TECHNICAL MANUAL

# MULTIPLEXING INPUT/OUTPUT PROCESSOR MODELS 8271/8471 AND 8272/8472

SCIENTIFIC DATA SYSTEMS

SDS 901515A
$5.25

TECHNICAL MANUAL

# MULTIPLEXING INPUT/OUTPUT PROCESSOR

## MODELS 8271/8471 AND 8272/8472

May 1968

SCIENTIFIC DATA SYSTEMS • 1649 Seventeenth Street • Santa Monica, Calif. • (213) 871-0960

# LIST OF EFFECTIVE PAGES

Total number of pages is 156, as follows:

| Page No. | Issue | Page No. | Issue |
|---|---|---|---|
| Title | Original | | |
| A | Original | | |
| i thru iv | Original | | |
| 1-1 thru 1-2 | Original | | |
| 2-1 thru 2-2 | Original | | |
| 3-1 thru 3-142 | Original | | |
| 4-1 thru 4-4 | Original | | |

# CONTENTS

CONTENTS (Cont.)



TABLES



ILLUSTRATIONS

ILLUSTRATIONS (Cont.)

# RELATED PUBLICATIONS

| Publication Title | Publication No. |
|---|---|
| SDS Sigma 5 Computer, Reference Manual | 900959 |
| SDS Sigma 7 Computer, Reference Manual | 900950 |
| SDS Sigma Computer Systems Interface Design Manual | 900973 |
| SDS Sigma 7 Computer, Technical Manual | 901060 |
| SDS Sigma 5 Computer, Technical Manual | 901172 |
| SDS Sigma 5 and 7 Systems Test Monitor | 901076 |
| SDS Sigma 5 and 7 Buffered Line Printer System Test | 901085 |
| SDS Sigma 5 and 7 Keyboard-Printer System Test | 901086 |
| SDS Sigma 5 and 7 Medium-Speed RAD File System Test | 901090 |
| SDS Sigma 5 and 7 9-Channel Magnetic Tape System Test | 901110 |
| SDS Sigma 5 and 7 Card Punch System Test | 901120 |
| SDS Sigma 5 and 7 Card Reader System Test | 901121 |
| SDS Sigma 5 and 7 Paper Tape Reader/ Punch System Test | 901122 |
| SDS Sigma 7 Multiplexing IOP Test | 901126 |

# SECTION I
# GENERAL DESCRIPTION

## 1-1 INTRODUCTION

This manual describes SDS Multiplexing Input/Output Processor (MIOP) Models 8471 and 8271 and optional subchannels, Models 8472 and 8272. The manual consists of four sections that provide general information, programming information, a functional description, maintenance information, and parts lists.

The MIOP provides independent control of data transfers between core memory and certain peripheral devices, and starts, tests, and acknowledges interrupts pertaining to certain peripherals under control of a Sigma 5 or 7 central processing unit.

Figure 1-1 shows the physical layout of the basic MIOP, Models 8471 (Sigma 7) and 8271 (Sigma 5), and the optional subchannels, Models 8472 (Sigma 7) and 8272 (Sigma 5).

Technical manuals describing equipment associated with the MIOP are referred to in the list of Related Publications in the front matter of this manual.

## 1-2 PHYSICAL DESCRIPTION

The basic MIOP consists of 98 modules installed in chassis A, B, C, and D, and includes eight subchannels. Each subchannel accommodates one device controller. Five fast-access memory modules (FT25) provide eight subchannels. Since one fifth of each subchannel is contained on each of the five FT25's, the FT25's must be installed in groups of five. Three additional groups of FT25's provide a total of 32 subchannels.

## 1-3 FUNCTIONAL DESCRIPTION

The MIOP contains input and output data storage registers and buffers, fast-access memory registers for command manipulation, a timing signal generator, and control logic. The function of the MIOP is to control and sequence input and output operations for a number of peripheral devices simultaneously, allowing the CPU to concentrate on program execution. The active devices time-share the hardware in the MIOP. Any input-output events that require CPU intervention are brought to the attention of the CPU by means of the interrupt system. The device controllers and devices are described in other manuals.

## 1-4 SPECIFICATIONS AND LEADING PARTICULARS

The general specifications for the MIOP are given in table 1-1.

Table 1-1. General Specifications

| | |
|---|---|
| Power requirements (supplied by PT16 power supply) | +8Vdc (9.0 amps)<br>-8Vdc (2.4 amps)<br>+4Vdc (20 amps)<br>Total watts: 171 |
| Logic signal levels | One, +4Vdc; Zero, 0v (low impedance to ground) |
| Data format | 8-bit byte, 32-bit word |
| Temperature | |
| Nonoperating: | -40°C to 60°C (-40°F to 140°F) |
| Operating: | 5°C to 50°C (41°F to 122°F) |
| Relative humidity (operating) | 10% to 95% |
| Altitude | |
| Nonoperating: | 20,000 feet maximum |
| Operating: | 10,000 feet maximum |

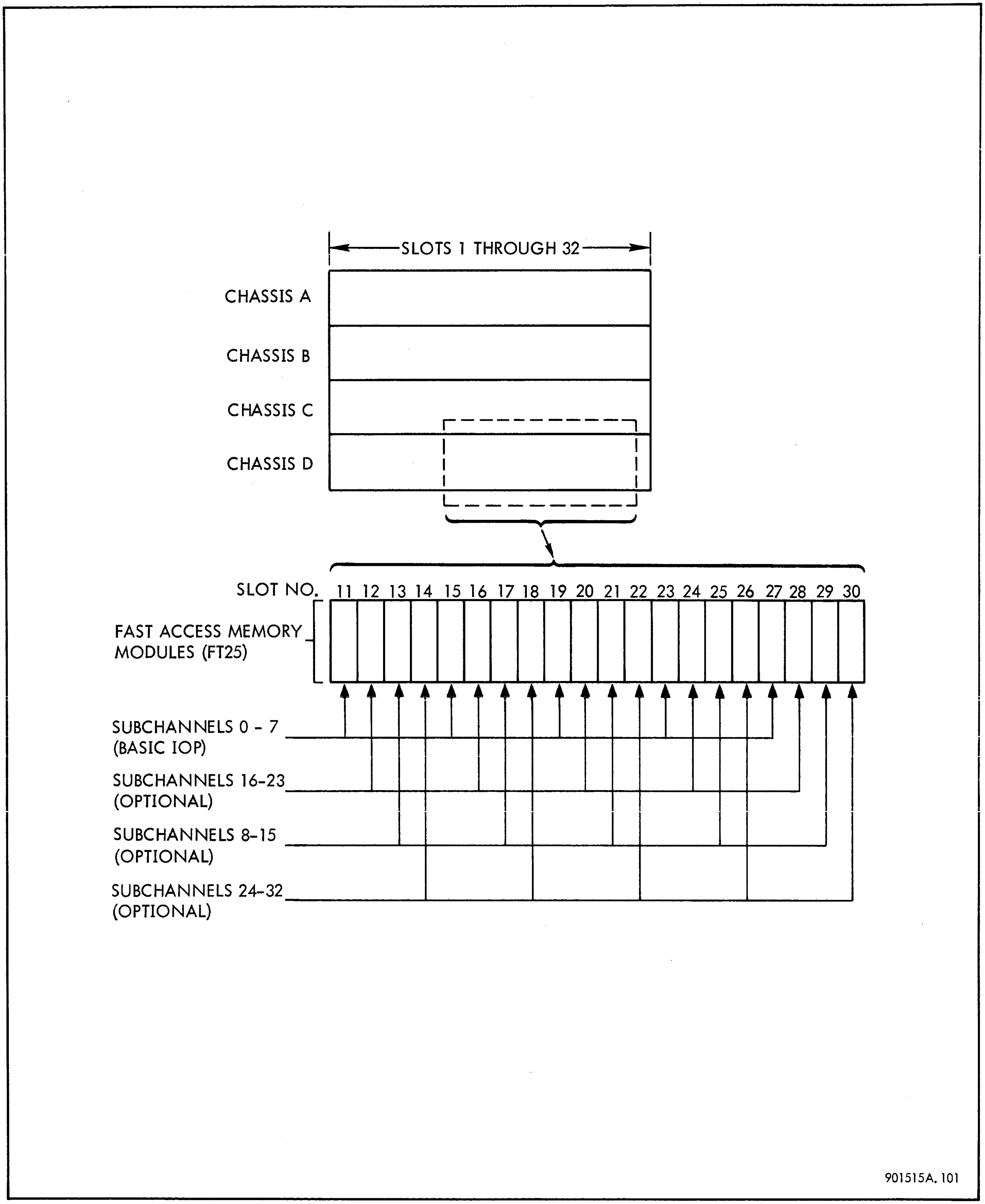



Figure 1-1. MIOP Basic and Optional Subchannel Locations

# SECTION II
# OPERATION AND PROGRAMMING

## 2-1 GENERAL

The MIOP contains no controls or indicators other than address switches and switch LASTONE, which is closed on all MIOPs connected to a CPU except the last one. These switches are contained on the switch comparator module (LT26) in slot 13 of chassis C. The module contains eight switches; the three address switches that apply to the MIOP are S1-1, S2-1, and S3-1, and the switch that applies to signal LASTONE is S1-2.

In the Sigma 5 and 7 I/O system, the CPU executes instructions, the MIOP executes commands, and the I/O device executes orders. For example, the CPU may execute a start input/output (SIO) instruction to initiate an I/O operation. During the course of the operation, the MIOP fetches a command doubleword (command) from the command list in core memory and stores it (except the order bits) in its own fast access memory. The command provides the MIOP with information needed to perform its functions; commands are, therefore, executed by the MIOP. The order bits of the command are transmitted to the device. The order defines the operation to be performed by the device; the I/O devices, therefore, execute orders.

For a description of I/O instructions, commands, and orders, see SDS Sigma 7 Computer Reference Manual 900950 and SDS Sigma 5 Computer Reference Manual 900959.

# SECTION III
# PRINCIPLES OF OPERATION

## 3-1 INTRODUCTION

This section describes the principles of operation of the MIOP on a general information level and on a detailed level in terms of the logic equations. The detailed description includes a description of each IOP register, a typical I/O operation, the interface signals, timing generation, interface signals, IOP phase sequences, and a glossary of logic signals. There is a phase sequence description for each CPU-initiated I/O instruction and for each of the four service cycles. Each phase sequence description includes a table that lists every logic operation that occurs in the MIOP relating to the specific instruction or service cycle, starting with the first timing signal of the first phase.

## 3-2 GENERAL PRINCIPLES OF OPERATION

### 3-3 GENERAL

The maximum number of devices that can be uniquely addressed by a computer with one MIOP installed is 152. Up to eight MIOP's can be connected to one computer. The basic MIOP is mechanized with eight subchannels that accommodate eight device controllers. Because of the addressing structure of the I/O instructions, only these first eight subchannels can be used for servicing multiunit device controllers. The multiunit device controllers are capable of controlling up to 16 devices each. When the multiunit device controllers are used, the subchannel is shared by all the devices controlled by that device controller. Each subchannel contains all the information necessary to control any I/O operation between core memory and the device.

Once an I/O operation has been started by the CPU, the operation is performed to completion by the MIOP, device controller, and core memory without intervention by the CPU. Timing between these units is asynchronous. The MIOP processes the I/O operation while the CPU is performing functions possibly unrelated to the I/O operation. The MIOP controls the I/O operations by executing a command list prepared by the CPU and stored in core memory (figure 3-1).

### 3-4 OVERALL OPERATION

An I/O operation starts when the CPU issues a start input/output (SIO) instruction addressed to a particular MIOP and device controller. The addressed MIOP, after receiving the address information, places the device controller address on the MIOP/device controller interface lines and waits for a response. The addressed device controller responds by sending condition code and status information to the MIOP.

The MIOP sends the condition code information to the CPU and, depending upon the coding of the SIO instruction, may or may not send status and other information related to the MIOP, device, and device controller to the CPU. The SIO instruction, if successful, causes the addressed device controller to make service requests from the MIOP. As a result of the service requests (after the SIO instruction has been concluded), the device controller electrically connects itself to the MIOP/device controller interface lines. The time the device controller is electrically connected is called a service cycle. It is during the service cycles that follow an SIO instruction that data is exchanged between the device controller and core memory. During these service cycles the MIOP performs core memory accesses as required by the device, continually updates information such as byte address and byte count, and controls the I/O operation until it is completed, aborted, or halted by a halt input/output (HIO) instruction.

During execution of the I/O instructions, core memory locations, X'20' and X'21' (hexadecimal 20 and 21), are used to exchange information between the MIOP and CPU. These are in addition to the MIOP/CPU interface lines. During the early phases (CPU phases) of an SIO instruction, the CPU writes the address of the device/device controller, the address of the first command doubleword, and the nature of the R field into core memory location X'20' (see figure 3-2).

The function code (SIO) and MIOP address are sent directly to the MIOP on the MIOP/CPU interface lines. The addressed MIOP then reads core memory location X'20', stores the address of the command doubleword in its internal registers, and places the device controller address on the MIOP/DC interface lines. After the addressed device controller responds by sending status and condition code information to the MIOP, the MIOP loads the status and other information (if the CPU so specifies by the coding of the R field of the instruction) into core memory location X'20' and X'21', where it is available to the CPU. The other information consists of information previously stored in the subchannel for the addressed device controller. The condition code is sent directly to the CPU.

At the conclusion of a successful SIO instruction, the device controller starts making service requests to the MIOP. The first service request, which will be for an order out, causes the MIOP to access the command list in core memory for the first command doubleword. The address of the command doubleword was obtained from the CPU during the SIO instruction. The order that is encoded in the command doubleword is sent to the device controller so that it will know what

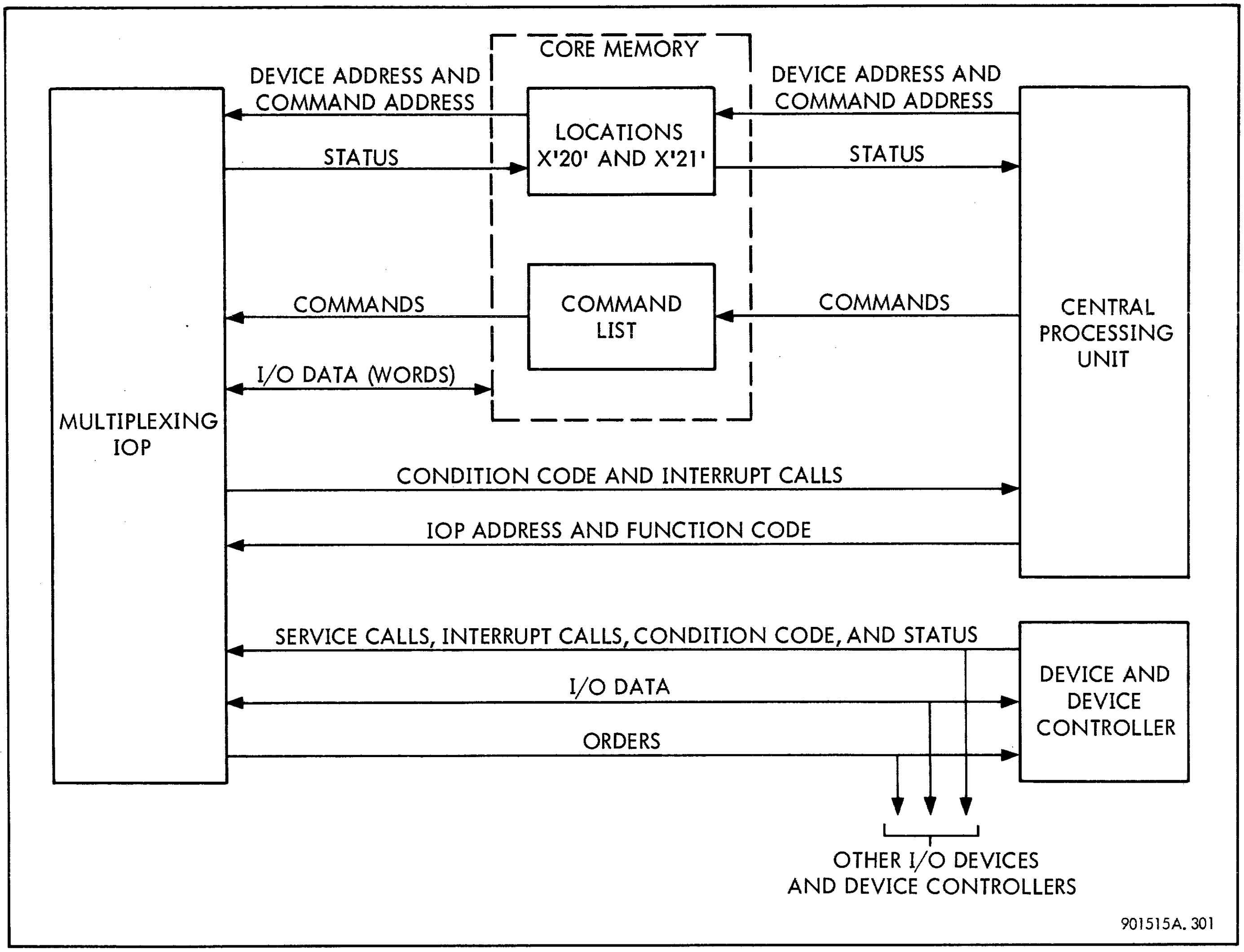



Figure 3-1. I/O System, Simplified Overall Block Diagram

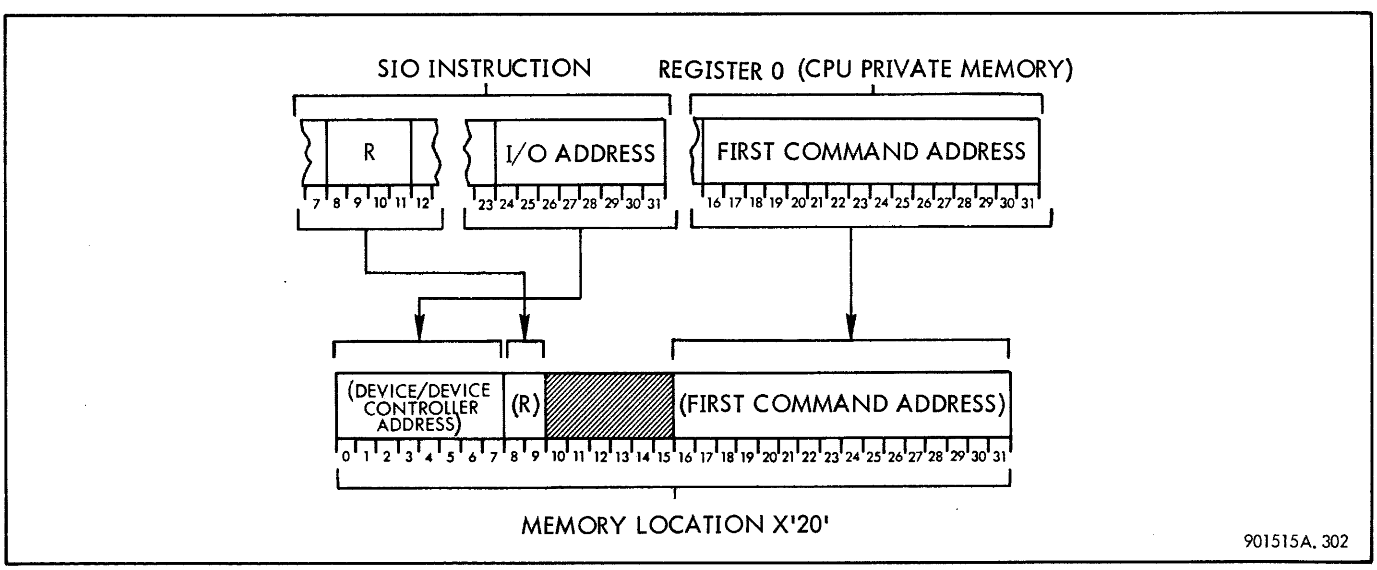



Figure 3-2. Loading Core Memory Location X'20' During an SIO Instruction

function to perform. The balance of the command doubleword is related to the I/O operation and is retained by the MIOP. This information directs the operations of the MIOP until the I/O operation is concluded. The exchange of I/O data takes place after the device controller has received the order.

As a result of subsequent service requests from the device controller, data is exchanged between core memory and the device through the MIOP. The exchange of data between core memory and the MIOP is on a word basis and between the IOP and the device controller on a byte basis. A maximum of four bytes of data may be exchanged between the MIOP and device controller for each service request. For example, if the order the device received initially was a write order, the operation would be an output operation. In this case, as a result of a service request from the device controller, the MIOP would access core memory for one word of data and store it in an MIOP register. The word is then fed to the device controller one byte at a time. If the order the device received was a read order, the operation would be an input operation. During an input operation, the MIOP accepts data from the device controller one byte at a time and stores it until it has a maximum of four bytes. The four bytes (one word) are then stored in core memory by the MIOP. In addition to transferring I/O data, the MIOP keeps track of the number of bytes of data transmitted and their core memory locations, records status for future interrogations by the CPU, checks parity, and performs other operations as required.

### 3-5 Interrupt Calls

Interrupt calls made by a device controller are passed along to the CPU by the MIOP. One standard interrupt level is provided to service all interrupts generated by all devices connected to all MIOP's controlled by a CPU. In response to an interrupt call, the CPU issues an acknowledge input/output interrupt (AIO) instruction. The primary purpose of the AIO is to determine the address of the interrupting IOP and device controller. The highest priority device controller with an interrupt pending sends its address (along with status and condition code information) to the MIOP. The MIOP writes its own address, the device controller address, and status information in memory location X'20' and sends the condition code information to the CPU. The CPU then reads memory location X'20' to acquire the address and status, and takes action based on the status and condition code information it has just received.

### 3-6 Chaining

Chaining is the term applied to the operation that permits an activity to continue after the functions specified by the current command have been completed. The MIOP employs both data chaining and command chaining. Both types of chaining are controlled by a flag setting of the current command and result in a new command being fetched by the MIOP when all data specified by the current command has been transferred. Each new command fetched by the MIOP is the next command in sequence in the command list in core memory.

Data chaining is used for scatter-read or gather-write operations, where the peripheral device is operating with a record of continuous data that may come from, or be delivered to, noncontiguous areas of core memory in subblocks of any size specified by the programmer.

Command chaining provides a means of writing a program to operate on several records without intervention by the CPU. Command chaining causes the MIOP to load a new command at the end of a record, send the new order to the device controller, and start processing the new record.

## 3-7 DETAILED PRINCIPLES OF OPERATION

## 3-8 MIOP REGISTERS

The various data and control registers within the MIOP are shown in figure 3-3. The MIOP address logic and timing is shown as a single block, since these two sections are described separately. For convenience, the formats of the subchannel registers, the data lines, and the function response lines are also included in figure 3-3. With the description of each register is a flow diagram showing the source of all input signals to that register. The transfer signal that enables the input signals to enter the registers is shown in the break of the line connecting the input source to the register.

### 3-9 A-Register

The A-register (see figure 3-4) consists of eight buffer latches, A0 through A7, and normally contains the device/device controller address. During execution of an instruction (except an AIO), the A-register receives the address information from the M-register. During an acknowledge service call (ASC) function and an AIO instruction, the A-register receives the address from a device controller by means of the FR lines. The address in the A-register is decoded by the address decoding logic (SPA3 through SPA7) to select one of the 32 possible subchannels (see figure 3-25). The A-register is cleared to zeros when signal AX0 is true.

### 3-10 C-Register

The C-register (see figure 3-5) consists of 15 buffer latches, C0 through C14, and one buffered latch, flip-flop C15. The C-register is the only input to the adder. The C-register, with the adder, is used as a common point for the distribution of data between various MIOP registers. The C-register is also used with the adder and two buffered latches (K15 and SUB), as a means of incrementing or decrementing a number by one. Input data to the C-register consists of status information which comes from various sources, the M-register, the BA lines, and the BC lines. When signal LS0 is true, the information on the BA lines is the output of the command address (CA) register, and when signal LS0 is false, the information on the BA lines is the output of the byte address (BA) register. When signal LS1 is true, the information on the BC lines is the output of the flags and status (FS) register, and when signal LS1 is false, the information on the BC lines is the output of the byte count (BC) register. The C-register is cleared to zeros when signal CX0 is true.

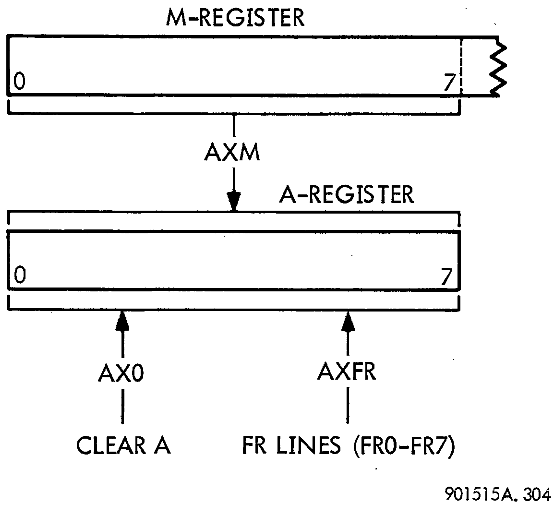



Figure 3-4. A-Register Inputs

3-11 CA-Register

The CA-register (see figure 3-6) is one of the six registers that comprise each of the 32 possible subchannels. Each CA-register consists of 16 fast-access flip-flops, CA0 through CA15. This register holds the current command doubleword plus one. During execution of an instruction (when the current command doubleword is to be sent back to the CPU as part of the response information), the contents of the CA-register are transferred through the adder and decremented by one. During chaining operations, the contents of the CA-register are circulated through the C-register and adder to increment them by one every time a command doubleword is accessed from core memory. The CA-register receives its input data from the adder. The output of the CA-register is available to the C-, M-, and S-registers. Both the input data lines and the output data lines of the CA-register are shared with the BA-register.

The command address (16 bits) is set into the 16 most significant bits (MSB) of the S-register whenever a new command doubleword is to be accessed. The 17th bit, S31, which is the least significant bit (LSB) of the S-register, is controlled separately. This permits the MIOP to access the first word of the command doubleword when S31 is false and to access the second word when it is true.

3-12 BA-Register

The BA-register (see figure 3-6) is one of the six registers that comprise each of the 32 possible subchannels. Each BA-register consists of 16 fast-access flip-flops, BA0 through BA15. This register contains the 16 MSB's of the current byte address. The three LSB's of the byte address are contained in the OF-register (see figure 3-46). Bit position OF2 is the LSB of the byte address and OF0 is the third LSB. The 16 bits (BA0 through BA15) of the BA-register and OF0 constitute a word address in core memory. The two LSB's (OF2 and OF1) define the particular byte of that word. Every time a byte of data is processed the byte address is incremented or decremented as required. After every fourth byte a carry to, or borrow from, the LSB of the word address (OF0) is required. The BA-register contents must, therefore, be updated when a carry to, or borrow from, the three LSB's in the OF-register occurs. Whether the byte address is incremented or decremented depends on the backward flag (BK) stored in the OF-register. This flag is true only if a read backward order was sent to the device. When the device is reading backward, data received from that device is written into descending core memory locations. Therefore, the byte address must be decremented by one with each byte processed. If the BK flag is false, the byte address is incremented by one with each byte processed.

3-13 BC-Register

The BC-register (see figure 3-7) is one of the six registers that comprise each of the 32 possible subchannels. Each BC-register consists of 16 fast-access flip-flops, BC0 through BC15, and contains the current byte count. During a data-in or data-out operation, the byte count is decremented by one by circulating it through the C-register and adder after each byte of data is processed. The BC-register receives its input data from the adder. The output of the BC-register is available to the C-register when decrementing during a data operation and the M-register as part of the response information during execution of an instruction. Both the input data lines and the output data lines of the BC-register are shared with the FS-register. The BC-register is addressed when signal LS1 is false.

