18	31	TICSTWT	16
HLTIO19	30	TSTCC	20
IBL	38	TSTCC1	19
IBLHLTIO	41	TSTDEB	26B
IBLLACK	40	TSTERR	26C
IBLTERM	41	TSTEXIT	8
IBLTERM1	41	TSTEXT	36
INFAST	7	TSTPRIO	39
INLOOP	26B	WRTEM	35
INSLOW	6	WRTSTAT1	18
INTSIST	16	WRTSTAT2	15
IOCBADR	3	WRTSTAT3	16
IOCOMP	14	WRTSTAT4	16
IOCMP1	14	WRTSTAT5	28
LINKACK	40	WRTSTAT6	29
MCLEAR	1	WRTWAIT	7
MODTC	13		

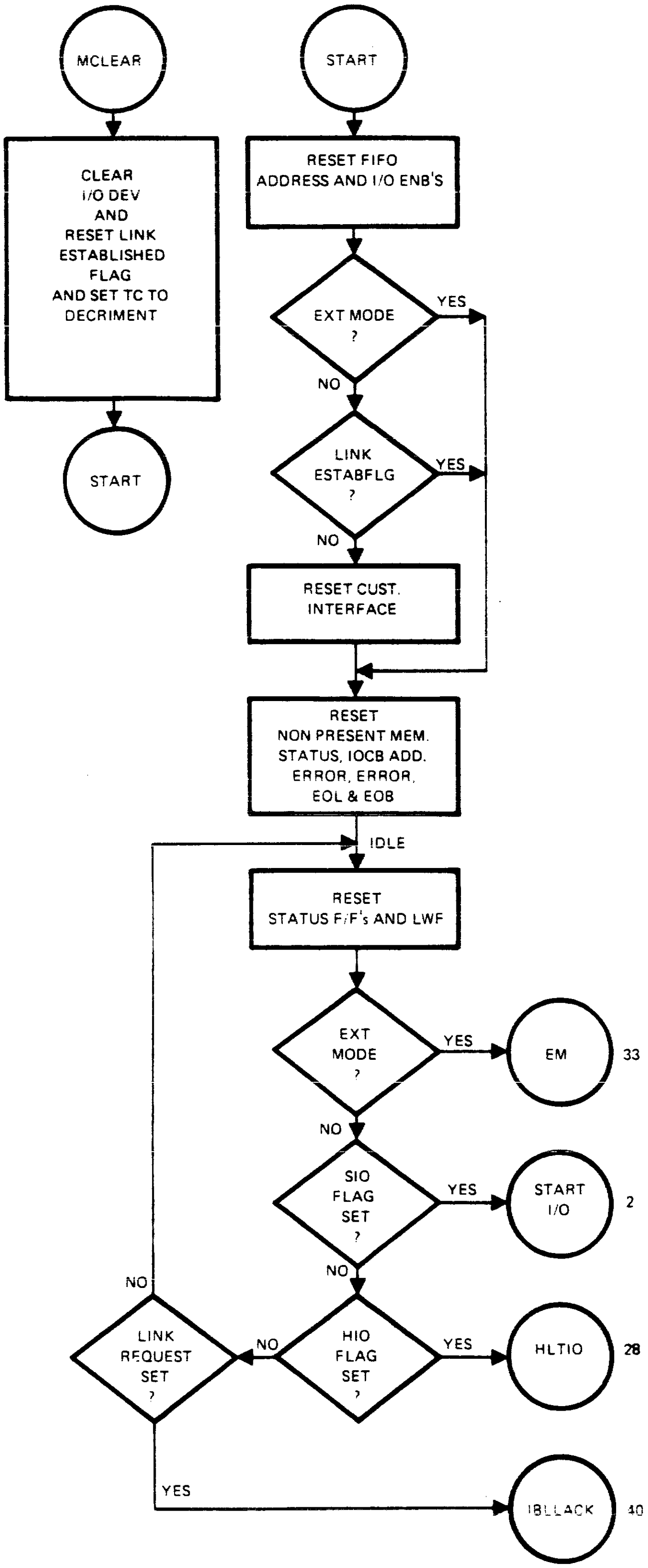

MCLEAR
CLEAR I/O DEV AND RESET LINK ESTABLISHED FLAG AND SET TC TO DECRIMENT
START
START
RESET FIFO ADDRESS AND I/O ENB'S
EXT MODE ?
YES
NO
LINK ESTABFLG ?
YES
NO
RESET CUST. INTERFACE
RESET NON PRESENT MEM. STATUS, IOCB ADD. ERROR, ERROR, EOL & EOB
IDLE
RESET STATUS F/F's AND LWF
EXT MODE ?
YES
EM
33
NO
SIO FLAG SET ?
YES
START I/O
2
NO
HIO FLAG SET ?
YES
HLTIO
28
NO
LINK REQUEST SET ?
NO
YES
IBLLACK
40

Figure 3-20. Flowchart - Memory Interface Control Sequencer
(Sheet 1 of 46)

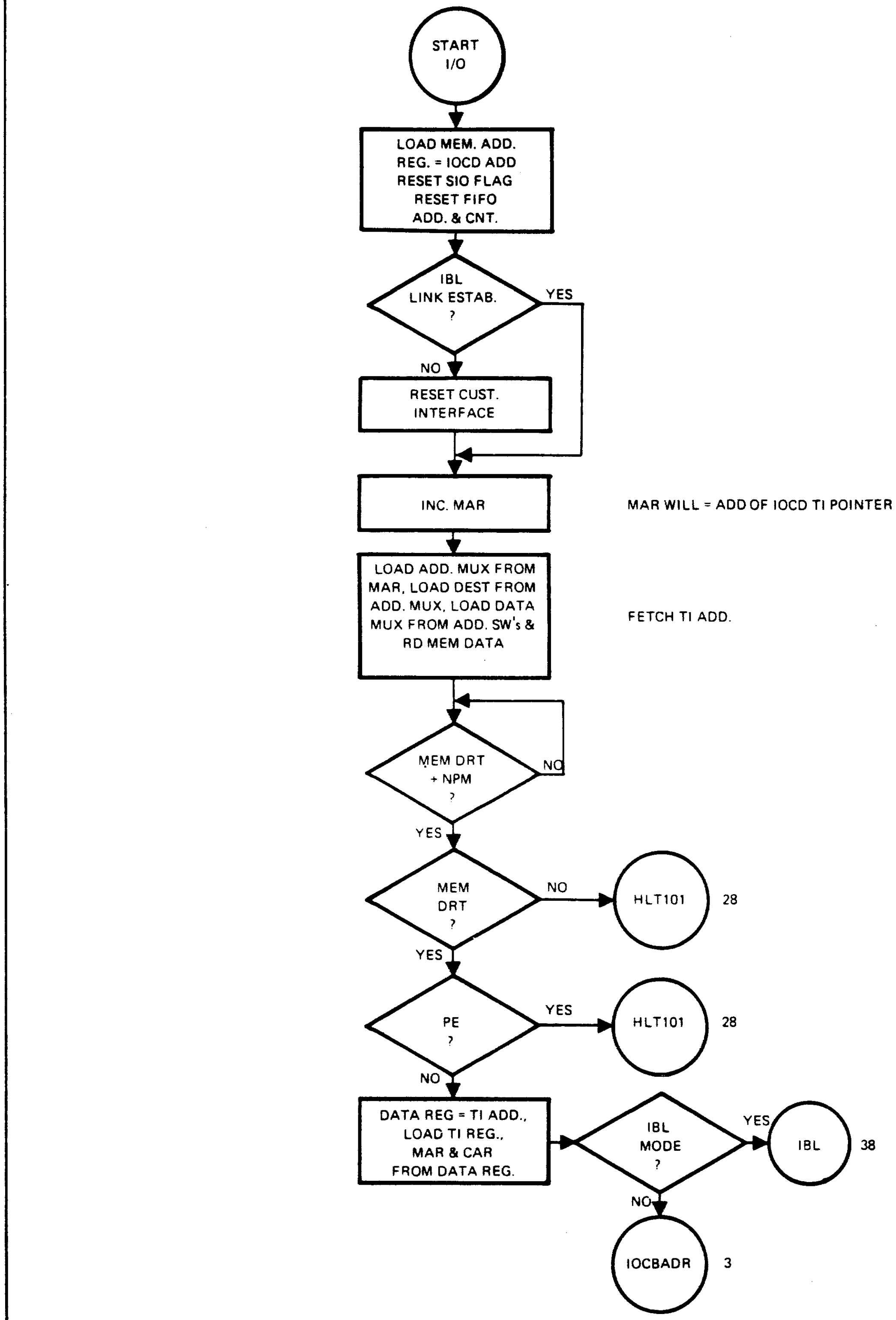


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 2 of 46)

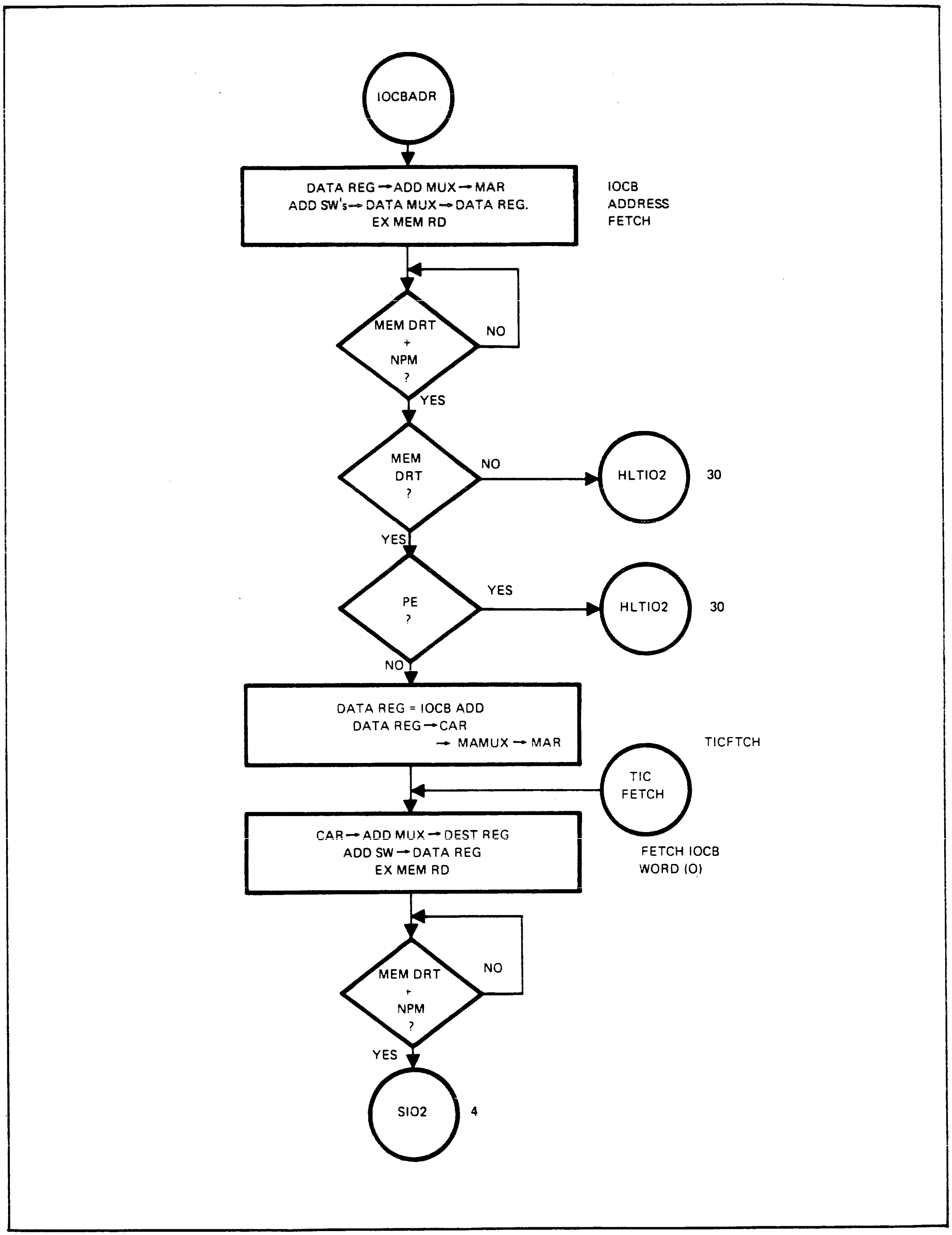


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 3 of 46)

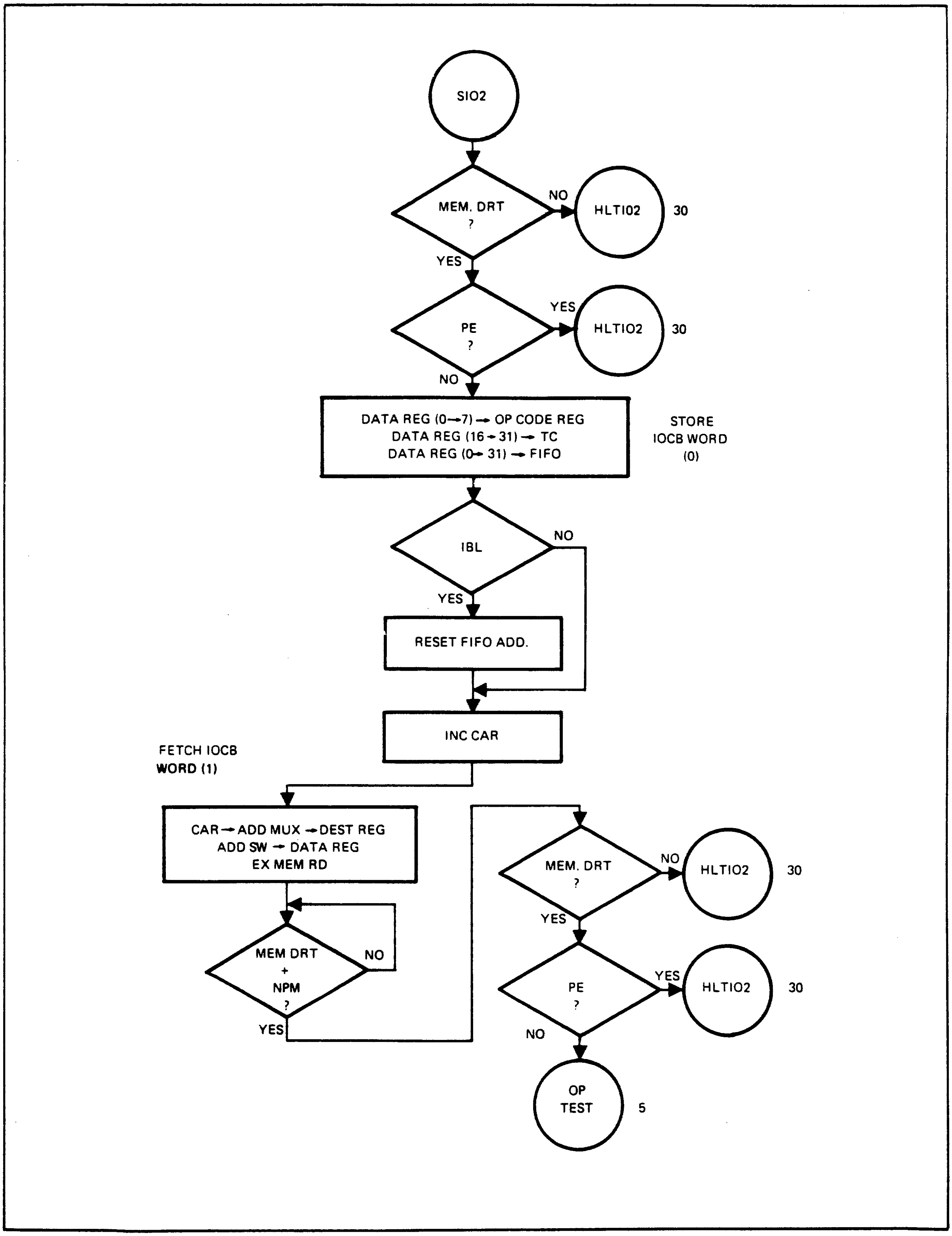


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 4 of 46)

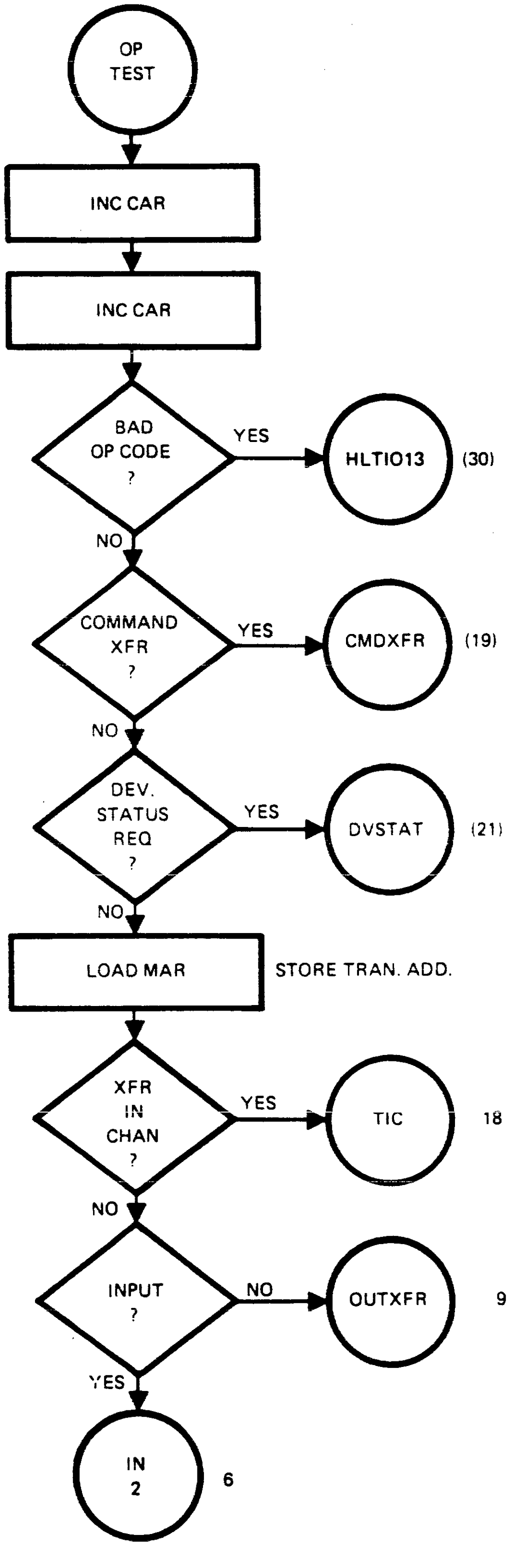


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 5 of 46)

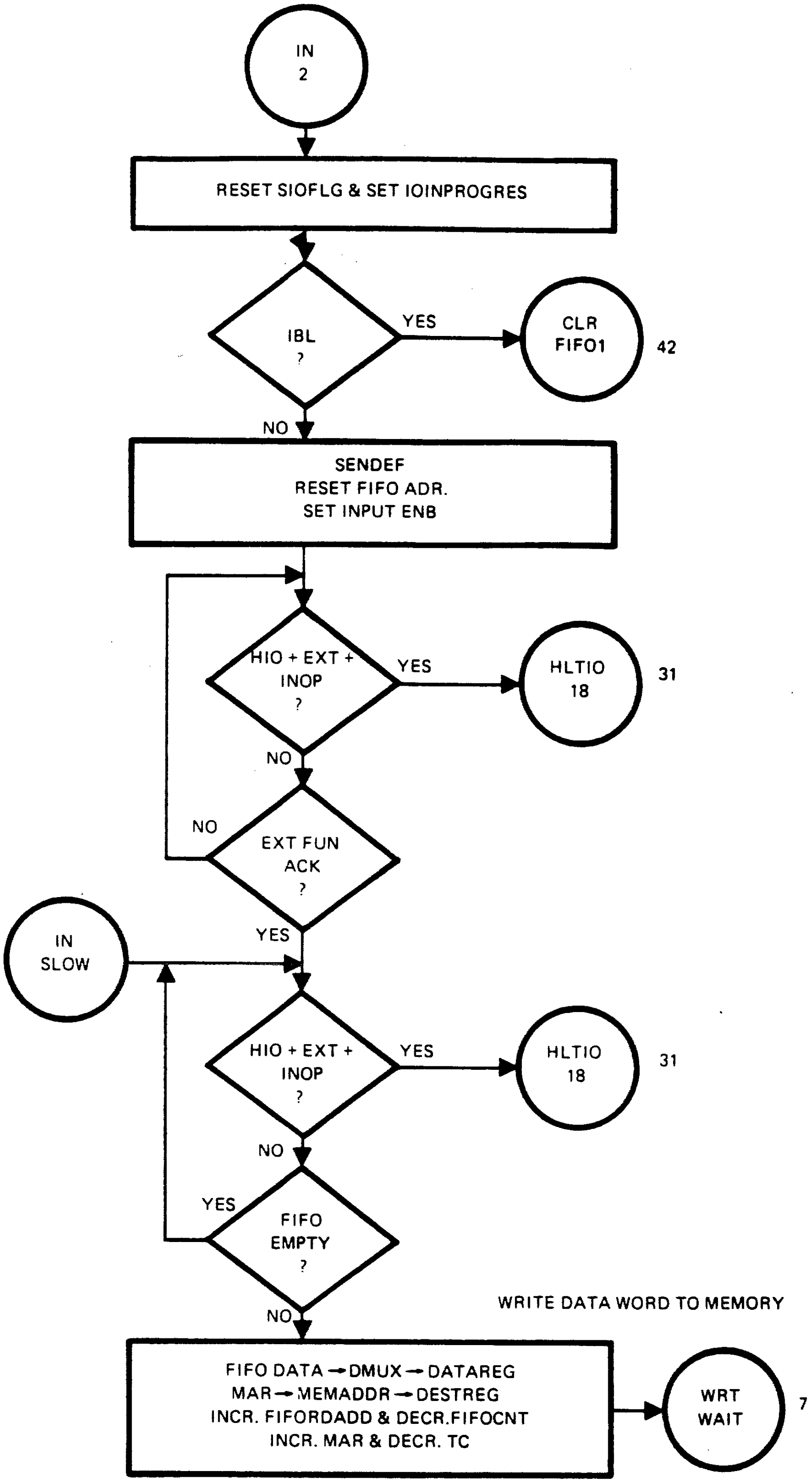


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 6 of 46)

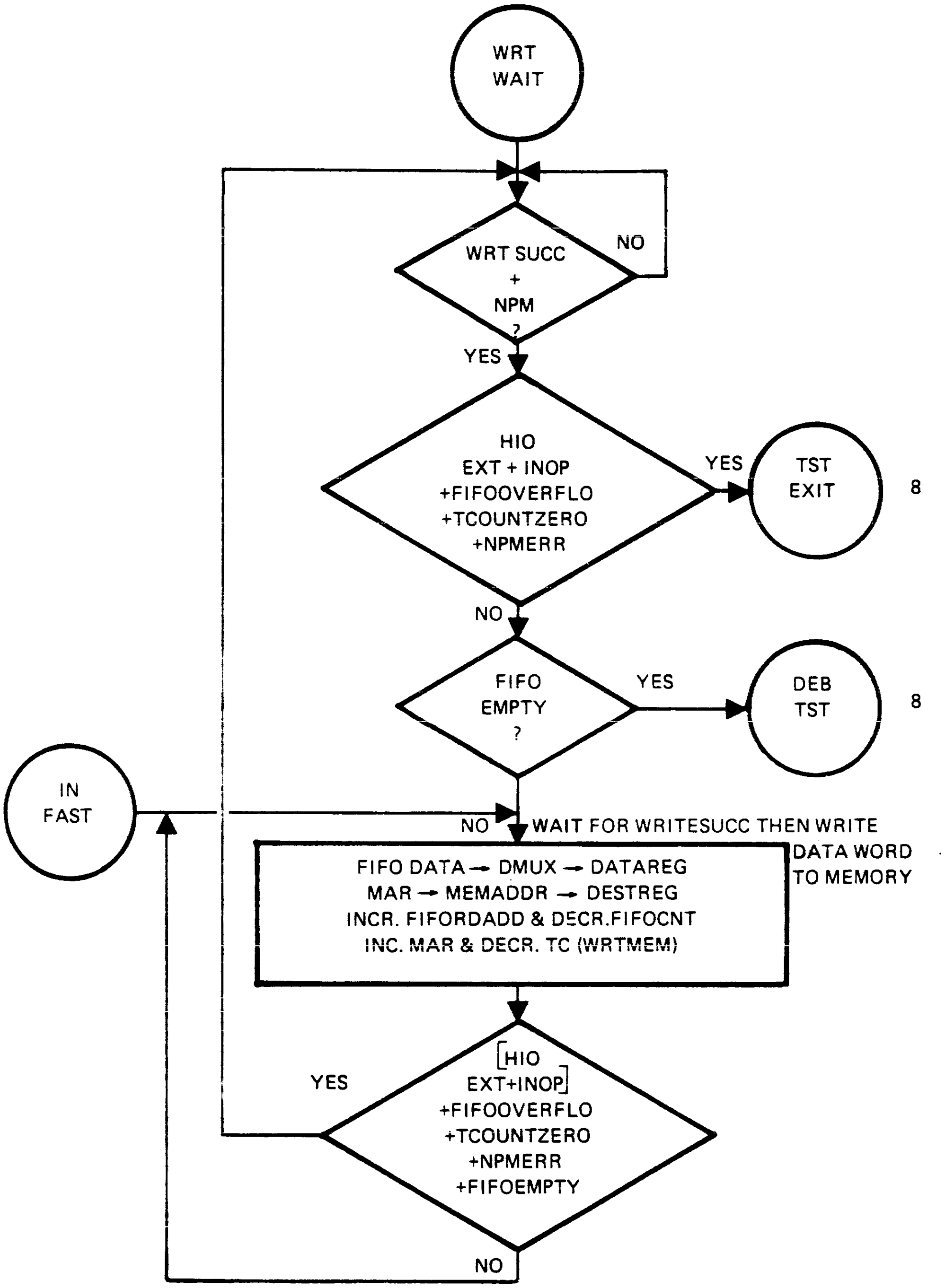


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 7 or 46)

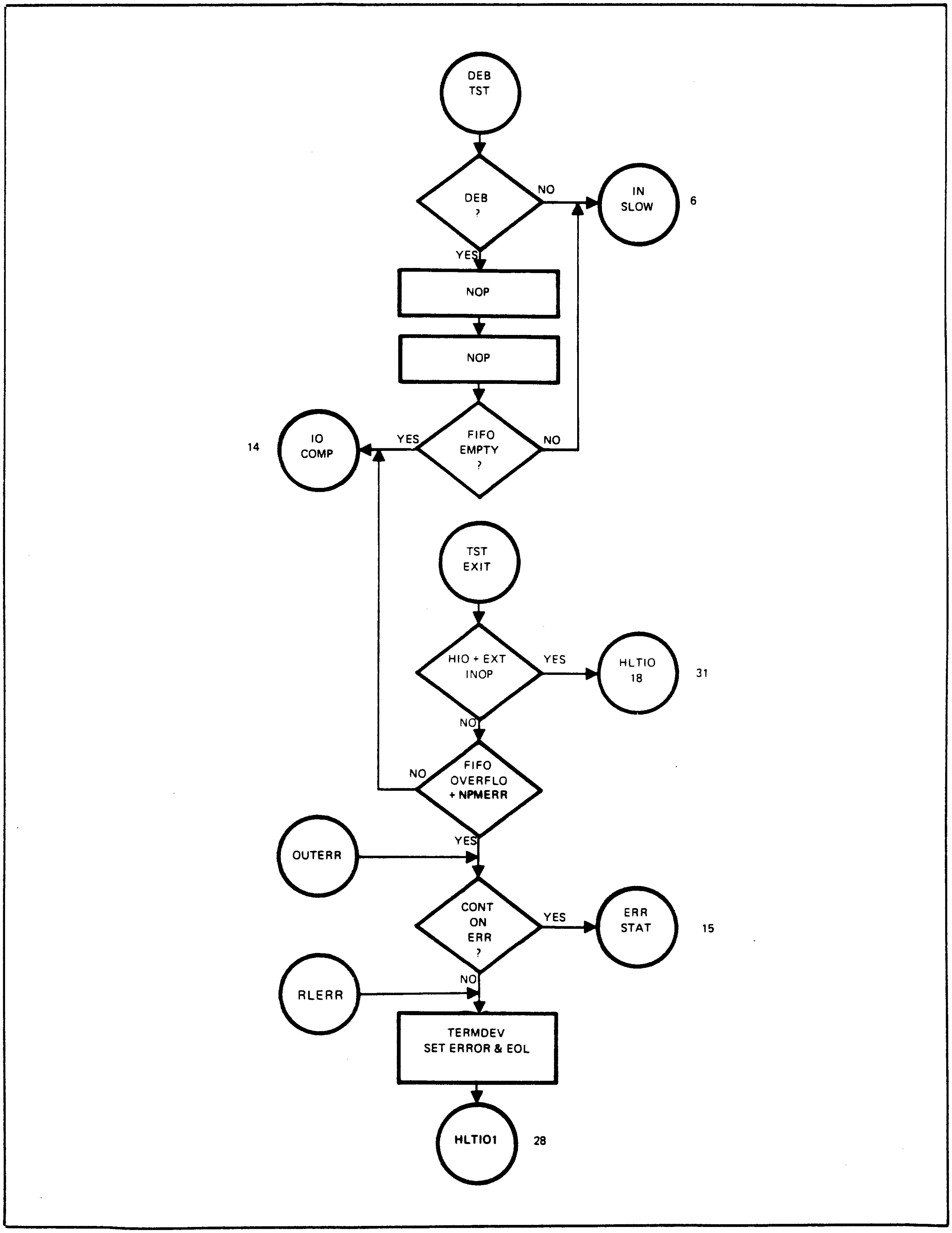


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 8 of 46)

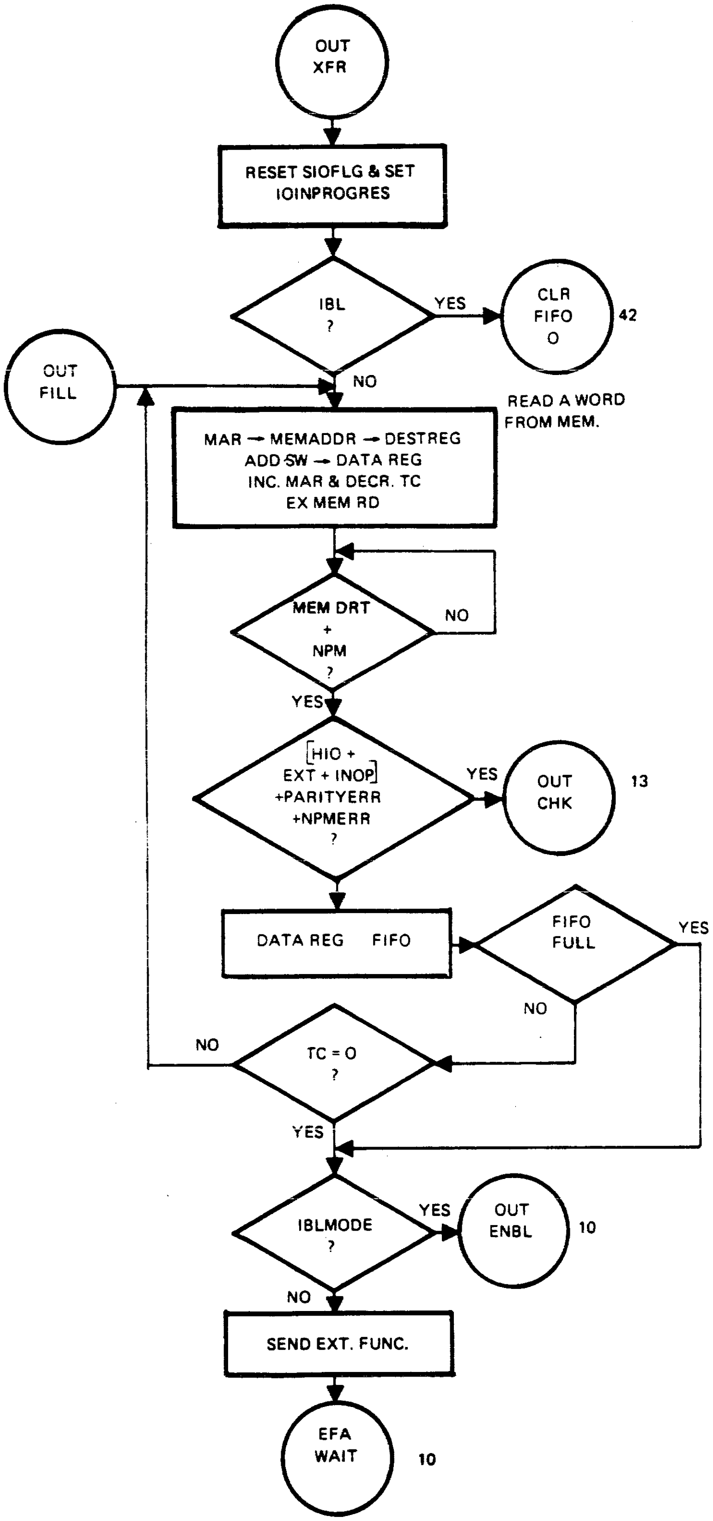


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 9 of 46)

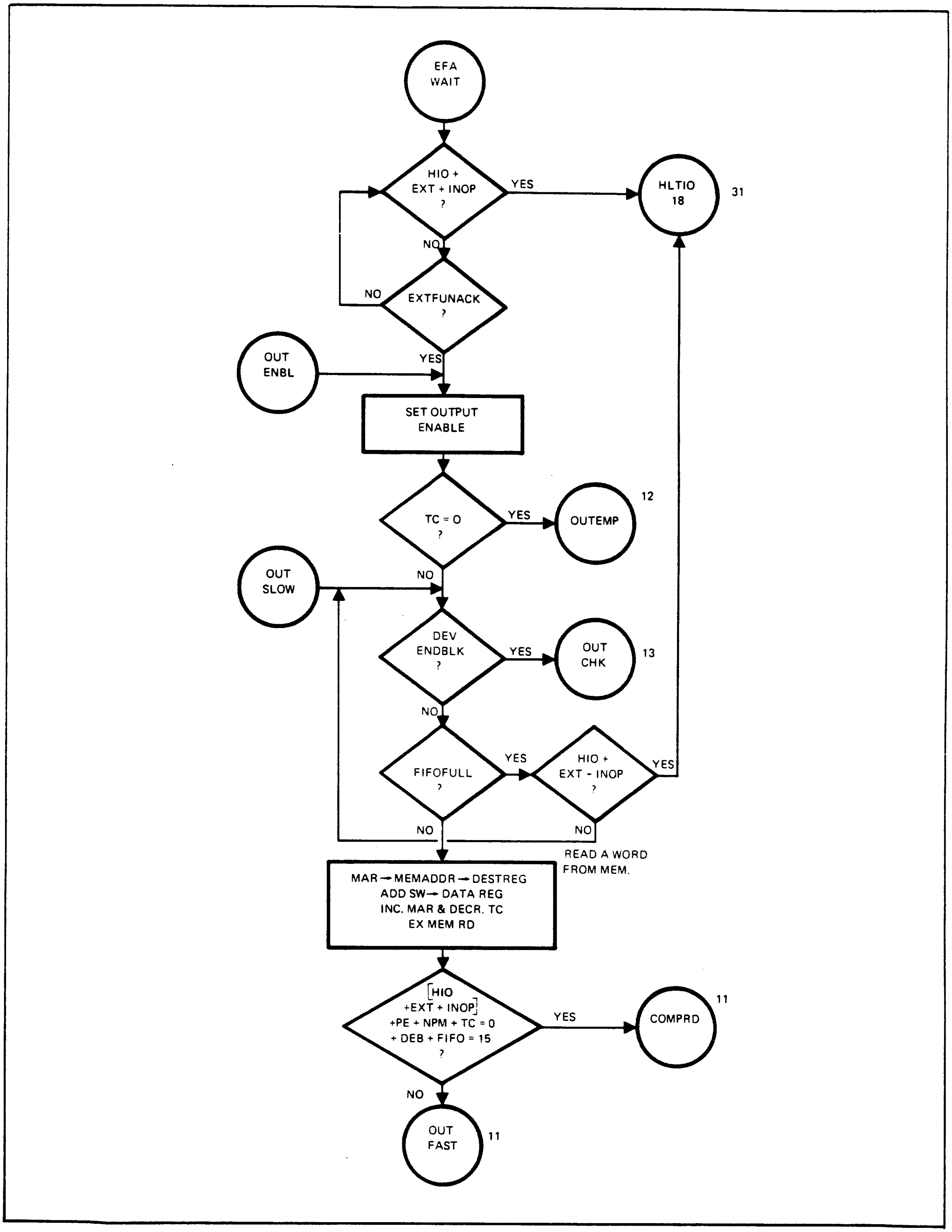


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 10 of 46)

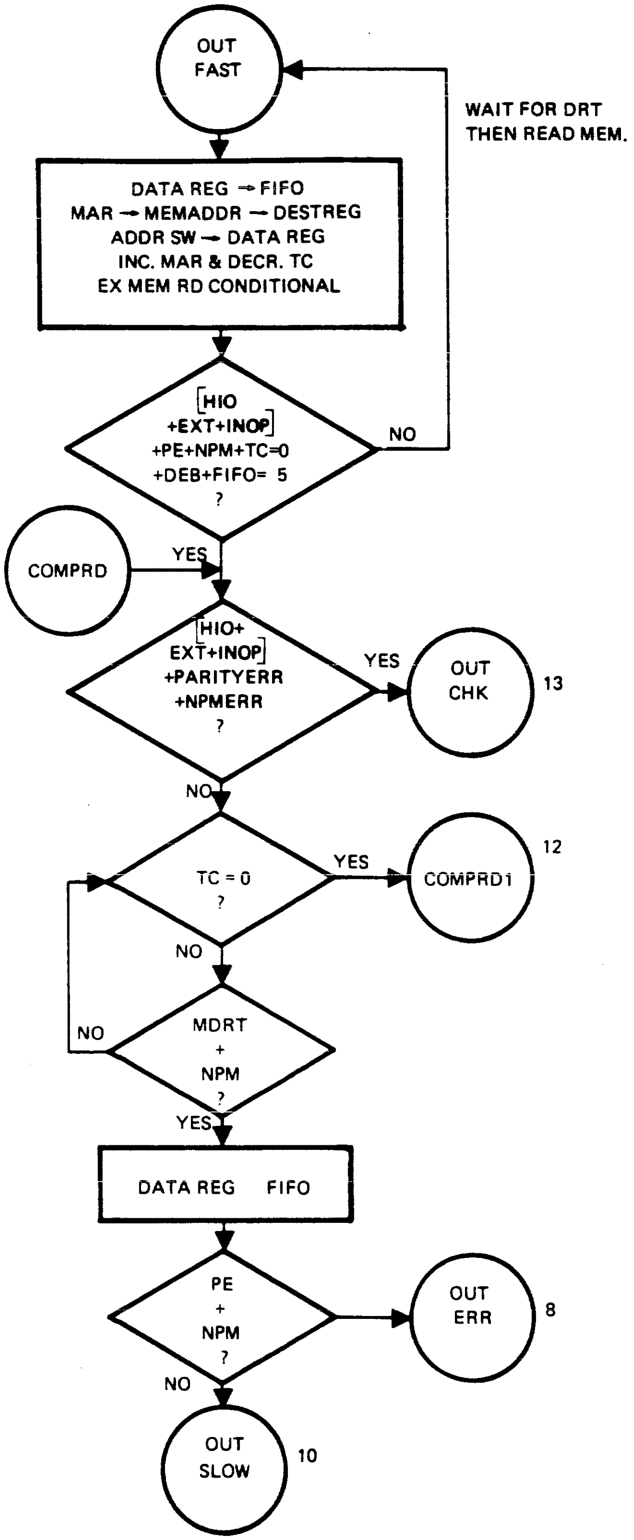


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 11 of 46)

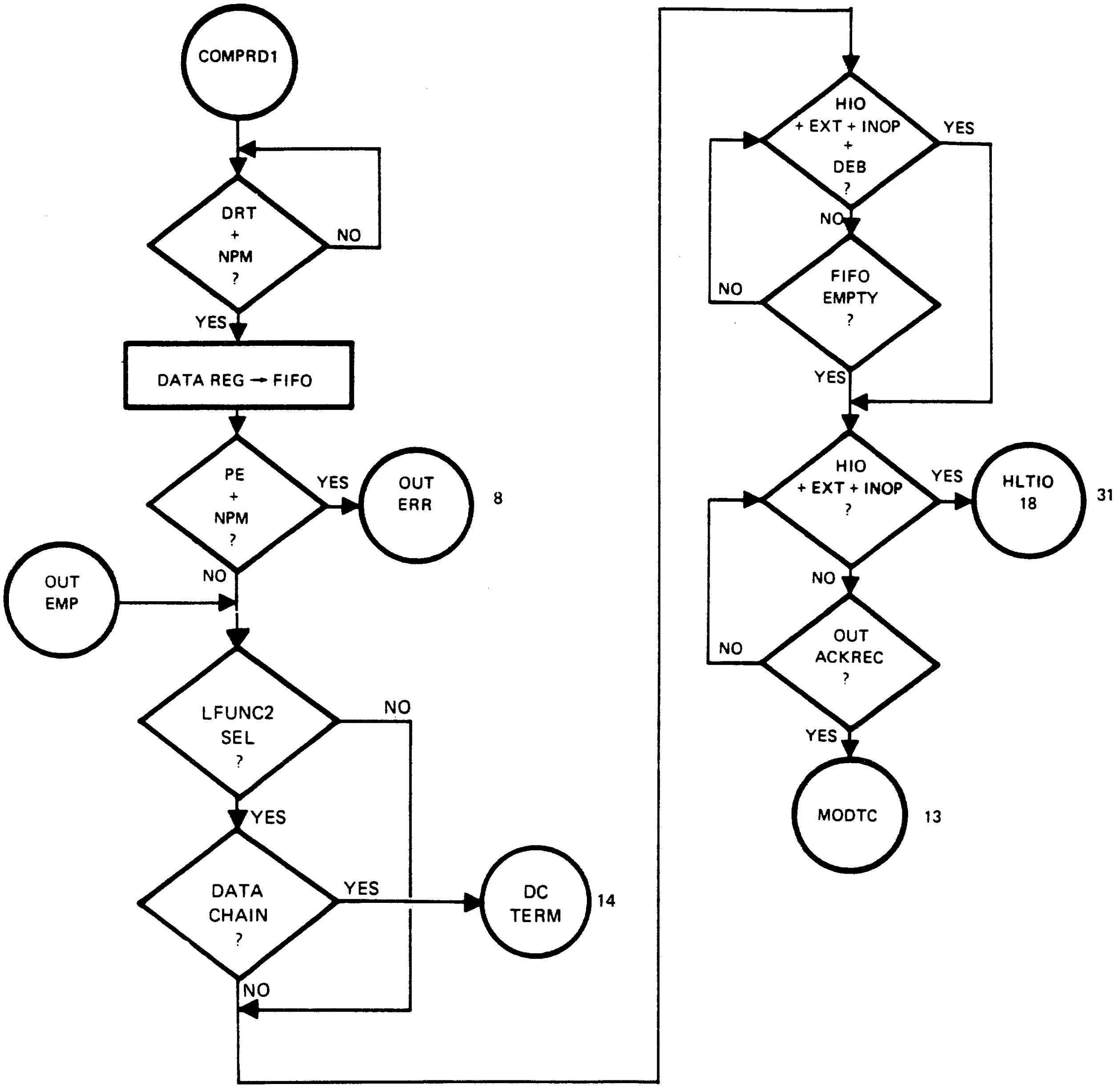


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 12 of 46)

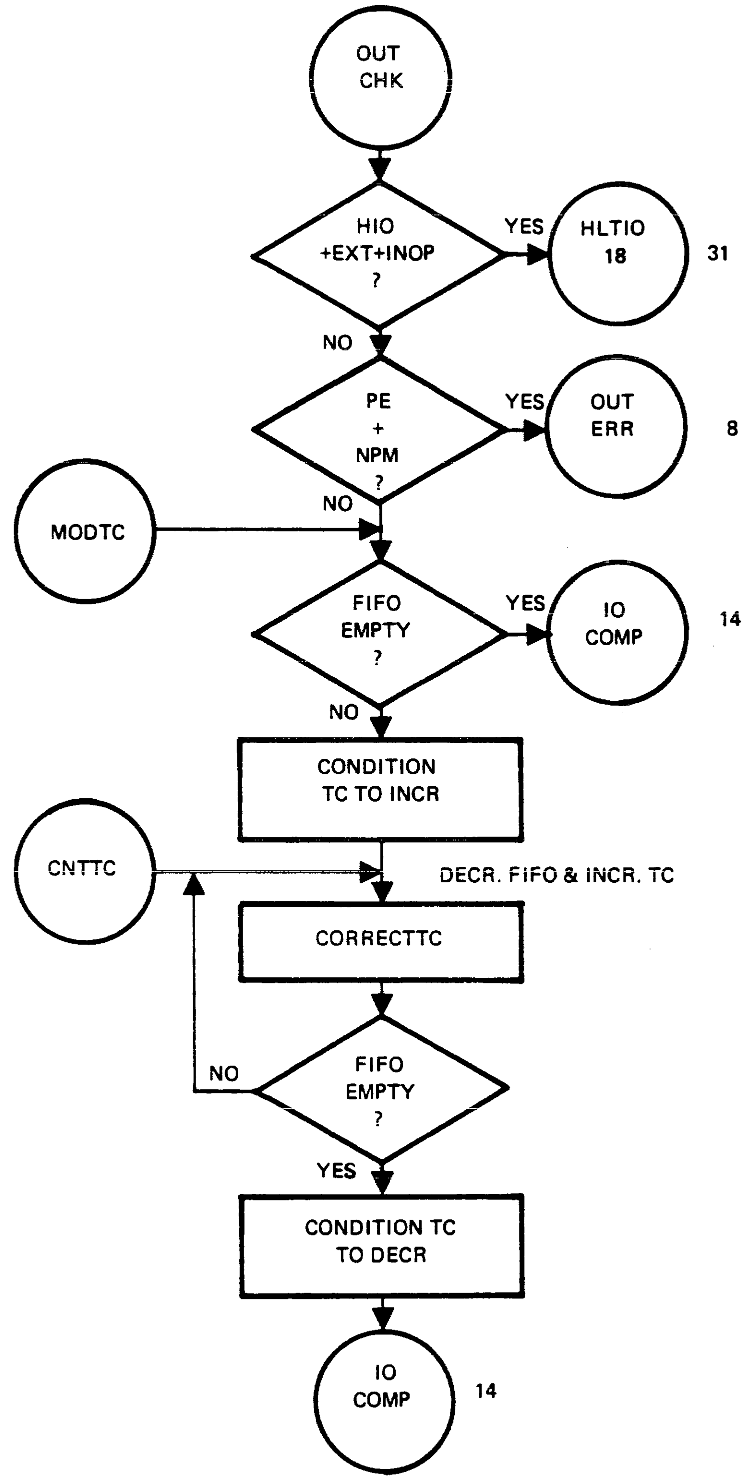


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 13 of 46)

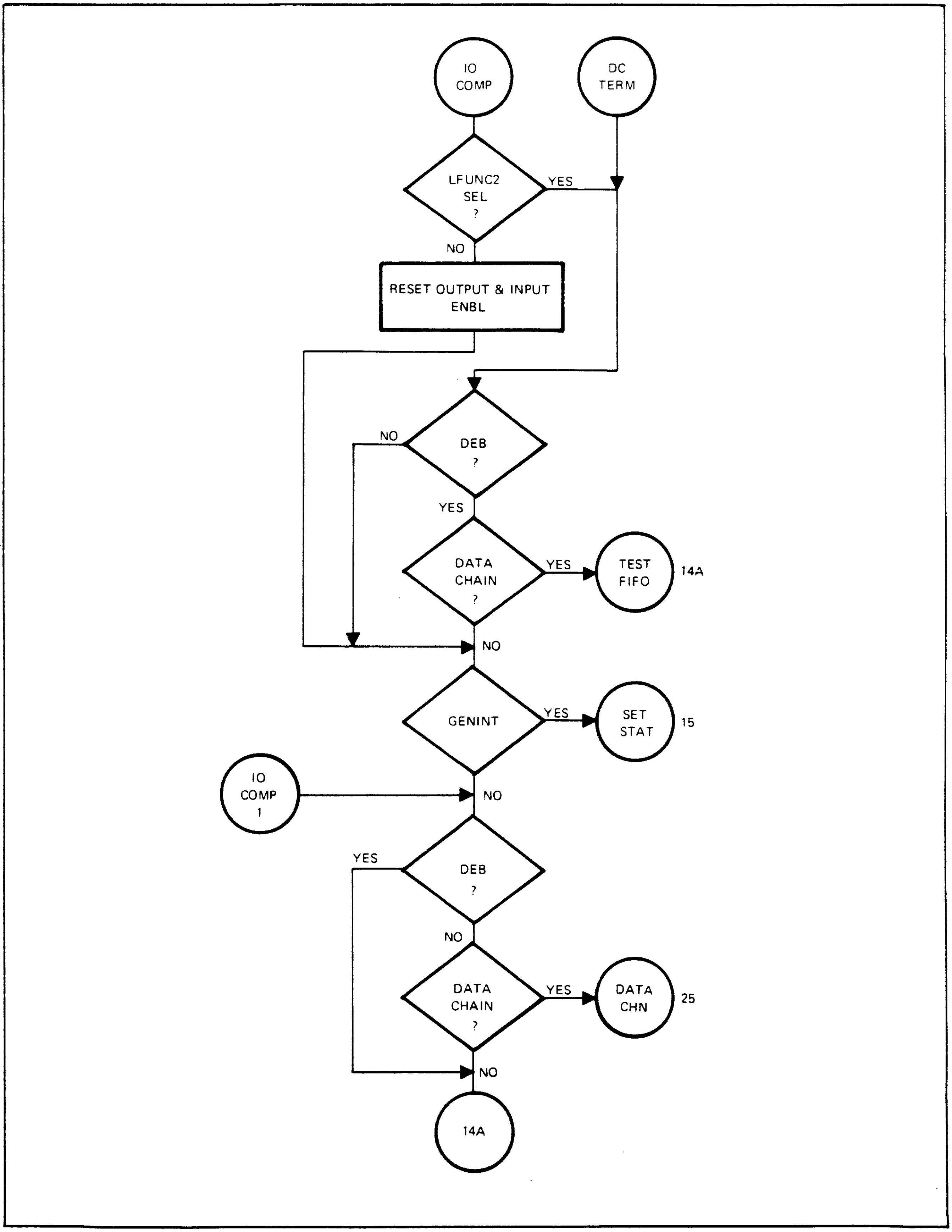


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 14 of 46)

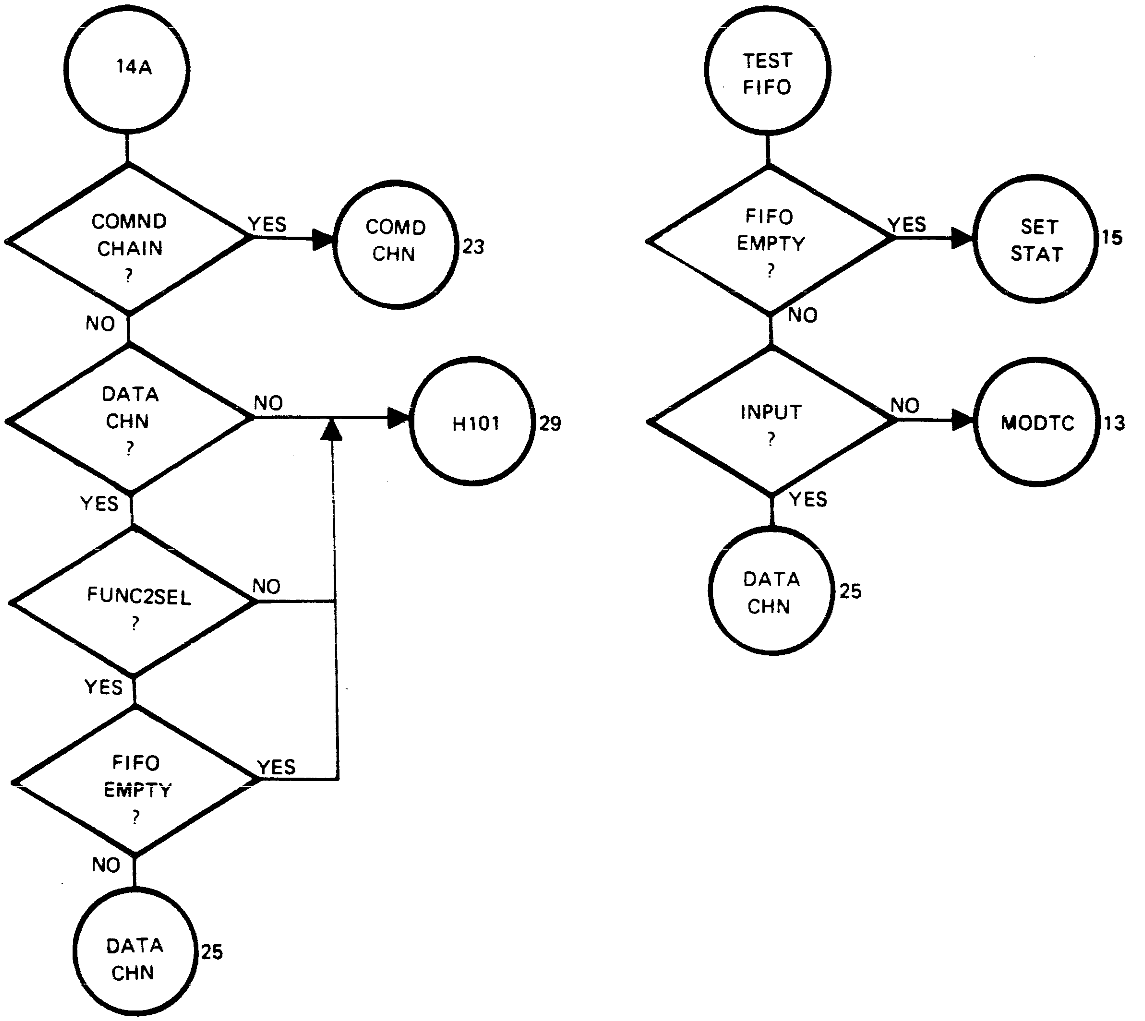


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 15 of 46)