3-14 FS-Register

The FS-register (see figure 3-7) is one of the six registers that comprise each of the 32 possible subchannels. Each FS-register consists of 16 fast-access flip-flops, FS0 through FS15. The upper half of the FS-register (bits FS0 through FS7) contains the flags specified by the command doubleword. During an order-out service cycle, the second word of the command doubleword is set into the M-register. During the termination phase of the service cycle, if there are no error conditions detected, the flags from the command doubleword are transferred from the M-register by means of the C-register and adder to the FS-register (see figure 3-38). If error conditions are detected, the old flags are effectively retained by the FS-register. During an order-in service cycle, the old flags are retained by the FS-register.

The lower half of the FS-register (bits FS8 through FS15) contains status information. The status information is updated during the service cycles. Figures 3-38 and 3-40 show the source of status update information during an order-out and order-in service cycle. An OR operation is performed on the new status, acquired during the service cycle, with part of the old status in the FS-register. The flags and status control the operations of the MIOP during the service cycles.

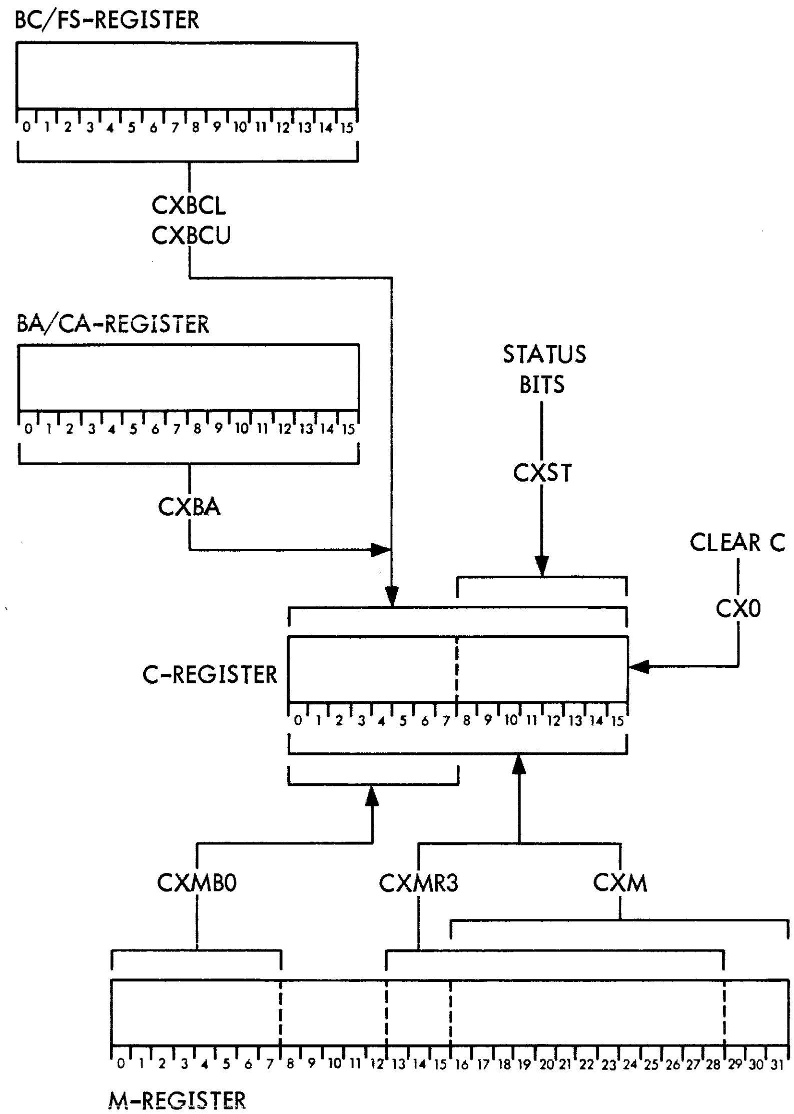



Figure 3-5. C-Register Inputs

Both the input data lines and the output data lines of the FS-register are shared with the BC-register. The FS-register is addressed when signal LS1 is true.

## 3-15 OF-Register

The OF-register (see figure 3-8) is one of six registers that comprise each of the 32 possible subchannels. Each OF-register consists of eight fast-access flip-flops, OF0 through OF7. Bit positions OF0 through OF2 contain the three LSB's of the current byte address. The 16 MSB's of the byte address are contained in the BA-register (see paragraph 3-12). Bit positions OF3 through OF7 contain operating flags. The flags in bits OF3, OF5, and OF6 are duplicates of the flags specified in the command doubleword. Bit OF4 is the backward (BK) flag. This flag is set during an order-out service cycle if the order bits of the command doubleword specify a read backward order. Figure 3-37 shows how the three LSB's of the byte address (bits OF0 through OF2) and the four flags in bits OF3 through OF6 are set. Bit OF7 contains the transmission error halt (TEH) flag. This flag is set during a data-in service cycle if an error condition is detected too late in that service cycle for the MIOP to report error halt to the device by means of a terminal order. In effect, the error is recorded by the TEH flag until the next communication with that device, at which time the error condition may be used to halt the device.

The OF- and IS-registers share the same input data lines and the same output data lines. The OF-register is addressed when LS2 is false.

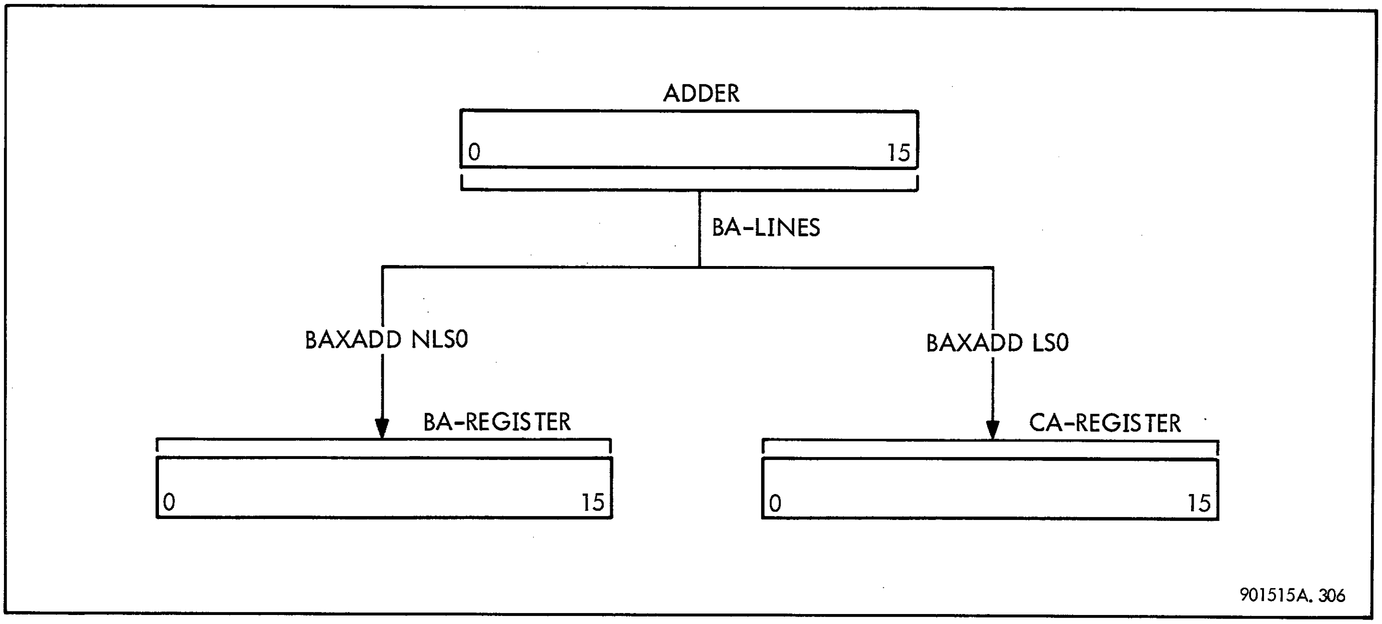
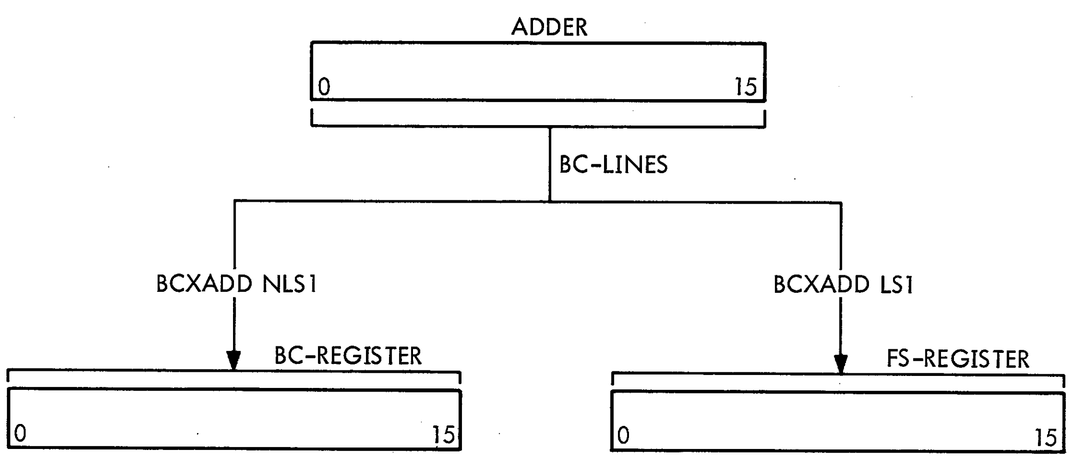



Figure 3-6. BA- and CA-Register Inputs




FS-REGISTER FORMAT

BIT POSITION

0 - DATA CHAIN (DC)
1 - INTERRUPT ON ZERO BYTE COUNT (IZC)
2 - COMMAND CHAIN (CC)
3 - INTERRUPT ON CHANNEL END (ICE)
4 - HALT ON TRANSMISSION ERROR (HTE)
5 - INTERRUPT ON UNUSUAL END (IUE)
6 - SUPPRESS INCORRECT LENGTH (SIL)
7 - SKIP (S)

BIT POSITION

8 - INCORRECT LENGTH (IL)
9 - TRANSMISSION DATA ERROR (TE)
10 - TRANSMISSION MEMORY ERROR (TMP)
11 - MEMORY ADDRESS ERROR (MAE)
12 - IOP MEMORY ERROR (CMP)
13 - IOP CONTROL ERROR (CE)
14 - IOP HALT (IOP UE)
15 - CHAINING MODIFIER (CM)

901515A. 307

Figure 3-7. BC- and FS-Register Inputs

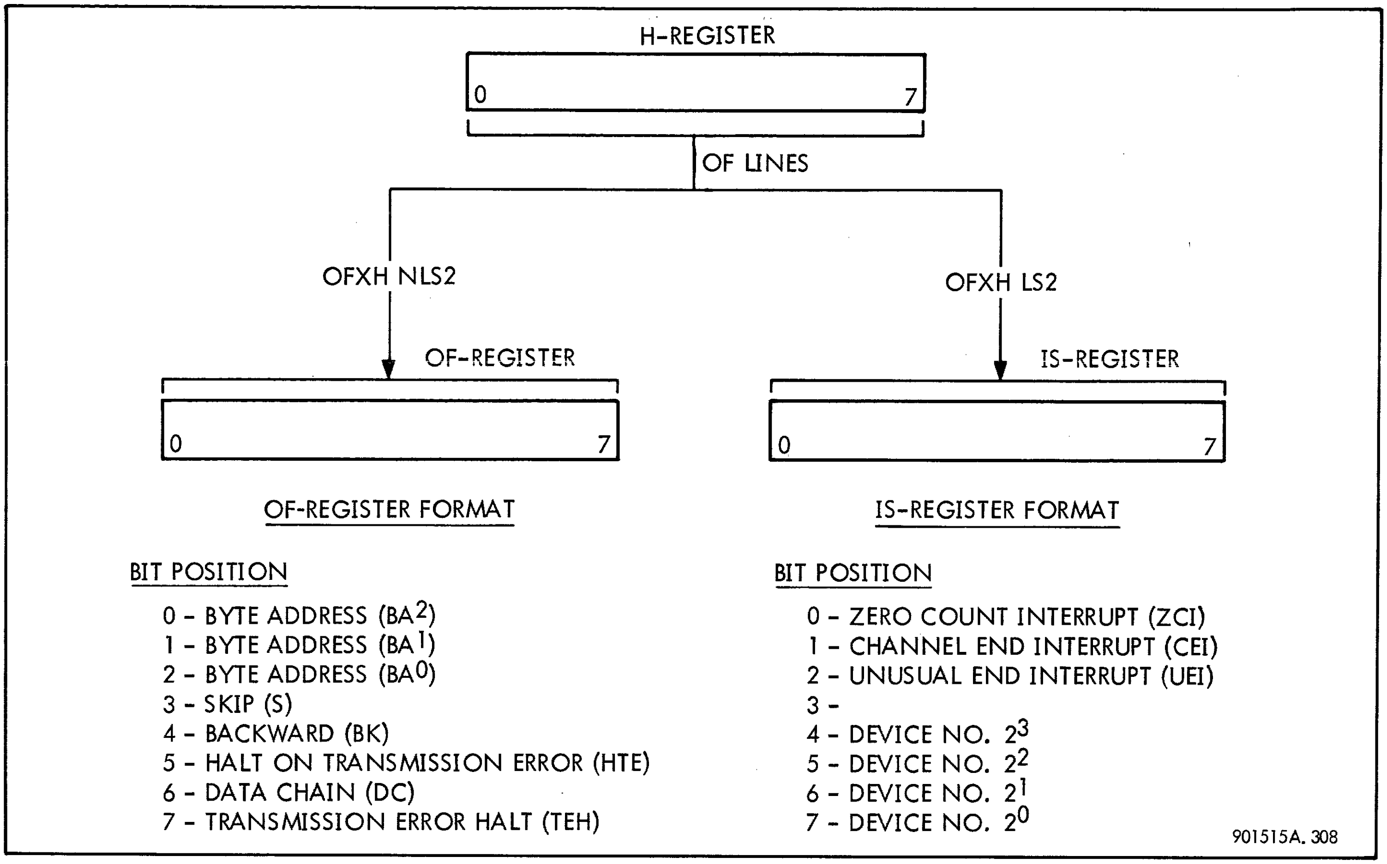



Figure 3-8. OF- and IS-Register Inputs

### 3-16 IS-Register

The IS-register (see figure 3-8) is one of six registers that comprise each of the 32 possible subchannels. Each IS-register consists of eight fast-access flip-flops, IS0 through IS7. Bits IS0 through IS2 contain the interrupt status. These three flags are sent to the CPU as part of the status information during an AIO instruction. The flags are updated as shown in figure 3-41 during an order-in service cycle. Bit IS3 is not used. Bits IS4 through IS7 contain the address of the last successfully started (by means of an SIO) device connected to the device controller associated with this subchannel. This address pertains only to subchannels associated with multiunit device controllers. All devices controlled by a multiunit device controller share the subchannel assigned to that device controller. It is, therefore, possible that the information stored in the subchannel is for a device other than the one requesting an interrupt. During an SIO instruction, the address in the IS-register is compared with the address presently being offered to the MIOP by the interrupting device. If they compare, the information stored in the subchannel is for that device, and can be sent to the CPU.

Both the input data lines and the output data lines are shared by the OF-register. The IS-register is addressed when LS2 is true.

### 3-17 F-Register

The F-register (see figure 3-9) consists of six buffered latch flip-flops, F0 through F5. Flip-flops F0, and F2 through F5 accept decoded information from the three CPU function code lines, FNC0 through FNC2, and provide on a single function indicator line, the function to be performed (SIO, HIO, TIO, TDV, or AIO). Function indicator line ASC is controlled by F1. Flip-flop F1 is set when one or more device controllers drive the service call (SC) line true. The F-register flip-flops are reset by signal FX0. Signal FNT is true during a CPU-initiated function that is addressed to this MIOP.

The function code lines are decoded by the function code decoding logic as follows:

| | Instruction | | | | |
|---|---|---|---|---|---|
| Function Code Line | SIO (F3) | TIO (F4) | TDV (F5) | HIO (F2) | AIO (F0) |
| FNC0 | 0 | 0 | 0 | 0 | 1 |
| FNC1 | 0 | 0 | 1 | 1 | 1 |
| FNC2 | 0 | 1 | 0 | 1 | 0 |

3-18 H-Register

The H-register (see figure 3-10) consists of eight buffered latch flip-flops, H0 through H7. The H-register is used primarily as a temporary storage register for the data in the OF- and IS-registers while the data is operated on. Figure 3-46 shows how the H-register is used with the J-register to increment or decrement the byte address during a data-in service cycle. The H-register is the only input to the OF- and IS-registers.

3-19 J-Register

The J-register (see figure 3-10) consists of three buffered latch flip-flops, J0 through J2. The J-register is used to increment the H-register when signal HXJP1 is true and decrement it when signal HXJM1 is true. Signal HXJM1 is true only when a device is performing a read backward operation. Signal HXJP1 may be true when a device is performing a read (forward) or write operation. The three upper bits of the H-register (H0 through H2) are cleared by signal HUX0, and the five lower bits (H3 through H7) are cleared by signal HX0.

3-20 I-Register

The I-register (see figure 3-11) consists of nine buffer latches, I0 through I7, and IP. The I-register accepts data from the device controller as shown in figure 3-46. Bit IP is also controlled by signal DAP during a data-in service cycle if the device is capable of transmitting a parity bit. If the device is not mechanized to check parity, the device controller drives the parity check (PC) line true. This causes the MIOP to ignore the parity information supplied by the device controller. During an order-in service cycle, the operational status byte from the device controller is set into bits I0 through I7 for subsequent storage in the sub-channel registers by means of the H-register, and the C-register and adder. During a data-in service cycle, the input data in the I-register is transmitted to the M-register for subsequent writing into core memory.

During an output operation (data out or order out), the I-register receives the data or order from core memory by means of the M-register and IMB buffers, and sends it to the O-register where it is available to the data lines and thus the device controller. Figure 3-43 shows the processing of output data. The I-register is cleared by signal IX0.

3-21 O-Register

The O-register (see figure 3-12) is a nine-bit register made up of buffer latches O0 through O7, and OP. The O-register is used only during output operations and during the termination phase of input operations. During an order-out service cycle, bits O0 through O7 receive the order from the I-register and transmit it to the device controller by means of the data lines. The parity bit (OP) is not used in this case. During a data-out service cycle, bits O0 through O7 receive the byte of data from the I-register. If the eight bits of data in the I-register are an even number, signal IEVEN will be true and will cause parity bit OP to be set. The data byte set on the data lines (DA0 through DA7), and the data parity line (DAP) will, therefore, have odd parity.

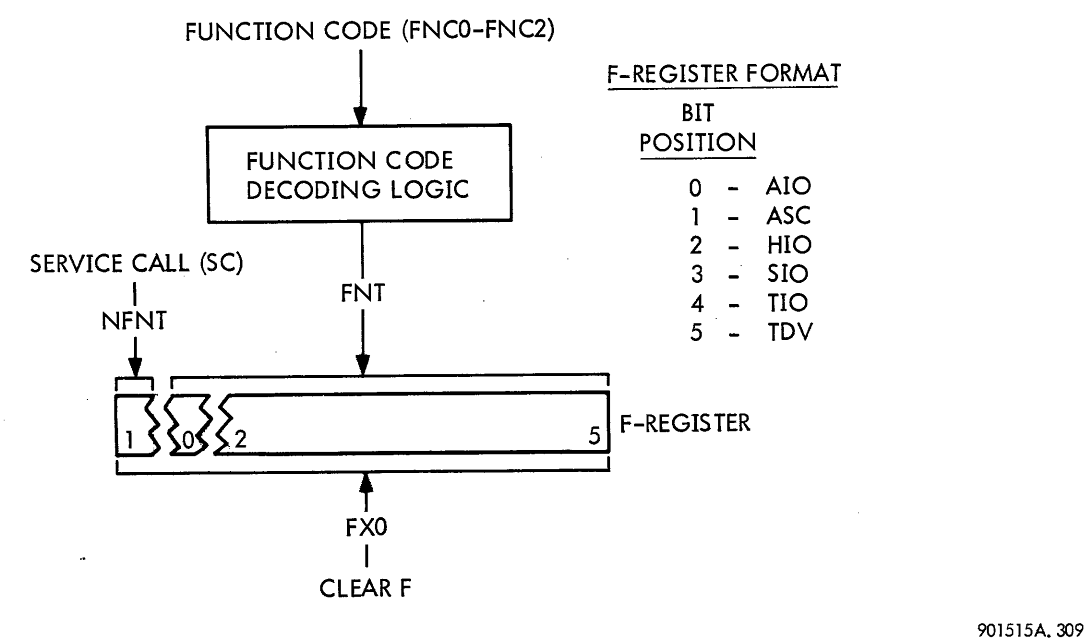



Figure 3-9. F-Register Inputs

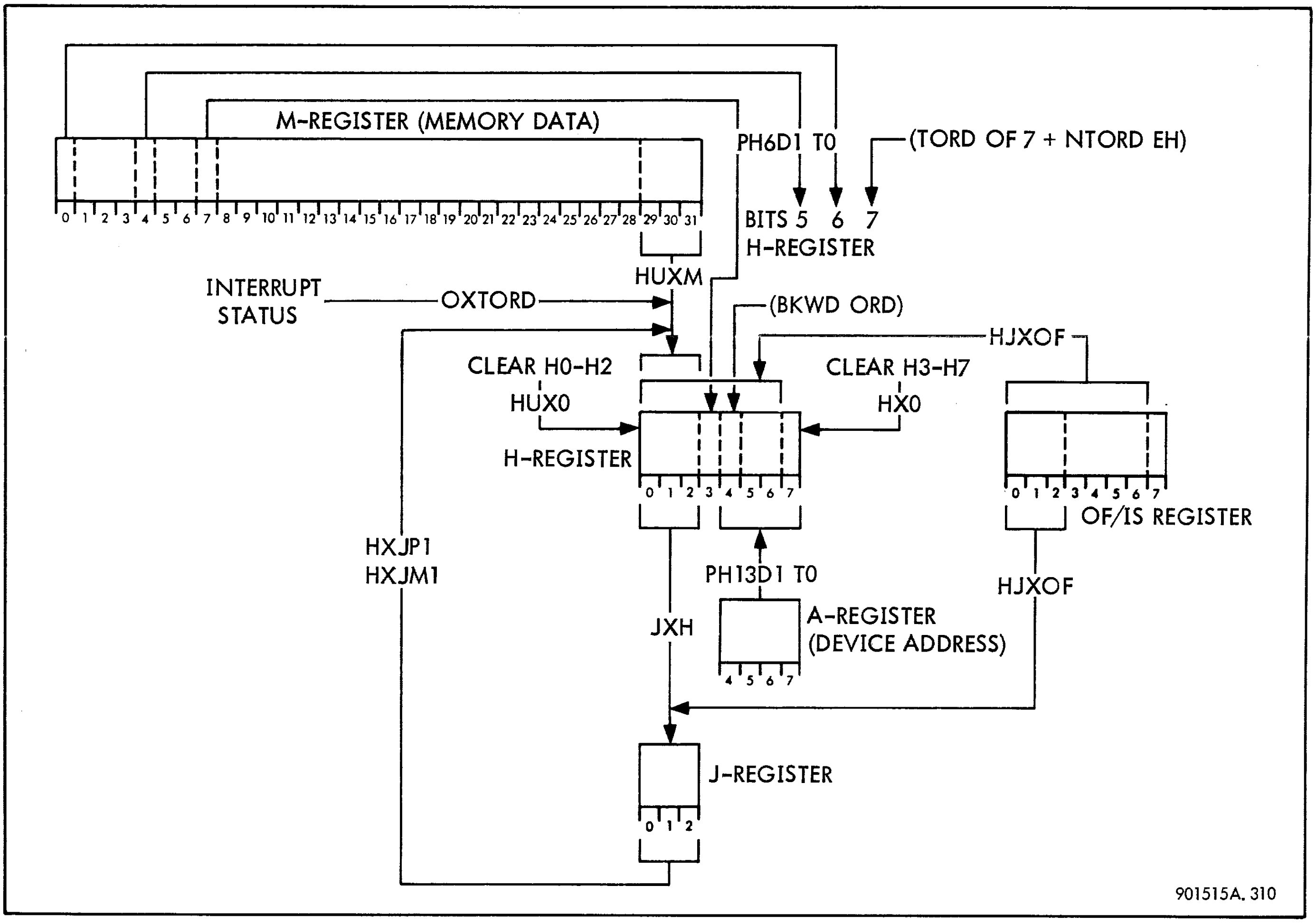



Figure 3-10. H- and J-Register Inputs

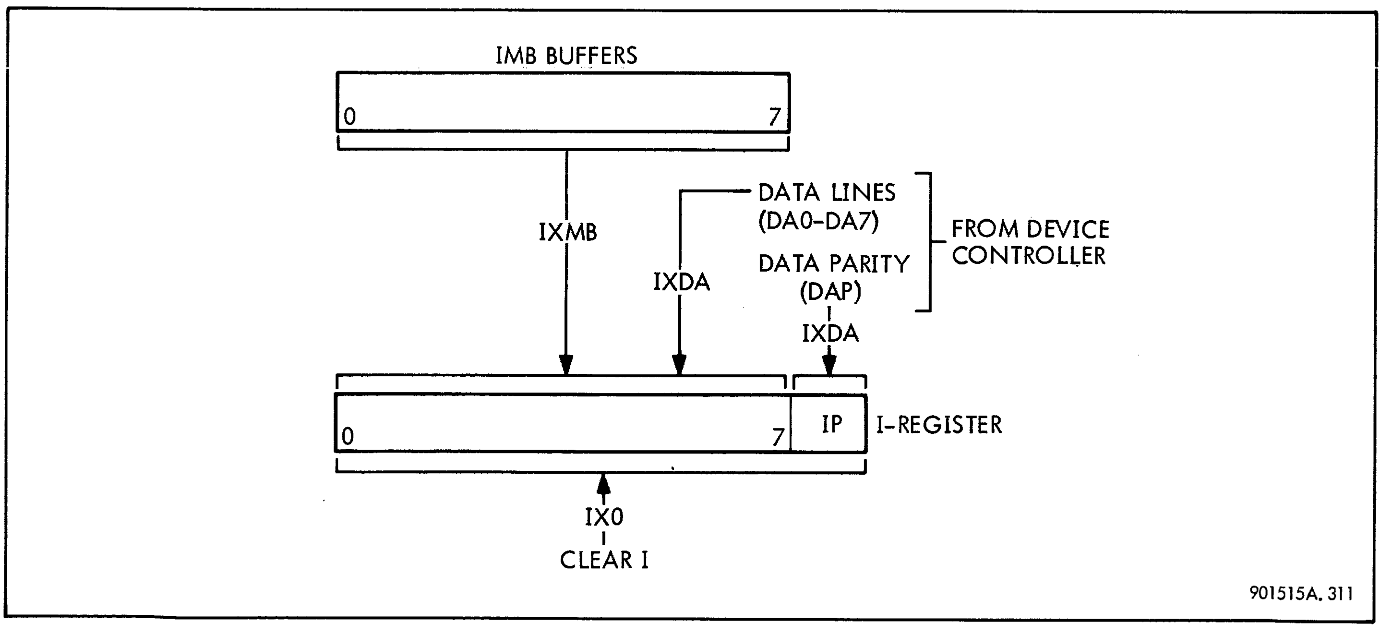



Figure 3-11. I-Register Inputs

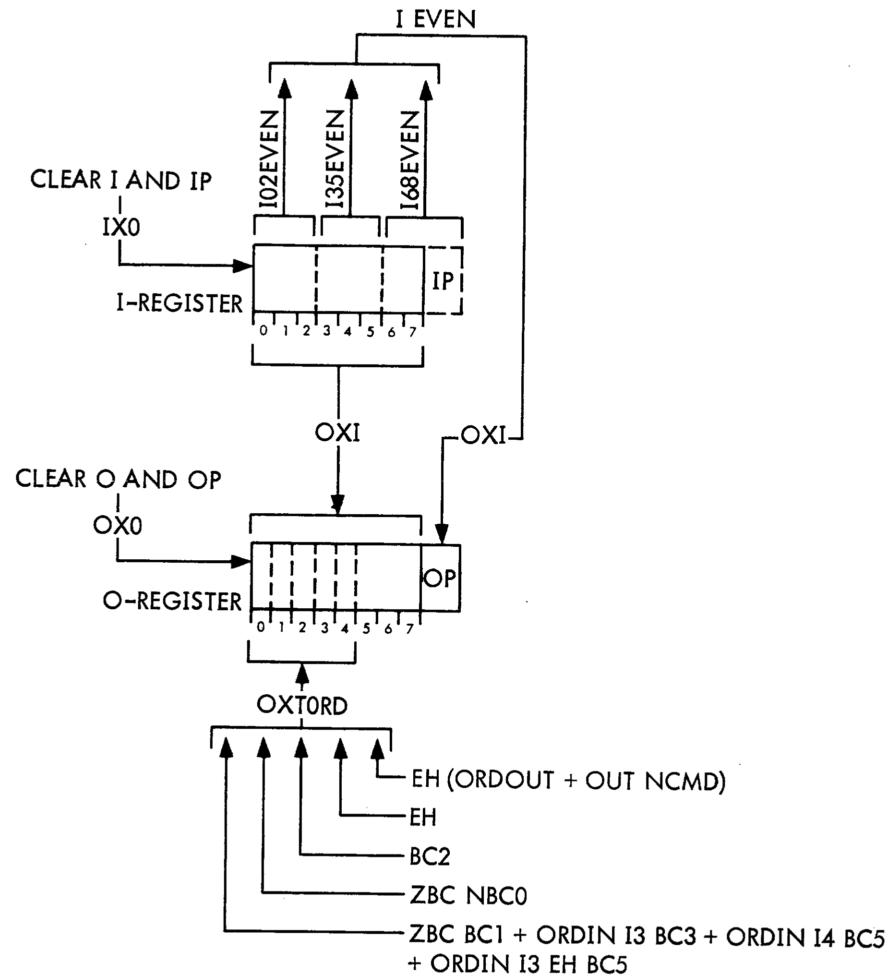



Figure 3-12. O-Register Inputs

During the termination phase of both input and output operations, the terminal order is assembled in bits O0 through O4 of the O-register. Bits O5 through O7 and OP are not used. The O-register is cleared by signal OX0.