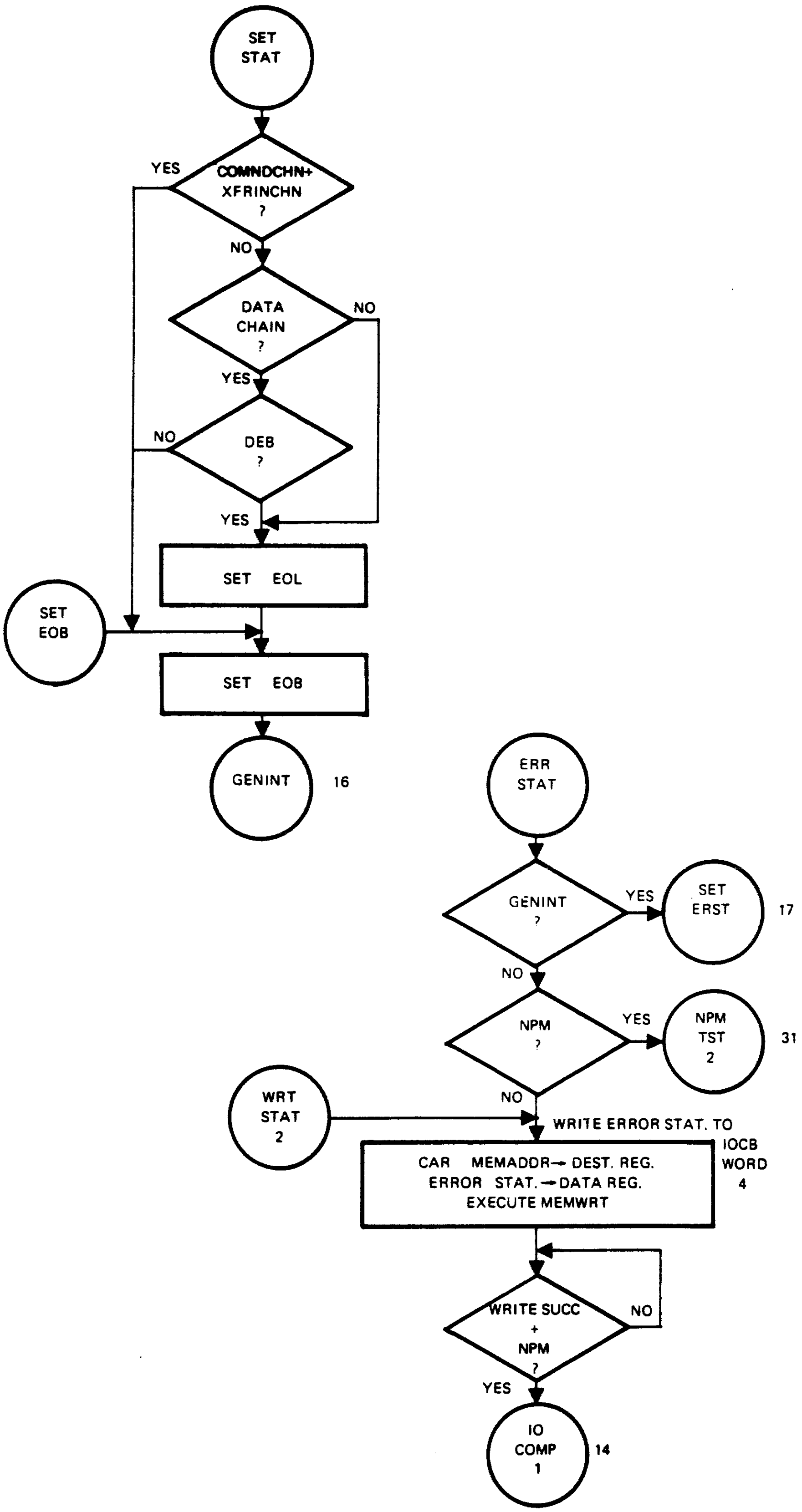


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 16 of 46)

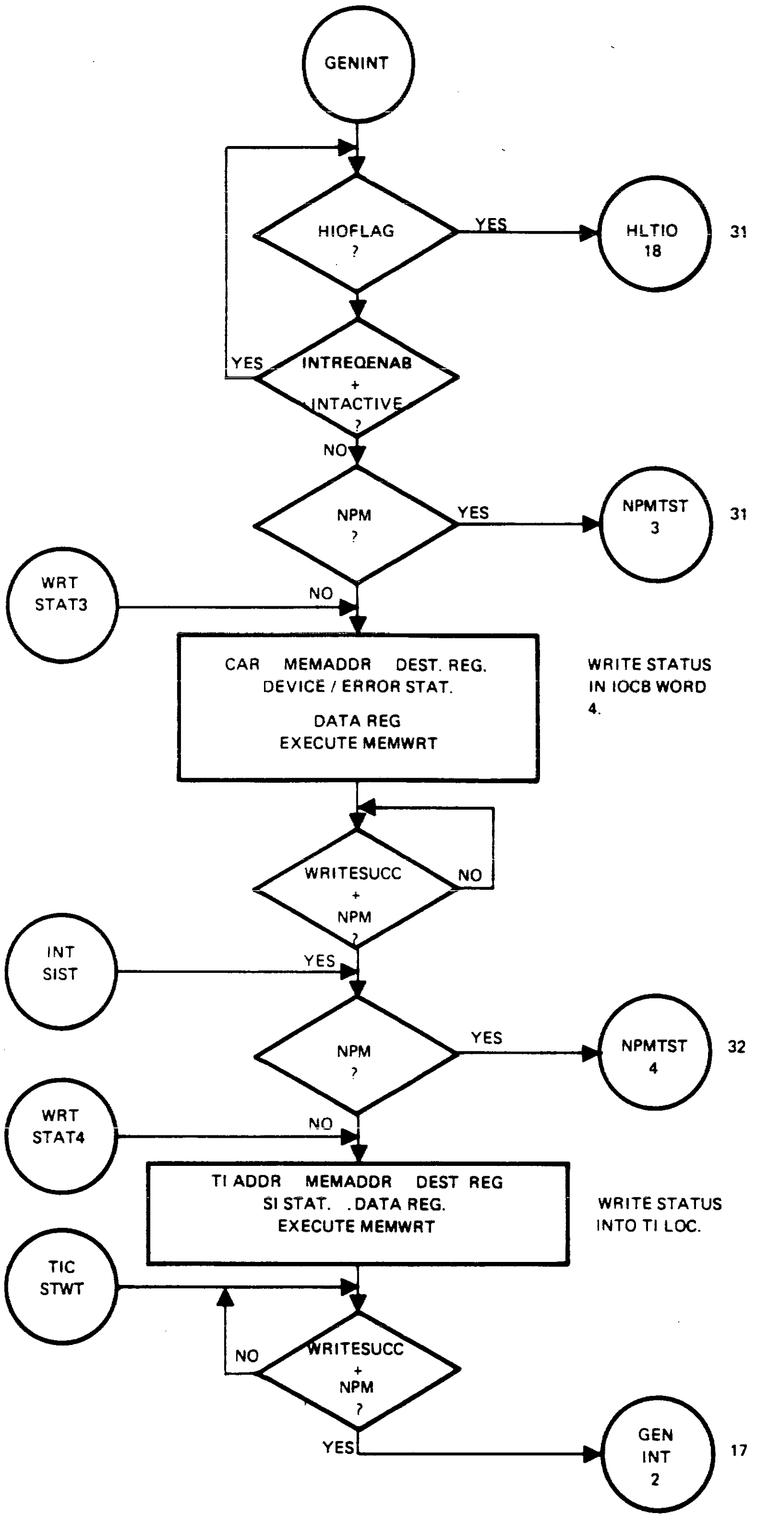


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 17 of 46)

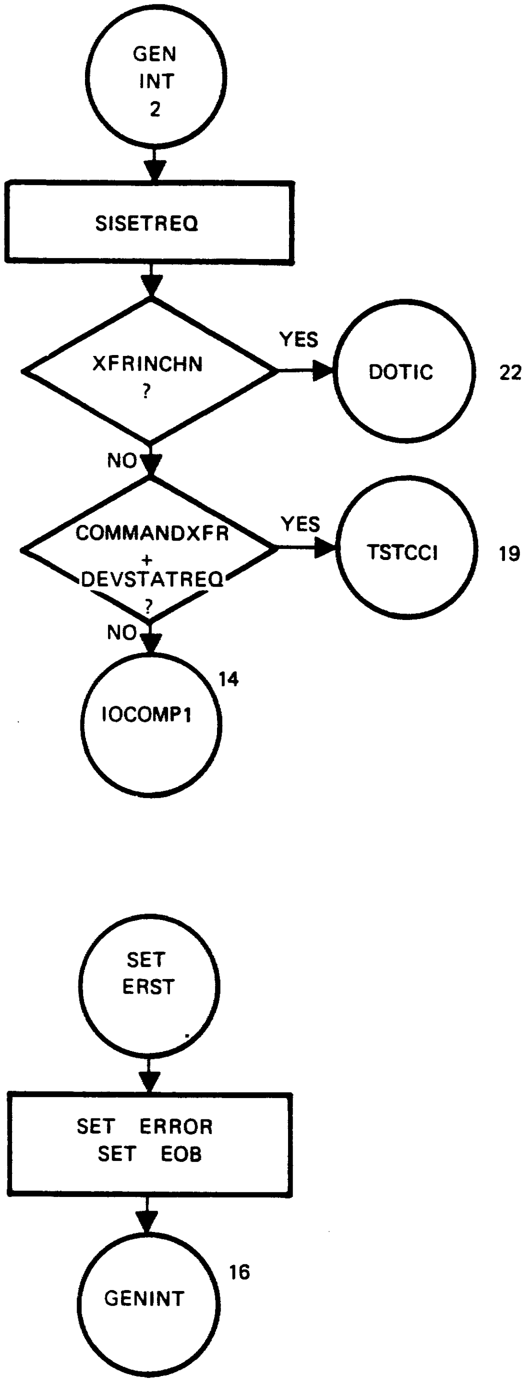


Figure 3-20. Flowchart - Memory Interface Control Sequencer
(Sheet 18 of 46)

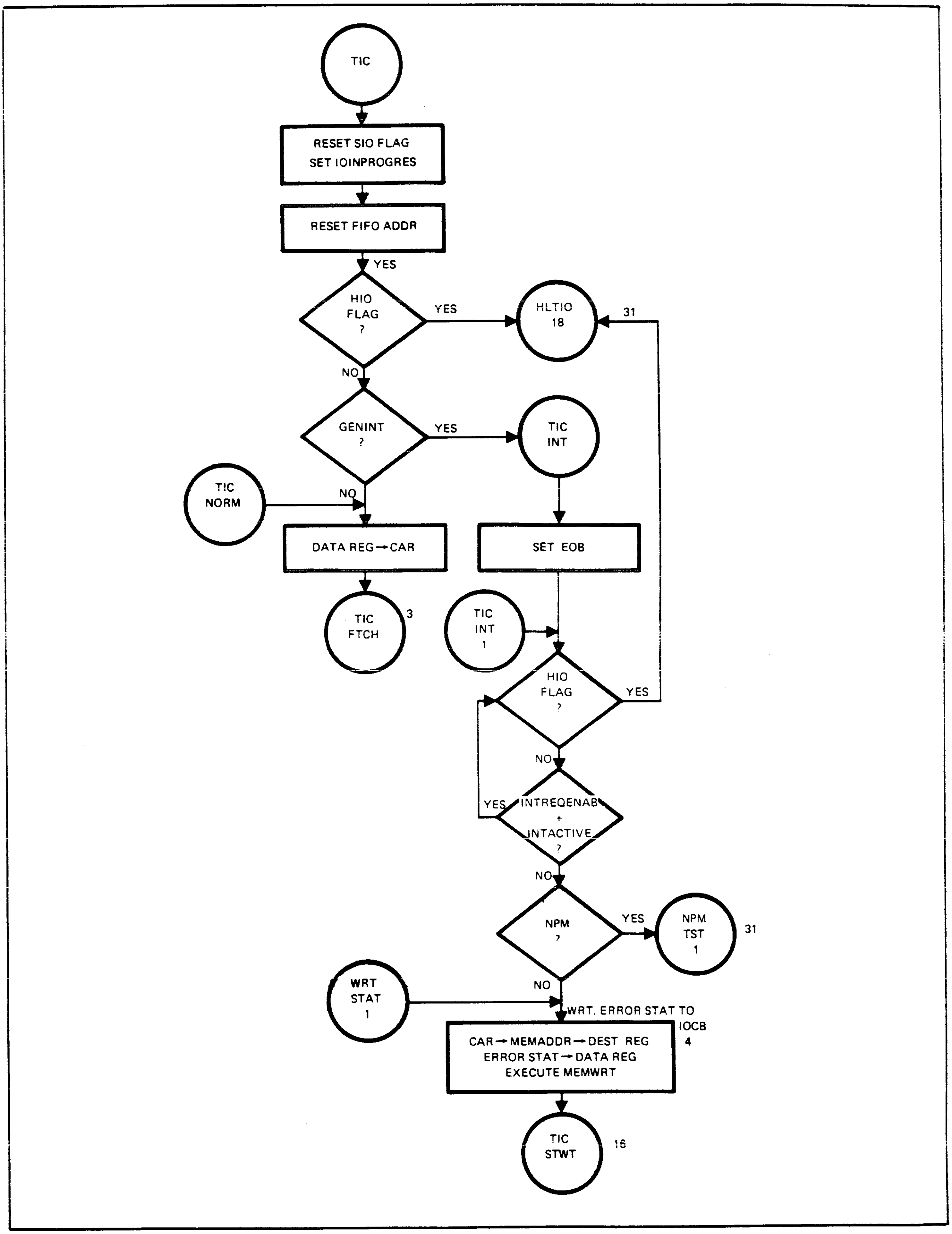


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 19 of 46)

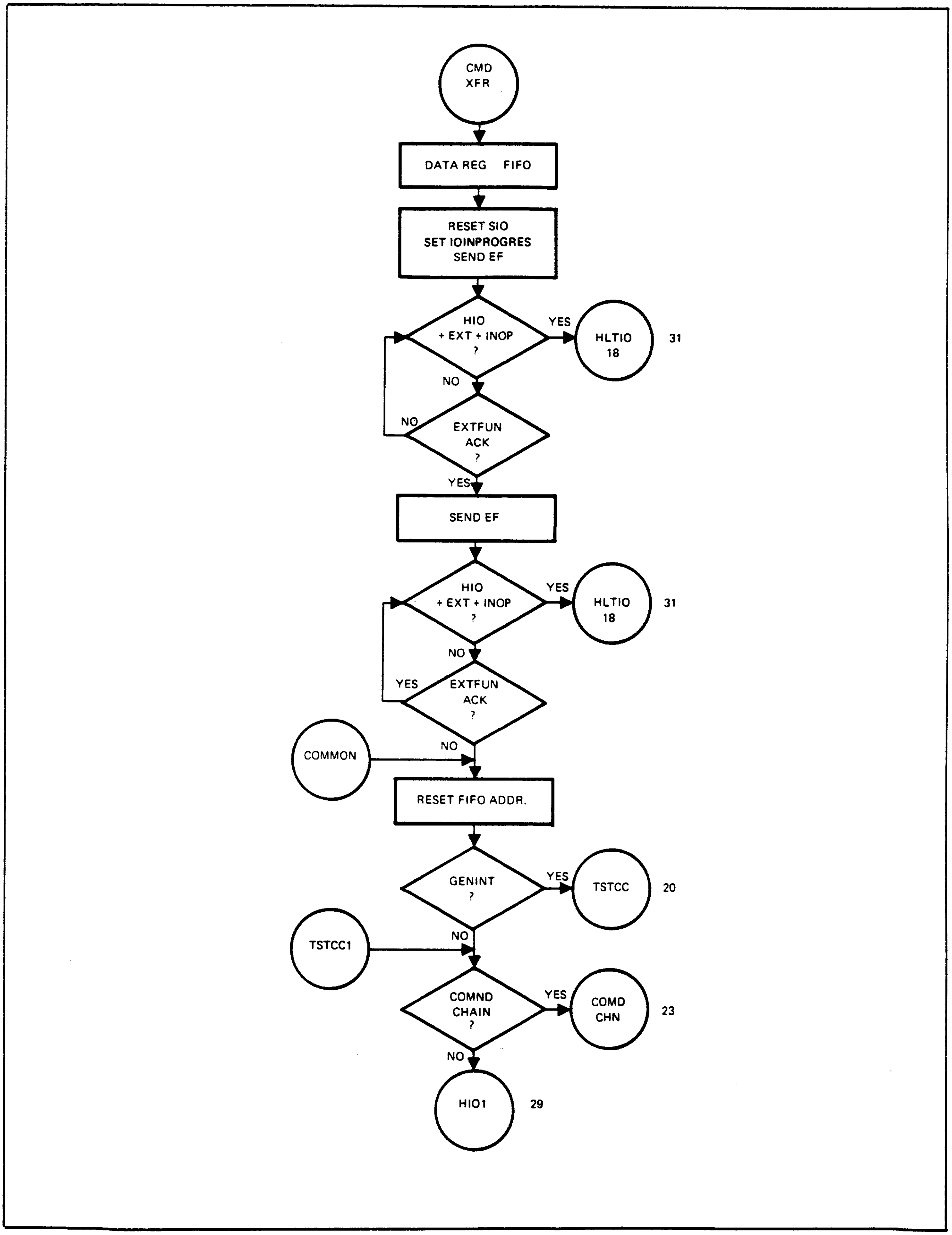


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 20 of 46)

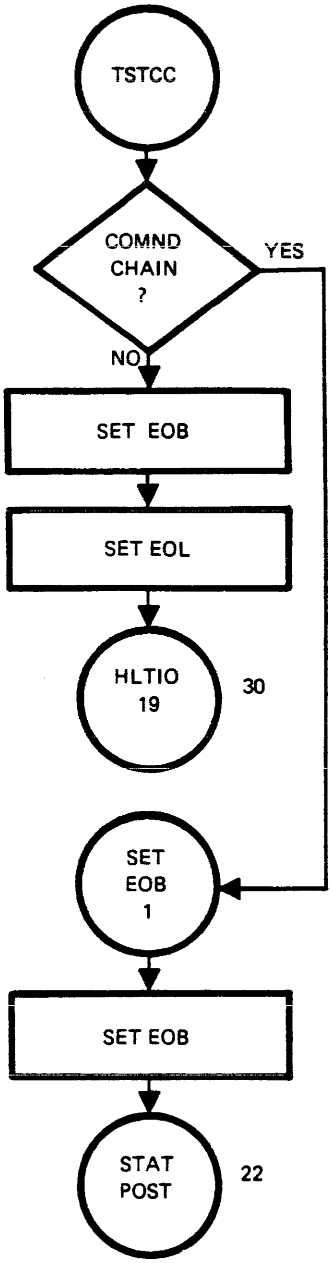


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 21 of 46)

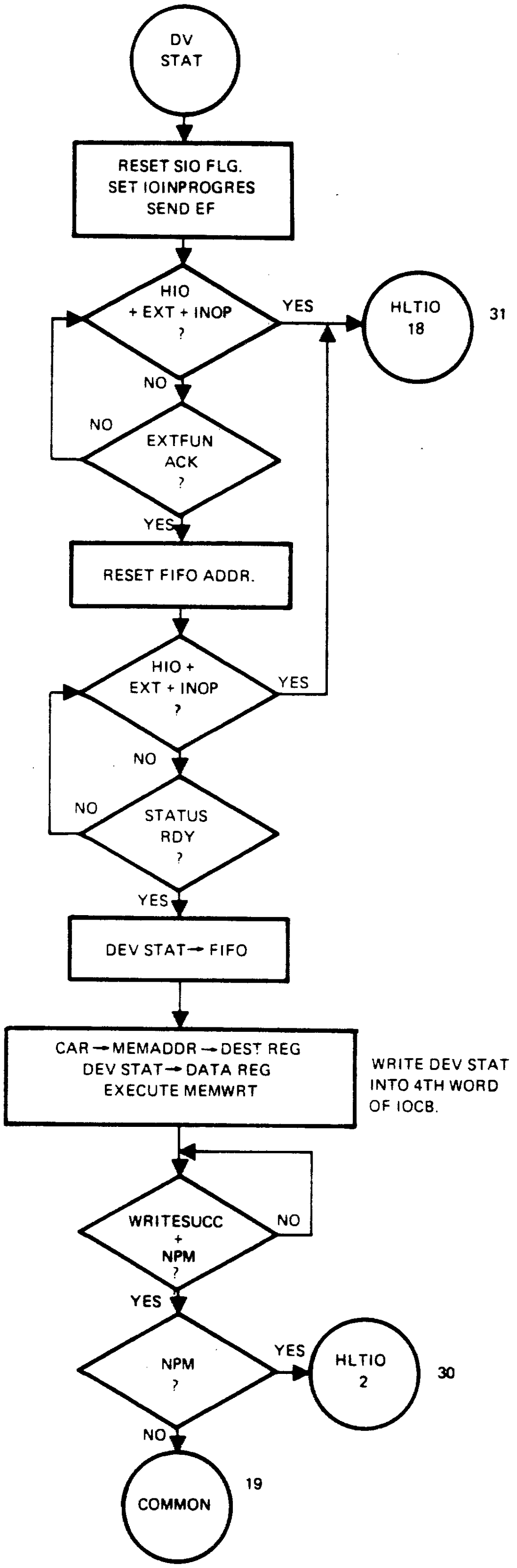


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 22 of 46)

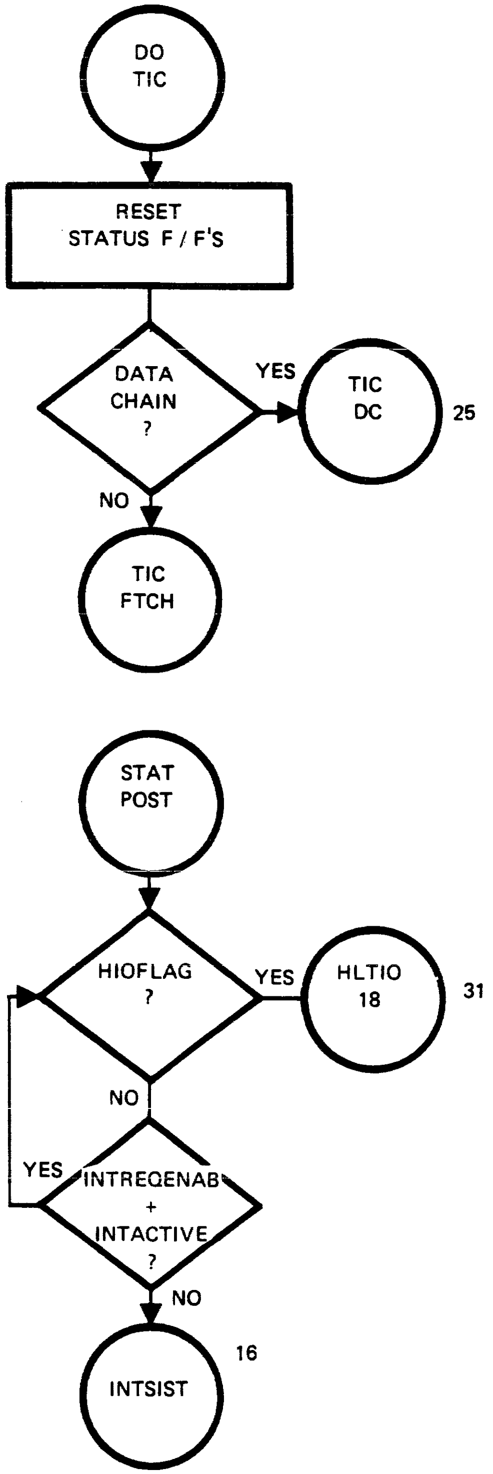


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 23 of 46)

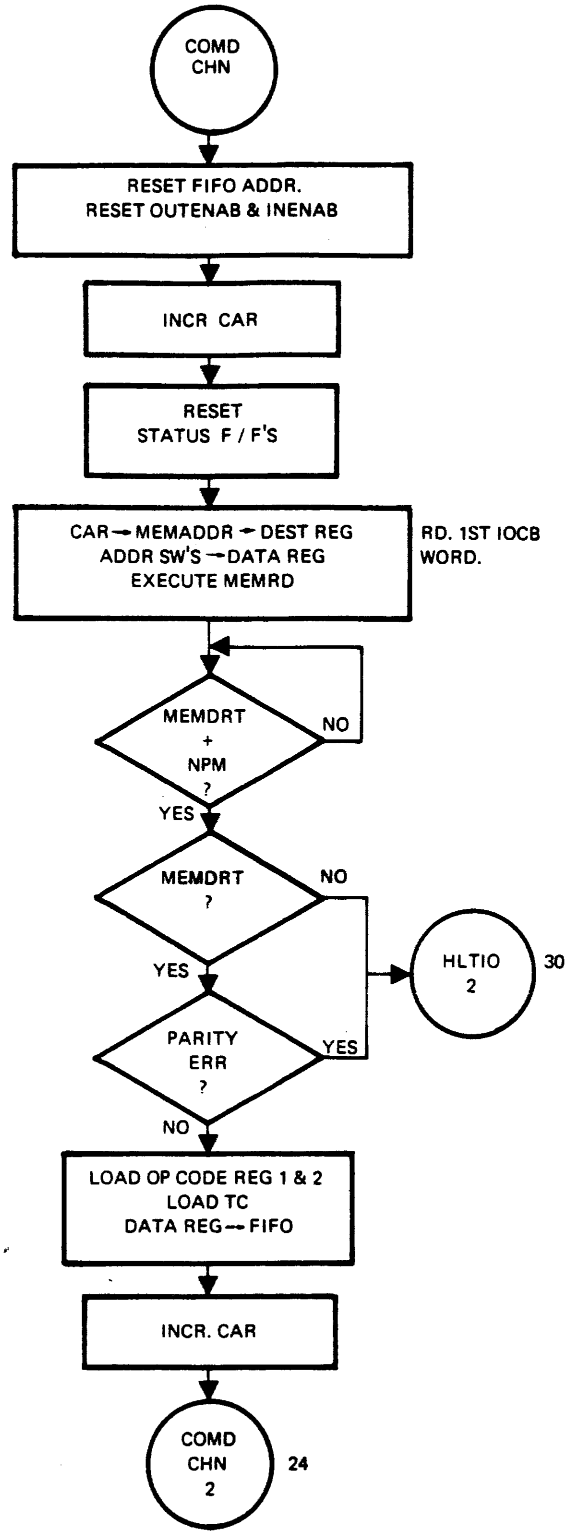


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 24 of 46)

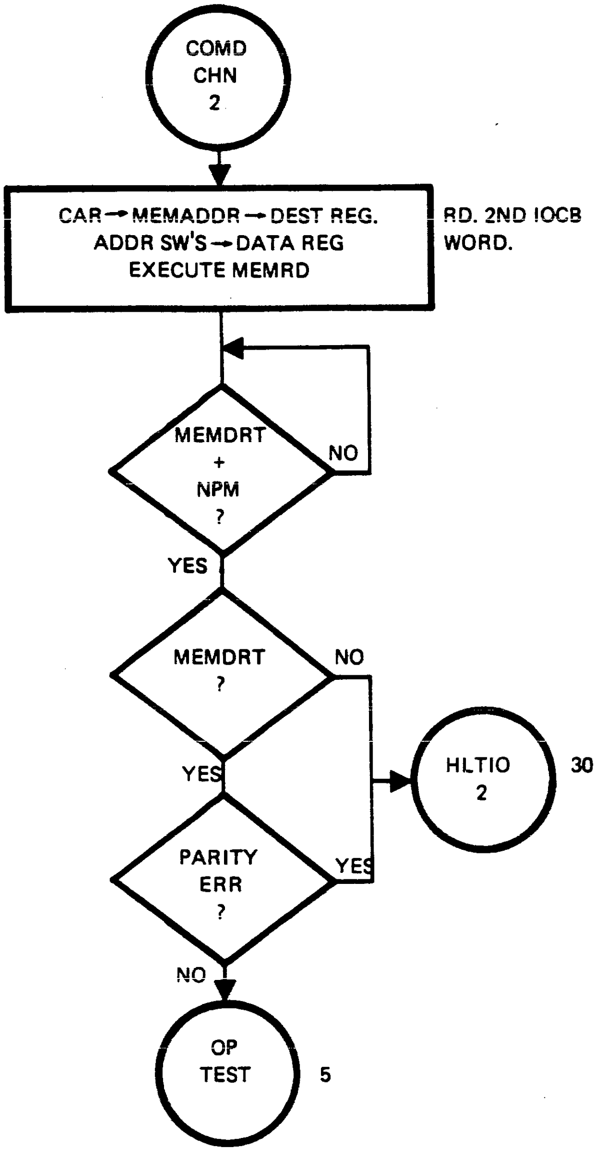


Figure 3-20. Flowchart - Memory Interface Control Sequencer
(Sheet 25 of 46)

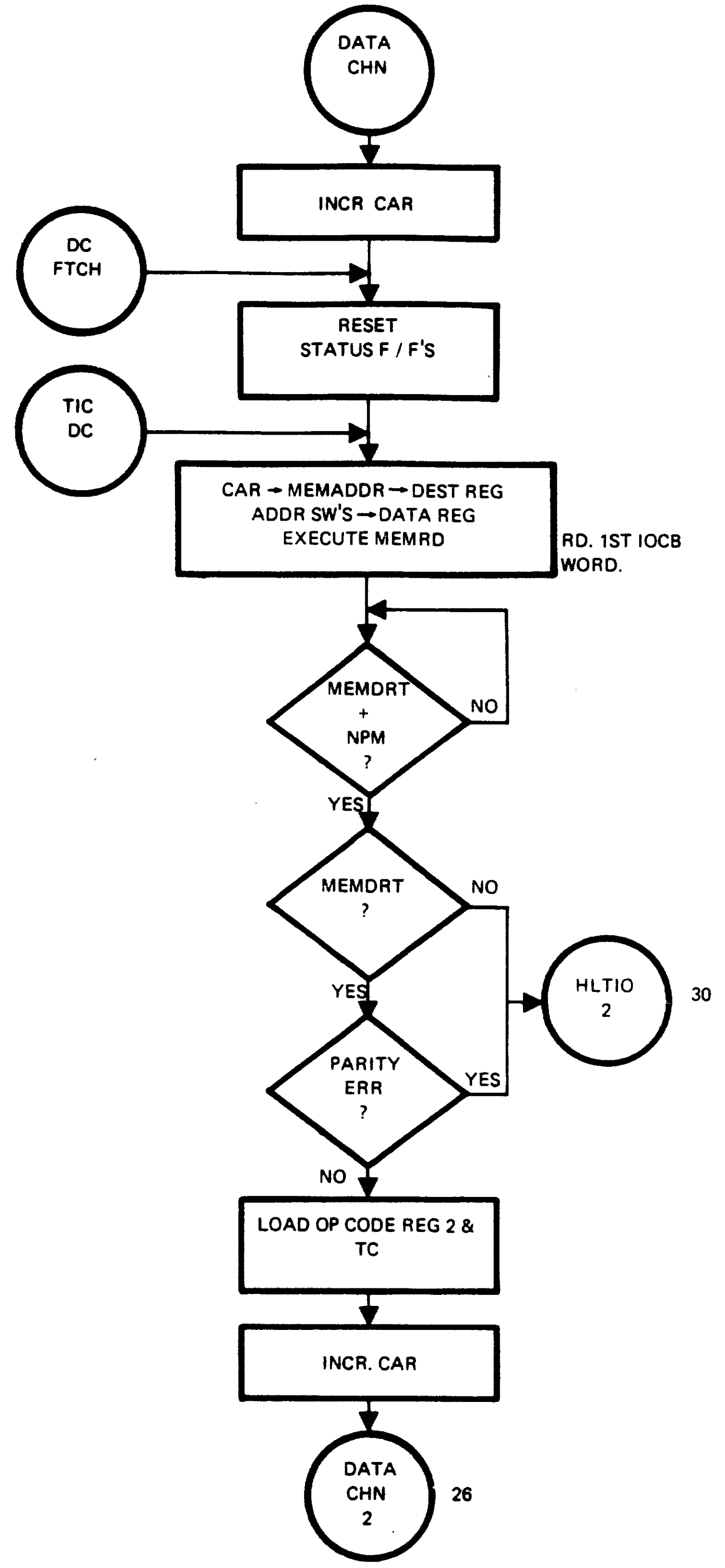


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 26 of 46)

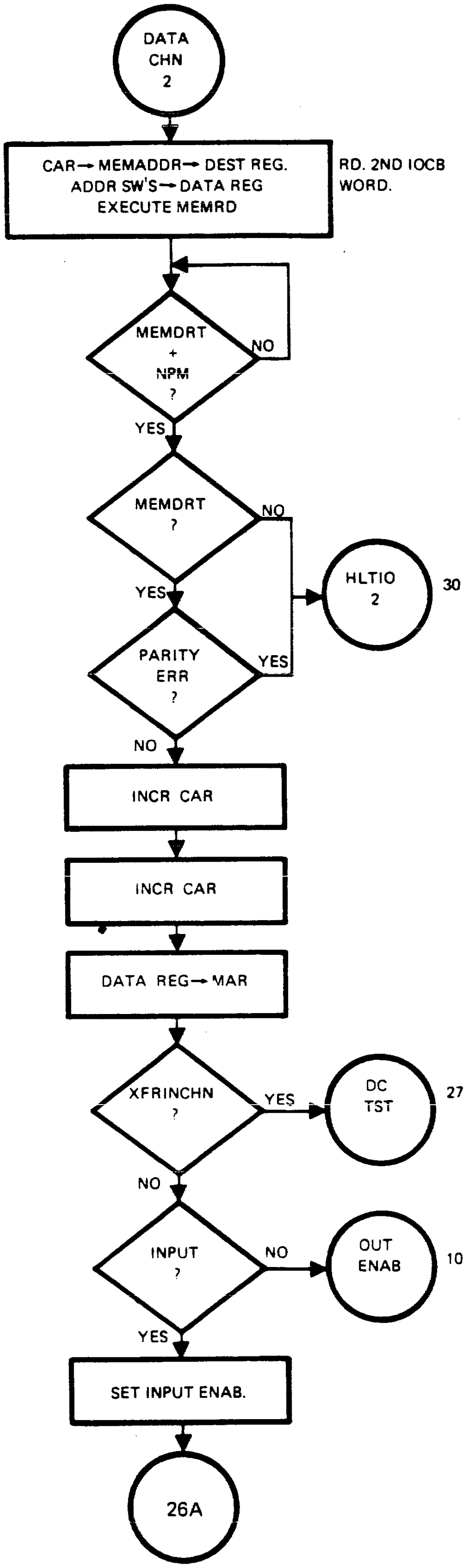


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 27 of 46)

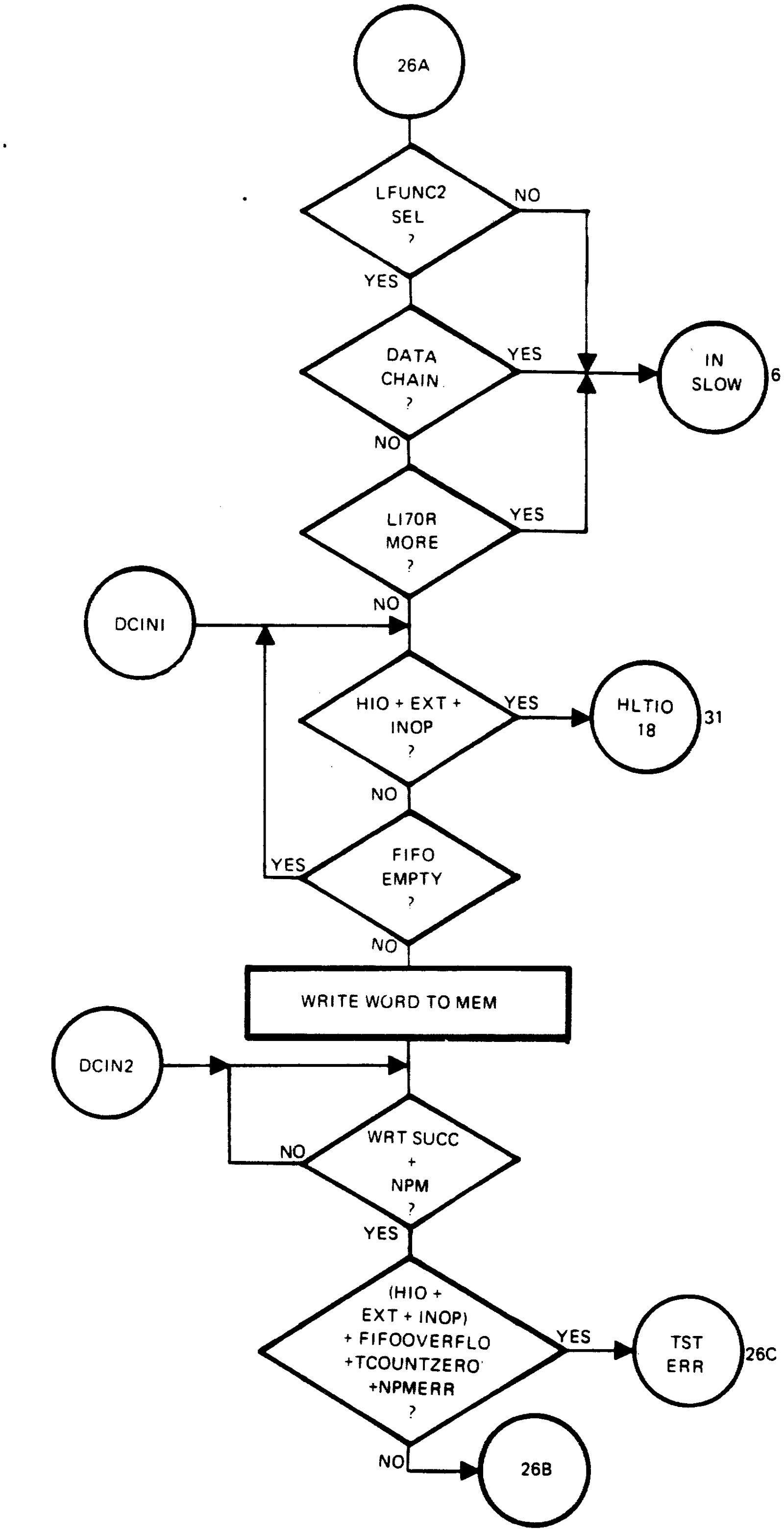


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 28 of 46)

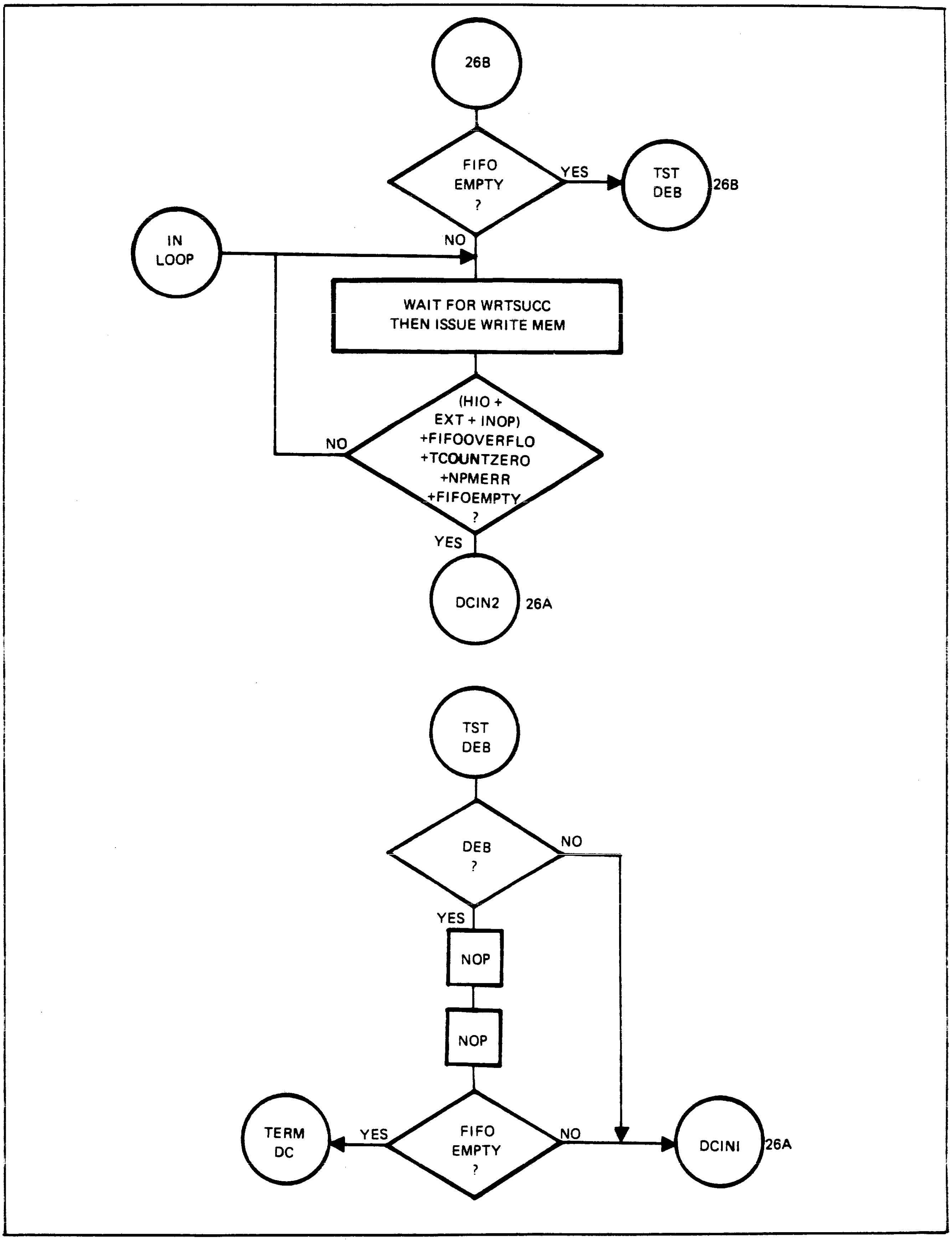


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 29 of 46)

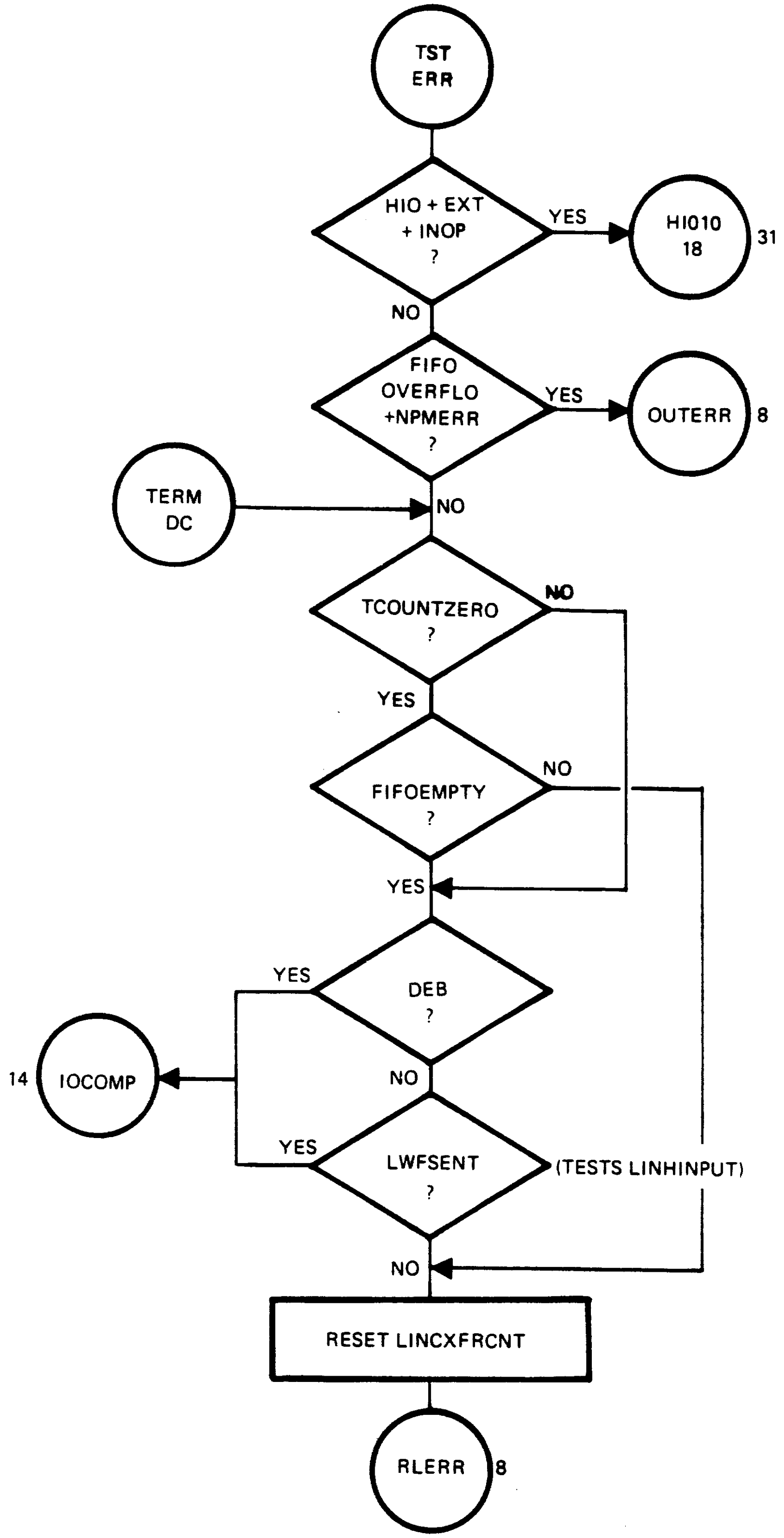


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 30 of 46)

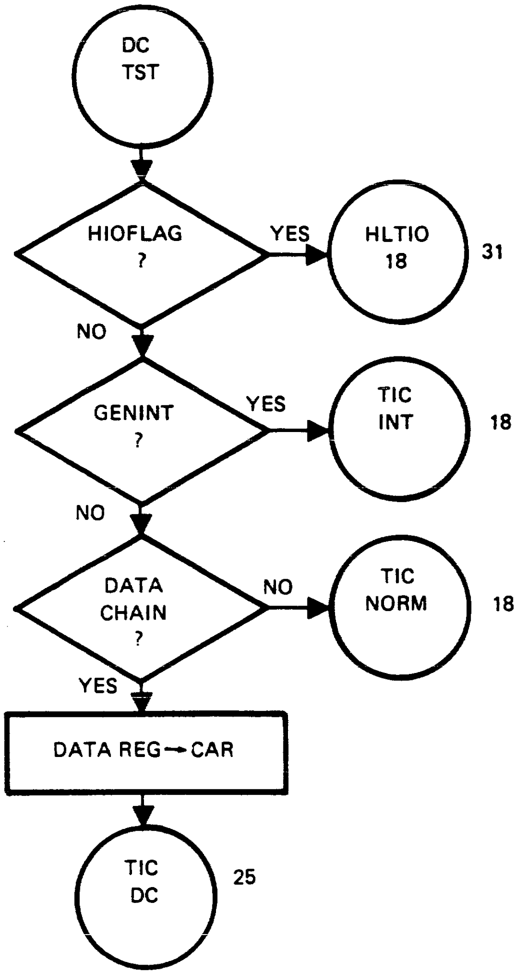


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 31 of 46)

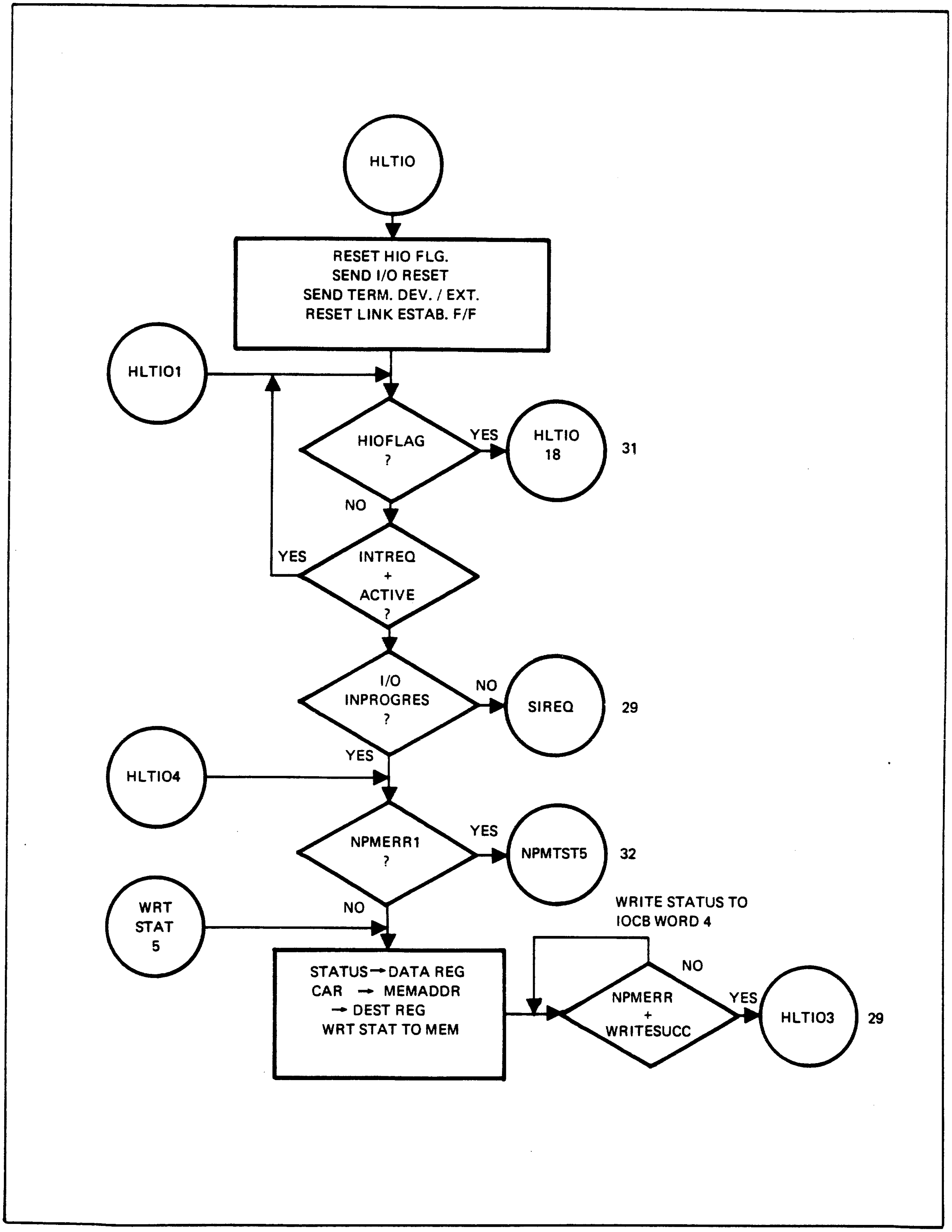


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 32 of 46)

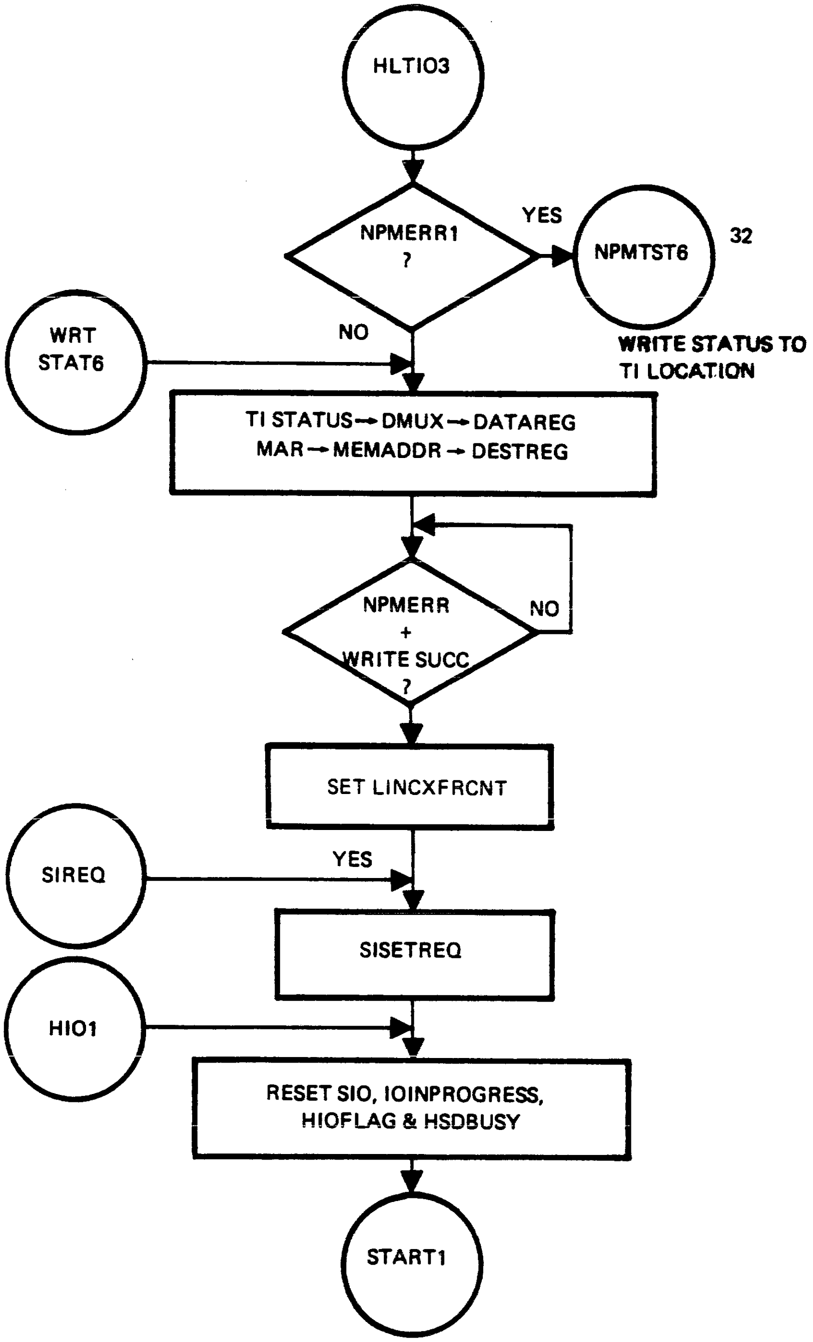


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 33 of 46)