3-22 S-Register

The S-register (see figure 3-13) consists of 17 buffer latches, S15 through S31. This register drives the 17 address lines to core memory. Bit S31 is the LSB of the word address and bit S15 is the MSB. During a CPU function, when the MIOP must read core memory location X'20', it clears the S-register and brings up signal SX20. This signal forces a one into the sixth LSB (S26), therefore specifying location X'20'. When the MIOP is required to respond to a CPU function by writing into location X'20' and X'21', it first specifies location X'20' as before, and stores the first word. Then, to store the second word, the MIOP forces a one into the LSB (S31). With S31 and S26 true and the rest of the S-register bits false, location X'21' will be written into when the next memory request is made.

During an order-out service cycle, the MIOP first clears the S-register and then transfers the command doubleword address to bits S15 through S30, leaving bit S31 (the LSB) false. When a memory request is made, the first word of the command doubleword is read out of memory. After the MIOP processes the first word, it forces bit S31 true and makes another memory request. The second word of the command doubleword, stored in an odd-numbered memory location, is therefore read out of memory.

During a data-in or data-out service cycle, the byte address MSB's from the BA-register are transferred into bits S15 through S30. The LSB of the word address, which is the third LSB of the total byte address, is stored in the OF-register. It is transferred to bit S31 by means of the J-register, where it is updated after every fourth byte has been processed, as shown in figure 3-46. (See paragraph 3-12.)

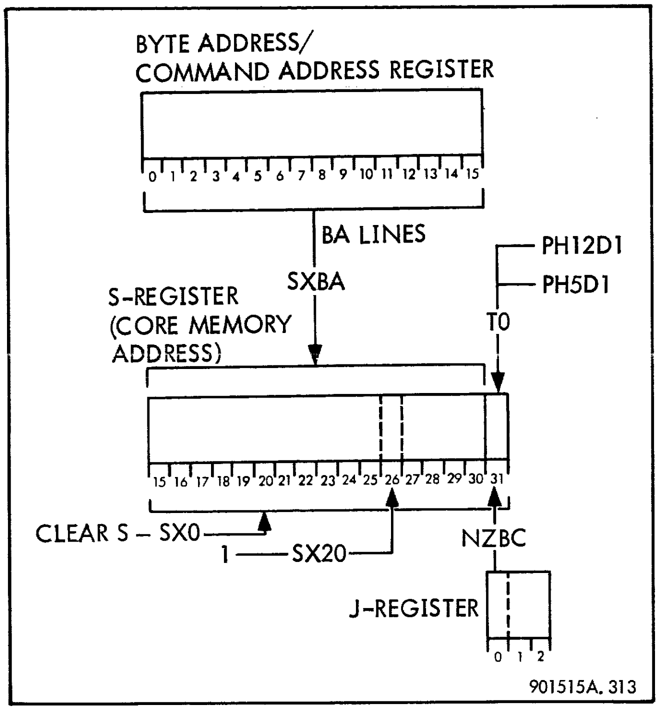



Figure 3-13. S-Register Inputs

3-23 M-Register

The M-register (see figure 3-14) consists of 32 buffer latches, M0 through M31. The M-register receives data from core memory during read operations and also drives the data lines to core memory during memory write operations. When a core memory access is made, the memory places the 32-bit word from the memory location specified by the S-register on the memory data lines. Core memory also drives the data gate line to gate the 32-bit word into the M-register. The MSB of the memory word on data line MX0 is set into bit M0 of the M-register, and the LSB on data line MX31 is set into bit M31 of the M-register.

Data in the M-register that is to be stored in the MIOP fast-access memory registers enters these registers by way of the C-register and adder, and the H-register (see figure 3-37). Data in the M-register that is to be sent to the device controller is placed on the device controller interface lines by way of the IMB buffers, the I-register, and the O-register (see figure 3-43). Data that is accepted from the device controller by the MIOP for subsequent storage into core memory enters the M-register by way of the I-register (see figure 3-46). Data that is stored in the MIOP registers that is to be written into core memory enters the M-register directly and also by way of the H-register, C-register, and adder.

The upper half of the M-register (bits M0 through M15) is cleared by signal MUX0. The lower half (bits M16 through M31) is cleared by signal MLX0.

3-24 W-Register

The W-register (see figure 3-15) consists of four buffer latches, W0 through W3. The W-register drives the write byte indicator lines to core memory. When the MIOP wants to perform a read operation, it clears the W-register by driving signal WX0 true. When a full write operation is to be performed, the MIOP sets the W-register bits to ones. When a partial write operation is to be made, the MIOP sets the appropriate W-register bits by decoding the two LSB's of the byte address as shown in figure 3-46.

3-25 ADDER

The adder (see figure 3-16) is composed of 16 buffer amplifiers, ADD0 through ADD15; 15 buffer amplifiers, K0 through K14, which act as carries; one precarry flip-flop K15; the add/subtract control flip-flop SUB; and several amplifiers and logic gates used primarily as look-ahead circuits for carry propagation.

The C-register is the only input to the adder. The adder performs three functions depending on the state of signals SUB and K15. With SUB true and K15 true, the contents of the C-register are decremented by one; with SUB false and K15 true, the contents of the C-register are incremented by one; with K15 false and SUB true or false, the contents of the C-register appear unchanged at the adder output.

Adder output terms ADD0 through ADD15 perform an exclusive OR operation on the contents of C-register bits C0 through C15 and carry bits K0 through K15 for both increment and decrement operations.

When incrementing the contents of the C-register by one (NSUB K15), the low-order carry bits will be true for all low-order one bits of the C-register and for the first low-order zero bit of the C-register. The remaining carry bits will then be false for all remaining high-order C bits after the first zero bit of C is encountered. See examples 1 and 2.

Example 1

C = 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 0

K = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

ADD = 0 0 0 0 1 1 0 0 1 1 1 1 1 0 0 1

Example 2

C = 0 0 0 0 1 0 0 0 0 0 1 1 1 1 1 1

K = 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1

ADD = 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0

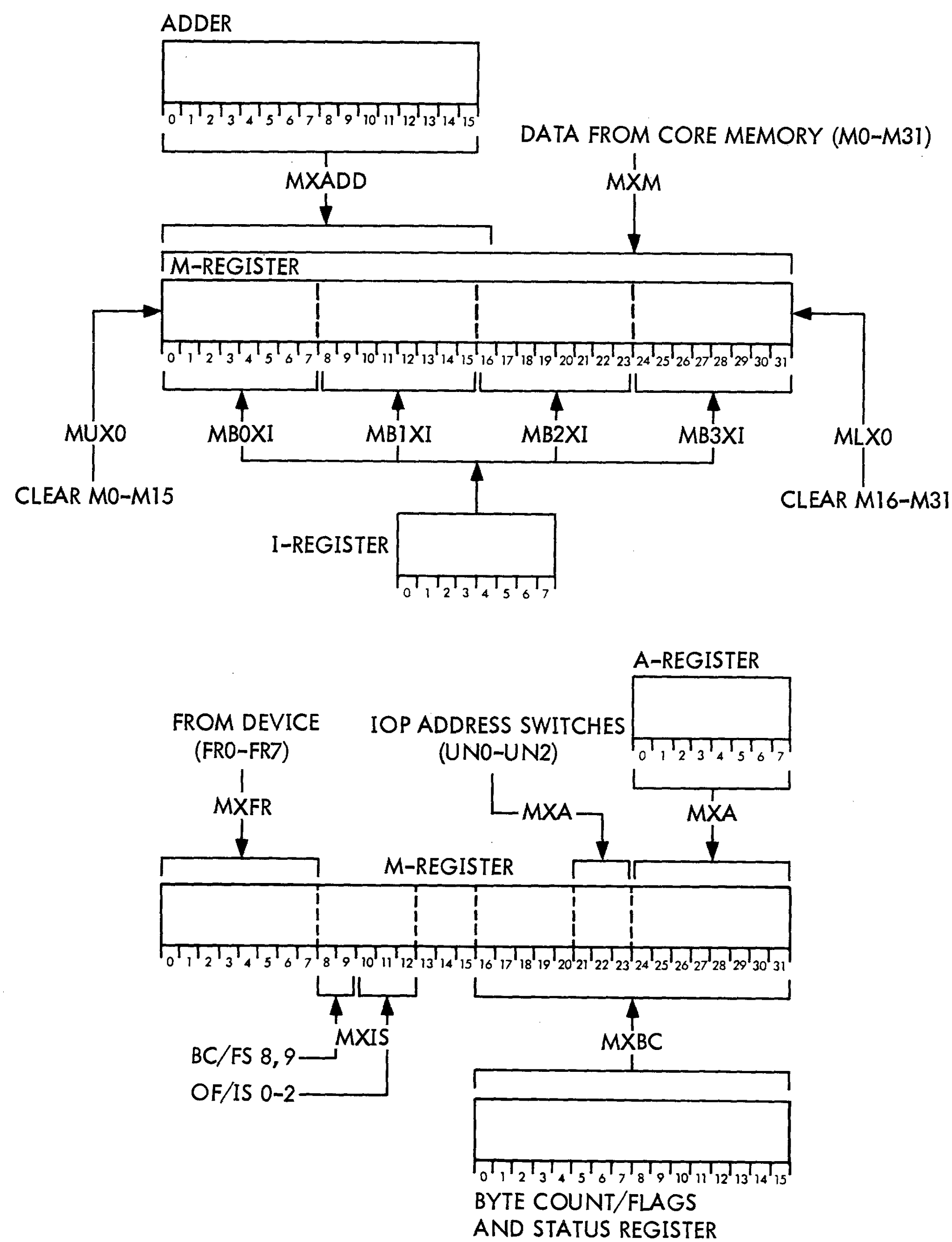



Figure 3-14. M-Register Inputs

When decrementing the contents of the C-register by one (SUB K15), the carry bits are true for the initial low-order zero bits of C and true for the first one-bit encountered. The carry bits will then be false for the remaining high-order bits of C. See examples 3 and 4.

Example 3

C = 0000110011111000
K = 0000000000001111
ADD = 0000110011110111

Example 4

C = 0000000011111111
K = 0000000000000001
ADD = 0000000011111110

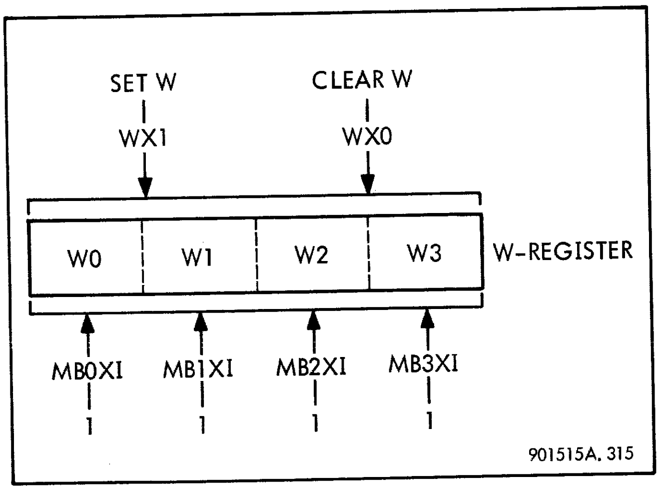



Figure 3-15. W-Register Inputs

## 3-26 TYPICAL I/O OPERATION STARTING WITH AN SIO INSTRUCTION

An I/O operation generally involves a number of communication cycles during which an I/O device is started by an SIO instruction and data is exchanged between the device and core memory. Figure 3-17 is a sequence diagram starting with an SIO instruction followed by service cycles during which an order is sent to the device controller. Following the order service cycle, data is sent to the device controller during the data-out service cycles at a maximum rate of four bytes per service cycle. Figure 3-18 shows the timing of I/O operation, and figure 3-19 is an overall block diagram of the MIOP.

When the CPU issues an SIO instruction to the MIOP, the CPU places the MIOP address on the address lines and the function code on the function code lines. The address (three bits) is compared with the MIOP address to determine if this is the MIOP to which the SIO is addressed. If it is, a delay line in the timing section is started. The delay line generates the timing signals needed to perform the I/O operation.

The function code (three bits) is decoded by the F-register input logic, and results in the SIO function indicator line at the device controller interface being driven true. The function indicator lets the device controller know how to handle information received on the other lines.

As the MIOP progresses to the next phase of the operation, a X'20' is forced into the memory address register (S-register), and a core memory access is made. As a result, the contents of core memory location X'20' are set into the M-register. Previously, the CPU has written into core memory location X'20' the device/device controller address, the nature of the R field, and the command doubleword address. The device/device controller address is transferred from the M-register to the O-register by means of the I-register. Information in the O-register is also present on the data

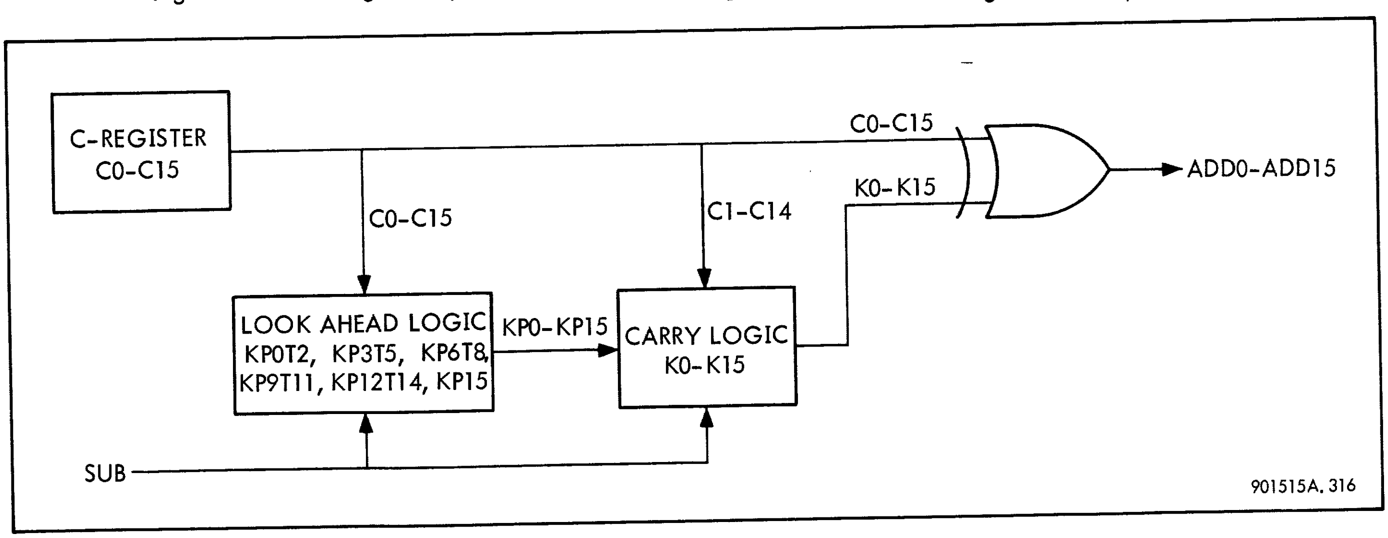



Figure 3-16. Adder, Overall Block Diagram

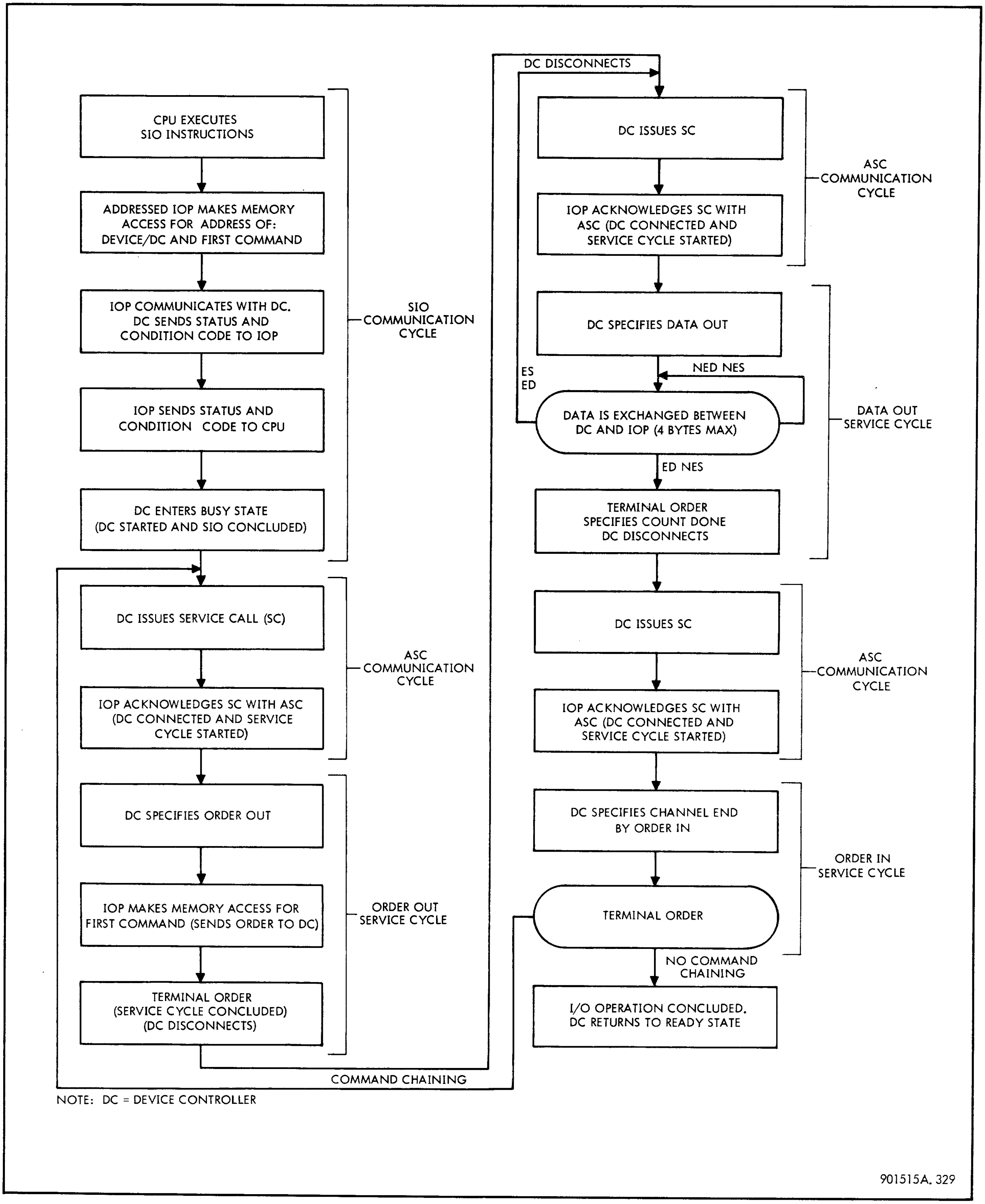



Figure 3-17. Typical I/O Operation, Overall Sequence Diagram

lines, where it is available to the device controllers. The address of the device/device controller is also transferred from the M-register to the A-register. The output of the A-register is decoded to select one of the 32 possible subchannels in the MIOP. The subchannels as a group are called the fast access memory. Any time information is transferred to or from the fast access memory, only the subchannel whose address (which is the same as the device controller address) is in the A-register is affected. Each subchannel consists of the following registers:

Byte Address (BA)

Command Address (CA)

Byte Count (BC)

Flags and Status (FS)

Operating Flags (OF)

Interrupt Status (IS)

The command address is transferred from the M-register to the CA-register by way of the C-register and adder. This is the address of the first command doubleword in the part of the command list for the addressed device. The address is retained by the CA-register until the SIO instruction has been completed and the service cycles begin.

The nature of the R field (whether it is odd, even, or zero) is retained by two latches. (For actions taken by the MIOP for the three different combinations of the R field see Input/Output Instructions in the Sigma 7 Computer Reference Manual.)

The addressed device controller places its status information (and that of the device) on the function response (FR) lines. The MIOP loads this status information, along with other information related to the I/O operation, in the M-register. Depending upon the nature of the R field, response information may be written into core memory locations X'20' and X'21'. Beside the status information the MIOP has just received from the device controller, the response information consists of the current command doubleword address, the current byte count, and status information that the MIOP has stored in its fast access memory during previous communications with the addressed device controller. The device controller also sends condition code information on the IOR and DOR lines.

If the SIO is successful, the device controller enters the busy state and drives its service call (SC) line true. (If the device controller is not in the ready state, or if an interrupt is pending, the SIO will be unsuccessful. The CPU is informed of an unsuccessful SIO by means of the state of the condition code lines.) The CPU is released from the I/O operation at the conclusion of the SIO instruction. The MIOP and device controller work together to execute the program (the portion of the command list) associated with the current I/O operation. The operation may consist of only one command (command chaining not specified), or more than one (command chaining specified).

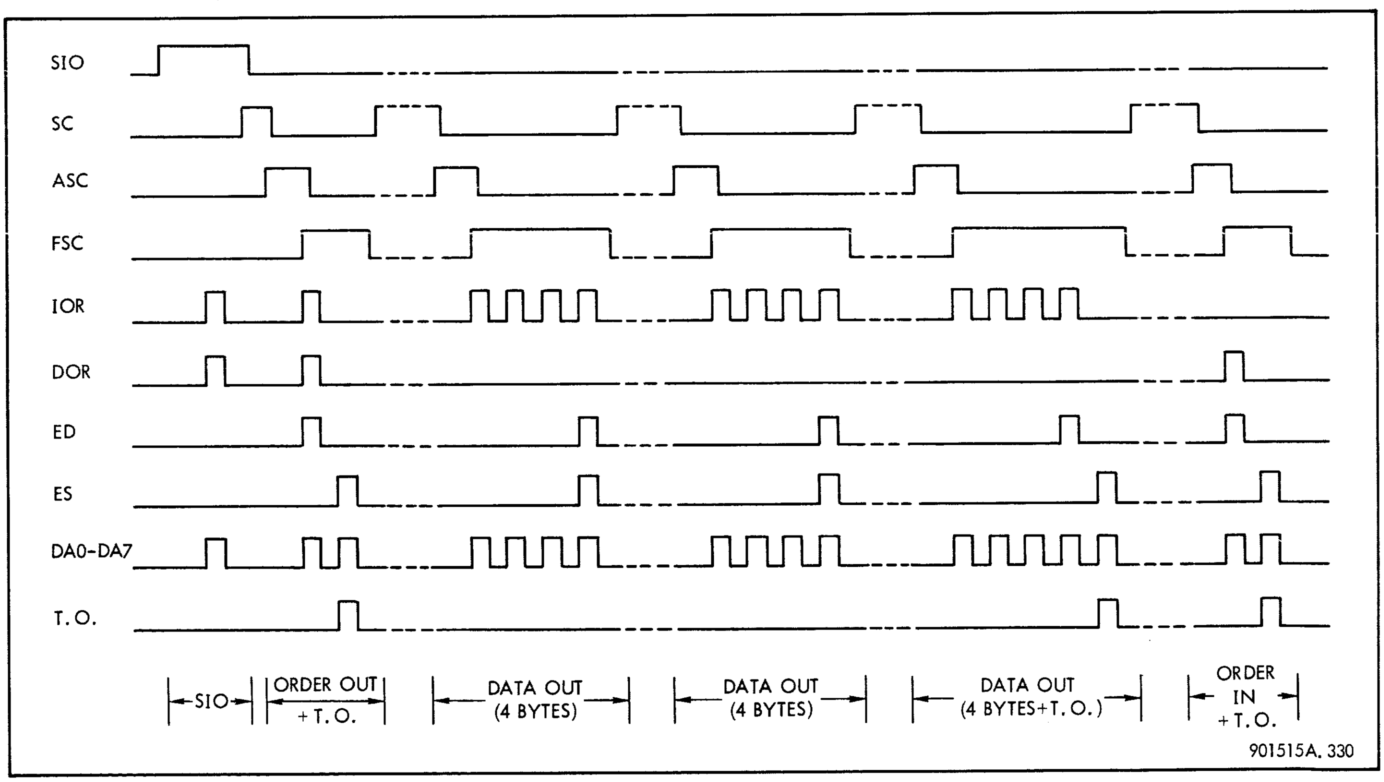



Figure 3-18. Typical I/O Operation, Timing Diagram

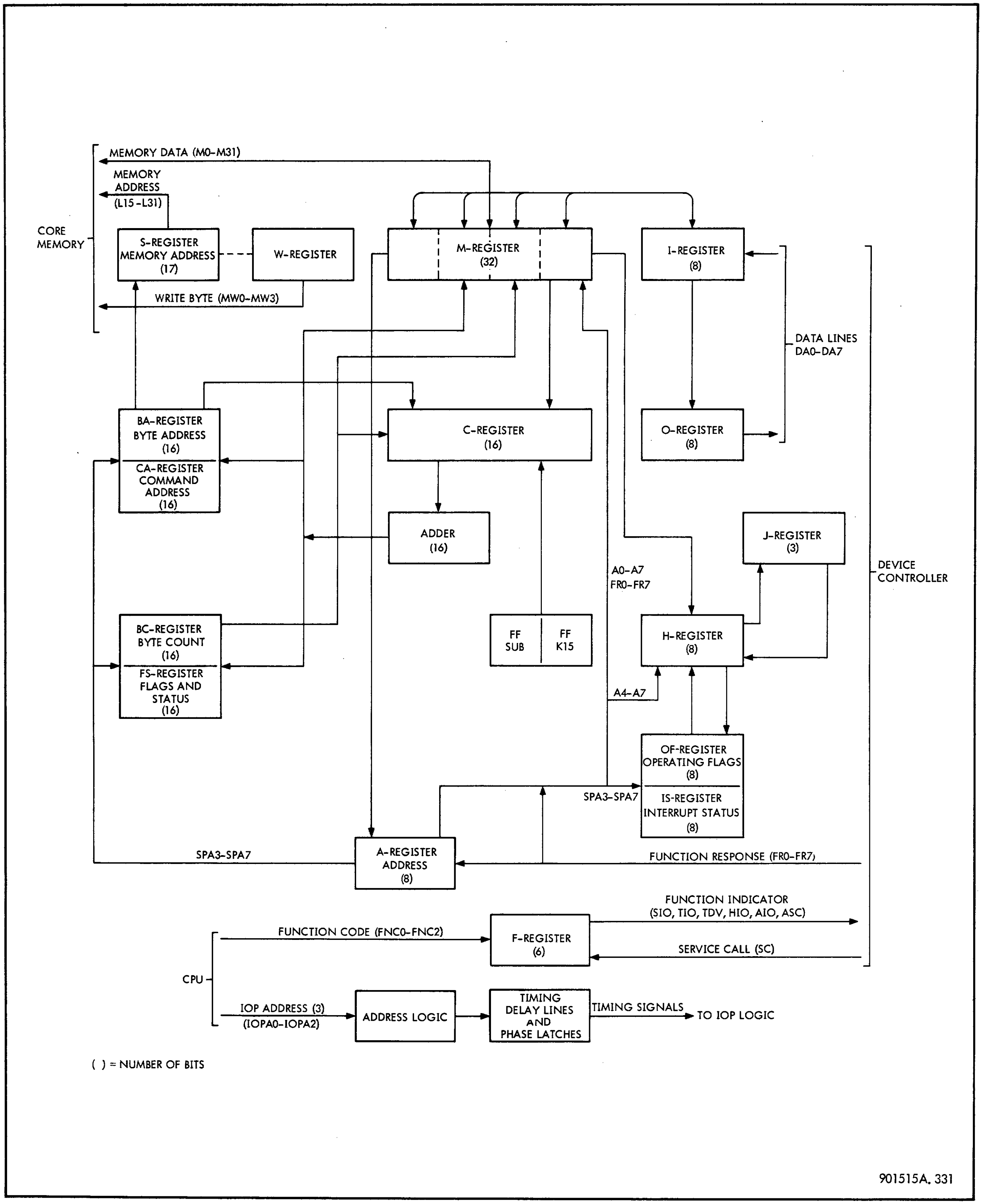



Figure 3-19. MIOP, Overall Block Diagram

The sequences following an SIO instruction consist of a number of service cycles. The four possible service cycles (order in, order out, data in, and data out) are defined by the device controller by coding the DOR and IOR lines. See the SDS Sigma Computer Systems Interface Design Manual for a complete description of the four service cycles.