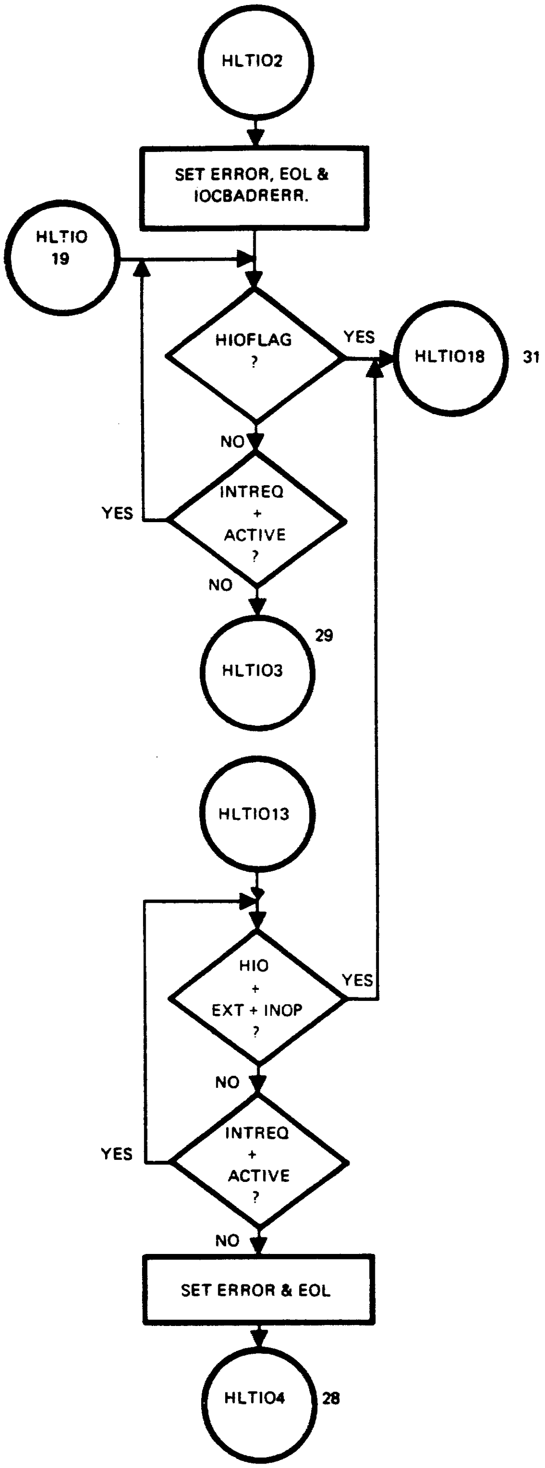


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 34 of 46)

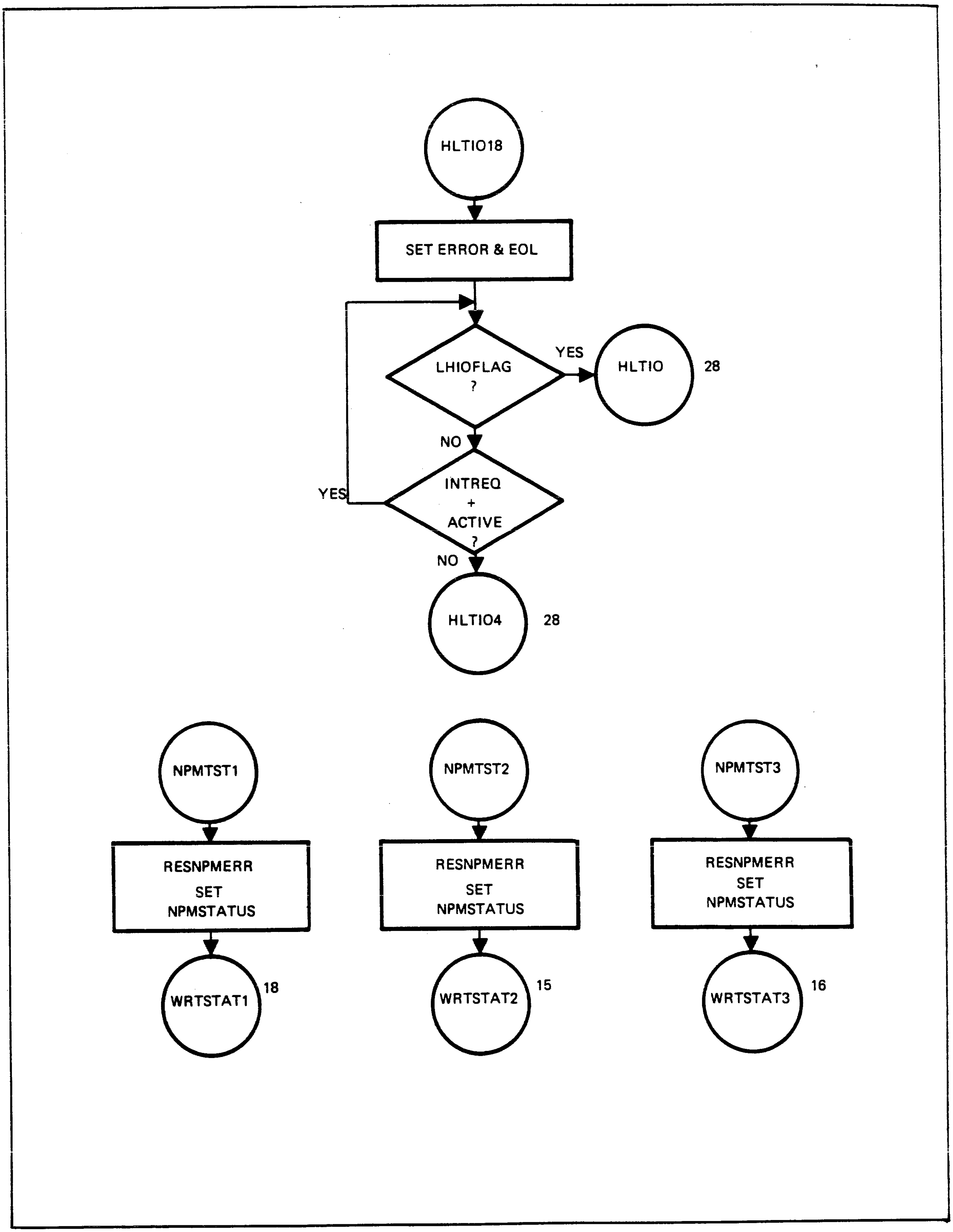


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 35 of 46)

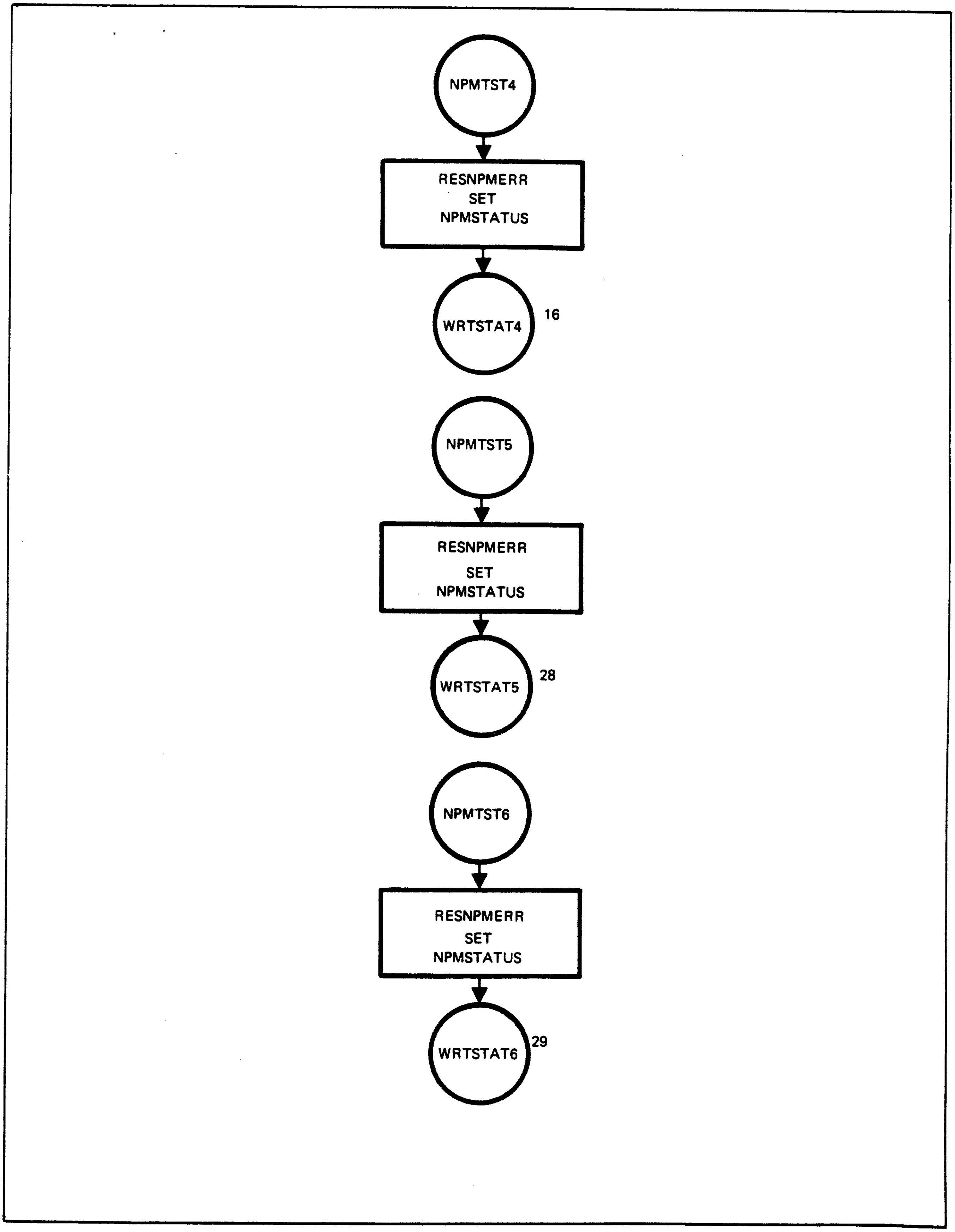


Figure 3-20. Flowchart - Memory Interface Control Sequencer
(Sheet 36 of 46)

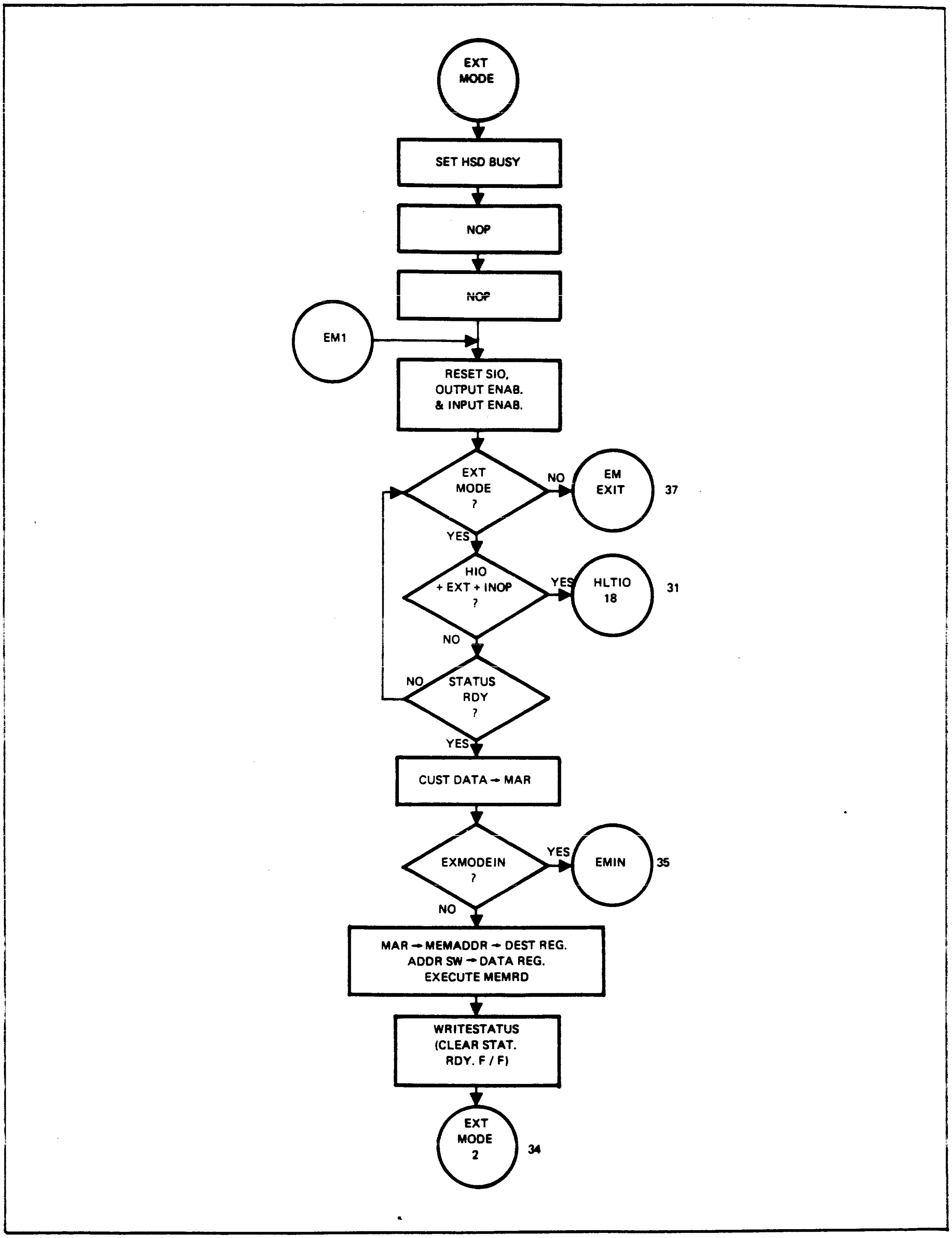


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 37 of 46)

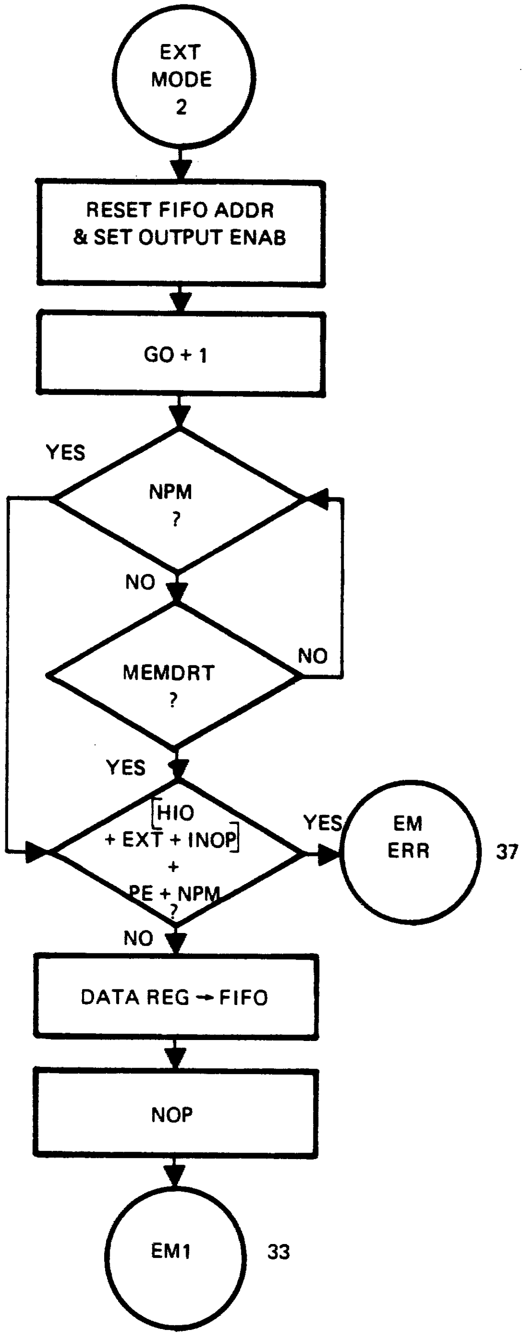


Figure 3-20. Flowchart - Memory Interface Control Sequencer
(Sheet 38 of 46)

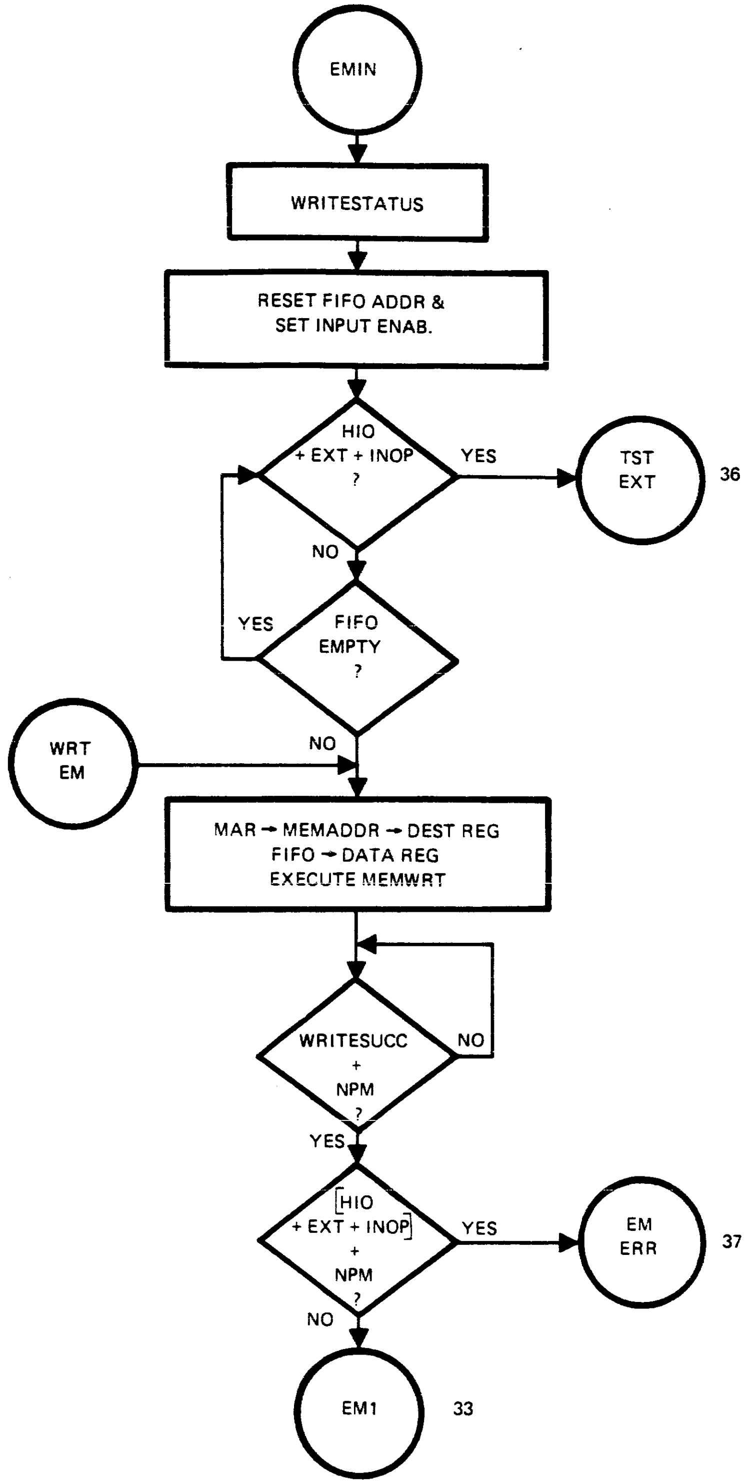


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 39 of 46)

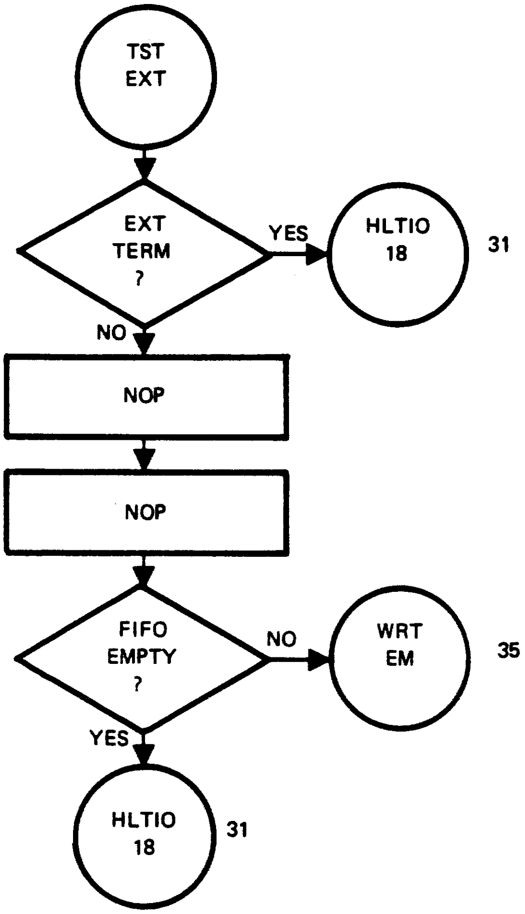


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 40 of 46)

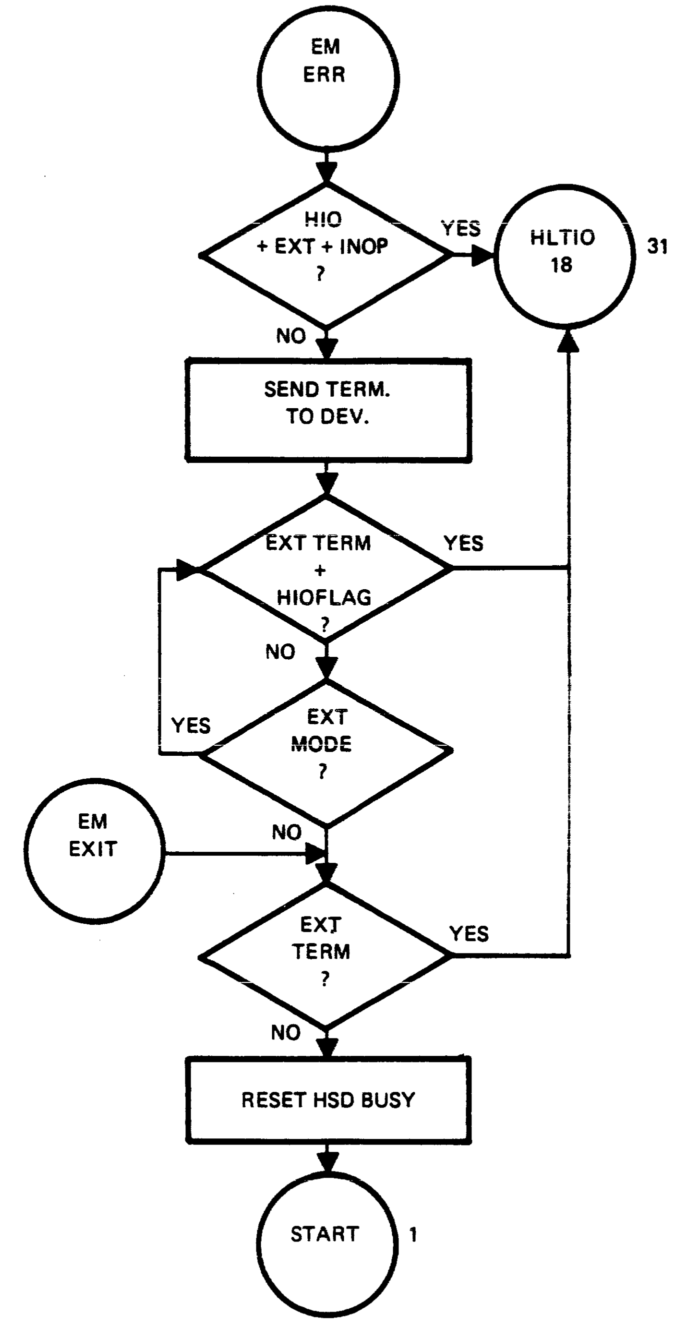


**Figure 3-20. Flowchart - Memory Interface Control Sequencer
(Sheet 41 of 46)**

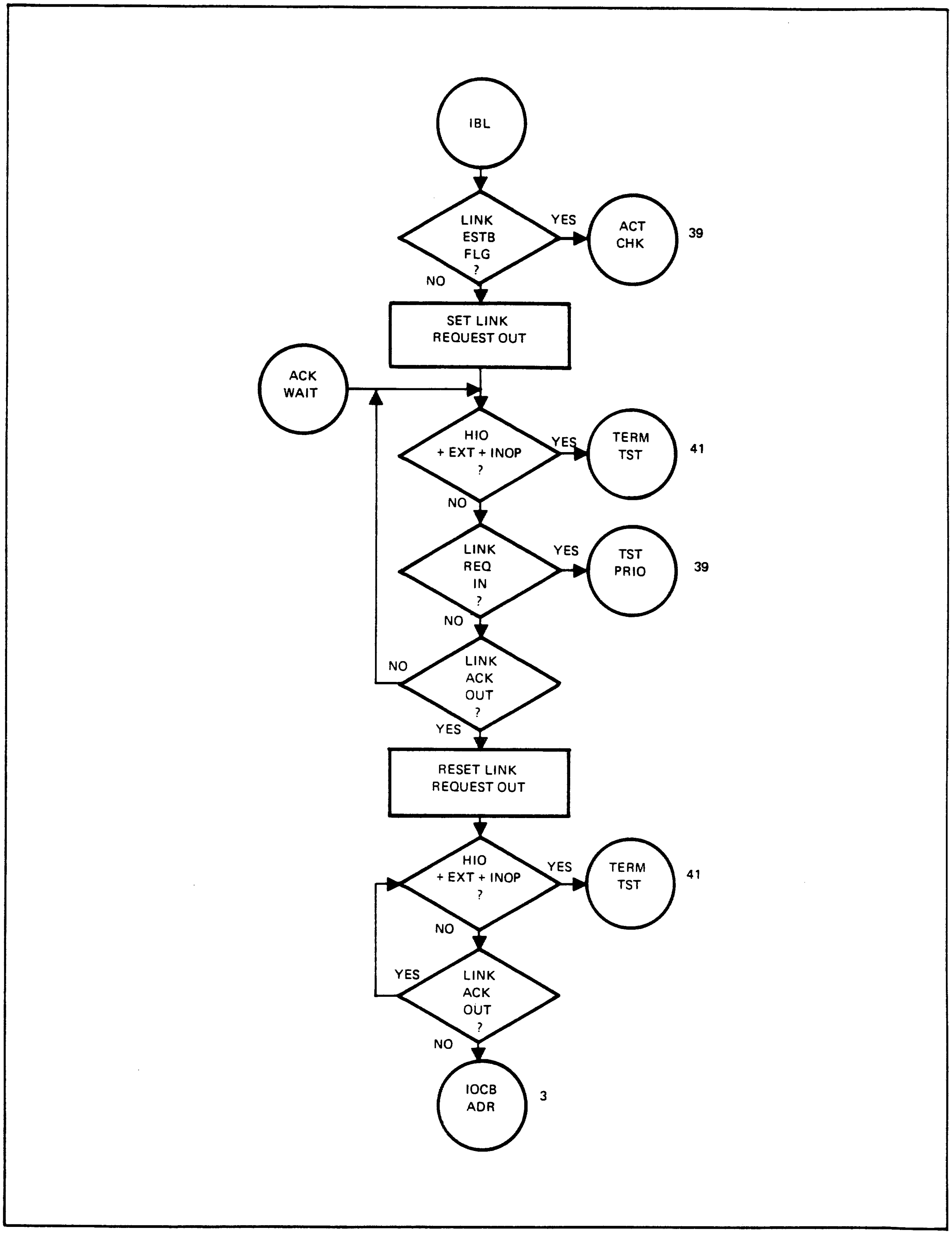


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 42 of 46)

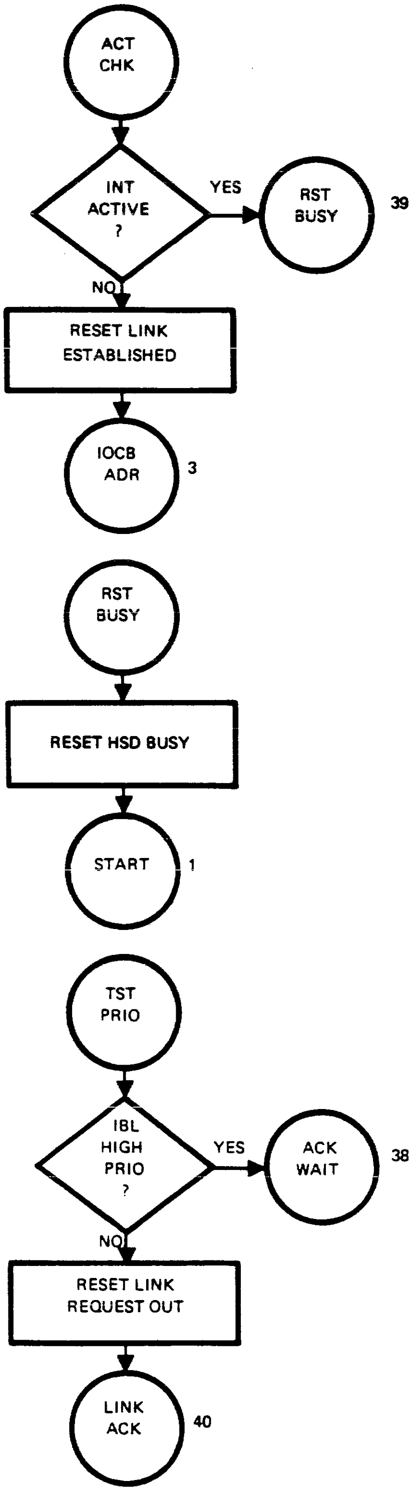


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 43 of 46)

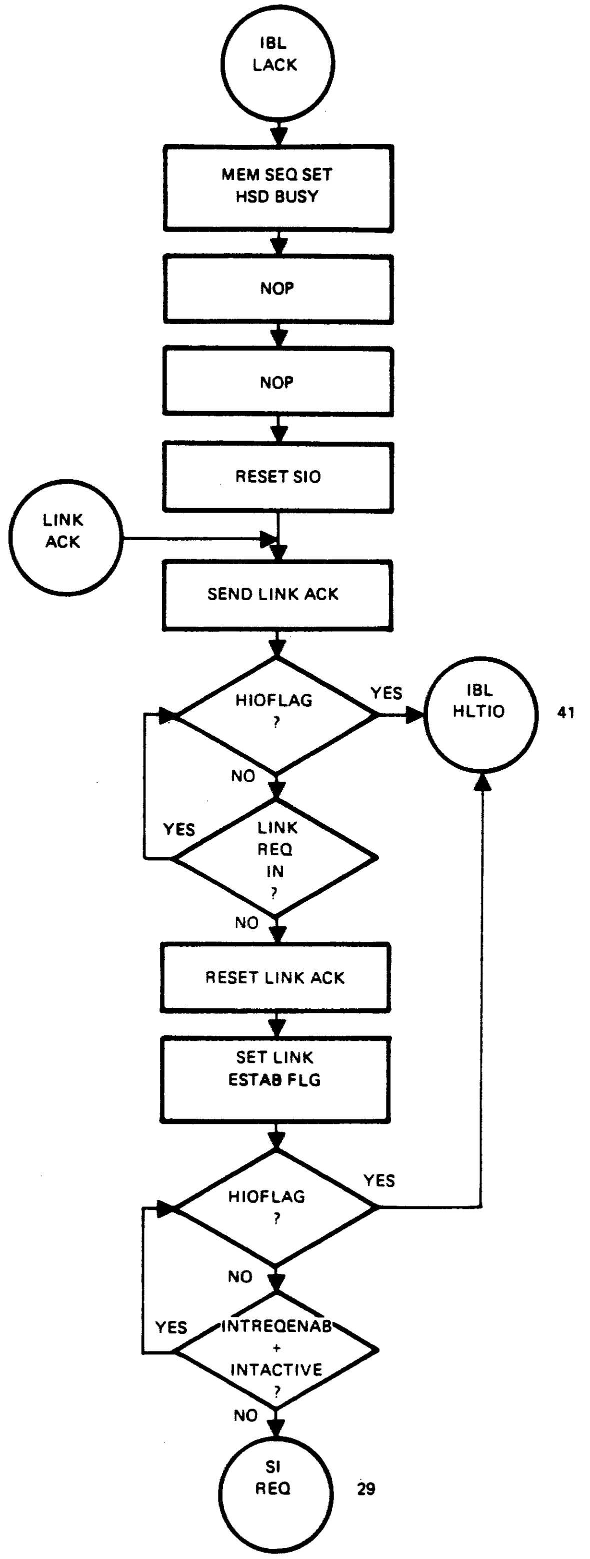


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 44 of 46)

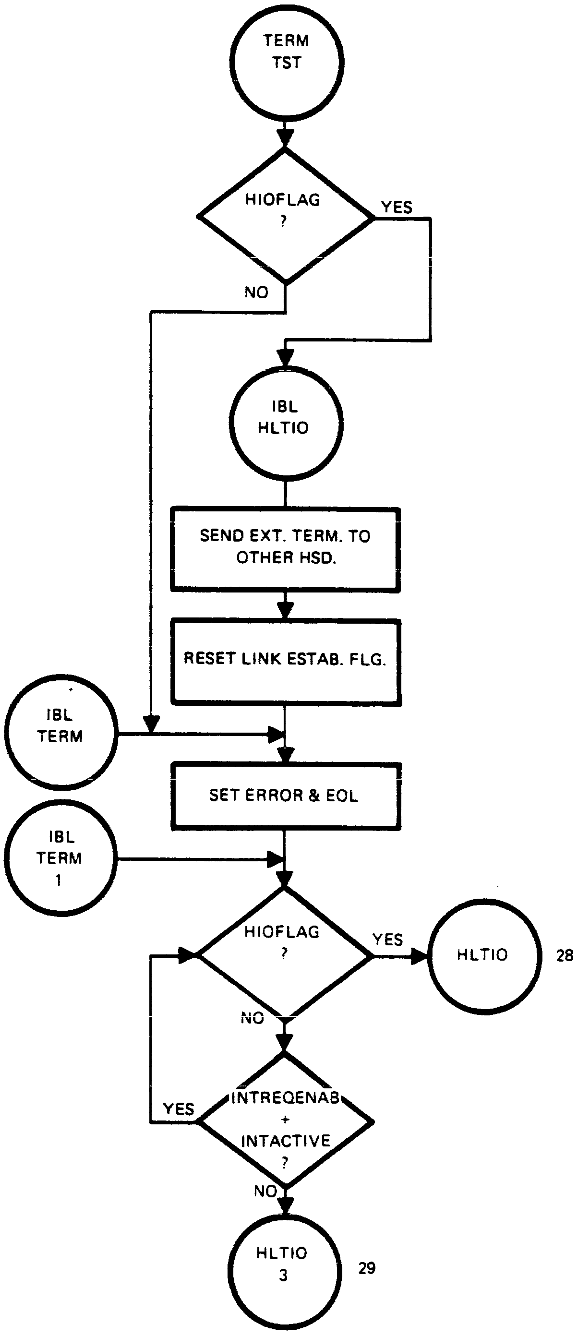


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 45 of 46)

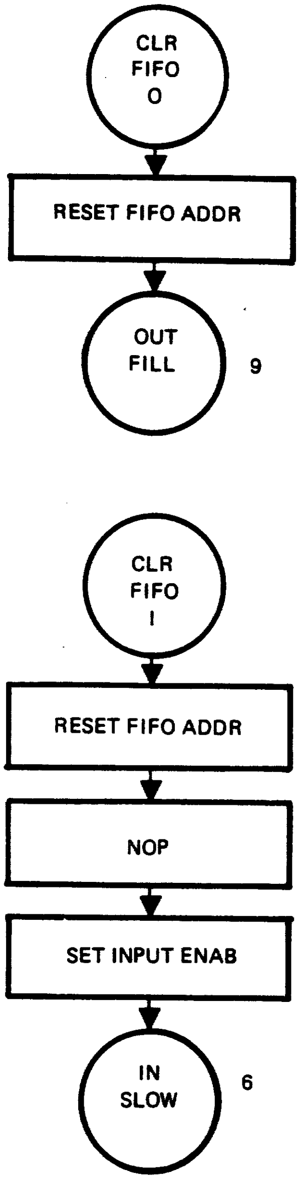


Figure 3-20. Flowchart - Memory Interface Control Sequencer (Sheet 46 of 46)

3.3.5 HSD/Customer Interface

3.3.5.1 Interface Signal Definition

A basic block diagram of the interface between the customer's device and the HSD is shown in Figure 3-21. A definition of each control signal and its intended use is described below:

1. IDR - signal applied to the HSD with the input data. Level. Remains true until the IA is returned by the HSD.

2. IA - signal from the HSD acknowledging receipt of the input data. Level. Remains true with IDR drops.

3. ODR - signal from the HSD with the output data. Level. Remains true until OA is returned by the customer's equipment.

4. OA - signal applied to the HSD acknowledging receipt of the output data. Level. Remains true until ODR drops.

5. EF - signal from the HSD with the function (command data) on the data lines. Level. Remains true until the EFA is returned by the customer's equipment.

6. EFA - signal applied to the HSD acknowledging receipt of the function data level. Remains true until EF drops.

7. ISR - signal applied to the HSD with the status information on input data lines. Level. Remains true until receipt of ISA from HSD.

8. ISA - signal from the HSD acknowledging receipt of the status information. Level. Remains true until ISR drops.

9. LWF - flag accompanying the last ODR in the output mode or the last IA in the input mode. This is a flag from the HSD to the customer's equipment signifying end-of-block. The customer must not try to send more data in the input mode or expect more data in the output mode.

10. EXT - asynchronous signal applied to the HSD to indicate an unusual terminate condition in the device controller. HSD will not attempt to send or receive more data after receipt of an EXT. The EXT causes storage of the residual transfer count, posting of the SI status, and generation of an interrupt.

11. EM - a level applied to the HSD indicating that the customer's equipment must provide the memory address for all ensuing data transfers. The customer's equipment maintains control until the EM level is removed.

12. IOR - standard I/O reset signal which occurs from the system reset of the halt I/O command.

13. DP - a level applied to the HSD indicating that the customer equipment is connected and on-line.

14. DEB - flag accompanying the last IDR in the input mode or the last OA in the output mode. This is a flag from the customer's equipment to the HSD signifying end-of-block. The HSD will not try to send more data in the output mode or expect more data in the input mode.

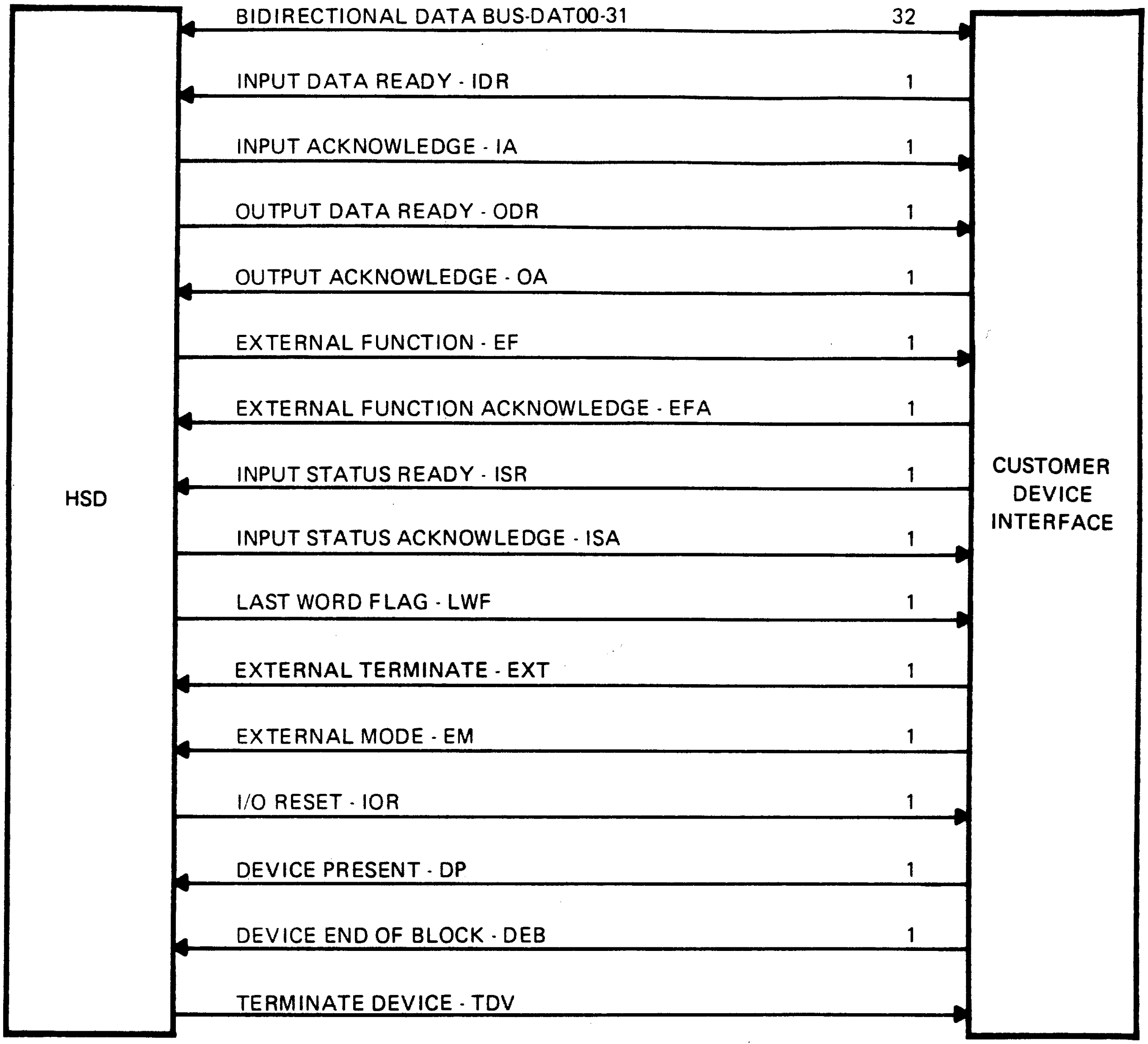


Figure 3-21. HSD/Customer Interface Block Diagram

15. TDV - asynchronous signal applied to the device to indicate an unusual terminate condition in the HSD. The device should not attempt to send or receive more data after receipt of a TDV. The HSD will store its normal error and SI status, and thus indicate the condition which caused the TDV.

3.3.5.2 Interface Timing Relationships

As indicated by the signal descriptions, the HSD/customer interface is of the simple handshake variety. Basic timing relationships between these handshake signals are shown in Figures 3-22 and 3-23. Interface logic diagrams illustrating the timing are under interface electrical characteristics. All relationships assume cable propagation delays as constants; therefore, they are omitted in the timing equations which are presented. All times are computed using the HSD interface as the reference, and any fixed times are to be measured at the +2.3-volt points. All interface signals are nominally ground for a logic one, and nominally +5.0 volts for a logic zero. Table 3-12 provides the interface timing definitions.

The minimum time from the trailing edge (positive-going) of the ODR, EF, IDR, or ISR signal to the leading edge (negative-going) of the next ODR, EF, IDR, or ISR signal is 300 nanoseconds. Only one interface function (ODR, EF, IDR, or ISR) may occur at a time.

The LWF signal has the same timing relationship to other interface signals as the ODR and the IA signals which LWF must accompany to provide an end-of-block indication. See Figures 3-22 and 3-23 for the ODR and IA timing.

The DEB signal has the same timing relationship to other interface signals as the IDR and the OA signals which DEB must accompany to inform the HSD that an end-of-block has occurred at the device. See Figure 3-24 and 3-26 for the IDR and OA timing.

The EXT and TDV signals may occur asynchronously with respect to any other interface activity. These signals must be at least 300 nanoseconds wide and no longer that 600 nanoseconds.

The EM signal is generated by the customer logic only after being enabled by a command transfer I/O operation. This signal is a level which may be raised any time after the reception of the device command information. This level must remain valid (logic one) until all externally controlled transfers have been completed, including the appropriate HSD response to the last transfer. While this level is valid, attempted execution of a macro level CD or TD will cause a program violation. Interrupt control instructions will execute normally while the EM signal is valid.

The IOR signal is a 300-nanosecond pulse generated by the HSD during all system reset conditions, or by reception of a CD terminate instruction from the macrolevel software.

The DP signal is a level generated by the customer equipment which indicates that the power is ON, and any other device operable conditions specified by the customer have been met.

3.3.5.3 Interface Electrical Characteristics

The customer should use the interface defined in Figures 3-24 through 3-29. At the interface, a logic one is defined as 0 volt to +1.5 volts, and a logic zero is defined as +3.2 volts to +5.0 volts. These seemingly pseudo-TTL levels exist because of the driver-receiver circuit (S/N 75138).

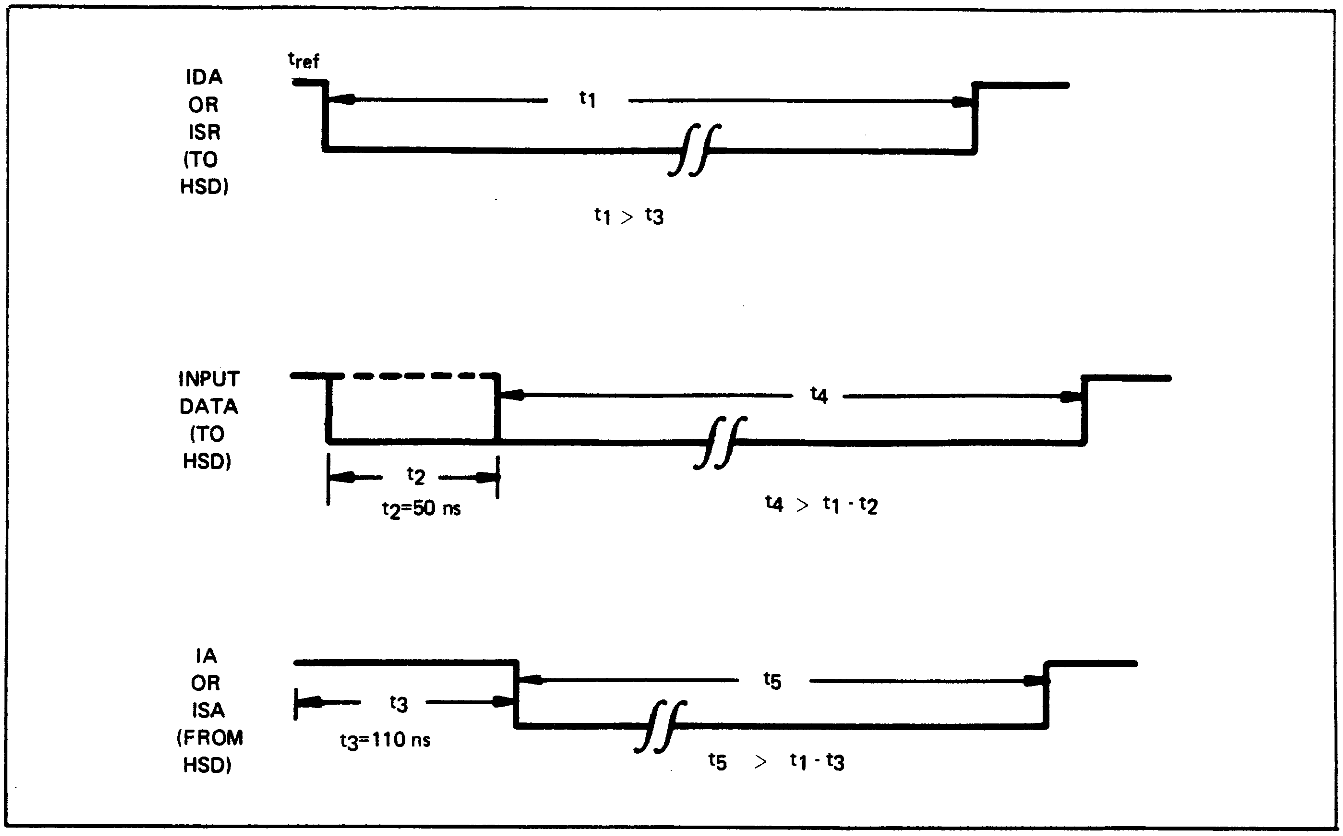


Figure 3-22. Input Data and Status Timing Relationships

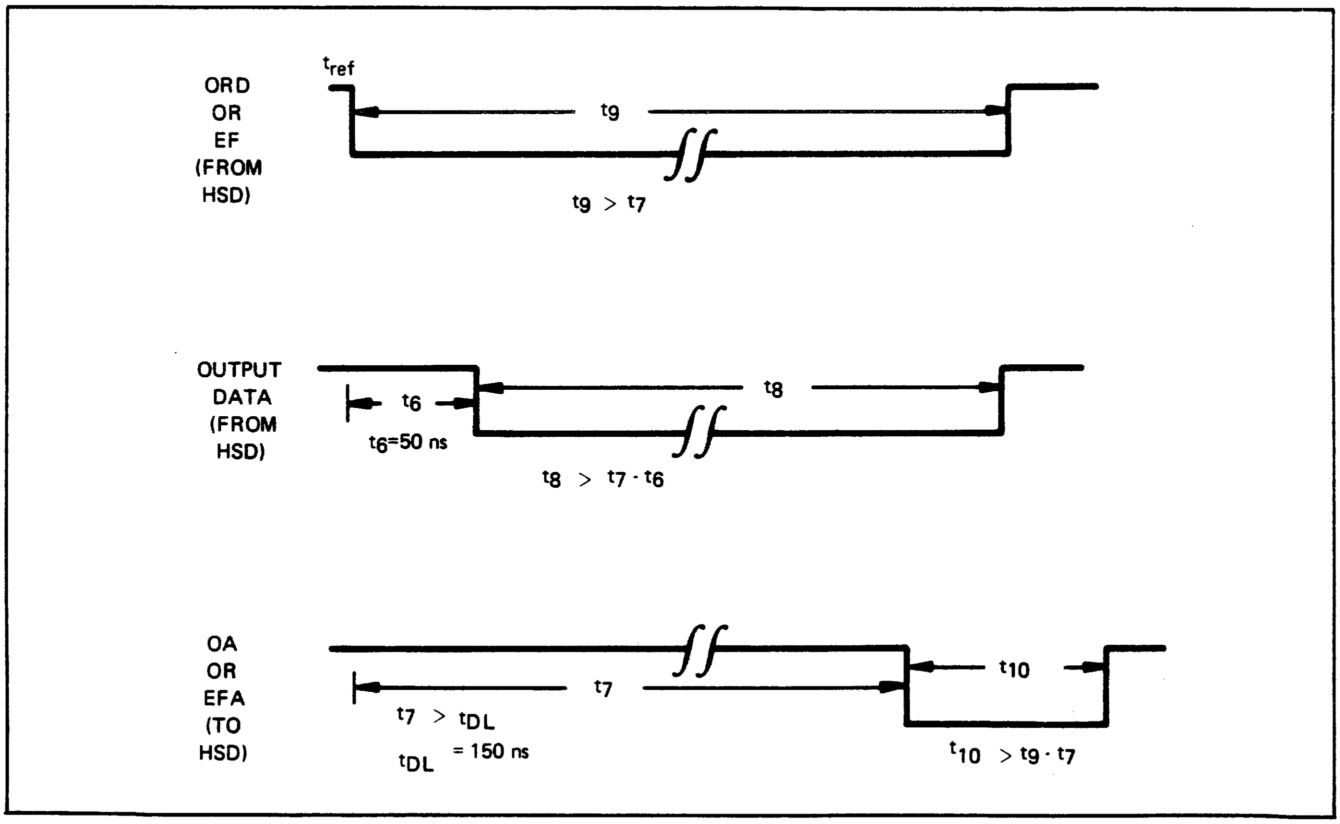


Figure 3-23. Output Data and External Function Timing Relationships

Table 3-12
Interface Timing Definitions

Timing Reference	Definition
t_1	Width of the IDR or ISR signal
t_2	Maximum time that data may lag the IDR signal at the HSD interface - (t_2 = 50 ns)
t_3	Minimum time before an IA or ISA may be returned to the customer device from the HSD
t_4	Width of the input data signal
t_5	Width of the IA or ISA signal
t_6	Maximum time that data may lag the ODR signal at the HSD interface (t_6 = 50 ns)
t_7	Time from the ODR or EF signal until the leading edge of the OA or EFA signal appears at the HSD interface
t_8	Width of the output data signal
t_9	Width of the ODR or EF signal
t_{10}	Width of the OA or EFA signal
t_{DL}	Maximum time permitted to load data into a storage media in the customer's device (t_{DL} = 150 ns)

Note: This time (t_{DL}) is measured from the time the ODR or EF signal appears at the customer device connector. t_{DL} must not exceed 150 ns if the HSD is to operate at rated speed, i.e., one transfer every 1.2 microsecond. t_{DL} may be longer if the device is to operate at a slower transfer rate.