The first service cycle following an SIO is always an order-out. The order-out service cycle causes the MIOP to fetch the first (and possibly only) command from the command list in core memory. The address of this command was sent to the MIOP by means of core memory from the CPU during execution of the SIO instruction. During the order-out service cycle, the MIOP stores the command (except the order bits) in the subchannel associated with the connected device controller. The order bits of the command are sent to the device controller on the data lines. The MIOP logic is such that a terminal order is always included with order-in and order-out service cycles, even though the MIOP may have nothing to report to the device controller. Along with the terminal order transmitted on the data lines, the MIOP drives the end service (ES) line true to conclude the service cycle.

After the device controller disconnects, it again drives the service call line true and again connects for service. During this service cycle, the device controller codes the DOR and IOR lines to specify a data-in or data-out service cycle. For example, if the order the device controller received during the order-out service cycle was a write order, the device controller would drive the IOR line true and hold the DOR line false.

The MIOP would interpret this coding as a data-out service cycle, and would, therefore, access core memory for the word of data specified by the byte address. (The byte address was received as part of the command doubleword during the order-out service cycle.) The data word from core memory is set into the M-register and transmitted to the device controller one byte at a time. A maximum of four bytes may be transmitted during any one service cycle. The MIOP will transmit up to four bytes per service cycle, continuing through as many service cycles as are required to send the number of bytes that are specified by the byte count. (The byte count was also received as part of the command doubleword during the order-out service cycle.) (Figures 3-17 and 3-18 show an I/O operation during which data is transmitted to the device controller.)

The state of signals ED and ES controls the transmission of each byte of data. If they are both false, another byte of data is transmitted during the current service cycle. If they are both true, the current service cycle will end and another data-out service cycle will follow. If ED is true and ES is false, a terminal order will be included as part of the current service cycle. When all of the bytes of data specified by the byte count have been transmitted (byte count equals zero), the MIOP will code the ED and ES lines so that a terminal order will be transmitted. During transmission of this terminal order the MIOP will specify count done. The device controller responds to count done by requesting service at the appropriate time, and when connected specifies an order-in service cycle. At this time, the device controller reports channel end to the MIOP, and if there are any unusual conditions they are also reported.

Some unusual conditions are capable of stopping a data exchange before count done. Upon sensing an error (unusual condition), the device controller defines the next service cycle as an order in instead of a data out. The error condition is reported during this order-in service cycle instead of at the end of the data exchange. The device controller inspects the contents of the terminal order following order in to determine what action must be taken.

If command chaining is specified, the device controller will again request service. After being connected, it will request another order from the MIOP by means of an order-out service cycle. The new order may begin another series of data exchanges similar to the preceding one, or it may start an entirely different operation. If command chaining is not specified, the device controller disconnects and enters the ready state.

### 3-27 I/O SYSTEM INTERFACE CONNECTIONS

Interconnection of the MIOP's in the system is shown in figure 3-20. Even though the figure shows only two MIOP's, a maximum of eight may be connected to a single CPU. Each additional MIOP is connected to the preceding one, with a single cable connecting the first MIOP to the CPU. Connection of each MIOP to the memory is by means of six cables. Each MIOP may or may not connect to more than one memory module; the type of I/O devices and other system requirements determine the memory interconnections. Device controllers connect to the MIOP by means of four cables. Three cables constitute the MIOP/device controller bus; the fourth is the priority cable. The routing of the priority cable is determined by the desired priority of the devices and is therefore not routed with the other three cables. The bus (three cables) is routed according to the physical placement of the device controllers. All device controllers connect to the MIOP in a trunk-tail fashion similar to the connection of the MIOP's to the CPU. Each MIOP may control a maximum of 32 device controllers (provided the MIOP is equipped with 32 subchannels). Only eight of the 32 are capable of controlling multiple devices.

### 3-28 MIOP INTERFACE SIGNALS

A description of the operation at each MIOP interface and a brief description of each interface signal is given under the three interface headings.

All MIOP interface signals are listed in tables 3-1 through 3-10. These tables include the signal name, designator, number of lines for signal groups, and direction of signal flow. The letter X in the signal reference designator refers to the applicable memory port – A, B, or C.

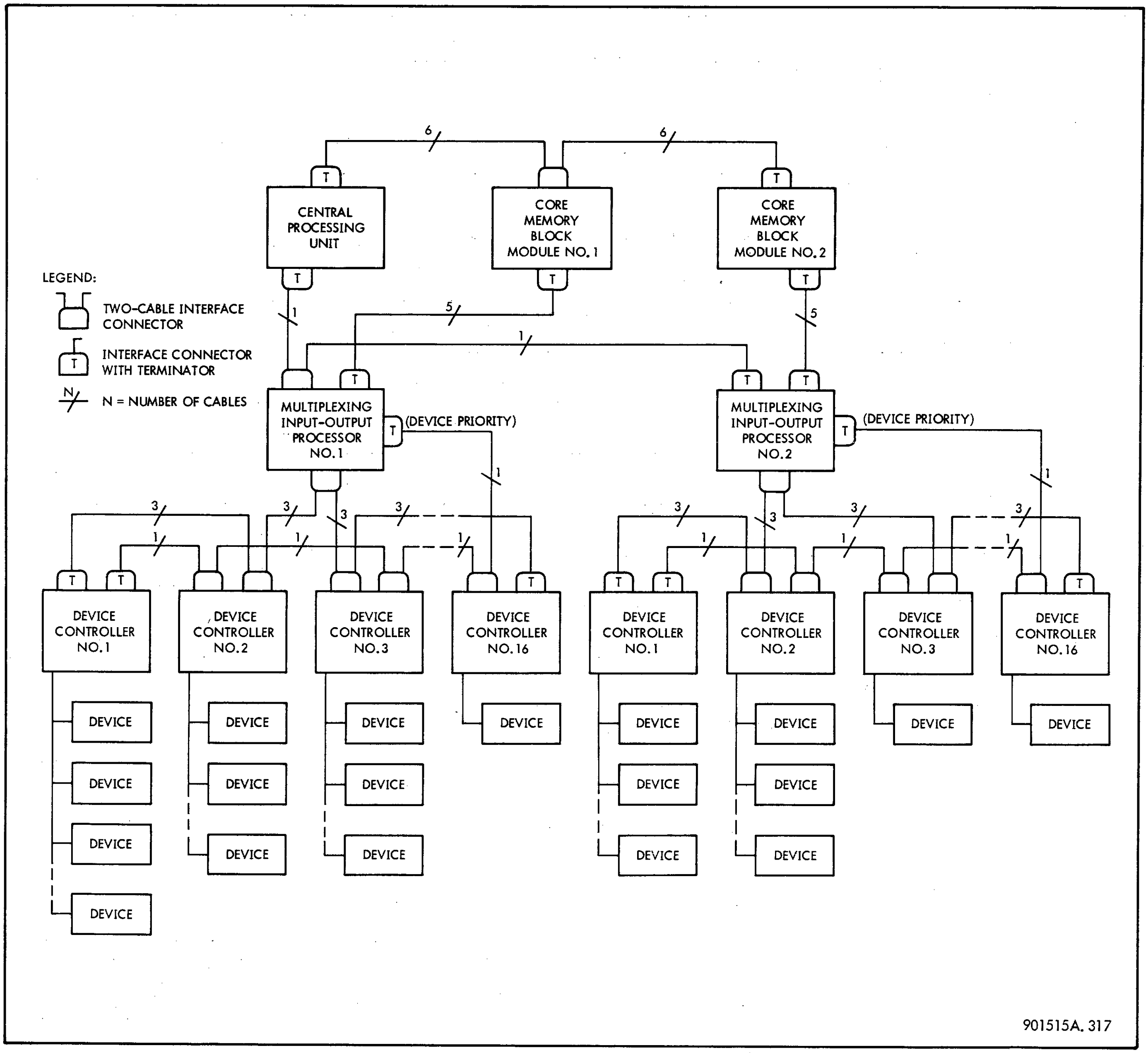



Figure 3-20. I/O System Interconnection Diagram

3-29 MIOP/Memory Interface

The MIOP normally connects to memory port A or B; however, it may be connected to port C if a lower priority is desired. Memory port A has the highest priority, port B the second highest, and port C the lowest.

When information is read from memory, a full 32-bit word is always transferred to the MIOP during the read operation. During a write operation, however, information may be written into memory from the MIOP on an 8-bit byte basis. One, two, three, or four selected bytes may be transferred to memory; the number of bytes is controlled by the MIOP.

A memory access is initiated when the MIOP drives memory request line MQX. (See figure 3-21.) Before driving the memory request line, the MIOP places a 17-bit address on memory address lines LX15 through LX31. During a memory write operation, the MIOP drives the memory request line and then supplies the data to be written on the data lines simultaneously with the write byte signals on write byte lines MW0X through MW3X. The memory request line is held true until the memory acknowledges by driving address release signal ARX true. If no further requests are to be made, the request line and address lines are released.

Table 3-1 provides a list of MIOP/memory interface signals and lines.

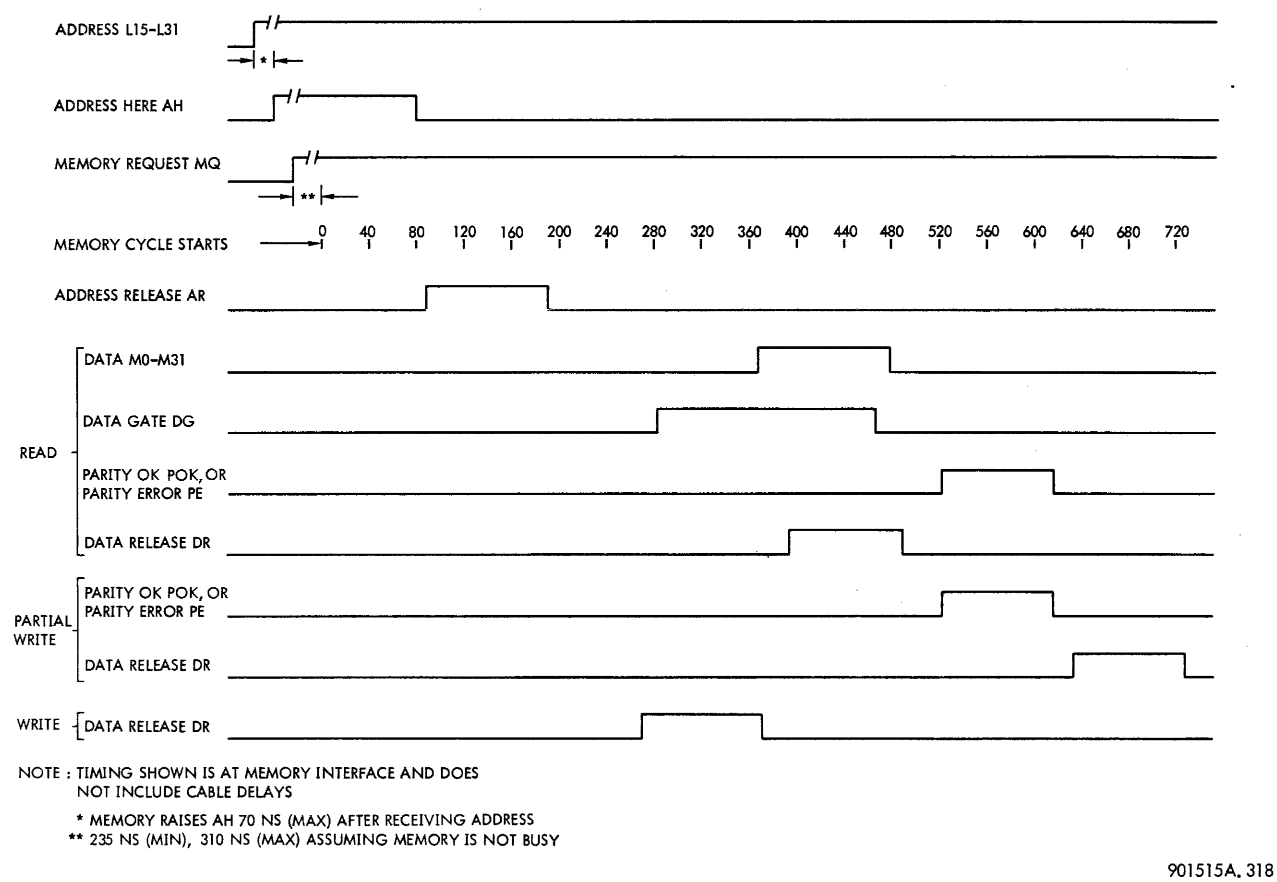



Figure 3-21. MIOP/Memory Interface Signal Timing Diagram (Port A or B)

Table 3-1. MIOP/Memory Interface Signals

| SIGNAL FUNCTION | SIGNAL | NO. OF LINES | DIRECTION OF SIGNAL FLOW | |
|---|---|---|---|---|
| | | | Memory to MIOP | MIOP to Memory |
| Data | /MX0/-/MX31/ | 32 | X | X |
| Address | /LX15/-/LX31/ | 17 | | X |
| Data release | /DRX/ | 1 | X | |
| Parity error | /PEX/ | 1 | X | |
| Parity OK | /POK/ | 1 | X | |
| Write byte | /MW0X/-/MW3X/ | 4 | | X |
| Memory request | /MQX/ | 1 | | X |
| Address release | /ARX/ | 1 | X | |
| Address here | /AHX/ | 1 | X | |
| Data gate | /DGX/ | 1 | X | |

A brief description of the MIOP/memory interface signals and lines follows:

a. Data lines MX0 through MX31: All data written into or read from memory (to and from the I/O devices) is transmitted over the 32 bidirectional data lines. When writing into memory, signals on the data lines are present and stable within 20 ns after the memory request signal is given and do not change until the data release signal is received by the MIOP. During a read operation, signals on the data lines are stable for at least 200 ns after the leading edge of the data release signal is received by the MIOP.

b. Address lines LX15 through LX31: The address of the memory location to be read from or written into is defined by the address lines. Signals on these lines are present for at least 60 ns before memory request MQX is true, and remain stable until address release ARX is received from the memory.

c. Data release DRX: The memory generates the data release signal and sends it to the MIOP to signify the following:

1. Another memory request may be made by the MIOP.

2. If the current operation is a memory write the data lines may be released.

3. If the current operation is a memory read, the data lines will remain true for 200 ns after the leading edge of the data release signal is received by the MIOP. (See figure 3-22.)

d. Parity error signal PEX: The memory drives the parity error line true when a parity error is detected during a memory read or partial write operation. The parity error signal goes true when signal DRX goes true, and stays true for 75 ns to 200 ns after signal DRX goes false.

e. Parity OK signal POKX: If no memory parity error is detected by core memory during a read or partial write operation, the memory drives the parity OK line. The timing is the same as for the parity error signal.

f. Write byte signals MW0X through MW3X: The memory write byte signals designate the memory byte or bytes into which new data is to be written during a full write or partial write operation. These signals go true at least 60 ns before the memory request signal and stay true until the address release signal is received by the MIOP.

g. Memory request signal MQX: The MIOP drives the memory request signal true when it must perform a memory read or write operation. The MIOP holds the memory request signal true until it receives the address release signal from the memory.

h. Address release signal ARX: The address release signal is generated by the memory to inform the MIOP that it may release the memory address lines and the memory request line.

i. Address here signal AHX: The memory drives the address here line true to inform the MIOP that the address encoded on the address lines is an address implemented in the memory or memories to which the MIOP is attached.

j. Data gate signal DGX: The memory drives the data gate line to permit the data on the data lines to be gated into the MIOP M-register.

## 3-30 MIOP/CPU Interface

Since much of the information transfer between MIOP and CPU is by means of the memory, only one 14-wire cable is required at the MIOP/CPU interface. MIOP/CPU interface operations resulting from a standard I/O instruction occur as three distinct phases as described below:

a. The CPU decodes the instruction and also stores information related to the particular operation in memory location X'20'. The CPU/MIOP control and address lines are energized and service is requested of the MIOP.

b. During servicing of the I/O instruction, the MIOP obtains the information stored in memory location X'20' and performs the operations defined by this information. The MIOP then returns I/O system response information to memory locations X'20' and X'21' and sends condition code information to the CPU. Following this, the MIOP sends the proceed signal to the CPU.

c. When the proceed signal is received, the CPU performs the necessary operations with the data received and terminates the operation.

The above three phases are described in more detail below. See table 3-2 for a list of MIOP/CPU interface signals.

When the CPU executes one of the five I/O instructions, it encodes the instruction on the three function code lines and the MIOP address on the three MIOP address lines. After a sufficient delay to allow the signals on these lines to stabilize, the CPU generates the control strobe. Each MIOP, upon receiving the control strobe, examines the MIOP address lines. If the MIOP address does not compare with the one on the address lines, the MIOP permits the control strobe to be passed along to the next MIOP. After the addressed MIOP has finished processing its currently connected device, the MIOP accesses memory location X'20' to obtain the new device address. An exception to this is the AIO instruction, since one purpose of this instruction is to determine the device address.

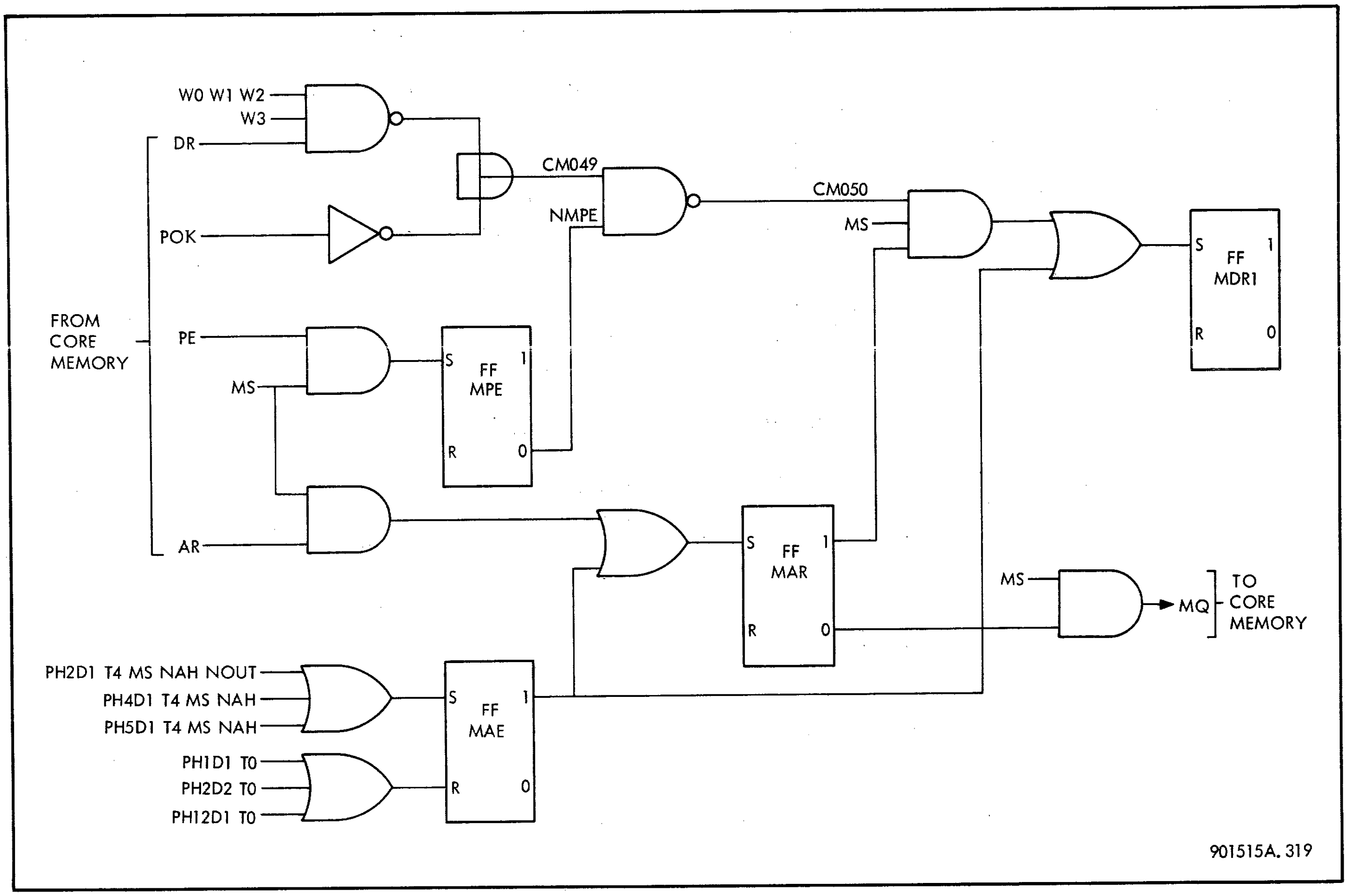



Figure 3-22. Setting Memory Data Release Flip-Flop MDR1

Table 3-2. MIOP/CPU Interface Signals

| SIGNAL FUNCTION | SIGNAL | NO. OF LINES | DIRECTION OF SIGNAL FLOW | |
|---|---|---|---|---|
| | | | MIOP to CPU | CPU to MIOP |
| Function code | /FNC0/-/FNC2/ | 3 | | X |
| MIOP address | /IOPA0/-/IOPA2/ | 3 | | X |
| Not condition code 1 | /NCOND1/ | 1 | X | |
| Not condition code 2 | /NCOND2/ | 1 | X | |
| 1-MHz clock | /CL1S/ | 1 | | X |
| Reset I/O | /RIO/ | 1 | | X |
| Interrupt request | /IR/ | 1 | X | |
| Proceed | /PR/ | 1 | X | |
| Control strobe | /CNST/ | 1 | | X |

The MIOP then communicates with the device. During this communication, the MIOP receives status information which it sends to memory locations X'20' and X'21' and condition code information which it sends directly to the CPU. The MIOP then sends the proceed signal to the CPU. Upon receipt of the proceed signal, the CPU sets its condition code according to the states of the two condition code lines and loads other required registers with the status information from memory locations X'20' and X'21'.

Also upon receipt of the proceed signal, the CPU releases the control strobe and after a short delay removes the MIOP address and function code signals from their respective lines.

If the address is not recognized by any MIOP, the last MIOP on the line returns the control strobe to the CPU on the proceed line. If the MIOP address is recognized, but the device address is not (MIOP to devices), the addressed MIOP sends the proceed signal to the CPU. The CPU is informed of address recognition (MIOP or device) by means of the condition code lines.

During execution of an AIO instruction, MIOP/CPU interface operation is somewhat different than for the other four I/O instructions. The AIO instruction is executed by the CPU when a device initiates a request to the MIOP for an interrupt, and the MIOP sends the request along to the CPU on the interrupt request line. When the CPU completes its current instruction (or between iterations of some long instructions), it will branch to the input/output interrupt location (5C), providing a higher priority interrupt is not waiting. This location contains an exchange program status doubleword instruction that transfers control to a standard I/O subroutine. Contained in this subroutine is an AIO instruction that identifies the source and reason for the interrupt. To identify the device, the CPU sends the AIO instruction and the control strobe to the MIOP's. The MIOP's that do not have an interrupt pending pass the control strobe to the following MIOP. When the strobe reaches an MIOP with an interrupt pending, it is routed sequentially according to priority to the devices attached to it until one is reached with an interrupt pending. This device then returns its address and status to memory location X'20' by means of the MIOP. The MIOP then sends the proceed signal to the CPU, thus permitting the CPU to examine the returned address to determine the interrupting device.

A brief description of the MIOP/CPU interface signals follows:

a. Function code lines FNC0 through FNC2: The CPU places a code representing the I/O instruction on these three lines. The signals on these lines are true before the control strobe is driven true, and remain true until after the control strobe is released.

b. MIOP address lines IOPA0 through IOPA2: The CPU places the coded signals representing the address of the MIOP to be connected during SIO, TIO, TDV, or HIO instructions. The signals on these lines are settled before the control strobe is driven true and remain settled until after the control strobe is released.

c. Not condition code 1 and not condition code 2: During execution of an I/O instruction, the condition code lines are coded as in table 3-3. The signals on the condition code lines are true before the proceed signal goes true and remain true until the control strobe is released.

d. 1-MHz clock signal CL1S: The CPU sends a 1-MHz clock to the MIOP, which it passes along to the device controllers.

e. Reset I/O line RIO: The I/O system is reset when the CPU drives the RIO line true.

f. Interrupt request signal IR: All requests for interrupts from device controllers are made to the CPU by means of the MIOP. When any device controller requests an interrupt, the MIOP drives the interrupt request line to the CPU. The CPU subsequently executes an AIO instruction to determine which device controller is requesting the interrupt. The interrupt request line goes false during the execution of an AIO instruction. If another interrupt is pending, the line is again driven true.

g. Proceed line PR: The MIOP drives the proceed line true at the termination of an I/O operation whether or not it was successful. For SIO, HIO, TIO, or TDV instructions, a successful I/O operation is dependent upon all of the following conditions:

1. MIOP address recognized
2. Memory location X'20' read without error
3. Device address recognized
4. Memory locations X'20' and X'21' written into by the MIOP as previously defined by the CPU
5. Not condition code lines NCOND1 and NCOND2 driven true by the MIOP.