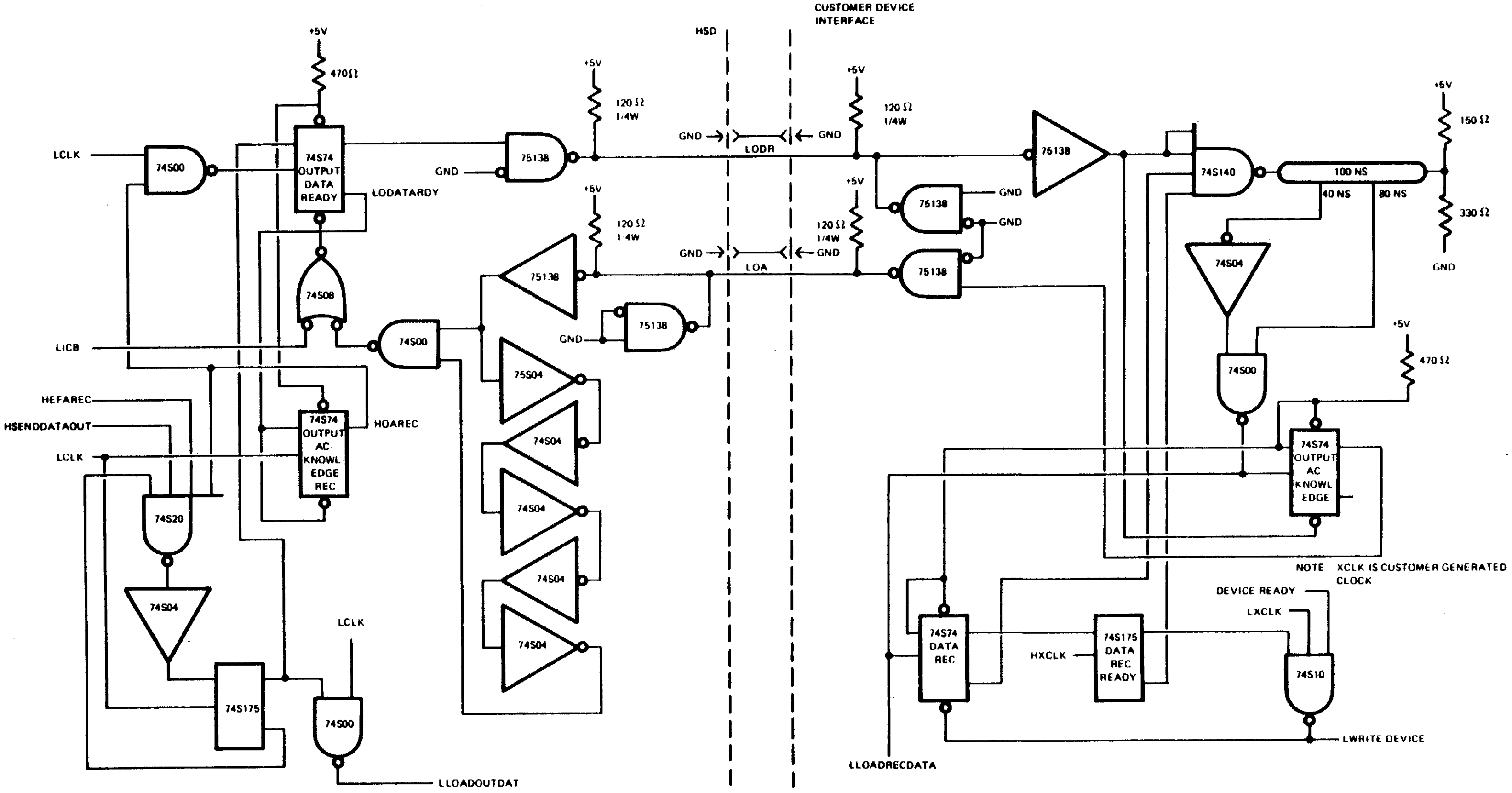


Figure 3-24. HSD/Customer Device Interface Output Data Control Logic

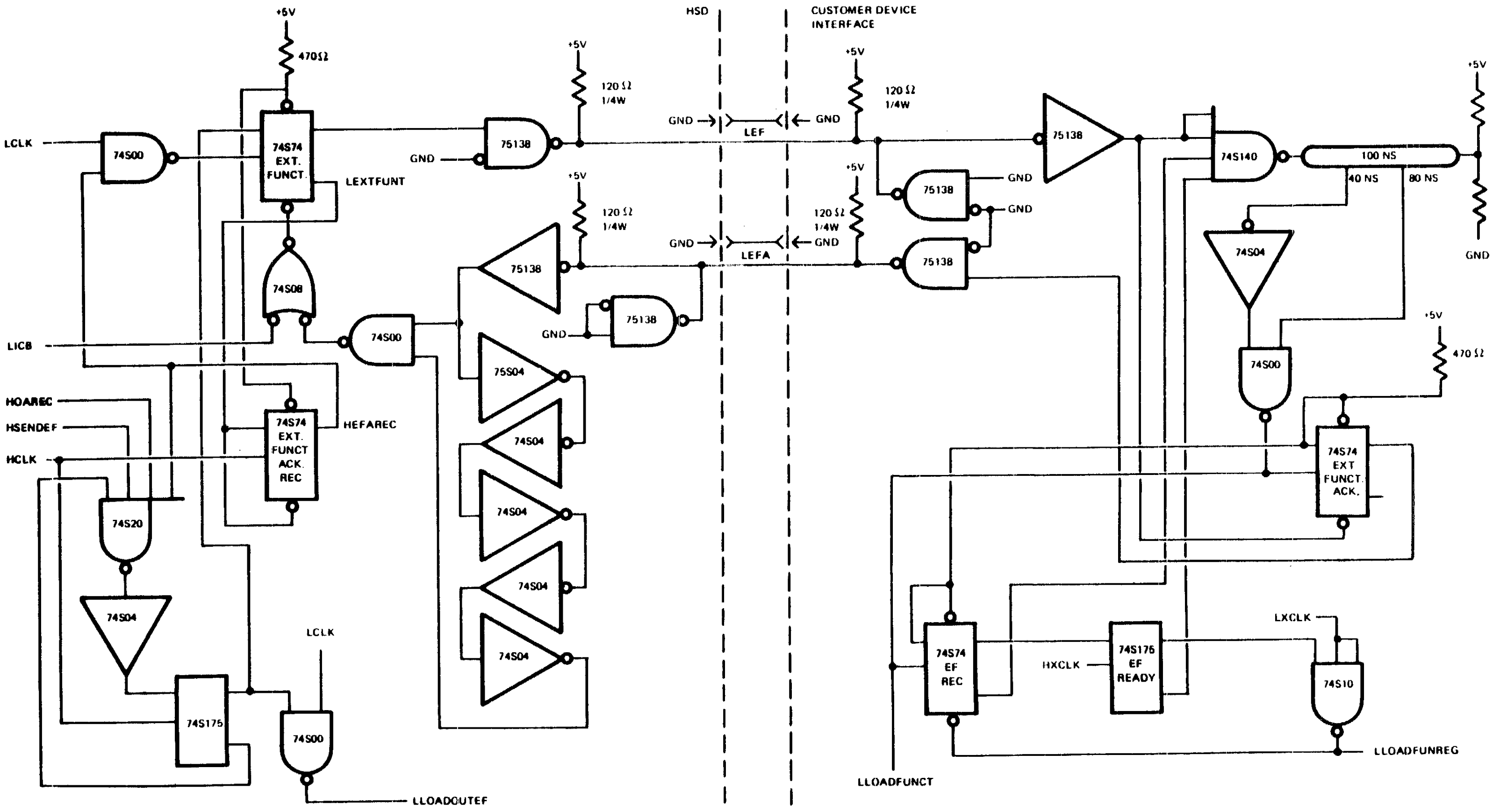


Figure 3-25. HSD/Customer Device Interface External Function Control Logic

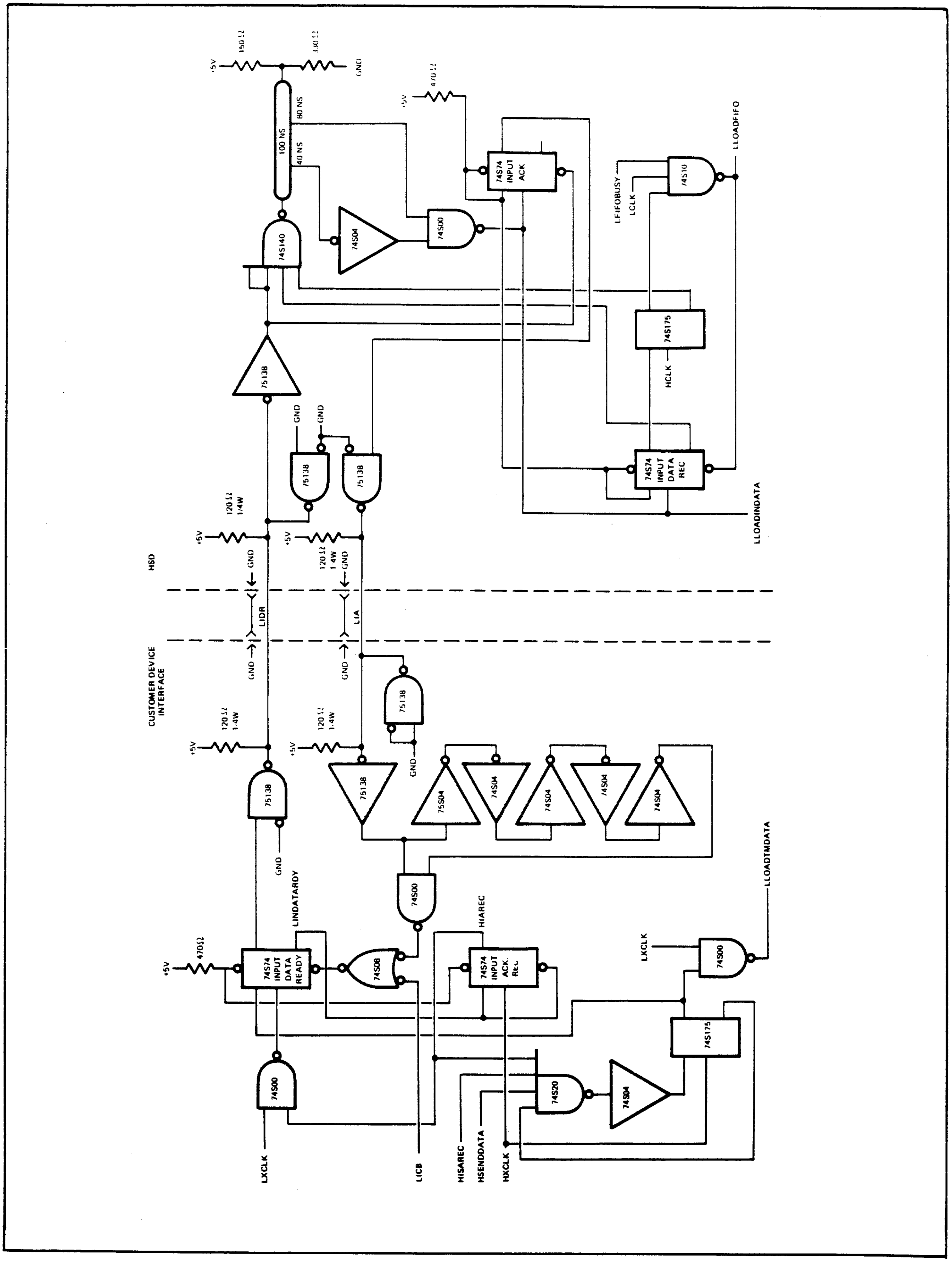


Figure 3-26. HSD/Customer Device Interface Input Data Control Logic

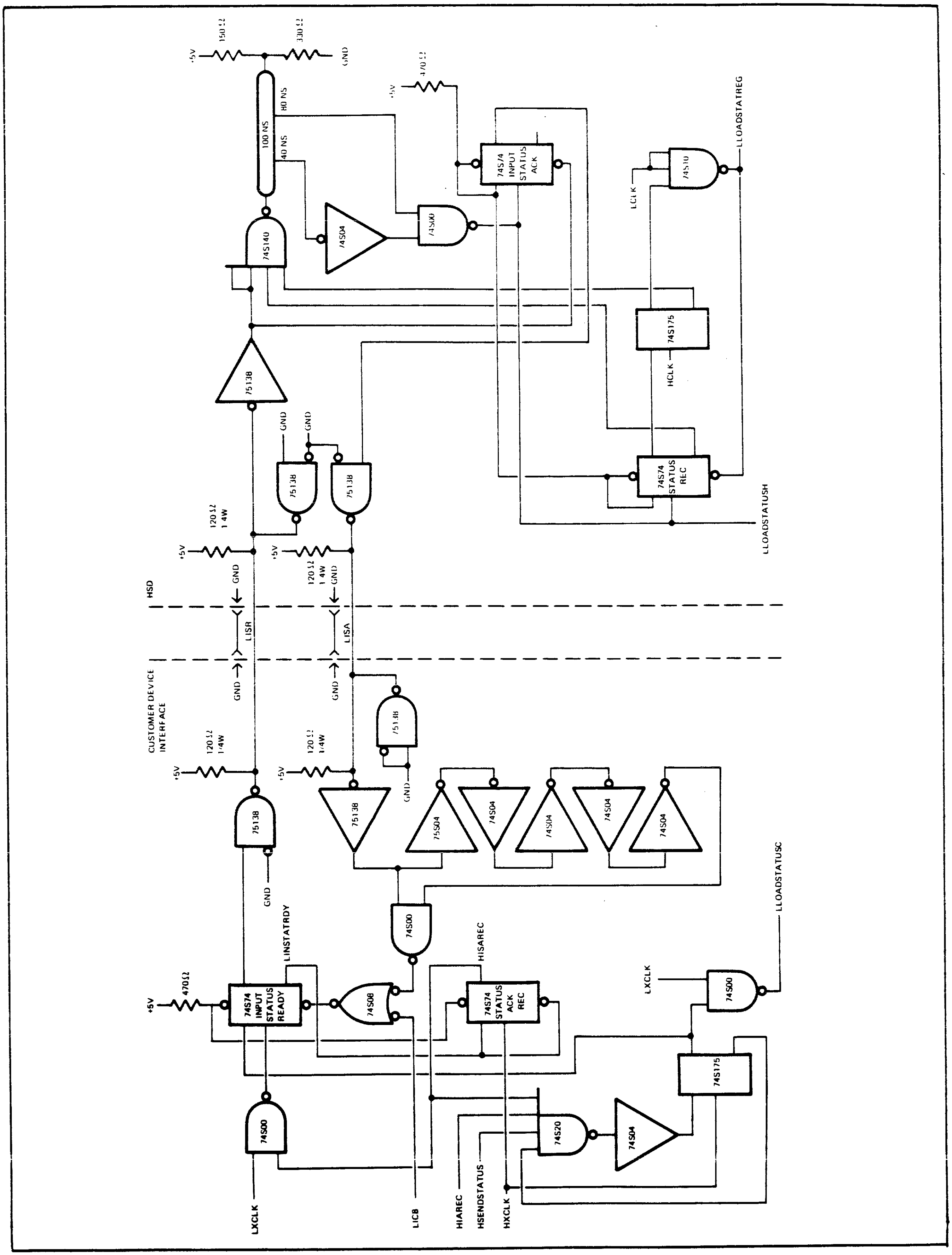


Figure 3-27. HSD/Customer Device Interface Input Status Control Logic

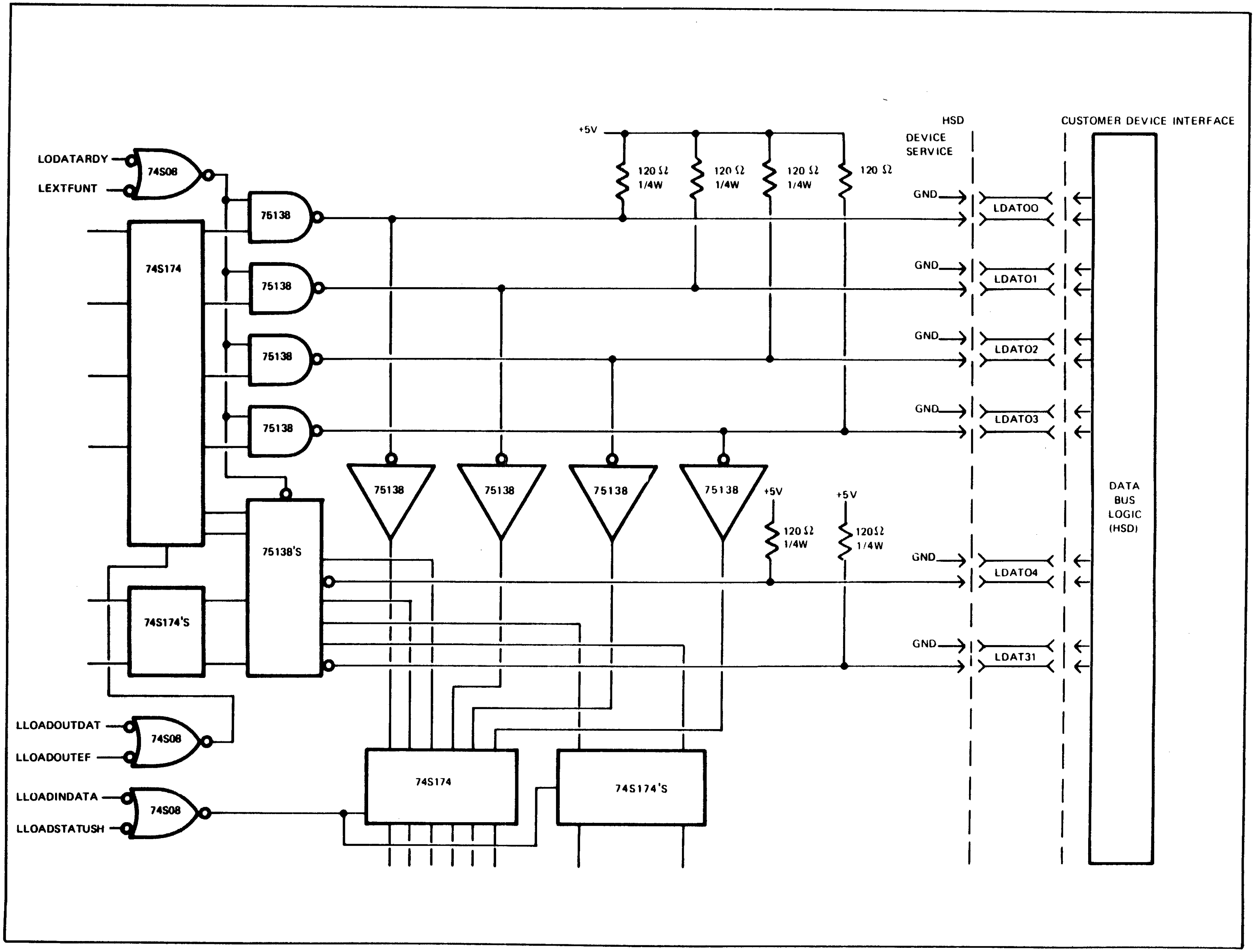


Figure 3-28. HSD/Customer Device Interface Data Bus Logic (Sheet 1 of 2)

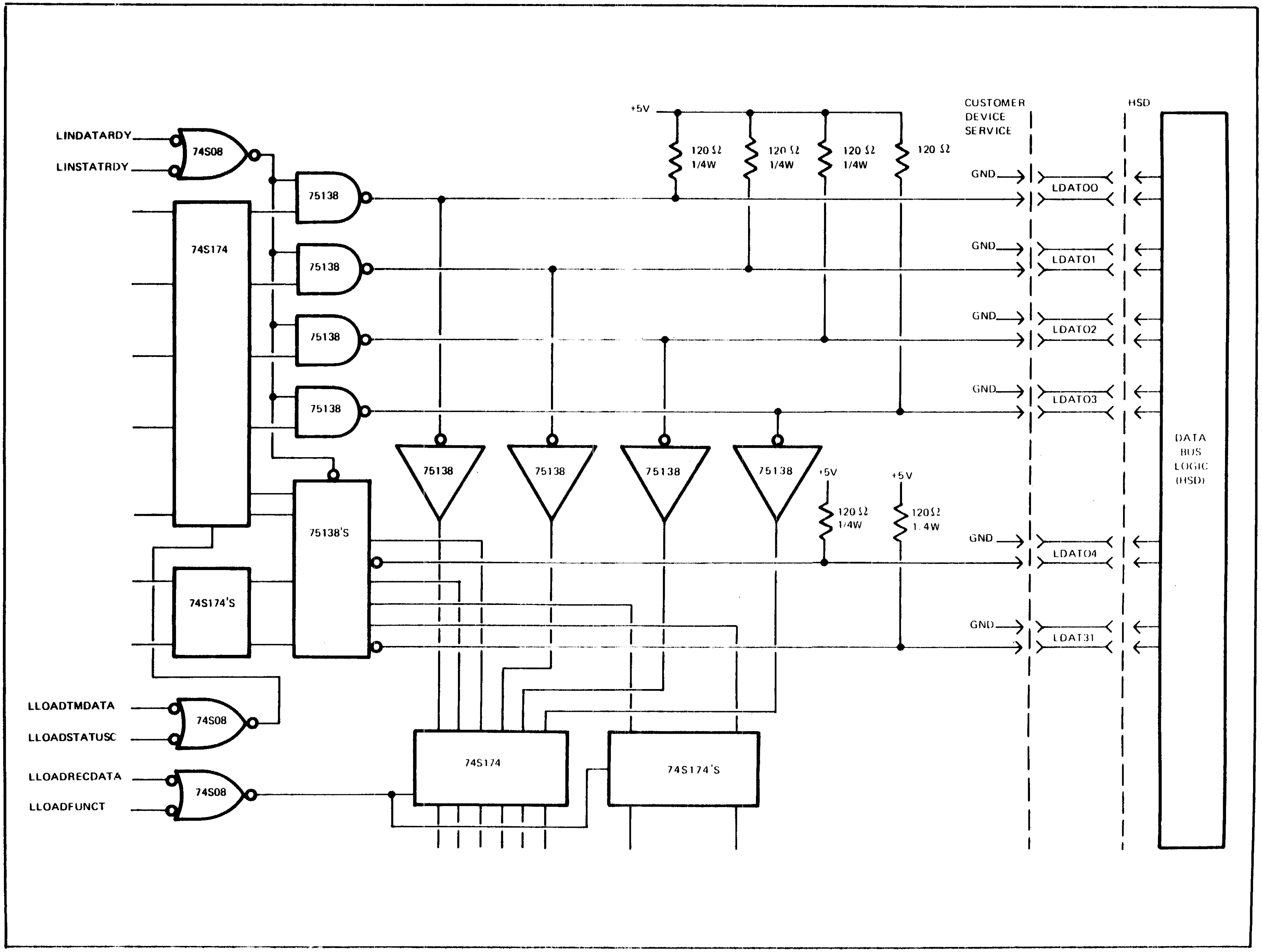


Figure 3-28. HSD/Customer Device Interface Data Bus Logic (Sheet 2 of 2)

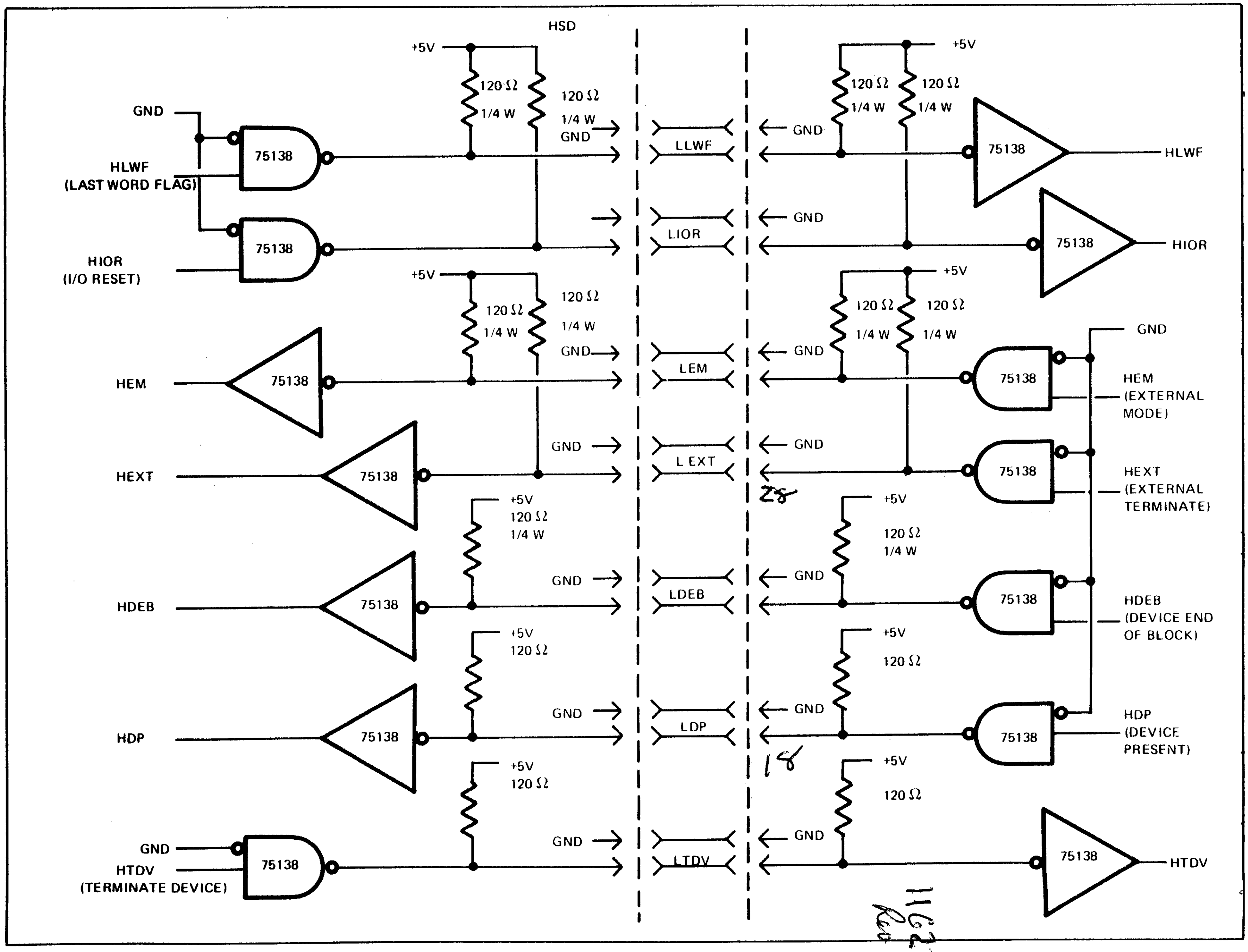


Figure 3-29. HSD/Customer Device Interface Control Signals Logic

116274-050
Rev A

The customer interface equipment must include high-frequency decoupling capacitors in the handshake interface logic. There should be at least one 0.01 microfarad ceramic capacitor for every two chips in the interface logic. A suggested capacitor for this function is Gould's part number 254-101192-010.

3.3.5.4 Interface Cable Definition

The HSD is designed to communicate with the customer equipment by two 50-conductor flat cables. Tests using 22- and 24-gauge twisted pair conductors have been made; although mismatching does occur, it does not seem to effect the 75138 receiver. Therefore, it appears feasible to use twisted pair cables in lieu of flat cable, if desired.

The two cables are plugged into headers on the HSD cable adapter board, which, in turn, plugs into the backplane P1A and P1C connector pins (refer to Table 3-13). The adapter board will accept either the standard amp-latch flat cable connector or the amp mod-4 twisted pair connector.