Table 3-3. Condition Code Settings

| Instruction | NCOND1 | NCOND2 |
|---|---|---|
| SIO | I/O address recognized | SIO successful |
| HIO | I/O address recognized | Device not operating at time of disconnect |
| TIO | I/O address recognized | SIO possible |
| TDV | I/O address recognized | Device operating |
| AIO | Interrupt pending | Normal interrupt |

If address recognition is not obtained, the control strobe is returned to the CPU by the lowest priority MIOP as a proceed signal. Also, both condition code lines NCOND1 and NCOND2 will be false.

If a parity error is obtained in reading memory location X'20', the MIOP terminates the operation by setting the condition code lines false and the proceed line true.

A successful AIO operation depends upon the following conditions:

1. MIOP has pending interrupt

2. MIOP successfully connects to interrupting device for status

3. MIOP writes in memory location X'20'

4. Condition code line NCOND1 is true.

h. Control strobe CNST: The CPU drives the control strobe line true when executing any I/O instruction. The line is driven true after the signals on the function code lines (FNC0 through FNC2) and address lines (IOPA0 through IOPA2) have settled. The control strobe line operates in a closed-loop manner with the proceed line; the CPU releases the control strobe line upon receipt of the proceed signal.

3-31 MIOP/Device Controller Interface

A brief description of the operation at the MIOP/device controller interface follows: An I/O operation is initiated when the program executes an SIO instruction to start a device. During this initial operation, the device controller sends condition code and status information back to the MIOP, enters the busy condition, and drives the service call line true. The device is now started, but no data has been transferred between the device controller and core memory.

Since any number of device controllers may have previously been started (in the busy condition and service call line true), the MIOP responds to the service call by generating the acknowledge service call function. This results in signals that activate a hard-wired priority chain between device controllers. The highest priority device controller that has been started puts its own address on the return lines along with an acknowledge signal. Following this, the service connect flip-flop in the highest priority device controller is set. The operations that take place during the time this flip-flop is set constitute a service cycle.

The first service cycle following an SIO instruction is always an order out. The order-out service cycle results in an order being transmitted from the MIOP to the device controller so that the device controller will know what operation to perform. Subsequent service cycles will then be data in or data out if the operation to be performed involves transfer of data.

If the I/O device happens to be a printer, for example, the order will either be a write order or a control order. If it is a write order, data will be transferred to the printer during subsequent data-out service cycles at a maximum rate of four bytes per service cycle. At the end of each service cycle, the device controller resets its service connect flip-flop and again drives the service call line true. It is between these service cycles that time-sharing of devices occurs. Priority must be determined each time a device is connected.

Table 3-4 provides a list of MIOP/device controller interface signals. Table 3-5 correlates each of the lines with the various operations performed at the interface.

Most of the interface lines serve more than one purpose. For example, the data input/output lines carry the device address during the SIO, HIO, TIO, and TDV instructions; status information during the AIO instruction; data during the data-in and data-out service cycles; and other types of information during other operations. Table 3-6 lists the data input/output lines (by reference designator) and indicates the function performed by each line during each operation. Table 3-7 indicates the function performed by the data order request and input/output request lines during each operation. Table 3-8 indicates the function performed by the function response lines during each operation.

The following paragraphs describe the MIOP/device controller interface lines.

FUNCTION INDICATOR LINES SIO, HIO, TIO, AND TDV. The MIOP informs the device controllers of the instruction being executed by the CPU by driving one of the respective function indicator lines true. At the same time, the MIOP encodes the device address on the data input/output lines. The function indicator lines are not used for any purpose during the transferring of data between device and memory.

FUNCTION INDICATOR LINE AIO. The AIO indicator line serves the same purpose as the other indicator lines; however, the highest priority interrupting device places its address on the function response lines and its status on the data input/output lines. The AIO line goes true in response to an AIO instruction executed by the CPU.

FUNCTION INDICATOR LINE ASC. The MIOP drives the ASC line true in response to a service call from any of the device controllers. The purpose of the ASC signal is to acknowledge the service call. The highest priority device controller requesting service places its address on the function response lines. Ultimately, the recognized service-calling device controller is connected to the MIOP for service.

INTERRUPT CALL LINE IC. The device controllers drive the interrupt call line true to request an interrupt. The two basic reasons a device controller requests an interrupt are that it is requested to do so by means of a terminal order from the MIOP, or because of an internal condition such as the device needing control information from the MIOP or the pressing of a pushbutton.

Table 3-4. MIOP/Device Controller Interface

| SIGNAL FUNCTION | SIGNAL | DIRECTION OF SIGNAL FLOW | |
|---|---|---|---|
| | | MIOP to Device Controller | Device Controller to MIOP |
| Data Input/Output | /DA0/-/DA7/ | X | X |
| Data Parity | /DAP/ | X | X |
| End Data | /ED/ | X | X |
| Parity Check | /PC/ | | X |
| Data/Order Request | /DOR/ | | X |
| Function Response | /FR0/-/FR7/ | | X |
| Request Strobe | /RS/ | | X |
| Input/Output Request | /IOR/ | | X |
| Function Strobe Acknowledge | /FSL/ | | X |
| Interrupt Call | /IC/ | | X |
| Service Call | /SC/ | | X |
| I/O Reset | /RST/ | X | |
| Clock, 1 Megacycle | /CL1/ | X | |
| End Service | /ES/ | X | |
| Request Strobe Acknowledge | /RSA/ | X | |
| Start I/O Function Indicator | /SIO/ | X | |
| Halt I/O Function Indicator | /HIO/ | X | |
| Test I/O Function Indicator | /TIO/ | X | |
| Test Device Function Indicator | /TDV/ | X | |
| Acknowledge Interrupt Function Indicator | /AIO/ | X | |
| Acknowledge Service Call Indicator | /ASC/ | X | |
| Function Strobe | /FS/ | X | |
| Available In | /AVI/ | X | |
| Available Out | /AVO/ | | X |

Table 3-5. MIOP/Device Controller Interface Line Utilization

| LINES | OPERATION | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SIO | HIO | TIO | TDV | AIO | ASC | Order Out | Order In | Data Out | Data In | Terminal Order |
| Start I/O Function Indicator | X | | | | | | | | | | |
| Halt I/O Function Indicator | | X | | | | | | | | | |
| Test I/O Function Indicator | | | X | | | | | | | | |
| Test Device Function Indicator | | | | X | | | | | | | |
| Acknowledge Interrupt Function Indicator | | | | | X | | | | | | |
| Acknowledge Service Call | | | | | | X | | | | | |

(Continued)

Table 3-5. MIOP/Device Controller Interface Line Utilization (Cont.)

| LINES | OPERATION | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | SIO | HIO | TIO | TDV | AIO | ASC | Order Out | Order In | Data Out | Data In | Terminal Order |
| Interrupt Call | | | | | X | | | | | | |
| Service Call | | | | | | X | | | | | |
| Function Strobe | X | X | X | X | X | X | | | | | |
| Function Acknowledge Strobe | X | X | X | X | X | X | | | | | |
| Function Response | X | X | X | X | X | X | | | | | |
| Data/Order Request | X | X | X | X | X | | X | X | X | X | |
| Input/Output Request | X | X | X | X | X | | X | X | X | X | |
| Data Input/Output | X | X | X | X | X | | X | X | X | X | X |
| Data Parity | | | | | | | | | X | X | |
| Parity Check (device dependent) | | | | | | | | | | X | |
| Request Strobe | | | | | | | X | X | X | X | X |
| Request Strobe Acknowledge | | | | | | | X | X | X | X | X |
| End Data | | | | | | | | | X | X | |
| End Service | | | | | | | X | X | X | X | X |
| Available In | X | X | X | X | X | X | | | | | |
| Available Out | X | X | X | X | X | X | | | | | |

Table 3-6. Data Input/Output Line Definition

| LINE | INSTRUCTION | | ORDER OUT | | | | | | ORDER IN | DATA OUT | DATA IN | TERMINAL ORDER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SIO, HIO, TIO, TDV | AIO | Control | Write | Read | Read Back | Sense | Stop | | | | |
| DA0 | MSB ↑ | Data overrun | M | M | M | M | M | I | Transmission error | MSB ↑ | MSB ↑ | Interrupt |
| DA1 | Device address | Status (unique to device con- | M | M | M | M | M | 0 | Incorrect length | | | Count done |
| DA2 | | troller) | M | M | M | M | M | 0 | Chaining modifier | Data | Data | Command chain |
| DA3 | | | M | M | M | M | M | 0 | Channel end | | | MIOP halt |
| DA4 | | | M | M | M | 1 | 0 | 0 | Unusual end | | | Ignore last byte |
| DA5 | | | M | M | M | 1 | 1 | 0 | | | | |
| DA6 | ↓ | ↓ | 1 | 0 | 1 | 0 | 0 | 0 | | ↓ | ↓ | |
| DA7 | LSB | | 1 | 1 | 0 | 0 | 0 | 0 | | LSB | LSB | |

Note: M is a modifier bit unique to the device.
I, when true, designates that an interrupt shall be requested.

Table 3-7. Data Order Request and Input/Output Request Line Definitions

| Line | Instructions (SIO, HIO, TIO, TDV, AIO) | Order Out | Order In | Data Out | Data In |
|---|---|---|---|---|---|
| Data/Order Request (DOR) | 1 = NCC1 | 1 | 1 | 0 | 0 |
| Input/Output Request (IOR) | 1 = NCC2 | 1 | 0 | 1 | 0 |

Table 3-8. Function Response Line Definition

| LINE | INSTRUCTION OR FUNCTION: SIO, HIO, or TIO | INSTRUCTION OR FUNCTION: TDV | AIO, ASC |
|---|---|---|---|
| FR0 | Interrupt Pending | Data overrun | MSB ↑ |
| | Device ready<br>Device not operational<br>Device unavailable<br>Device busy | | |
| FR1 | 0 0 1 1 | ↑ | |
| FR2 | 0 1 0 1 | | |
| FR3 | Device auto | | |
| FR4 | Device unusual end | | Device address |
| | DC ready<br>DC not operational<br>DC unavailable<br>DC busy | Not assigned | |
| FR5 | 0 0 1 1 | | |
| FR6 | 0 1 0 1 | | |
| FR7 | Not assigned | ↓ | ↓ LSB |

SERVICE CALL LINE SC. The service call line is driven true by device controllers requesting service. The device controllers which have been started by an SIO instruction raise their service call line and keep it high during the course of the entire input/output operation, except during the time the device controller's service connect flip-flop is set and the device is connected to the MIOP.

INPUT/OUTPUT REQUEST LINE IOR AND DATA ORDER REQUEST LINE DOR. These two lines are controlled by the device controller. During execution of an instruction, they supply condition code information to the MIOP. During a service cycle, they specify whether the transfer of information is an order or data, and whether that order or data information is to be transferred into or out of memory. Refer to table 3-7 for coding of the DOR/IOR lines.

END DATA LINE ED AND END SERVICE LINE ES. These two lines designate the action to be taken by the MIOP and device controller during data and order exchanges. The end data line may be controlled by either the MIOP or the device controller. When the end data line is true it signifies that no more data is to be exchanged. The end service line is controlled only by the MIOP, which drives it true during a data exchange either at the time the last data byte is being exchanged or at the time the terminal order is transmitted to the device controller. During the initital byte of an order in/out operation the MIOP causes the device controller to request a terminal order by holding the end service line false. The end service line is driven true when the terminal order is transmitted during an order in/out operation. The coding of the end data and end service lines during data in/out operations is as shown in table 3-9. The permitted next states of the end data and end service lines are shown in table 3-10.

Table 3-9. Coding of End Data and End Service Lines During Data In/Out Operations

| STATE OF LINE | | RESULTING OPERATION |
|---|---|---|
| End Data | End Service | |
| 0 | 0 | Device controller requests another data byte |
| 1 | 0 | Device controller requests a terminal order |
| 1 | 1 | Device controller disconnects from MIOP (last data byte) |
| X | 1 | Defines terminal order. Device controller disconnects from MIOP. X signifies that the state of the end data line is meaningless during a terminal order |

Table 3-10. Permitted Next States of End Data and End Service Lines

| PRESENT STATE | | NEXT STATE | |
|---|---|---|---|
| End Data | End Service | End Data | End Service |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | X | 1 |

The coding of the end data line during an order in/out operation (X, table 3-9) has no meaning. The coding of the end service line during an order in/out operation has the following meaning:

a. End service line false: Indicates initial byte of an order in/out operation.

b. End service line true: Defines the order as a terminal order. The device controller disconnects from the MIOP.

The MIOP always holds the end service line false during transmission of the initial byte of an order in/out operation; therefore, the device controller always requests a terminal order to conclude the service cycle.

DATA INPUT/OUTPUT LINES DA0 THROUGH DA7. During execution of the SIO, HIO, TIO, and TDV instructions, the MIOP places the device address on the data input/output lines. (During an AIO instruction, the device places its address on the function response lines.) During an order-out operation, the lines carry the order (read, write, sense, read backward, and so forth) from the MIOP to the device. During an order-in operation, the operational status byte (transmission error, incorrect length, chaining modifier, and so forth) is sent from the device to the MIOP on the data lines. During a data-in operation, the device places the data to be transferred to core memory (by means of the MIOP) on the data lines. During a data-out operation, the MIOP places data from core memory on the lines for transfer to the device. During a terminal order, the MIOP places the information to be conveyed (interrupt, command chain, count done, and so forth) on the data lines for transfer to the device.

FUNCTION RESPONSE LINES FR0 THROUGH FR7. During execution of an SIO, HIO, or TIO instruction, the device controller places status information on the function response lines. During execution of a TDV instruction, the device controller may report a rate error (data overrun) condition to the MIOP by means of the function response lines. During the AIO and ASC functions, the device controller places its address on these lines.

DATA PARITY LINE DAP. The data parity line carries the parity bit; parity is odd. This line is used during the data-in or data-out operations. A parity bit is generated by the MIOP for each data byte presented to the device controllers. Also, each data byte supplied by the device controllers, along with a parity bit, is checked by the MIOP.

PARITY CHECK LINE PC. The parity check line is used only during a data-in operation. If the device controller drives the parity check line true the MIOP checks the parity information supplied by the device controller on the data parity line against the parity generated by the MIOP on the input data byte.

1-MHZ CLOCK SIGNAL CL1. The 1-MHz clock signal is sent to the device controllers for timing purposes. The MIOP receives the clock signal from the CPU and sends it along to the device controllers; it is not used by the MIOP.

I/O RESET SIGNAL RST. The MIOP receives reset signal RIO from the CPU and passes it along to the device controllers as signal RST. Signal RIO is generated by the CPU as a function of either SYS RESET, I/O RESET, the ST signal from the power monitor, or through special maintenance logic in the CPU when the MUSIC flip-flop is reset. When signal RST is true, all devices receiving the signal are halted, and all status and control indicators in the input/output system are reset.

FUNCTION STROBE SIGNAL FS. The MIOP issues a function strobe during each of the five computer instructions and also during the acknowledge service call function. The addressed device controller acknowledges by sending the function strobe acknowledge (FSL) signal back to the MIOP and by not driving the priority line (AVO) to the next lower priority device controller. The function strobe (with other signals) permits MIOP communication with:

a. The addressed device during an SIO, HIO, TIO, or TDV function, or

b. The highest priority device with its service call or interrupt call line high during the ASC or AIO functions.

FUNCTION STROBE ACKNOWLEDGE SIGNAL FSL. The function strobe acknowledge signal is used by the device controller to acknowledge the function strobe issued by the MIOP when it drives any of the function indicator lines.

AVAILABLE IN SIGNAL AVI AND AVAILABLE OUT SIGNAL AVO. The priority signals (AVI and AVO) are used by the device controllers to determine the highest priority device requesting an interrupt or service call. As the priority signal enters a device, it is designated as available in (AVI); as it leaves a device, it is designated as available out (AVO).

REQUEST STROBE SIGNAL RS AND REQUEST STROBE ACKNOWLEDGE SIGNAL RSA. The request strobe and request strobe acknowledge signals are used after a device has been started by an SIO instruction and has been connected to the MIOP for service. The connected device issues a request strobe along with data specifying the type of service required. When the MIOP receives the request strobe, a sequence of events is started. One of these is sending the request strobe acknowledge signal to the device controller. When the controller senses signal RSA true, it drops signal RS. When the MIOP senses signal RS false, it responds by dropping signal RSA.

## 3-32 SUBCHANNEL ADDRESSING

A full complement of fast access memory modules (FT25) provides 32 subchannels; each subchannel requires 80 bits. The characteristics of the FT25 modules are such that each module provides 16 of the 80 bits for eight different subchannels. The modules must, therefore, be installed in groups of five. The basic MIOP configuration, model 8471 (Sigma 7) and model 8271 (Sigma 5), contains in addition to the other 93 modules, five FT25 modules. Models 8472 (Sigma 7) and 8272 (Sigma 5), which are optional, provide for 15 additional FT25 modules, installed in groups of five, to accommodate 8, 16, or 24 additional device controllers.

Following is an explanation relating the 32 subchannels shown in the MIOP block diagram (figure 3-3) to the physical location of the modules shown in figure 3-23. Bits 0 through 7 of the BA- (or CA-) register for subchannels 0 through 7 (group 0) are located on the module in slot 23 of chassis D (D23). Bits 8 through 15 of the BA- (or CA-) register for subchannels 0 through 7 (group 0) are located on the module in slot D27. Likewise, bits 0 through 7 and 8 through 15 of the BC- (or FS-) register for group 0 are located on the modules in slots D15 and D19, respectively, and bits 0 through 7 of the OF- (or IS-) register for group 0 are located on the module in slot D11. The next group of five modules, located in slots D24 and D28 (BA/CA-register), D16 and D20 (BC/FS-register), and D12 (OF/IS-register) accommodate device controllers 16 through 23 (group 1). The third group, located in slots D25 and D29 (BA/CA-register), D17 and D21 (BC/FS-register), and D13 (OF/IS-register) accommodate device controllers 8 through 15 (group 2). The fourth group, located in slots D26 and D30 (BA/CA-register), D18 and D22 (BC/FS-register), and D14 (OF/IS-register) accommodate device controllers 24 through 31 (group 3). The subchannel groups are defined by signals SPA3 and SPA4 as shown in figure 3-24.

The address register (A-register) bits are decoded to produce signals SPA1 through SPA7 (see figure 3-25). Figure 3-26 indicates the manner in which the state of signal SPA5 selects eight of the 16 SDS 304 memory elements (integrated circuits) on each of the FT25 modules, thus dividing each group in half. (The figure shows the FT25 modules in slots D23 and D27. These two modules contain bits 0 through 15 of the BA- and CA-register, group 0.) The state of signal LS0 selects the bits in each memory element applicable to either the BA- or CA-register. The state of signals SPA6 and SPA7 further narrows the selection to the 16 bits applicable to one of the four subchannels defined by the state of SPA5 and LS0. Figure 3-24 relates the subchannel number with the state of signals SPA3 through SPA7. Figure 3-27 shows the electrical layout of a typical FT25 module.

Signals SPA3 and SPA4 both have to be false to select this module as one of five in group 0. Memory bits may be read or written only if the control input (terminal 1) of the memory element is true. Terminal 1 of each memory element is controlled by signals SPA3, SPA4, and SPA5. The particular bit in each memory element is selected by the address inputs (terminals 2, 3, and 4). These terminals are controlled by signals SPA6, SPA7, and LS0. Decoding logic for address terminals 2, 3, and 4 is contained in each memory element. The read/write clock input (terminal 8) must be true (in addition to the control input) for any bit to be changed. This input (for the BA- and CA-registers) is controlled by signal BAXADD.

The logic equations for the fast access memory inputs are divided into three different categories: Read/write clock input, data input, and address input. The coding of the left-hand term of the input equations is described in steps a through c. (The examples apply to the BA/CA-registers. The coding for the BC/FS-registers and the OF/IS-registers is identical except that BC or OF, as applicable, are used in place of BA.)

| SPA3 | SPA4 | SPA5 | SPA6 | SPA7 | SUBCHANNEL (DEVICE NO.) | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 1 | 1 | |
| 0 | 0 | 0 | 1 | 0 | 2 | |
| 0 | 0 | 0 | 1 | 1 | 3 | GROUP 0 (NSPA3 NSPA4) → L/BA007 - 0/1* |
| 0 | 0 | 1 | 0 | 0 | 4 | |
| 0 | 0 | 1 | 0 | 1 | 5 | |
| 0 | 0 | 1 | 1 | 0 | 6 | |
| 0 | 0 | 1 | 1 | 1 | 7 | |
| 0 | 1 | 0 | 0 | 0 | 8 | |
| 0 | 1 | 0 | 0 | 1 | 9 | |
| 0 | 1 | 0 | 1 | 0 | 10 | |
| 0 | 1 | 0 | 1 | 1 | 11 | GROUP 2 (NSPA3 SPA4) → L/BA007 - 2/1* |
| 0 | 1 | 1 | 0 | 0 | 12 | |
| 0 | 1 | 1 | 0 | 1 | 13 | |
| 0 | 1 | 1 | 1 | 0 | 14 | |
| 0 | 1 | 1 | 1 | 1 | 15 | |
| 1 | 0 | 0 | 0 | 0 | 16 | |
| 1 | 0 | 0 | 0 | 1 | 17 | |
| 1 | 0 | 0 | 1 | 0 | 18 | |
| 1 | 0 | 0 | 1 | 1 | 19 | GROUP 1 (SPA3 NSPA4) → L/BA007 - 1/1* |
| 1 | 0 | 1 | 0 | 0 | 20 | |
| 1 | 0 | 1 | 0 | 1 | 21 | |
| 1 | 0 | 1 | 1 | 0 | 22 | |
| 1 | 0 | 1 | 1 | 1 | 23 | |
| 1 | 1 | 0 | 0 | 0 | 24 | |
| 1 | 1 | 0 | 0 | 1 | 25 | |
| 1 | 1 | 0 | 1 | 0 | 26 | |
| 1 | 1 | 0 | 1 | 1 | 27 | GROUP 3 (SPA3 SPA4) → L/BA007 - 3/1* |
| 1 | 1 | 1 | 0 | 0 | 28 | |
| 1 | 1 | 1 | 0 | 1 | 29 | |
| 1 | 1 | 1 | 1 | 0 | 30 | |
| 1 | 1 | 1 | 1 | 1 | 31 | |

NOTES: BIT 0 OF A-REGISTER = 0

* REPRESENTS TYPICAL FAST-ACCESS MEMORY ADDRESS TERM

901515A. 321

Figure 3-24. Assignment of Subchannel Groups

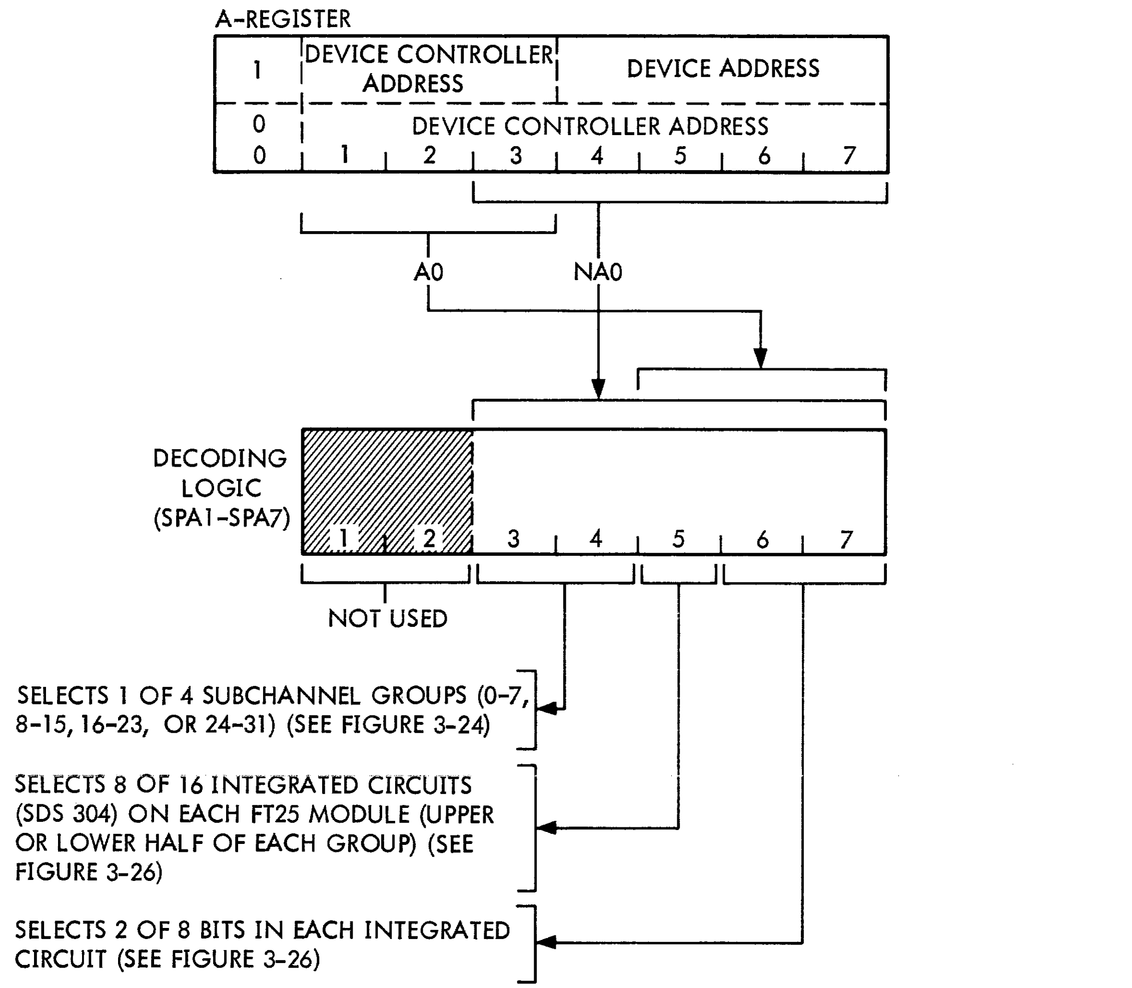



Figure 3-25. Decoding the Address Register

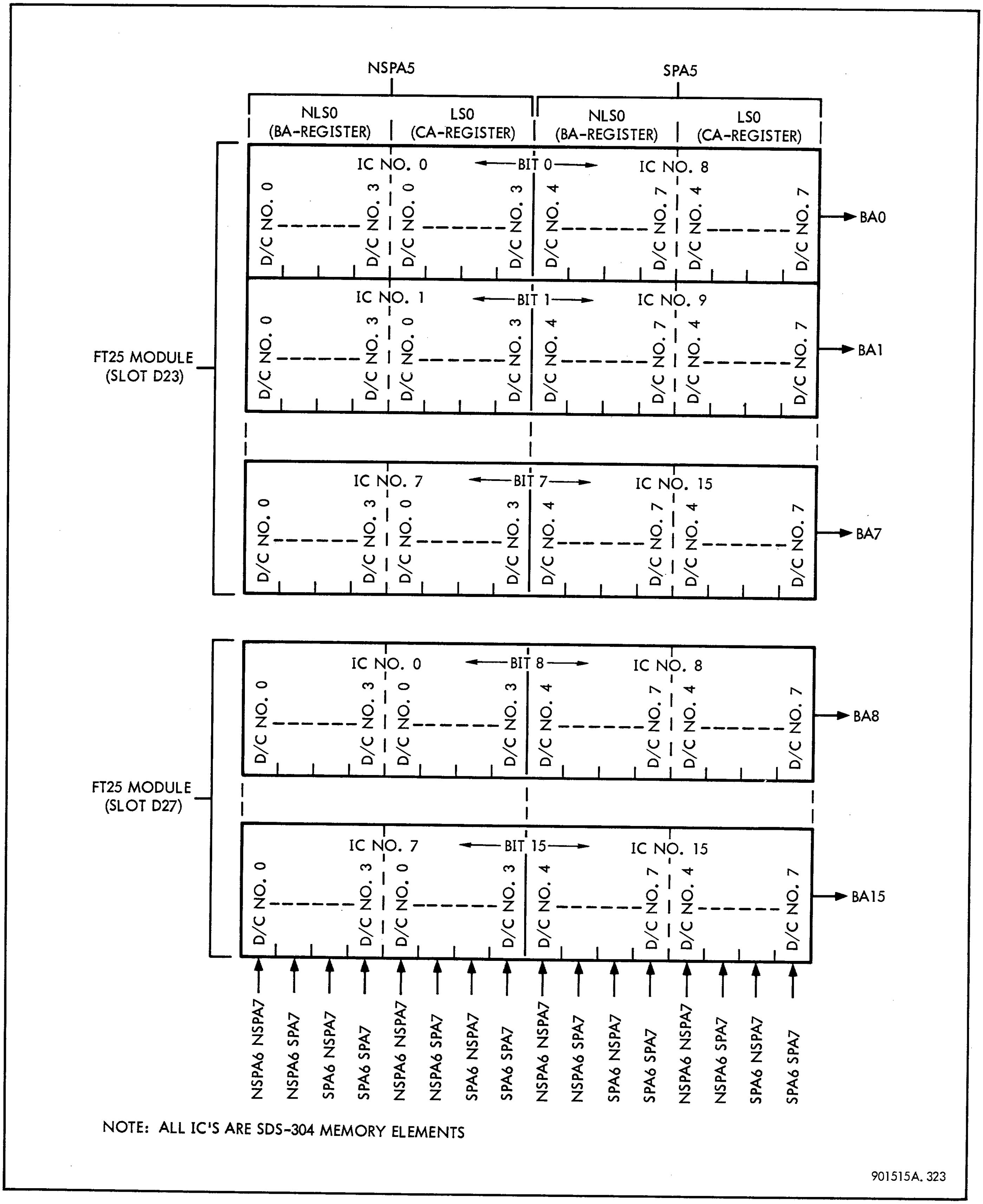
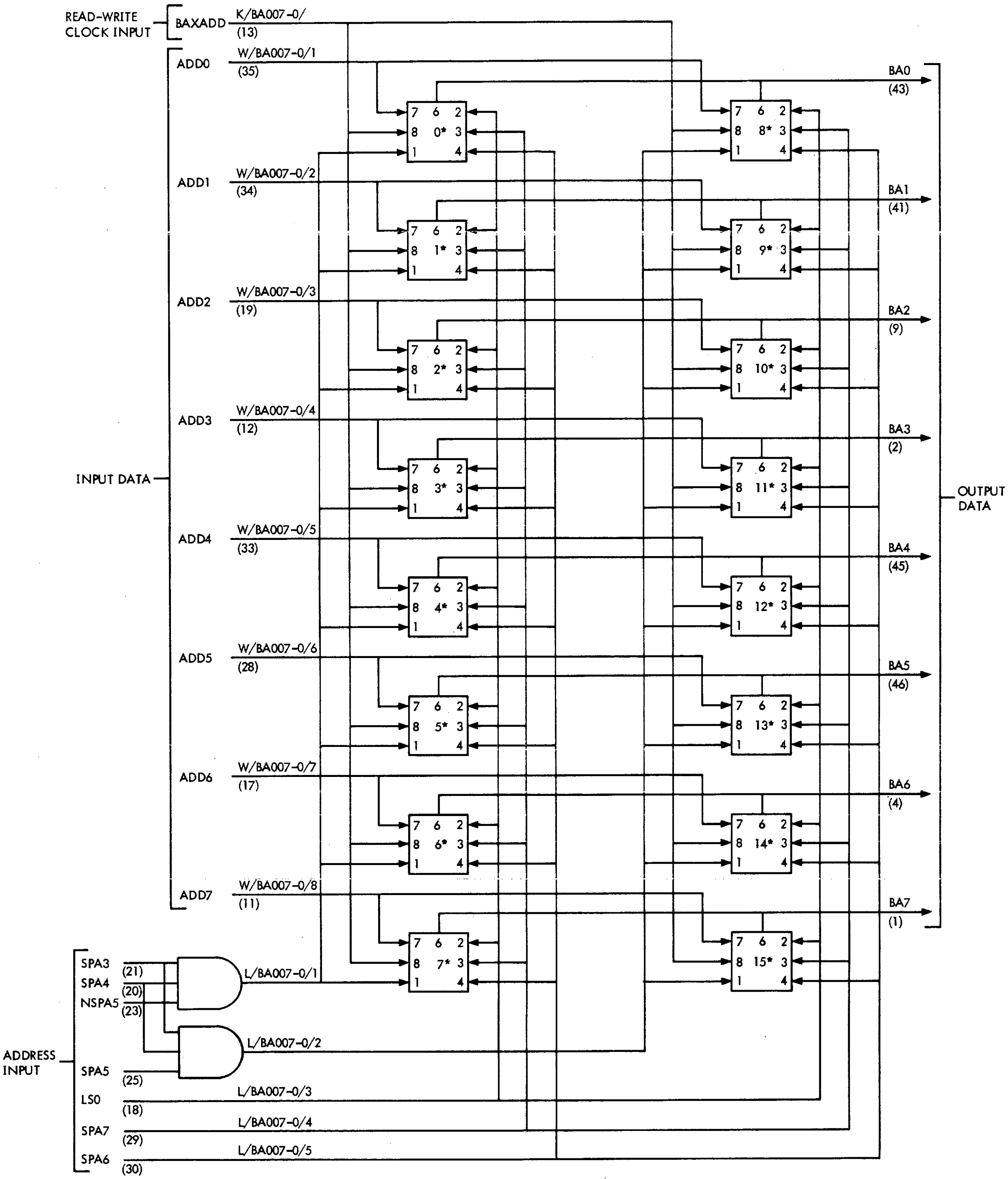



Figure 3-26. Decoding SPA5, SPA6, and SPA7 Address Signals to Select One Subchannel in Group Zero




NOTES: *1. 1 OF 16 SDS 304 MEMORY ELEMENTS; EACH CONTAINS 8 BITS.
2. NUMBERS IN PARENTHESES ARE PIN NUMBERS ON FT25 MODULE.
3. NUMBERS SHOWN INSIDE SDS 304 MEMORY ELEMENTS ARE DEFINED BELOW:

1 - CONTROL
2, 3, 4 - ADDRESS (SELECTS 1 OF 8 BITS IN EACH MEMORY ELEMENT)
5 - GROUND (NOT SHOWN)
6 - DATA OUTPUT
7 - DATA INPUT
8 - READ/WRITE INPUT
9 - NOT USED
10 - $V_{CC}$ (NOT SHOWN)

4. SIGNALS SHOWN ARE FOR BA OR CA REGISTER, BITS 0 THROUGH 7, SUBCHANNEL GROUP 0 (SLOT D23)

901515A. 324

Figure 3-27. Typical FT25 Fast Access Memory Module

a. Read/write clock:

K/BA007-0/
⋮
K/BA007-3/

- Subchannel group (groups 0 through 3)
- Bit group 0 through 7
- BA lines (BA-register if LS0 = 0, CA-register if LS0 = 1)
- Read/write clock

K/BA815-0/
⋮
K/BA815-3/

- Bit group 8 through 15

b. Data input:

W/BA007-0/1
⋮
W/BA007-3/8

- Bits 0 through 7
- Subchannel group
- Bit group 0 through 7
- BA lines
- Write operation

W/BA815-0/1
⋮
W/BA815-3/8

- Bit group 8 through 15

c. Address input (see figure 3-27):

L/BA007-0/1
⋮
L/BA007-3/1

- Memory elements 0 through 7 (subchannels 0 through 3)
- Subchannel group
- Bit group 0 through 7
- BA lines
- Address input

L/BA007-0/2
⋮
L/BA007-3/2

- Memory elements 8 through 15 (subchannels 4 through 8)

L/BA007-0/3
⋮
L/BA007-3/5

- Three address inputs on each memory element

L/BA815-3/5

- Bit group 8 through 15

## 3-33 MIOP TIMING SIGNALS

Timing signals for all MIOP input/output operations are supplied by two delay lines and their associated delay line sensors. The triggering of a delay line starts a series of timing pulses during which certain logic operations are permitted to occur. The combination of a particular timing signal generated by a delay line and the signal from a particular phase latch associated with that delay line provides a unique timing signal. This signal is used, either directly or indirectly, to enable the transfer of data and control signals among the various MIOP registers and logic elements and to control the transfer of data and control signals at the MIOP interfaces. The triggering of the delay line is a function of the appropriate phase and also of incoming signals to the MIOP. The two delay lines and their associated phase latches operate independently of each other; therefore, certain operations can overlap in the MIOP.

### 3-34 Delay Line Operation

The two delay lines (D1 and D2) are identical; however, the delay line sensor elements associated with delay line D1 are connected to different taps than those of delay line D2. Each delay line has 32 taps, electrically spaced at 20-ns intervals; however, only six of the taps are used. Figure 3-28 is a logic diagram of delay line D1 and associated sensor elements, and figure 3-29 is a timing diagram of delay line D1 timing signals. The timing signals shown assume 0-ns delay through all circuits except the delay line.

A pulse is started down the delay line when it is triggered. As the pulse wave front reaches each of the taps (SENSE 0D1 through SENSE 5D1), the voltage level at that tap rises from -4 volts to +4 volts. As the trailing edge of the pulse passes by the tap, the voltage level at that tap again

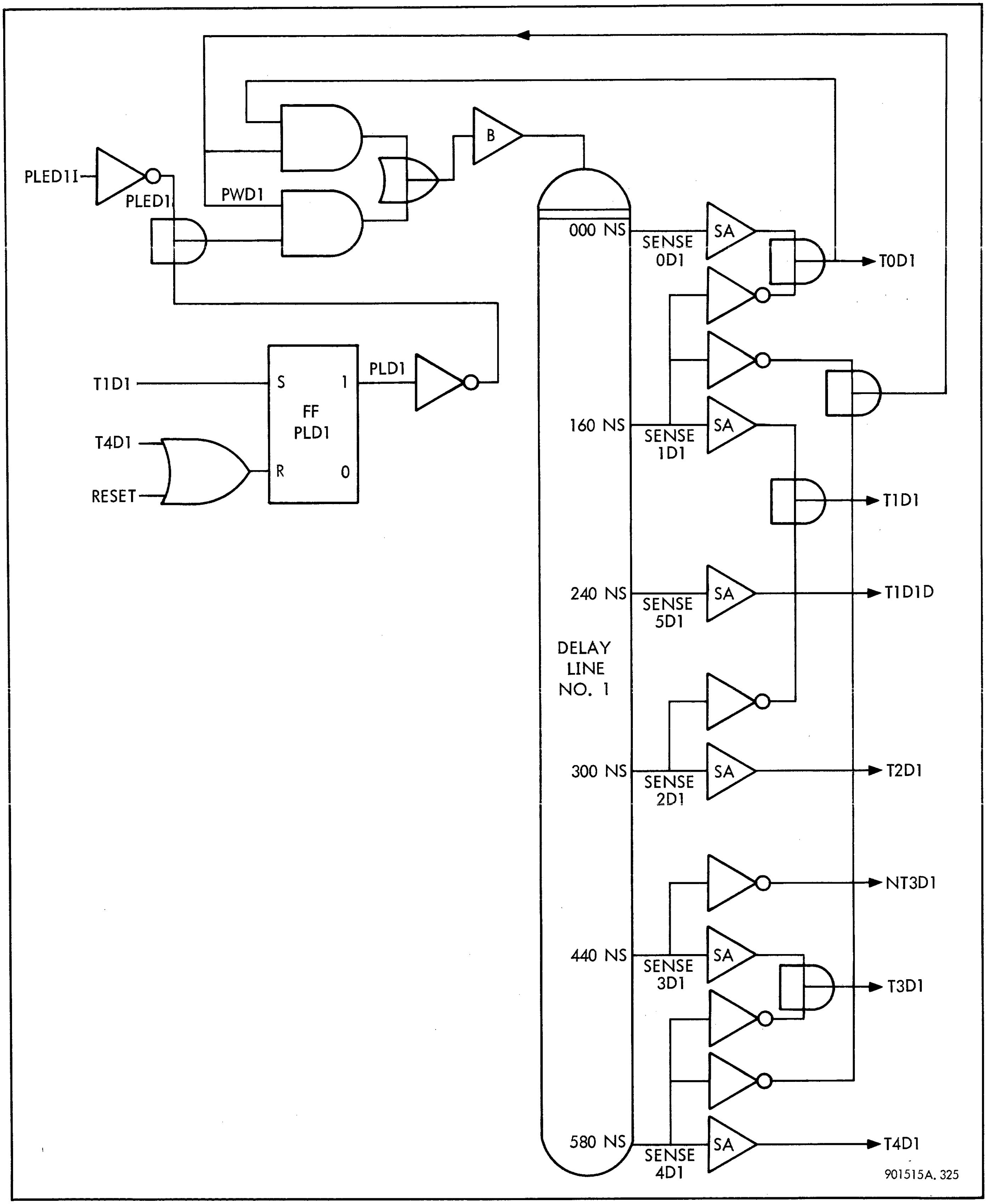



Figure 3-28. Delay Line 1 Logic Diagram

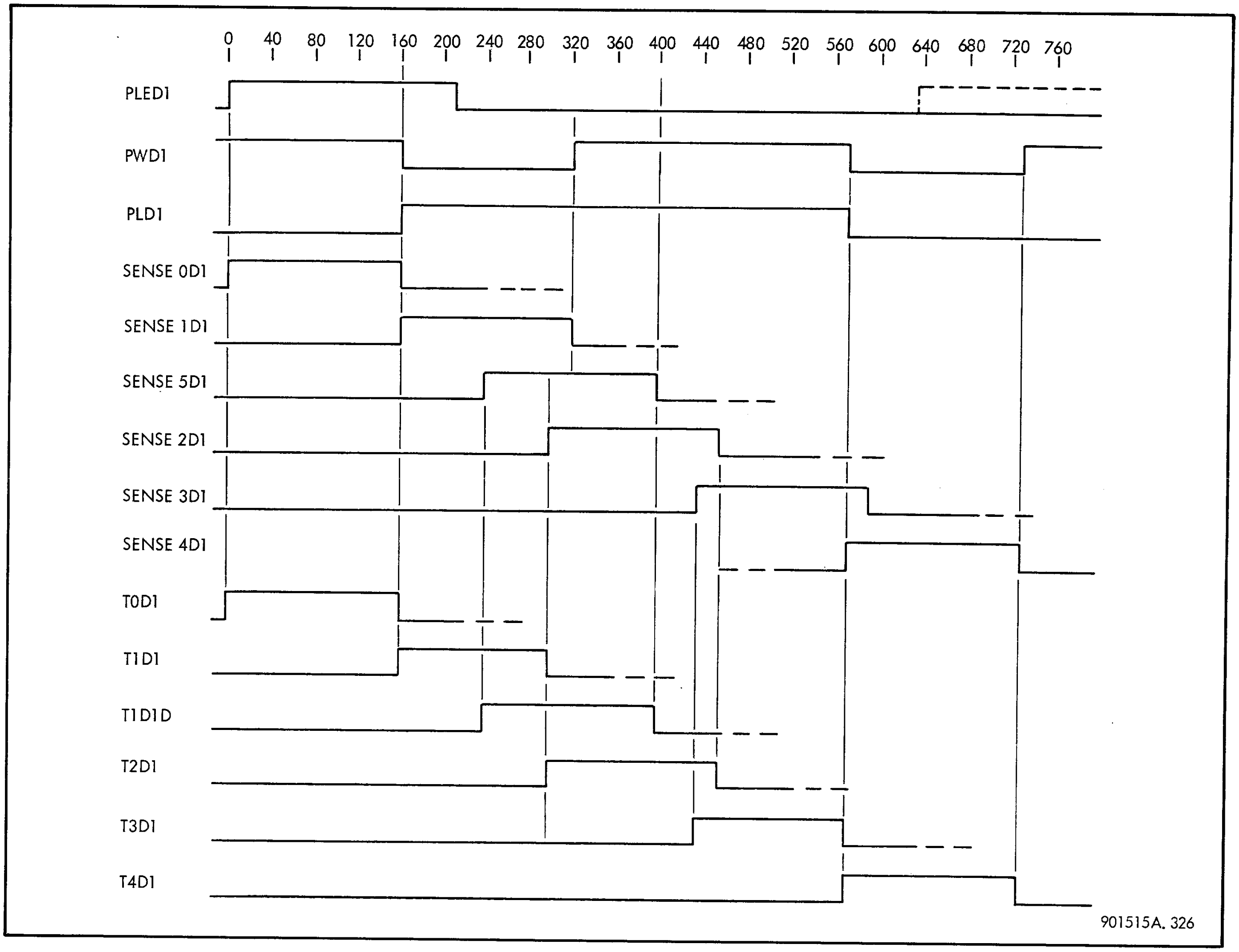



Figure 3-29. Delay Line 1 Timing Signals

assumes the low state (-4 volts). The logic that controls the pulse is such that only one pulse can be in the line at any given time, and the width of the pulse is 160 ns in duration regardless of the duration of input signal PLED1 that starts the line. Once the pulse is started down the line, it is cut off when it reaches the 160-ns tap (SENSE 1D1) since signal PWD1 (initially high) goes low when SENSE 1D1 goes high. If the input pulse, signal PLED1, is less than 160 ns in duration, signal I/SENSED 1 is held high by signals PWD1 and T0D1 for 160 ns.

During some operations, signal PLED1 goes true for the next phase during T4D1 of the current phase, that is, while the pulse is still traveling down the delay line. The line is not triggered again, however, until the end of T4D1, when signal PWD1 again goes true.

Delay line D2 operation is similar to delay line D1 except that the delay line sensors and taps are such that the pulse traveling down the delay line is 140 ns in duration instead of 160 ns. Figure 3-30 is a logic diagram of delay line D2 and associated sensor elements, and figure 3-31 is a timing diagram of delay line D2 timing signals. The timing signals shown assume 0-ns delay through all circuits except the delay line.

3-35 Phase Latches

The phase latches associated with each delay line are divided into ranks according to the timing periods during which they are used. There are 24 phase latches associated with delay line D1. The first rank of 12 consists of latches PH1D1 through PH14D1 (PH8D1 and PH9D1 do not exist). The second rank of 12 consists of latches PH1PD1 through PH14PD1. All phase latches except PH2D2 (a flip-flop) are buffer latches. The first rank is used during timing periods T0D1, T1D1, and T2D1, and the second rank is used during T3D1 and T4D1. At time T0D1 the second rank latches are cleared by signal PHPD1X0. The output of the first rank phase latches, in conjunction with other control

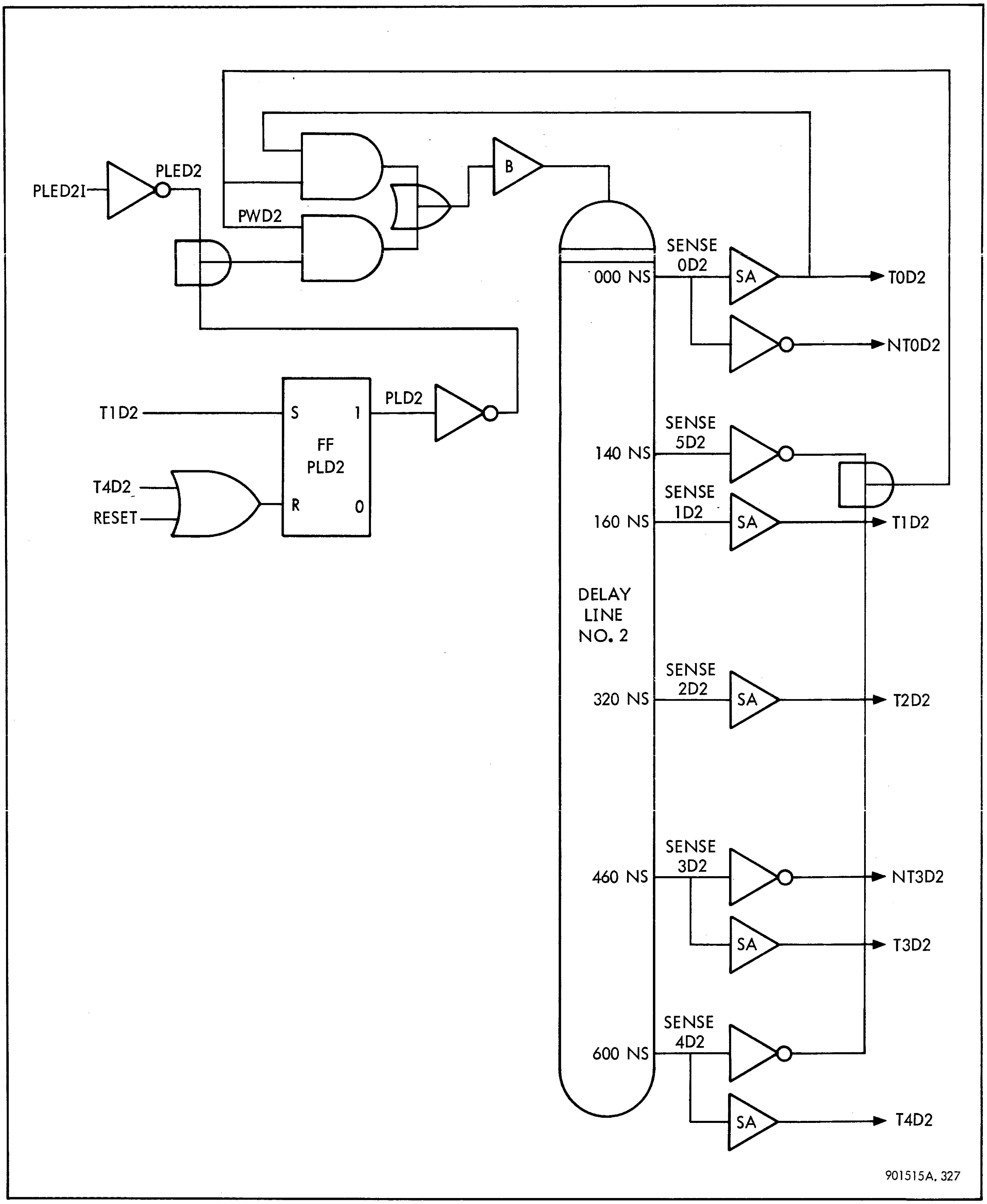



Figure 3-30. Delay Line 2 Logic Diagram

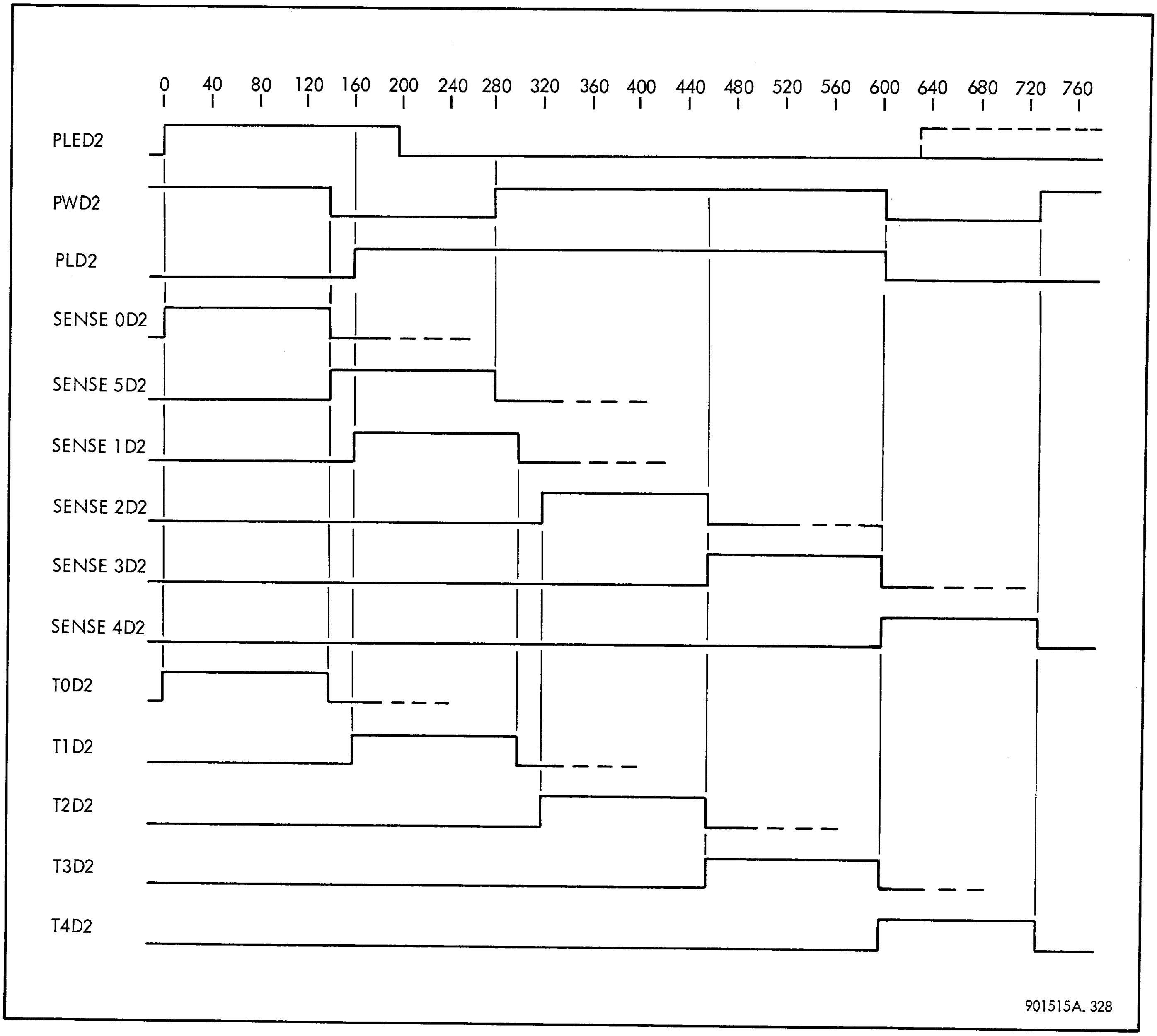



Figure 3-31. Delay Line 2 Timing Diagram

logic, is used to start delay line D1. At time T1D1 the contents of the first rank phase latches are transferred to the second rank phase latches (PH1D1 through PH14D1 to PH1PD1 through PH14PD1, respectively). At time T3D1 the first rank phase latches are cleared. Setting of the first rank phase latches is a function of incoming signals, delay line timing signals, and MIOP logic.

There are four phase latches associated with delay line D2. The first rank consists of PH1D2 and PH2D2, and the second rank consists of PH1PD2 and PH2PD2. All four are buffer latches. These four phase latches operate with delay line D2 in much the same manner that delay line D1 and its associated phase latches operate.

## 3-36 PHASE SEQUENCES

The phase sequences for the five computer I/O instructions and the four service cycles are described separately in the following paragraphs. Each description includes a flow diagram showing the sequences of phases, a description of the most significant functions performed during each phase, and a phase sequence table for each instruction and service cycle.

The tables include the functions performed during each phase, the signals involved, and a brief comment relating to each function. Table 3-11 outlines the functions performed by the MIOP during a core memory cycle. (See also figures 3-21 and 3-22.) Tables 3-12 through 3-19 are the phase sequence tables.