If flat cable is desired, the customer end of the cable should use the following connector:

Amp Latch	Amp Part No.	Gould's Part No.
Housing	86872-4	259-200008-010
Cover	86972-4	259-200008-011

If twisted pair is desired, the customer end of the cable may use any commercially available connector capable of terminating 22- or 24-gauge twisted pair conductors. Gould S.E.L., suggests one of these standard connector types or their equivalents, depending on customer connector mating requirements:

		Amp Part No.	Gould's Part No.
1.	Amp Mod-4	Amp Part No.	Gould's Part No.
	Body	2-96177-1	259-500034-020
	Contact	86016-1	259-900042-002
2.	Amp M Series	Amp Part No.	
	Pin Block	200277-2	
	Pin	661073-3	

Appropriate hood and jack screw combination.

3. SAE Series 70000 Edgeboard Connector 25/50.

3.3.6 External Control of the HSD

3.3.6.1 External Mode Initialization

Should the customer desire to control data transfers to and from memory by the HSD, he must assign one of his device-dependent operation codes to the external mode enable function. Execution of a command transfer I/O instruction with the external mode enable bit set should cause the customer controller to react by raising the external mode interface control line. The

Table 3-13
Interface Signal Pin List

Cable A		Cable C	
Backplane Connector Pin	Signal Name	Backplane Connector Pin	Signal Name
1	GND	1	GND
2	LDAT00	2	LDAT24
3	GND	3	GND
4	LDAT01	4	LDAT25
5	GND	5	GND
6	LDAT02	6	LDAT26
7	GND	7	GND
8	LDAT03	8	LDAT27
9	GND	9	GND
10	LDAT04	10	LDAT28
11	GND	11	GND
12	LDAT05	12	LDAT29
13	GND	13	GND
14	LDAT06	14	LDAT29
15	GND	15	GND
16	LDAT07	16	LDAT31
17	GND	17	GND
18	LDAT08	18	LDP
19	GND	19	GND
20	LDAT09	20	LTDV
21	GND	21	GND
22	LDAT10	22	LLWF
23	GND	23	GND
24	LDAT11	24	LDEB
25	GND	25	GND
26	LDAT12	26	LIOR
27	GND	27	GND
28	LDAT13	28	LEXT
29	GND	29	GND
30	LDAT14	30	LEM
31	GND	31	GND
32	LDAT15	32	LIDR
33	GND	33	GND
34	LDAT16	34	LIA
35	GND	35	GND
36	LDAT17	36	LISR
37	GND	37	GND
38	LDAT18	38	LISA
39	GND	39	GND
40	LDAT19	40	LEF
41	GND	41	GND
42	LDAT20	42	LEFA
43	GND	43	GND
44	LDAT21	44	LODR
45	GND	45	GND
46	LDAT22	46	LOA
47	GND	47	GND
48	LDAT23	48	UNUSED
49	GND	49	GND
50	UNUSED	50	UNUSED

external mode control line may be raised any time after the reception of the device command information; however, the input status ready (ISR) line, indicating the first memory address is on the interface bus, must not be raised until the device has responded with an external function acknowledge (EFA) to the second device-dependent command word constituting the command transfer I/O instruction. This ensures that the HSD is ready to accept the memory address for the first external mode transfer. No further action is required by the CPU other than the initial command transfer I/O instruction.

3.3.6.2 Execution

After the initialization function has been executed, external mode transfers occur by the following procedure:

1. The external mode (EM) interface line must remain valid during all externally controlled data transfers.
2. Upon receiving the EM level, the HSD will expect an ISR control line from the device. Reception of this signal indicates that the data bus contains the memory address (bits 8 through 31) and an input/output designator (bit 0). The HSD loads the address into the memory address register and returns an ISA to the device. The I/O bit is saved in the control logic to determine the HSD action with respect to memory access.
3. If the I/O bit specifies input, the device will raise an IDR, and the HSD will gate the contents of the data bus into the data buffer file and return an IA to the device.
4. The HSD will then initiate a poll for the SelBUS to write the data into memory at the location specified by the contents of the memory address register.
5. If the I/O bit specifies output, the HSD will initiate a poll for the SelBUS to read data from the memory location specified by the memory address register. A DRT returns the data to the HSD data buffer from memory.
6. The HSD will then initiate an ODR to the device. The device will load the data into its storage register and return an OA to the HSD to complete the transaction.
7. Upon completion of each data transfer, the device will again raise its ISR line to begin the next transfer. On output, the device must not raise the ISR line until it has acknowledged the receipt of the last data word with an OA. On input, the device may not raise its ISR line until the reception of the IA from the HSD signifying the acceptance of the device data.

3.3.6.3 External Mode Errors

Should a parity error or a nonpresent memory error occur during an external mode operation, the HSD will send a terminate device (TDV) signal to the device controller. This signal should be used to clear the external mode condition in the device controller. The controller may respond with an external terminate (EXT) to the HSD which will cause both error status and SI status to be stored and an interrupt to be generated. This feature may be used to inform the CPU that the external mode has been terminated.

3.3.6.4 CPU Ramifications

Upon completion of the last transfer in the external mode, the device must remove the EM line to return control to the CPU. The CPU is unaware of data transfers made in the external mode. If a macrolevel CD or TD is issued while the HSD is in the external mode, the program violation status flip-flop is set, and the CD or TD is ignored.

If the software knowing that an external mode data transfer has occurred is desirable, the device may generate an external terminate (EXT) before removing the EM line. This causes error status to be posted in the current IOCB word 3, SI status to be posted in the TI dedicated location, and an interrupt to be generated. The initialization function for the external mode guarantees that the TI dedicated location address and the IOCB address are valid.

3.3.7 Throughput Capabilities

3.3.7.1 Fundamental Throughput Problem

The throughput associated with the HSD cannot be explicitly defined since it is highly dependent on the system configuration, the HSD bus priorities, the actual data rates of the devices attached to the HSD, the cable lengths, the number of FSDs in the system, and where in memory the data is being processed in relation to where the CPU is executing. To provide some basic guidelines relating to throughput, certain assumptions will be made, and the best guess maximum number of HSDs and their rates will be defined. These numbers must be used only in the context in which they are defined when configuring systems incorporating the HSD.

The HSD cannot sustain its maximum transfer rate of 1.2 microseconds into the same module as the CPU and other I/O devices.

3.3.7.2 Data Rate Guidelines

The data rates described below assume that only MBCs have a higher SelBUS priority than HSDs, and the cable length between the HSD and the device controller is 20 feet to 50 feet long.

1. One of two HSDs may access the same memory module as the CPU and other I/O devices at a maximum rate of 1.2 microseconds or 834 KW each.
2. Up to N (where N is greater than 2) HSDs may access the same memory module as the CPU and other I/O devices if their aggregate rate does not exceed 1.36 MW, and no individual HSD exceeds 834 KW.

   NOTE

   These two situations represent the worst system problem with respect to locking up the CPU and other I/O devices. For example, if two HSDs were simultaneously running at maximum rate into the same memory module from which the CPU was executing instructions, and the number of transfers was 2K words for each HSD, the CPU and other I/O devices will be virtually locked out for 2K x 1.2 microseconds, which equals 2.4 milliseconds. If only one HSD was running, the lockout time would be approximately 800 microseconds or less, depending on I/O activity other than that of the HSD.

3. Two pairs of HSDs (four total) may each access different memory modules than the CPU and other I/O devices at a maximum rate of 1.2 microseconds or 834 KW each (i.e., two HSDs may access memory module 0 at 834 KW each, two may access memory module 1 at 834 KW each, and the CPU and other I/O devices may access memory module 2). The individual rate (834 KW) remains the same even if each of the four HSDs accesses a different memory module (i.e., HSD 1 accesses memory module 0, HSD 2 accesses memory module 1, HSD 3 accesses memory module 2, HSD 4 accesses memory module 3, and the CPU and other I/O devices access memory module 4).

4. Up to N (N is greater than 2) HSDs may access a different memory module than the CPU and other I/O devices at an aggregate rate not to exceed 1.36 MW.

3.3.8 Data Rates with Data Chaining and Function 2 Selected

When data chaining has been invoked in the IOCB operation code, the maximum sustained data rates are defined below, assuming that only MBCs have a higher SelBUS priority than HSDs, the cable length between the HSD and the device controller is 20 feet to 50 feet long, and there is no SI status posting or interrupt generated at the end of the current IOCB which specifies data chaining.

1. One HSD may access the same memory module as the CPU and other I/O devices at a maximum rate of 1.2 microseconds or 834 KW.

2. Up to N (N is greater than 1) HSDs may access the same memory module as the CPU and other I/O devices such that the first HSD operates at 834 KW, and the aggregate rate of the second to the Nth HSD must be 500 KW. If SI status must be posted and an SI generated, the aggregate rate of the second to the Nth HSD must be 215 KW; but the first HSD may continue to operate at 834 KW.

3. For HSDs accessing different memory modules than the CPU and other I/O devices, the throughput rates are the same as those defined in items 3 and 4 of paragraph 3.3.8.

3.3.9 Data Rates with Data Chaining and Function 1 Selected (Function 2 Not Selected)

When operating with function 1 selected, the HSD is not capable of sustaining an uninterrupted data rate, while fetching the next IOCB, of greater than 76.9 kilowords. The aggregate rate remains at 834 kilowords if the transfer block size is 64 words or greater and the interface is running on a system with a memory speed of 600 nanoseconds per word.

If a sustained rate of 834 kilowords is required, function 2 operation should be selected. Refer to Section II of this manual for a description of the programming differences between function 1 and 2 operation.

NOTE

> All measurements given assume the HSDs are running in a system using 600-nanosecond core memory exclusively. When run on systems using 900-nanosecond core memory, 900-nanosecond MOS memory, or a mixture of 900-nanosecond and 600-nanosecond memories, these figures do not apply. In general, the HSD's data rates are dependent on the total throughput load of the system, including the customer supplied external device or devices.

3.3.10 Data Rate for External Mode Operation

When the HSD is operating in the external mode, the maximum throughput rate is 256 KW.

3.3.11 Data Rates for Cable Lengths Greater than 50 Feet

For cable lengths greater than 50 feet between the HSD and the device, the throughput is a function of the cable length itself. Table 3-14 defines the maximum rate for a single HSD accessing the same memory module as the CPU and other I/O devices. These rates may be substituted for previously defined rates when cable lengths are greater than 50 feet. The new aggregate rate associated with previously defined aggregate rates is also listed.

Table 3-14
Cable Length Versus HSD Maximum Rate

Cable Length (Feet)	Maximum Rate	Aggregate Rate
50	834 KW	1.36 MW
100	799 KW	1.29 MW
150	689 KW	1.08 MW
200	606 KW	912 KW
250	540 KW	780 KW

SECTION IV

HIGH-SPEED INTER-BUS LINK (IBL)

4.1 Introduction

This section contains or references the information necessary for the installation, operation, and maintenance of the high-speed data interface configured for a high-speed inter-bus link (IBL) mode.

The information in this section is presented in the following order:

1. IBL mode
2. IBL interface signal definition
3. Interface electrical characteristics
4. Interface cable definition
5. Interface signal pin list
6. IBL software protocol
7. Throughput rate

Two 50-conductor cables interconnect two high-speed data interfaces to make up the high-speed inter-bus link (IBL); therefore, operation and maintenance personnel should be thoroughly familiar with Sections I through III. This section will only cover the modifications to the high-speed data interface (HSD) needed to configure the HSD for high-speed inter-bus link operation.

4.2 High-Speed Inter-Bus Link (IBL) Mode

The high-speed inter-bus link (IBL) mode is enabled by installing a jumper from B20-8 to B20-9 and B20-7 to B20-10 on the high-speed data interface board, as indicated on logic drawing (sheet 21). The jumper must be installed on two HSD boards, and the appropriate C cable must be installed between the two boards before the IBL function can be performed. Jumper B20-10 to 7 is the overflow inhibit jumper and should be installed to inhibit overflow when in the IBL mode.

To allow priority to be resolved should both computers request to transmit data simultaneously, one HSD board must have the high priority jumper installed between C20-8 and C20-9 (sheet 15).

Once the HSD boards are configured as an IBL, they can not be operated as an HSD since it is impossible to connect a customer device to an HSD which is configured as an IBL.

4.3 IBL Interface Signal Definition

A basic block diagram of the IBL interconnection is shown in Figure 4-1. The control signals are basically the same as those described in the previous sections, except that they are not connected differently, and some signals are not required for IBL operation.

A definition of each control signal and its intended use is described in Table 4-1. Note that HSD A and HSD B are used to distinquish the HSDs which make up the high-speed inter-bus link.

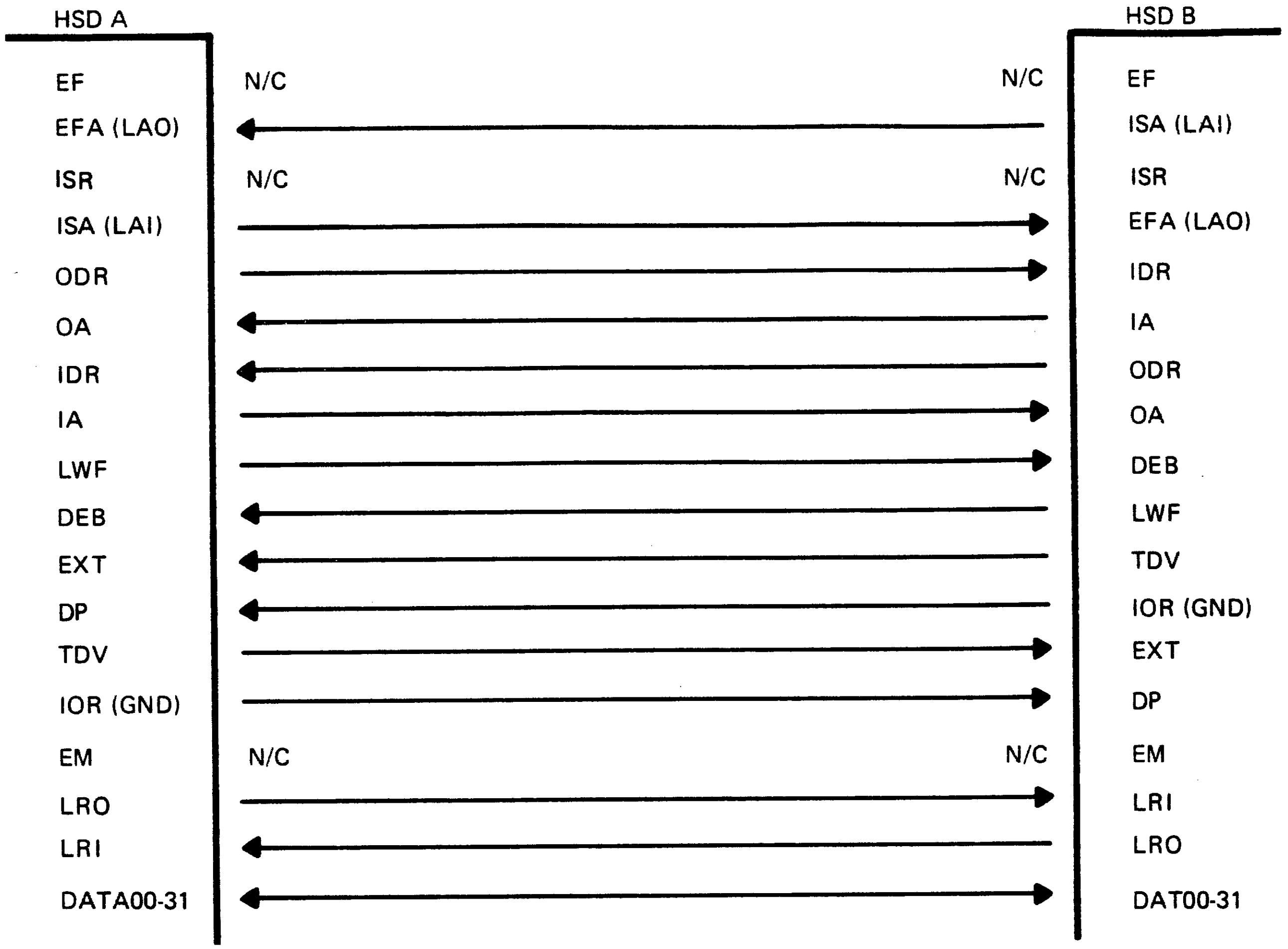


Figure 4-1. High-Speed Inter-Bus Link Signal Interconnection

Table 4-1
IBL Interface Signal Definitions

Signal	Definition
IDR	Level signal sent to HSD A (or B) with the input data. This level remains true until the IA signal is returned by HSD A (or B).
IA	Level signal from HSD A (or B) acknowledging receipt of the input data. This level remains true until IDR drops.
ODR	Level signal sent from HSD A (or B) with the output data. This level remains true until OA is returned by HSD B (or A).
OA	Level signal applied to HSD A (or B) acknowledging receipt of the output data. This level remains true until ODR drops.
LWF	Flag accompanying the last ODR in the output mode or the last IA in the input mode. Sent from HSD A to HSD B or from HSD B to HSD A signifying end-of-block. The HSD which sends LWF will not try to send more data in the output mode or expect to receive more data in the input mode.
DEB	Flag accompanying the last IDR in the input mode or the last IA in the output mode. Sent from HSD A to HSD B or from HSD B to HSD A signifying end-of-block. The HSD which receives DEB will not try to send more data in the output mode or expect to receive more data in the input mode.
EXT	Asynchronous signal applied to HSD A (or B) to indicate an unusual terminate condition in HSD B (or A). The HSD receiving the EXT signal will not attempt to send or receive more data. EXT causes storage of the residual transfer count and the SI status and causes an interrupt to be generated.
DP	Level signal applied to HSD A (or B) indicating that HSD B (or A) is connected and on line.
TDV	Asynchronous signal sent to HSD A (or B) to indicate an unusual terminate condition in HSD B (or A). The HSD receiving this signal should not attempt to send or receive more data. The HSD sending this signal will store its normal error and SI status; thus, indicating the condition which caused the TDV.
IOR	Level (gnd) sent to HSD A (or B) indicating that HSD B (or A) is connected and on line.
LRO	Level signal sent from HSD A (or B) indicating a desire to transfer data. The level remains true until LAO is returned by HSD B (or A). LRO may also be reset by the firmware in the sending HSD.
LAO	Level signal received by HSD A (or B) acknowledging receipt of the LRO signal. LAO remains true until LRO drops.

Table 4-1
IBL Interface Signal Definitions (Cont.)

Signal	Definition
LRI	Level signal received by HSD A (or B) indicating a desire by HSD B (or A) to tranfer data. The level remains true until LAI is returned by HSD B (or A).
LAI	Level signal sent from HSD A (or B) acknowledging receipt of the LRI signal. The level remains true until LRI drops.

NOTE: The roles of sending and receiving HSD dynamically change from block transfer to block transfer. The sending HSD is always the HSD which successfully sends a data request and receives an acknowledgement from the other HSD.

4.4 Interface Electrical Characteristics

Since the logic at both ends of the IBL cables is the actual HSD interface driving and receiving circuitry, the HSD interface is defined in logic drawing 130-103071 applies. A logic one is defined as zero volt to +1.5 volts and a logic zero is defined as +3.2 volts to +5 volts.

4.5 Interface Cable Definition

Two 50-conductor cables interconnect the two HSDs to make up the IBL. The cables are plugged into headers located on a cable adapter board which plugs into the backplane A and C connectors.

The A cable may be either a flat or twisted pair cable. The C cable must consist of 22- or 24-gauge twisted pair conductors mated to the Mod 4 twisted pair connector as indicated below:

Amp Mod 4	Amp Part No.	Gould's Part No.
Body	2-86177-1	259-500034-020
Contact	86016-1	259-900041-002

Table 4-2 lists the signal pin connections for the A cable, and Table 4-3 lists the signal pin connections for HSD A and B on each end of the C cable.

4.6 IBL Software Protocol

4.6.1 Protocol Definition

Any computer (A) wishing to transmit data to another computer (B) by the high-speed inter-bus link (IBL) must perform the following:

1. Computer A sets up an input/output command clock (IOCB) to output data. Computer A must activate the HSD interrupt level before the command device start I/O (CD SIO) is issued. A deactivate interrupt (DAI) instruction may follow the CD SIO instruction.

2. The output SIO causes a hardware link request to be generated and sent to computer B. This causes a service interrupt (SI) at the HSD's interrupt level in computer B.

3. That an interrupt occurred, and that no status is posted in bits 0 through 7 of the TI dedicated location, is interpreted by the software in computer B as a legitimate link request by computer A.

4. Computer B now sets up an IOCB to input data and uses a word count of maximum value (or a value greater than or equal to computer A's word count). The IOCB also specifies an interrupt to be generated upon completion of the transfer.

5. At this point, the calling computer (A) is performing an output, and the answering computer (B) is performing an input. When computer A's last word leaves the FIFO buffer, and if data chaining was not specified, the last word flag (LWF) signal is generated and appears as a device end-of-block (DEB) signal to computer B.

6. If the IBLs are not data chaining or command chaining, both computers will now logically terminate their respective I/O operations, and both will cause interrupts to occur if the interrupt on end of block bit is set in the command.

The roles of the calling and answering computer dynamically change from block transfer to block transfer. The calling computer is always the computer which successfully sends a link request and receives an acknowledgement from the other computer. (In this example, computer A is designated the calling computer.)

The calling computer must always perform its CD SIO with the interrupt level of the HSD active. The answering computer must never perform its CD SIO while its HSD interrupt level is active.

When in the IBL mode, an answering HSD must perform its start I/O command with the interrupt level of the HSD deactivated. The following procedure is recommended to satify this requirement after a link request interrupt has occurred.

The interrupt routine should contain the following:

- Restore all registers
- Activate the attention interrupt level (13)
- Deactivate the HSDs interrupt level
- Issue the start I/O (SIO) command
- Issue the branch and reset interrupt (BRI) indirectly through the return address

This procedure will prevent a higher level interrupt from taking place between the SIO command and the BRI instruction.

Simultaneous requests by both computers are handled by the firmware in conjunction with a jumper which assigns high priority to one end of the link. High priority may only be assigned to one of the two HSDs constituting an IBL, and this designation is only used to resolve simultaneous requests to become the calling computer for a given block transfer. The high priority jumper is located a C20, pins 8 and 9. This jumper is installed to select high priority.

4.6.2 Software Interrupt Receiver Requirements

Successful operation of the IBL requires that the HSD interrupt receiver be handled properly. The interrupt receiver must test for the following occurrences:

1. Was the interrupt caused by an SIO input being completed? The following conditions are present.

   a. SI status reveals proper termination (refer to Service Interrupt (SI) Status Format in Section II).

   b. Last SIO attempted was an input.

2. Was the interrupt caused by the computer at the other end of the link requesting to send data? The following conditions are present.

   a. SI status bits 0 through 7 are not changed.

   b. Last SIO attempted was an output.

Table 4-2
Interface Signal Pin List Cable A

Backplane Connector Pin	Signal
1	GND
2	LDAT00
3	GND
4	LDAT01
5	GND
6	LDAT02
7	GND
8	LDAT03
9	GND
10	LDAT04
11	GND
12	LDAT05
13	GND
14	LDAT06
15	GND
16	LDAT07
17	GND
18	LDAT08
19	GND
20	LDAT09
21	GND
22	LDAT10
23	GND
24	LDAT11
25	GND
26	LDAT12
27	GND
28	LDAT13
29	GND
30	LDAT14
31	GND
32	LDAT15
33	GND
34	LDAT16
35	GND
36	LDAT17
37	GND
38	LDAT18
39	GND
40	LDAT19
41	GND
42	LDAT20
43	GND
44	LDAT21
45	GND
46	LDAT22
47	GND
48	LDAT23
49	GND
50	USUSED

Table 4-3
Interface Signal Pin List Cable B

HSD A Backplane Connector Pin	Signal	HSD B Backplane Connector Pin	Signal
1	GND	1	GND
2	LDAT24	2	LDAT24
3	GND	3	GND
4	LDAT25	4	LDAT25
5	GND	5	GND
6	LDAT26	6	LDAT26
7	GND	7	GND
8	LDAT27	8	LDAT27
9	GND	9	GND
10	LDAT28	10	LDAT28
11	GND	11	GND
12	LDAT29	12	LDAT29
13	GND	13	GND
14	LDAT30	14	LDAT30
15	GND	15	GND
16	LDAT31	16	LDAT31
17	GND	25	GND
18	LDP	26	LIOR
19	GND	27	GND
20	LTDV	28	LEXT
21	GND	23	GND
22	LLWF	24	LDEB
23	GND	21	GND
24	LDEB	22	LLWF
25	GND	17	GND
26	LIOR	18	LDP
27	GND	19	GND
28	LEXT	20	LTDV
29	N/C	29	N/C
30	N/C	30	N/C
31	GND	43	GND
32	LIDR	44	LODR
33	GND	45	GND
34	LIA	46	LOA
35	N/C	35	N/C
36	N/C	36	N/C
37	GND	41	GND
38	LISA	42	LEFA
39	N/C	39	N/C
40	N/C	40	N/C
41	GND	37	GND
42	LEFA	38	LISA
43	GND	31	GND
44	LODR	32	LIDR
45	GND	33	GND
46	LOA	34	LIA
47	GND	49	GND
48	LLRO	50	LLRI
49	GND	47	GND
50	LLRI	48	LLRO

3. Was the interrupt caused by an SIO output being completed? The following conditions are present.

   a. SI status reveals proper termination (same as 1.a.).

   b. Last SIO attempted was an output.

4. Was the interrupt caused by an error condition in either an input or output transfer? The following conditions are present.

   a. SI status reveals improper termination; i.e., ERROR bit set (1.a.)

   b. Error status word defines error condition (see Error Status Format in Figure 2-8).

   c. Last SIO attempted was an input/output.

5. Was the interrupt caused by a CD terminate from the lost interrupt processor? The following conditions are present.

   a. SI status reveals improper termination; i.e., error bit set (1.a.).

   b. Error status word contains zero.

   c. Last CD issued was terminate.

   d. There are specific conditions under which a CD terminate will post no status. These conditions are described under CD terminate execution and CD terminate error conditions in Section III.

4.6.3 Estimated Time to Establish the Link

Certain fundamental functions must be performed by the software in both computers before the data is actually tranferred. They are:

1. Calling computer must issue CD SIO to output.

2. Answering computer must be interrupted; i.e., store, place, and branch (SPB).

3. Answering computer must execute interrupt subroutine, issue a CD SIO to input data, and exit interrupt subroutine.

These basic functions appear to introduce (depending on the length of the interrupt routine) from 100 to 150 microseconds overhead before transferring data.

4.7 Throughput Rate

Throughput rates will vary, depending on the activity of the memory module into which data is being transferred. The activity in turn is a function of system configuration, SelBUS priorities, cable length, and program execution. The optimum transfer rate occurs when both the calling computer and answering computer are moving data from/to memory modules other than those the program and other I/O devices are accessing. This rate is a maximum of 1.33 MW/sec or 5.33 MB/sec over 50 feet of cable. Proportionately slower rates will occur for longer cable lengths. This rate does not include software overhead.

If the calling computer outputs data from a memory module other than the one the program and other I/O devices are accessing, and the answering computer inputs data to the same memory module as the program is accessing, the rate is ≈ 1.12 MW/sec or 4.48 MB/sec over 50 feet of cable. This rate will be proportionately slower if other I/O devices are accessing the same memory module as the CPU and the IBL controller.