Table 3-11. MIOP/Memory Phase Sequence

| Signal | Remarks |
|---|---|
| Set MS | Signal MS is generated by the MIOP to start a memory cycle |
| MS NMAR ⇒ Set MQ | Flip-flop memory address release MAR is reset before a memory cycle occurs. When the MIOP sets MS, memory request MQ goes true. Signal MQ drives the memory request line to memory |
| MS AR ⇒ Set MAR | When the MIOP receives address release AR from memory, MAR is set. Signal AR indicates to the MIOP that the memory no longer needs the memory address and memory request signals. The memory transmits AR approximately 80 ns after the memory cycle starts |
| MAE ⇒ Set MAR | If the address the MIOP sends to the memory is implemented in the memory block to which the MIOP is connected, the memory sends address here AH to the MIOP. If AH is not received in the specified period of time, memory address error flip-flop MAE is set. (The memory will not send AR if it does not first send AH.) Signal MAE then sets flip-flop MAR so that the memory request signal MQ will go false. Signal MAE also sets memory data release flip-flop MDR1 so that the MIOP will not hang up if a nonexistent memory location was addressed |
| MS DG ⇒ MXM | Signal data gate DG is transmitted by the memory during a memory read operation to let the MIOP know the data is on the data lines. Signal MXM strobes in the data from memory |
| MS PE ⇒ Set MPE | Memory parity error flip-flop MPE is set if parity error PE is true when MS is true. During a read or partial write operation, memory generates PE if it detects a parity error, or POK if no parity error is detected. |
| MAE + [MS MAR (MPE + POK + W0 W1 W2 W3 DR)] ⇒ MDR1 | Flip-flop MDR1 is set when all the required signals have been received from memory for the particular operation (read, partial write, or full write), or when a nonexistent memory location has been addressed |

3-37 SIO Instruction

The phases follow the sequence shown in figure 3-32 during an SIO instruction. Whether the sequence includes PH11D1 or not depends upon the status information that is returned to the CPU by means of core memory locations X'20' and X'21'. If only one word or no words of status are to be returned, signal CMD is false, and the MIOP advances to PH12D1 from PH10D1. If two words of status information are to be returned, signal CMD is true, and the MIOP includes PH11D1 in the sequence.

PH1D2. During this phase the MIOP determines whether the operation is being initiated by a request from the CPU or a device controller. A request for service from the CPU takes priority over a request from a device controller. Since the operation is an SIO, the MIOP decodes the function lines from the CPU and drives the SIO function indicator line at the device controller interface.

PH2D2. During this phase, the MIOP clears various registers and flip-flops in preparation for a core memory access, forces the address of core memory location X'20' into the S-register, and makes a core memory access.

PH10D1. The delay line starts for PH10D1 after the contents of core memory location X'20' have been set into the M-register. During this phase the address bits are transferred from the M-register to both the A- and O-registers. The address in the A-register is decoded by the SP logic for the purpose of selecting the appropriate MIOP subchannel in the fast access memory. The address in the O-register is also present on the data lines, where it is available to the device controllers (see figure 3-33).

The coding of bits M8 and M9 reflects the nature of the R field of the SIO instruction. Flip-flop CMD is set if M8 is true, and flip-flop TPE is set if M9 is true. If CMD is true, the first word (address of the current command doubleword in the CA-register) is set into the M-register (bits M0-M15), and a core memory cycle is started to store the first word into memory location X'20'. Also, the MIOP will advance to PH11D1, during which time the new command doubleword address (currently in M16-M31) will be stored in the

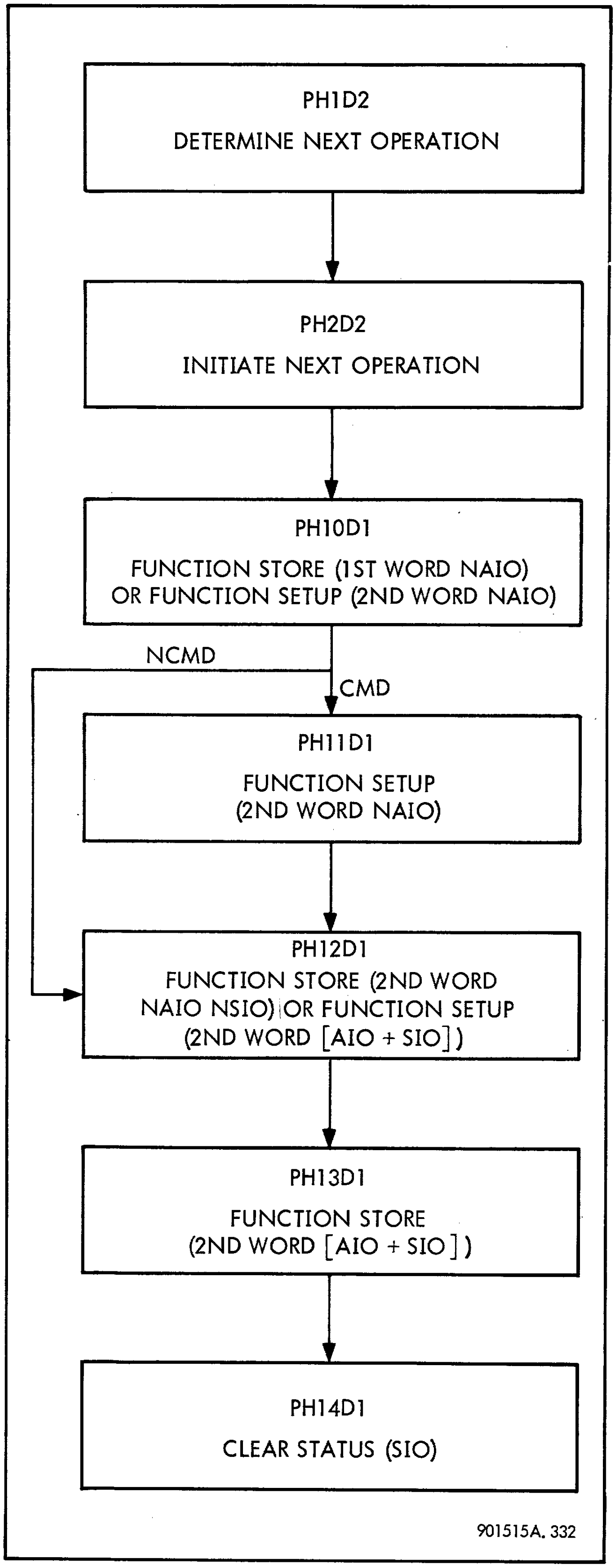



Figure 3-32. Phase Diagram for SIO, HIO, TIO, and TDV Instructions

CA-register. If CMD is false, the new command doubleword address is stored in the CA-register during this phase (PH10D1), and PH11D1 is skipped. If TPE is true, the status and byte count will be written into core memory location X'21' during PH13D1.

PH11D1. This phase is entered only if CMD is true. The current command doubleword address is cleared from the C-register and adder, and the new command doubleword address, currently in bits M16 through M31 of the M-register, is transferred to the C-register. This new command doubleword address will replace the current command doubleword address in the CA-register during PH12D1.

PH12D1. During this phase, the MIOP clears the M-register and proceeds to load it with status information from the device controller (by means of the FR lines), and the byte count from the appropriate MIOP subchannel.

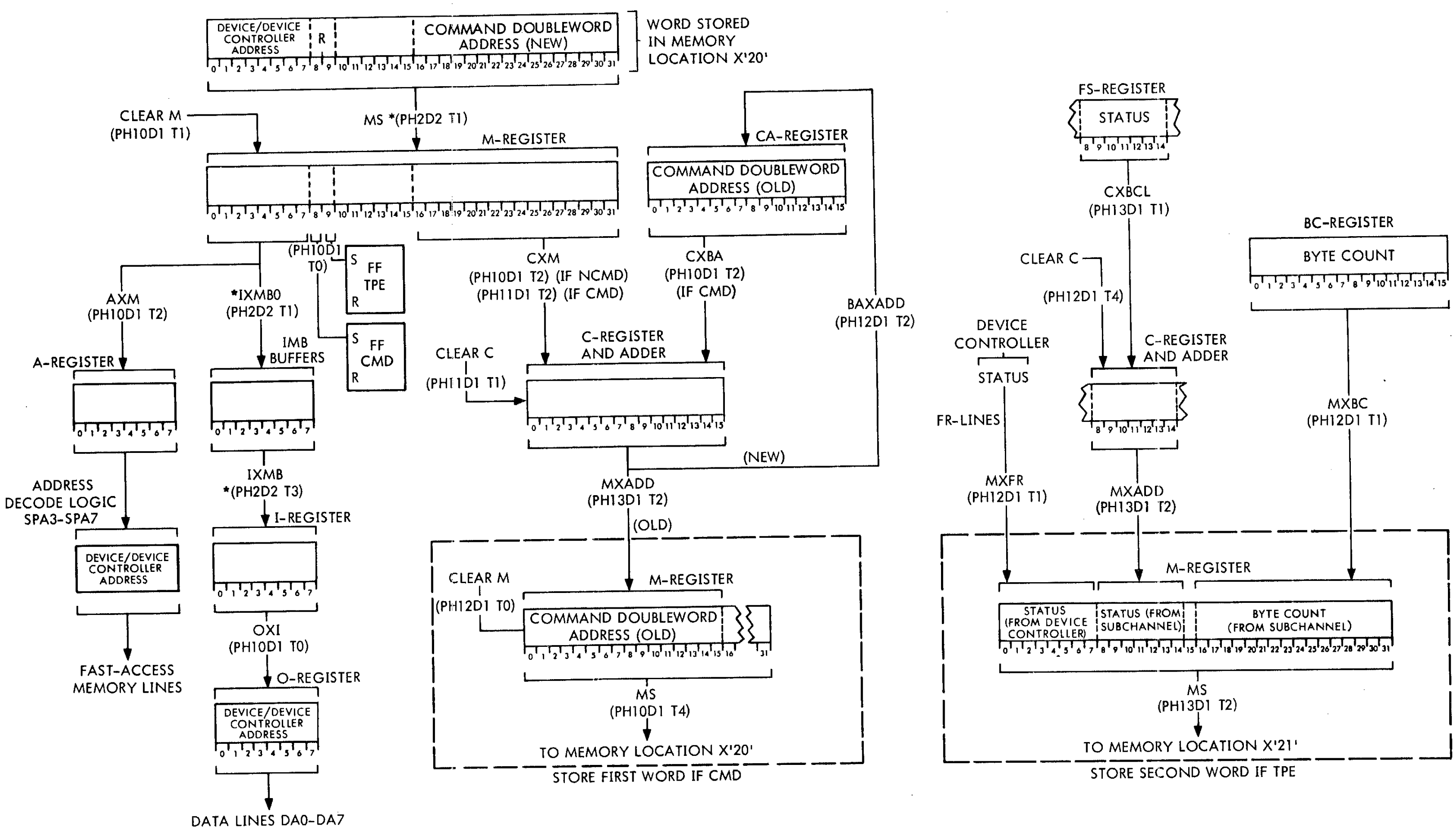
The MIOP accepts condition code information from the device controller and controls the condition code flip-flops accordingly. If the condition code specifies a successful start and I/O address recognition, the MIOP stores into the appropriate command address register the new command doubleword address that is currently in the C-register and adder. The MIOP then drops the SIO function indicator line to the device controller.

PH13D1. During this phase, the balance of the status information is set into the M-register from the FS-register. If TPE is true (R field of SIO instruction $\neq 0$) a core memory cycle is started to store the contents of the M-register into memory location X'21'.

Also, the 4-bit device address in the A-register is stored in the appropriate IS-register (bits IS4 through IS7), and the interrupt status (bits IS0 through IS3) is cleared.

PH14D1. The status is cleared from the appropriate OF-register and the flags and status are cleared from the appropriate FS-register if the condition code specifies a successful start and I/O address recognition. At the conclusion of the SIO instruction, if successful, the MIOP subchannel for the addressed device contains the following type of information:

| Register | Information |
|---|---|
| BA | Old byte address |
| CA | New command address |
| BC | Old byte count |
| FS | Zeros |
| OF | Zeros |
| IS (bits 0-3) | Zeros |
| IS (bits 4-7) | New device address |




*SIGNALS GO TRUE DURING PH2D2, HOWEVER, DATA IS NOT RECEIVED FROM MEMORY UNTIL AFTER PH2D2. PH10D1 STARTS AFTER DATA HAS BEEN RECEIVED FROM MEMORY

901515A. 333

Figure 3-33. Data Flow During an SIO Instruction

Table 3-12. SIO Instruction Phase Sequence

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D2 | Delay line D2 is started | PLED2 | = | PH1D2 NFSL NAVO CM027 + ... | Signal ME goes true when address signals IOPA0-IOPA2 compare with MIOP address switch settings |
| | | CM027 | = | CNST ME PR2 | |
| T0D2 | Set flip-flop FNT | S/FNT | = | PH1D2 T0D2 CM027 | Signal FNT implies a CPU initiated function |
| | Reset latch PH1PD2 | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Set latch PH1PD2 | PH1PD2 | = | PH1D2 T1D2 + ... | |
| T2D2 | MIOP drives SIO function indicator line | S/F3 | = | FXFN NFNC0 NFNC1 NFNC2 | Flip-flop F3 drives the SIO function indicator line when the function lines are decoded and enable signal FXFN goes true |
| | | FXFN | = | PH1T2D2 FNT | |
| | Reset flip-flop PR1 | R/PR1 | = | PH1T2D2 FNT | Prevents PR from being returned to the CPU prematurely |
| | | PR | = | PR1 NPR2 | |
| T3D2 | Reset flip-flop PR2 | R/PR2 | = | PH1PD2 T3D2 FNT | Primes PR |
| | Reset PH1D2 | PH1D2 | = | PH1D2 NT3D2 | |
| T4D2 | Set PH2D2 | S/PH2D2 | = | PH1PD2 T4D2 | Next phase in sequence |
| PH2D2 | Delay line D2 started | PLED2 | = | PH2D2 FIN CM028 + ... | Delay line is started at end of T4D2 of preceding phase if preceding I/O operation is not still in progress |
| | | CM028 | = | CM028I2 (NFI + ...) | |
| | | CM028I2 | = | FNT NF0 (NMS + ...) + ... | |
| T0D2 | A-register cleared | AX0 | = | PH2T0D2 | A-register cleared in preparation for new address from instruction |
| | | A0-A7 | = | A0-A7 NAX0 + ... | |
| | X'20' set into S-register | SX0 | = | PH2D2 T0D2 FNT NF0 | S-register set to X'20' by setting S26 and clearing all other bits |
| | | S26 | = | SX20 | |
| | Reset M- and W-register latches and flip-flop MS | MLX0 | = | MUX0 = WX0 = PH2D2 T0D2 NFT NF0 | Signals W0 through W3 define a read operation |
| | | R/MS | = | PH2T0D2 FNT NF0 | |
| | Reset flip-flops MPE, MAE, and latch PH1PD2 | R/MAE | = | R/MPE = PH2T0D2 | |
| | | PH1PD2 | = | PH1PD2 NT0D2 | |
| T1D2 | Set flip-flops LS0, LS1, LS2 | S/LS0 | = | S/LS1 = | Selects CA-, FS-, and IS-registers of appropriate subchannel |
| | | S/LS2 | = | PH2D2 FNT T1D2 | |
| | Set flip-flops CC1 and CC2 | S/CC1 | = | S/CC2 = PH2D2 FNT T1D2 | CC1 and CC2 are reset upon receipt of condition code from device controller |
| | Reset flip-flops MAR and MDR1 | R/MAR | = | R/MDR1 = PH2D2 T1D2 FNT NF0 | |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D2 (Cont.) | Set flip-flop MS | S/MS = PH2D2 T1D2 FNT NF0 NPR2 | MIOP accesses core memory location X'20' |
| | Reset flip-flops CMD, EH, EHE, IXMB, ORD OUT, TPE, TRA1, and latches TORD and ZBC | R/CMD = R/EH = R/EHE =<br>R/IXMB = R/ORD = R/OUT =<br>R/TPE = R/TRA1 = PH2T1D2<br>TORD = TORD NPH2T1D2, ZBC = ZBC NPH2T1D1 | Old information is cleared from these flip-flops |
| | Set flip-flop FP and latch PH2PD2 | S/FP = PH2T1D2<br>PH2PD2 = PH2D2 T1D2 | |
| T2D2 | Clear C-,and I-register latches and H-register flip-flops | HX0 = HUX0 = CX0 = IX0 = PH2T2D2<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ...<br>R/H0-R/H2 = HUX0 + ...<br>R/H3-R/H7 = HX0 + ...<br>I0-I7 = I0-I7 NIX0 + ... | Old information cleared from these registers by signal PH2T2D2 |
| | Reset J-register | R/J0 = R/J1 = R/J2 = PH2T2D2 | |
| | Set flip-flops K15 and SUB | S/K15 = S/SUB = PH2T2D2 | A one will be subtracted from all data that is transferred through the adder |
| T3D2 | Set flip-flop IXMB | S/IXMB = PH2PD2 T3D2 CM044 + ...<br>CM044 = FNT NF0<br>I0-I7 = IMB0-IMB7 IXMB + ...<br>IMB0-IMB7 = M0-M7 IXMB0 + ...<br>IXMB0 = NH1 NH2 | Device/device controller address gated into I-register by means of IMB buffers after it is received from core memory |
| | Set latch PH10D1 | PH10D1 = PH10D1S0 + ...<br>PH10D1S0 = PH2PD2 T3D2 CM044 | Next phase in sequence |
| | Reset flip-flop PH2D2 | R/PH2D2 = T3D2 + RESET | |
| PH10D1 | Delay line started | PLED1 = PH10D1 RSA2 CM023<br>CM023 = MDR1 + NMS | The delay line starts when the response signals have been received from core memory (MDR1 set) and the previous phase has been completed |
| T0D1 | Set latch OXI, signal OX0 goes true | OX0 = PH10D1 T0D1 + ...<br>S/OXI = PH10D1 T0D1 + ...<br>O0-O7 = NOX0 O0-O7 + OXI I0-I7 | Signal OX0 clears the O-register latches and signal IX0 gates the I-register contents (device controller address) to the O-register |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 (Cont.) | | /DA0/-/DA7/ = CD 00- CD 07 | The device controller address is therefore on the data lines |
| | Signal AXM goes true | AXM = PH10T0<br>A0-A7 = AXM M0-M7 | The device controller address (bits M0-M7) is gated to the A-register to select the designated subchannel |
| | Set flip-flops CMD and TPE | S/CMD = PH10T0 M8 + ...<br>S/TPE = PH10T0 M9 + ... | Bits M8 and M9 reflect the nature of the R field of the instruction. Signals CMD and TPE gate the first and second word of response information into core memory locations X'20' and X'21', respectively |
| | Reset flip-flop MS | R/MS = MCD1 T0D1 + ...<br>MCD1 = PH10D1 + ... | |
| | Signal WX1 goes true | WX1 = PH10T0 + ...<br>W0-W3 = W1 + ... | A full write core memory operation is defined when W0-W3 are true |
| | Second rank phase latches associated with delay line 1 are reset | PHPD1X0 = T0D1 + ... | |
| T1D1 | Reset flip-flops MAR and MDR1; signal MUX0 goes true | R/MAR = R/MDR1 = MCD1 + ...<br>MUX0 = MUX0I + ...<br>MUX0I = PH10D1 T1D1<br>M0-M15 = NMUX0 M0-M15 | The upper half of the M-register (M0-M15) and flip-flops MAR and MDR1 are reset |
| | Set latch PH10PD1 | PH10PD1 = PH10D1 T1D1 | |
| T2D1 | Reset flip-flop OXI | R/OXI = T2D1 + ... | |
| | Set flip-flop FS | S/FS = PH10D1 T2D1 NMPE NPR2 + ... | The function strobe (FS) is sent to the device controller if core memory location X'20' was read without error (NMPE) and if the CPU has not dropped the control strobe and the RESET signal is false (NPR2) |
| | Signal CXBA goes true | CXBA = PH10D1 T2D1 CMD<br>S/CMD = PH10T0 M8 | The contents of the CA-register (minus one) are gated to the C-register and adder. The CA-register presently contains the current command address plus one |

(Continued)

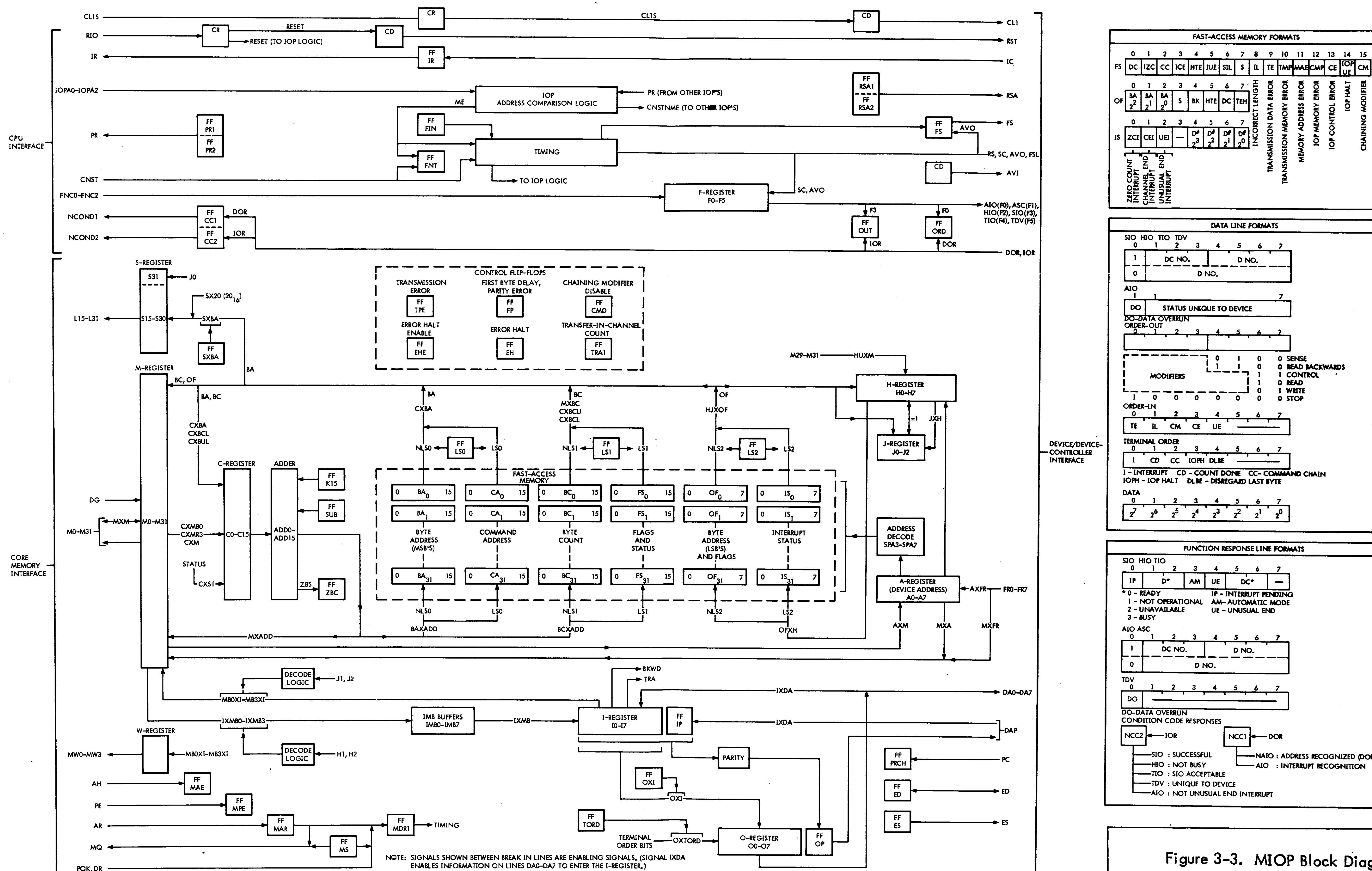



Figure 3-3. MIOP Block Diagram

901515A. 303

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T2D1 (Cont.) | Reset flip-flop K15 | R/K15 = CM006 + ...<br>CM006 = PH10D1 NCMD + ... | Information transferred through the C-register and adder will not be altered. This information will consist of the new command doubleword address currently in bits M16-M31 of the M-register if signal CMD is false |
| | Signal CXM goes true | CXM = CM006 T2D1 F3 + ... | The new command doubleword address is transferred from M16-M31 to the C-register and adder if CMD is false |
| T3D1 | Reset latch PH10D1 | PH10D1 = PH10D1 NPHD1X0<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signal MXADD goes true | MXADD = PH10PD1 T4D1 + ...<br>M0-M15 = ADD0-ADD15 MXADD | The contents of the adder are gated to M0-M15 of the M-register. If CMD is true, the adder contains the current command doubleword address minus one. If CMD is false, the adder contains the new command doubleword address supplied by the CPU by means of memory location X'20'. The M-register contents is stored in core memory location X'20' only if CMD is false |
| | Set flip-flop MS | S/MS = MSSET + ...<br>MSSET = CM043 NMPE NPR2<br>CM043 = PH10PD1 T4D1 CMD | The first word of response information is stored in core memory location X'20'. This word consists of the current command doubleword address minus one |
| | Reset flip-flop LS1 | R/LS1 = CM009 T4D1 + ...<br>CM009 = PH10PD1 NCMD + ... | The BC-register is selected |
| | Set latch PH11D1 | PH11D1 = PH11D1S0 T4D1 + ...<br>PH11D1S0 = PH10PD1 CMD | PH11D1 is the next phase if two words of response information are to be written into core memory |
| | Set latch PH12D1 | PH12D1 = CM009 T4D1 + ... | The MIOP skips PH11D1 and proceeds to PH12D1 if only one word or no words of response information are to be written into core memory |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| PH11D1 | Delay line started | PLED1 = PH11D1 + ... | Delay line 1 starts unconditionally at the end of T4D1 of the preceding phase |
| T0D1 | Reset latch PH10PD1 | PH10PD1 = PH10PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Signal CX0 goes true | CX0 = PH11D1 T1D1 + ...<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 | The current command address is cleared from the C-register and adder |
| | Set latch PH11PD1 | PH11PD1 = PH11D1 T1D1 | |
| T2D1 | Reset flip-flop K15 | R/K15 = CM006 T2D1 + ...<br>CM006 = PH11D1 + ... | The new command doubleword address will not be altered when it is stored in the CA-register (by means of the C-register and adder) |
| | Signal CXM goes true | CXM = T2D1 F3 (CM006 + ...)<br>CM006 = PH11D1 + ...<br>C0-C14 = M16-M30 CXM + ...<br>S/C15 = M31 CXM + ... | The new command doubleword address is gated to the C-register from M16-M31 of the M-register |
| T3D1 | Reset latch PH11D1 | PH11D1 = PH11D1 NPHD1X0<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Reset flip-flop LS1 | R/LS1 = CM009 T4D1<br>CM009 = PH11D1 + ... | The byte count stored in the BC-register will appear on the BC-lines |
| | Set latch PH12D1 | PH12D1 = CM009 T4D1 + ... | PH12D1 is the next phase in sequence for an SIO instruction |
| PH12D1 | Delay line started | PLED1 = PH12D1 CM023 CM026<br>CM023 = NMS + MDR1<br>CM026 = NFS + FSL | If PH12D1 is being entered from PH10D1 (MS false), the delay line starts when FSL is received from the device controller (or if no address recognition, when AVO is received instead, and FS is reset). If PH12D1 is being entered from PH11D1 (MS true), the delay line starts after the response signals have been received from core memory (MDR1 true), in addition to the response signals at the device controller interface as above |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
| --- | --- | --- | --- |
| T0D1 | Set S31 | S31 = PH12D1 T0D1 + ... | S-register is set to address core memory location X'21' when S26 is also set |
| | Reset flip-flops K15, FIN, MS, MDR1, and MAE | R/K15 = PH12T0 + ...<br>R/FIN = PH12T0 + ...<br>R/MS = MCD1 T0D1 + ...<br>MCD1 = PH12D1 + ...<br>R/MDR1 = MCD1 T0D1 + ...<br>R/MAE = PH12T0 + ... | |
| | Signals MLX0, MUX0, and WX1 go true | MLX0 = MUX0 = WX1 = PH12T0<br>M0-M15 = M0-M15 NMUX0 + ...<br>M16-M31 = M16-M31 NMLX0 + ...<br>W0-W3 = WX1 + ... | Signals MUX0 and MLX0 clear the M-register. Signals W0-W3 specify a full write operation |
| | Reset latch PH12PD1 | PH12PD1 = PH12PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Signal MXFR goes true | MXFR = PH12T1 NF0<br>M0-M7 = FR0-FR7 MXFR + ... | The status on the FR lines is gated into M0-M7 of the M-register |
| | If signal DOR is true, reset flip-flop CC1. If signal IOR is true, reset flip-flop CC2 | R/CC1 = PH12T1 DOR FS<br>R/CC2 = PH12T1 IOR FS<br>R/FS = AVO + ... | The device controller drives DOR and IOR true, indicating address recognition and a successful start |
| | Set latch SX20 | SX20 = PH12PD1<br>S26 = SX20 + ... | S-register is set to address core memory location X'21' (S31 is true) |
| | Signal MXBC goes true | MXBC = PH12T1 NF0<br>M16-M31 = BC0-BC15 MXBC + ... | The byte count part of the status information is gated to M16-M31 of the M-register from the BC-register |
| | Set flip-flop OUT | S/OUT = CM035 + ...<br>CM035 = PH12T1 F3 | Signal OUT is used to gate the conclusion of an SIO instruction |
| | Reset flip-flops MAR and MDR1 | R/MAR = MCD1 T1D1 + ...<br>MCD1 = PH12D1 + ...<br>R/MDR1 = MCD1 T1D1 + ... | |
| | Set latch PH12PD1 | PH12PD1 = PH12D1 T1D1 + ... | |
| T2D1 | Reset flip-flop FS | R/FS = PH12D1 T2D1 + ... | |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T2D1 (Cont.) | Signal BAXADD goes true | BAXADD = PH12D1 T2D1 F3 CM046<br>CM046 = NCC1 NCC2 | The new command doubleword address is stored in the CA-register if the condition code flip-flops indicate I/O address recognition and successful start |
| T3D1 | Reset flip-flop FNT | R/FNT = PH12T3 + ... | |
| | Set flip-flop LS1 | S/LS1 = CM001 + ...<br>CM001 = PH12T3 | Selects FS-register |
| | Set flip-flop PR1 | S/PR1 = PH12T3 NORD (NTPE + MPE) + ...<br>PR = PR1 NPR2 | The proceed signal PR is sent to the CPU since signal NPR2 is true |
| | Reset latch PH12D1 | PH12D1 = PH12D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ... | The C-register is cleared |
| | Signal FX0 goes true | FX0 = PH12T4 + ...<br>R/F3 = FX0 | F-register is cleared. SIO function indicator line goes false |
| | Set latch PH13D1 | PH13D1 = PH12PD1 T4D1 + ... | PH13D1 is the next in sequence for an SIO |
| | Set latch PH1D2 | PH1D2 = PH12PD1 NF1 T4D1 + ... | A new I/O operation may begin. If it does, it will proceed through PH1D2 and wait for signal FIN. Signal FIN is generated when the MIOP no longer needs access to the subchannel associated with the current operation |
| PH13D1 | Delay line D1 started | PLED1 = PH13D1 + ... | Delay line D1 starts unconditionally at the end of T4 of the preceding phase |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 | Signal CXBCL goes true | CXBCL = NCXBCLI + ...<br>CXBCLI = NPH13D1 + NT0D1 + ... | Previously stored status information from bits FS8-FS14 is gated to C8-C14 of the C-register for subsequent storage in core memory location X'21' |
| | Bits A4-A7 are transferred to H4-H7 | S/H4-S/H7 = A4-A7 PH13T0 + ... | The device address specified by the SIO instruction set into the H-register for subsequent storage in the IS-register |
| | Reset latch PH13PD1 | PH13PD1 = PH13PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set latch PH13PD1 | PH13PD1 = PH13D1 T1D1 | |
| T2D1 | Signal MXADD goes true | MXADD = PH13D1 T2D1 OUT + ...<br>M0-M15 = ADD0-ADD15 MXADD + ... | Status presently in bits C8-C14 (ADD8-ADD14) is gated to M8-M14 of the M-register. (Bits C0-C7 are all zeros.) The second word of response information has now been assembled in the M-register |
| | Set flip-flop MS | S/MS = PH13D1 T2D1 CM041 OUT + ...<br>CM041 = NPR2 NMPE TPE | Signal MS starts the memory cycle that stores the second word in location X'21' of core memory |
| | Signal OFXH goes true | OFXH = PH13D1 T2D1 OUT CM046 + ...<br>CM046 = NCC1 NCC2 | Contents of H-register are set into IS-register. IS0-IS3 is cleared. IS4-IS7 contain device address |
| T3D1 | Reset latch PH13D1 | PH13D1 = PH13D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signals CX0, HUX0, and HX0 go true | HUX0 = HX0 = PH13PD1 T4D1<br>CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1 | The C- and H-registers are cleared |
| | Reset flip-flop LS2 | R/LS2 = PH13PD1 T4D1 + ... | Selects OF-register |
| | Set latch PH14D1 | PH14D1 = PH13PD1 T4D1 + ... | PH14D1 is the next phase in sequence for an SIO |
| PH14D1 | Delay line started | PLED1 = PH14D1 + ... | Delay line D1 starts unconditionally at the end of T4 of the preceding phase |

(Continued)

Table 3-12. SIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 | Reset latch PH14PD1 | PH14PD1 | = | PH14PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Signal BXADD goes true | BCXADD | = | CM022 + ... | The flags and status are cleared from the FS-register. The C-register and adder, which contain zeros, are transferred to the FS-register |
| | | CM022 | = | PH14D1 T1D1 CM046 + ... | |
| | | CM046 | = | NCC1 NCC2 | |
| | Signal OFXH goes true | OFXH | = | CM022 + ... | The byte address 3 LSB's and operating flags are cleared from the OF-register. The H-register, which contains zeros, is transferred to the OF-register |
| | Set latch PH14PD1 | PH14PD1 | = | PH14D1 T1D1 + ... | |
| T2D1 | Set flip-flop FIN | S/FIN | = | PH14D1 T2D1 + ... | The current operation no longer needs the MIOP's fast access memory |
| T3D1 | Reset latch PH14D1 | PH14D1 | = | PH14D1 NPHD1X0 + ... | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |

3-38 <u>HIO, TIO, and TDV Instructions</u>

The phases for the HIO, TIO, and TDV instructions follow the same sequence as for an SIO instruction (figure 3-32). As in the case of an SIO instruction, the R field coding of the HIO, TIO, and TDV instructions controls the state of signal CMD, which determines whether or not PH11D1 is included in the sequence.

<u>PH1D2 AND PH2D2</u>. The functions that occur during these two phases are generally similar to an SIO.

<u>PH10D1</u>. The functions that occur during this phase are similar to an SIO except for the manner in which the response information is manipulated when signal CMD is false.

<u>PH11D1</u>. During this phase, the MIOP reads status information out of the FS-register for subsequent storage in core memory. During an SIO the MIOP is manipulating the command doubleword address which is to be stored in the CA-register.

<u>PH12D1</u>. During this phase, the MIOP clears the M-register and proceeds to load it with status information from the device controller by means of the FR lines and the status and byte count from the appropriate subchannel. If the coding of the R field of the instruction was not zero (TPE true), the MIOP generates the memory start signal to store the contents of the M-register in core memory location X'21'.

The MIOP accepts condition code information from the device controller and controls the condition code flip-flops accordingly. It then drops the function indicator line (TIO, HIO, or TDV) to the device controller.

PH13D1 AND PH14D1. The functions performed during these two phases for an HIO, TIO, or TDV instruction are insignificant. The information in the addressed subchannel is not altered.

Note

The phase sequence table for HIO, TIO, and TDV instructions during PH1D2 and PH2D2 are identical to the table for an SIO instruction, with the following exceptions that occur during PH1D2 T2D2:

a. During an HIO instruction, function indicator line HIO is driven true when flip-flop F2 is set (S/F2 = FXFN NFNC0 FNC1 FNC2).

b. During a TIO instruction, function indicator line TIO is driven true when flip-flop F4 is set (S/F4 = FXFN NFNC0 NFNC1 FNC2).

c. During a TDV instruction, function indicator line TDV is driven true when flip-flop F5 is set (S/F5 = FXFM NFNC0 FNC1 NFNC2).

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| PH10D1 | Delay line started | PLED1 = PH10D1 RSA2 CM023<br>CM023 = MDR1 + NMS | The delay line starts when the response signals have been received from core memory (MDR1 set) and if the previous phase has been completed |
| T0D1 | Set flip-flop OX1; signal OX0 goes true | OX0 = PH10D1 T0D1 + ...<br>S/OXI = PH10D1 T0D1 + ...<br>O0-O7 = NOX0 O0-O7 + OXI I0-I7<br>/DA0/-/DA7/ = CD O0-CD O7 | Signal OX0 clears the O-register latches, and signal OX0 gates the I-register contents (device controller address) to the O-register. The device controller address is, therefore, on the data lines |
| | Signal AXM goes true | AXM = PH10T0<br>A0-A7 = AXM M0-M7 | The device controller address (bits M0-M7) is gated to the A-register to select the designated subchannel |
| | If bit M8 is true, set flip-flop CMD; if bit M9 is true, set flip-flop TPE | S/CMD = PH10T0 M8 + ...<br>S/TPE = PH10T0 M9 + ... | Bits M8 and M9 reflect the nature of the R field of the instruction. Signals CMD and TPE gate the first and second status word into core memory |
| | Reset flip-flop MS | R/MS = MCD1 T0D1 + ...<br>MCD1 = PH10D1 + ... | |
| | Signal WX1 goes true | WX1 = PH10T0 + ...<br>W0-W3 = W1 + ... | A full write core memory operation is defined when W0-W3 are true |
| | Second rank phase latches associated with delay line 1 are reset | PHPD1X0 = T0D1 | |

(Continued)

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 | Reset flip-flops MAR and MDR1; signal MUX0 goes true | R/MAR = R/MDR1 = MCD1 + ...<br>MUX0 = MUX0I + ...<br>MUX0I = PH10D1 T1D1<br>M0-M15 = NMUX0 M0-M15 | The upper half of the M-register (M0-M15) and flip-flops MAR and MDR1 are reset |
| | Set PH10PD1 | PH10PD1 = PH10D1 T1D1 | |
| T2D1 | Reset flip-flop OXI | R/OXI = T2D1 + ... | |
| | Set flip-flop FS | S/FS = PH10D1 T2D1 NMPE NPR2 | The function strobe (FS) is sent to the device controller if core memory location X'20' was read without error (NMPE) and if the CPU has not dropped the control strobe, and the RESET signal is false (NPR2) |
| | Signal CXBA goes true | CXBA = PH10D1 T2D1 CMD<br>S/CMD = PH10T0 M8 | The contents of the CA-register (minus one) are gated to the C-register and adder. The CA-register presently contains the current command address plus one |
| | Reset flip-flop K15 | R/K15 = CM006 + ...<br>CM006 = PH10D1 NCMD + ... | Information transferred through the C-register and adder will not be altered. This information will consist of the current status and byte count if CMD is false |
| | Signal CXBCL goes true | CXBCL = CM006 T2D1 NF3<br>CM006 = PH10D1 NCMD + ... | Signal CXBCL gates the status from bits 8-14 of the FS-register to bits 8-14 of the C-register |
| | Reset latch PH10D1 | PH10D1 = PH10D1 NPHD1X0<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signal MXADD goes true | MXADD = PH10PD1 T4D1 + ...<br>M0-M15 = ADD0-ADD15 MXADD | The contents of the adder are gated to M0-M15 of the M-register. If CMD is true, the adder contains the current command doubleword address (from the CA-register) minus one. If CMD is false, the adder contains status from the FS-register. The M-register contents are stored in core memory location X'20' during PH10D1 only if CMD is true |

(Continued)

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 (Cont.) | Set flip-flop MS | S/MS = MSSET + ...<br>MSSET = CM043 NMPE NPR2<br>CM043 = PH10PD1 T4D1 CMD | The first word of response information is stored in core memory location X'20'. This word consists of the current command doubleword address less one |
| | Reset flip-flop LS1 | R/LS1 = CM009 T4D1 + ...<br>CM009 = PH10PD1 NCMD + ... | The BC-register is selected |
| | Set latch PH11D1 | PH11D1 = PH11D1S0 T4D1 + ...<br>PH11D1S0 = PH10PD1 CMD | PH11D1 is the next phase if two words of response information are to be written into core memory |
| | Set latch PH12D1 | PH12D1 = CM009 T4D1 + ... | The MIOP skips PH11D1 and proceeds to PH12D1 if only one word or no words of response information are to be written into core memory |
| PH11D1 | Delay line started | PLED1 = PH11D1 + ... | Delay line 1 starts unconditionally at the end of T4D1 of the preceding phase |
| T0D1 | Reset latch PH10PD1 | PH10PD1 = PH10PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Signal CX0 goes true | CX0 = PH11D1 T1D1 + ...<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 | The C-register is cleared |
| | Set latch PH11PD1 | PH11PD1 = PH11D1 T1D1 | |
| T2D1 | Reset flip-flop K15 | R/K15 = CM006 T2D1 + ...<br>CM006 = PH11D1 + ... | |
| | Signal CXBCL goes true | CXBCL = CM006 T2D1 NF3 + ...<br>CM006 = PH11D1 + ... | Signal CXBCL gates the status information from bits 8-14 of the FS-register to bits 8-14 of the C-register |
| T3D1 | Reset latch PH11D1 | PH11D1 = PH11D1 NPHD1X0<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Reset flip-flop LS1 | R/LS1 = CM009 T4D1<br>CM009 = PH11D1 + ... | The byte count stored in the BC-register will appear on the BC lines |
| | Set latch PH12D1 | PH12D1 = CM009 T4D1 + ... | PH12D1 is the next phase in sequence |

(Continued)

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| PH12D1 | Delay line started | PLED1 = PH12D1 CM023 CM026<br>CM023 = NMS + MDR1<br>CM026 = NFS + FSL | If PH12D1 is being entered from PH10D1 (MS false), the delay line starts when FSL is received from the device controller (or if no address recognition, when AVO is received instead, and FS is reset). If PH12D1 is being entered from PH11D1 (MS true), the delay line starts after the response signals have been received from core memory (MDR1 true), in addition to the response signals at the device controller interface as above |
| T0D1 | Set latch S31 | S31 = PH12D1 T0D1 + ... | S-register is set to address core memory location X'21' when S26 is also set |
| | Reset flip-flops K15, FIN, MS, MDR1, and MAE | R/K15 = PH12T0 + ...<br>R/FIN = PH12T0 + ...<br>R/MS = MCD1 T0D1 + ...<br>MCD1 = PH12D1 + ...<br>R/MDR1 = MCD1 T0D1 + ...<br>R/MAE = PH12T0 + ... | |
| | Signals MLX0, MUX0, and WX1 go true | MLX0 = MUX0 = WX1 = PH12T0<br>M0-M15 = M0-M15 NMUX0 + ...<br>M16-M31 = M16-M31 NMLX0 + ...<br>W0-W3 = WX1 + ... | Signals MUX0 and MLX0 clear the M-register. Signals W0-W3 specify a full write operation |
| | Reset latch PH11PD1 | PH11PD1 = PH11PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Signal MXFR goes true | MXFR = PH12T1 NF0<br>M0-M7 = FR0-FR7 MXFR + ... | The status on the FR lines is gated into M0-M7 of the M-register |
| | If signal DOR is true, reset flip-flop CC1. If signal IOR is true, reset flip-flop CC2 | R/CC1 = PH12T1 DOR FS<br>R/CC2 = PH12T1 IOR FS<br>R/FS = AVO + ... | The device controller drives DOR and IOR true, indicating address recognition and a successful start |
| | Set latch SX20 | SX20 = PH12PD1<br>S26 = SX20 + ... | S-register is set to address core memory location X'21' (S31 is true) |
| | Signal MXBC goes true | MXBC = PH12T1 NF0<br>M16-M31 = BC0-BC15 MXBC + ... | The byte count part of the response information is gated to M16-M31 of the M-register from the BC-register |

(Continued)

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 (Cont.) | Signal MXADD goes true | MXADD = PH12D1 T1D1 NF0 NF3<br>M0-M15 = MXADD ADD0-ADD15 + ... | Signal MXADD gates the status information currently in the adder to bits 0-15 of the M-register (adder bits 8-14 contain status information taken from the FS-register, and bits 0-7 and 15 equal zero) |
| | Reset flip-flops MAR | R/MAR = MCD1 T1D1 + ...<br>MCD1 = PH12D1 + ...<br>R/MDR1 = MCD1 T1D1 + ... | |
| | Set flip-flop MS | S/MS = PH12T1 NF3 CM041 + ...<br>CM041 = NPR2 NMPE TPE<br>S/MPE = MS PE<br>S/TPE = PH10T0 M9 + ... | Starts a core memory write operation that stores the contents of the M-register in core memory location X'21' |
| | Set PH12PD1 | PH12PD1 = PH12D1 T1D1 + ... | |
| T2D1 | Reset flip-flop FS | R/FS = PH12D1 T2D1 + ... | |
| T3D1 | Reset flip-flop FNT | R/FNT = PH12T3 + ... | |
| | Set flip-flop LS1 | S/LS1 = CM001 + ...<br>CM001 = PH12T3 | Selects FS-register |
| | Set flip-flop PR1 | S/PR1 = PH12T3 NORD (NTPE + MPE) + ...<br>PR = PR1 NPR2 | The proceed signal PR is sent to the CPU since signal NPR2 is true |
| | Reset latch PH12D1 | PH12D1 = PH12D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ... | The C-register is cleared |
| | Signal FX0 goes true | FX0 = PH12T4 + ...<br>R/F3 = FX0 | F-register is cleared. Appropriate function indicator line goes false |
| | Set latch PH13D1 | PH13D1 = PH12PD1 T4D1 + ... | PH13D1 is the next in sequence for HIO, TIO, and TDV instructions |

(Continued)

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 (Cont.) | Set latch PH1D2 | PH1D2 = PH12PD1 NF1 T4D1 + ... | A new I/O operation may begin. If it does, it will proceed through PH1D2 and wait for signal FIN. Signal FIN is generated when the MIOP no longer needs access to the subchannel associated with the current operation |
| PH13D1 | Delay line D1 started | PLED1 = PH13D1 + ... | Delay line D1 starts unconditionally at the end of T4 of the preceding phase |
| T0D1 | Signal CXBCL goes true | CXBCL = NCXBCLI + ...<br>CXBCLI = NPH13D1 + NT0D1 + ... | Previously stored status information from bits FS8-FS14 is gated to C8-C14 of the C-register |
| | Bits A4-A7 are transferred to H4-H7 | S/H4-S/H7 = A4-A7 PH13T0 + ... | The device address specified by the instruction is set into the H-register |
| | Reset latch PH13PD1 | PH13PD1 = PH13PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set latch PH13PD1 | PH13PD1 = PH13D1 T1D1 | |
| | Reset latch PH13D1 | PH13D1 = PH13D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signals CX0, HUX0, and HX0 go true | HUX0 = HX0 = PH13PD1 T4D1<br>CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1 | The C- and H-registers are cleared |
| | Reset flip-flop LS2 | R/LS2 = PH13PD1 T4D1 + ... | Selects OF-register |
| | Set latch PH14D1 | PH14D1 = PH13PD1 T4D1 + ... | PH14D1 is the next phase in sequence |
| PH14D1 | Delay line started | PLED1 = PH14D1 + ... | Delay line D1 starts unconditionally at the end of T4 of the preceding phase |
| T0D1 | Reset latch PH14PD1 | PH14PD1 = PH14PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set latch PH14PD1 | PH14PD1 = PH14D1 T1D1 + ... | |
| T2D1 | Set flip-flop FIN | S/FIN = PH14D1 T2D1 + ... | The current operation no longer needs the IOP's fast access memory |

(Continued)

Table 3-13. HIO, TIO, and TDV Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T3D1 | Reset latch PH14D1 | PH14D1 | = | PH14D1 NPHD1X0 + ... | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |

3-39 AIO Instruction

The AIO instruction phases follow the sequence shown in figure 3-34.

PH1D2. During this phase the MIOP determines whether the operation is being initiated by a device controller or the CPU. Since during an AIO instruction, it is CPU initiated, the MIOP decodes the function code lines and drives the AIO function indicator line at the device controller interface.

PH2D2. During this phase the MIOP clears various registers and flip-flops, and drives the function strobe line at the device controller interface.

PH12D1. The highest priority device controller with an interrupt pending places its address on the function response lines and its status information on the data lines. The MIOP loads this information, along with its own address, into the M-register. An X'20' is forced into the S-register.

PH13D1. The 4-bit device address stored in the IS-register is compared with the device portion of the address from the device controller if a multiunit device controller is supplying its address. If the address compares, or if a single-unit device controller is offering its address, the MIOP loads the status information it has previously stored in its subchannel into the M-register and stores the contents of the M-register into core memory location X'20'. Also, the 4-bit device address from the responding device controller is stored in the subchannel, and the interrupt status is cleared.

PH14D1. The functions performed during this phase are insignificant. The information in the addressed subchannel is not altered except as noted during PH13D1.

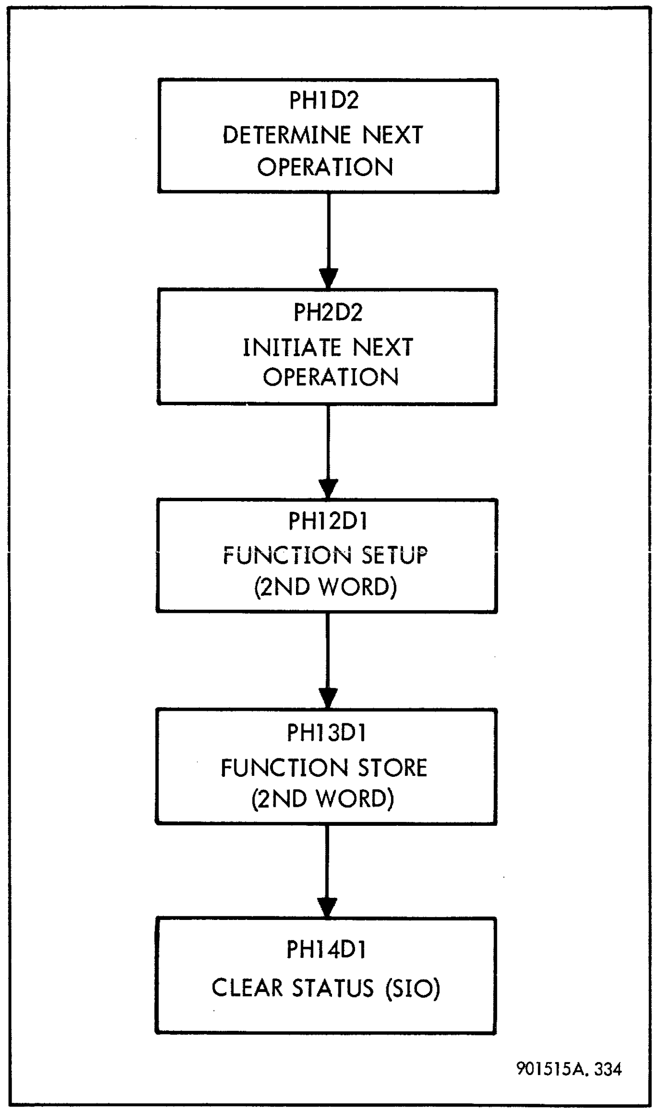



Figure 3-34. Phase Diagram for an AIO Instruction

Table 3-14. AIO Instruction Phase Sequence

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D2 | Delay line D2 started | PLED2 | = | CNST ME PR2 + ... | Signal ME goes true when the function is an AIO and this is the highest priority MIOP with an interrupt call pending |
| T0D2 | Set flip-flop FNT | S/FNT | = | PH1D2 T0D2 CM027 | FNT implies a CPU-initiated function |
| | | CM027 | = | CNST ME PR2 | |
| | Reset latch PH1PD2 | PH1PD2 | = | PH1PD = NT0D2 | |
| T1D2 | Set latch PH1PD2 | PH1PD2 | = | PH1D2 T1D2 | |
| T2D1 | Function indicator line AIO driven true | S/F0 | = | FXFN AIOFN | Signal FXFN enables setting of flip-flop F0, which drives the AIO function indicator line |
| | | AIOFN | = | FNC0 FNC1 NFNC2 | |
| | Reset flip-flop PR1 | R/PR1 | = | PH1T2D2 FNT | Prevents PR from being returned to the CPU prematurely |
| | | PR | = | PR1 NPR2 | |
| T3D1 | Reset flip-flop PR2 | R/PR2 | = | PH1PD2 T3D2 FNT | Primes PR |
| | Reset latch PH1D2 | PH1D2 | = | PH1D2 NT3D2 | |
| T4D2 | Set PH2D2 | S/PH2D2 | = | PH1PD2 T4D2 | Next phase in sequence |
| PH2D2 | Delay line D2 started | PLED2 | = | FIN NF1 F0 NRSA2 + ... | Delay line D2 is started at the end of T4 of the preceding phase if the preceding operation is not still in progress |
| T0D2 | A-register cleared | AX0 | = | PH2T0D2 | A-register cleared in preparation for new address from device controller |
| | Set flip-flop FS | S/FS | = | PH2T0D2 F0 + ... | The function strobe line at the device controller interface is driven true |
| | Reset flip-flops MPE, MAE, and latch PH2PD2 | R/MAE | = | R/MPE = PH2T0D2 | |
| | | PH2PD2 | = | PH2PD2 NT0D2 | |
| T1D2 | Set flip-flops LS0, LS1, and LS2 | S/LS0 | = | S/LS2 = S/LS2 = PH2D2 FNT T1D2 | Selects CA-, FS-, and IS-registers of appropriate subchannel |
| | Set flip-flops CC1 and CC2 | S/CC1 | = | S/CC2 = PH2D2 FNT T1D2 | Flip-flops CC1 and CC2 are reset upon receipt of condition code information from device controller |

(Continued)

Table 3-14. AIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D2 (Cont.) | Reset flip-flops CMD, EH, EHE, IXMB, ORD OUT, TORD, TPE, TRA1, and ZBC | R/CMD = R/EH = R/EHE = R/IXMB = R/ORD = R/OUT = R/TORD = R/TPE = R/TRA1 = R/ZBC = PH2T1D2 | Old information is cleared from these flip-flops |
| | Set flip-flop FP | S/FP = PH2T1D2 | |
| | Set latch PH2PD2 | PH2PD2 = PH2D2 T1D2 | |
| T2D2 | Clear C-, H-, and I-registers | HX0 = HUX0 = CX0 = IX0 = PH2T2D2 | Old information is cleared from these registers |
| | Reset J-register | R/J0 = R/J1 = R/J2 = PH2T2D2 | |
| | Set flip-flops K15 and SUB | S/K15 = S/SUB = PH2T2D2 | A one will be subtracted from all data that is transferred through the adder |
| | Signal OX0 goes true | OX0 = PH2T2D2 NCM044 + ...<br>NCM044 = NFNT + F0 | The O-register is cleared in preparation for receipt of status information from device controller |
| | Set flip-flops K15 and SUB | S/K15 = S/SUB = PH2T2D2 | A one will be subtracted from all data transferred through the adder |
| T3D2 | Set latch PH12D1 | PH12D1 = PH12D1S0 + ...<br>PH12D1S0 = PH2PD2 T3D2 D0 | Next phase in sequence |
| | Reset PH2D2 | R/PH2D2 = T3D2 + RESET | |
| PH12D1 | Delay line D1 started | PLED1 = PH12D1 CM023 CM026<br>CM023 = NMS + MDR1<br>CM026 = NFS + FSL | The delay line is started when signal FSL is received at the device controller interface |
| T0D1 | Reset flip-flops K15, FIN, MS, MDR1, and MAE | R/K15 = PH12T0 + ...<br>R/MS = MCD1 T0D1 + ...<br>MCD1 = PH12D1 + ...<br>R/MDR1 = MCD1 T0D1 + ...<br>R/MAE = PH12T0 + ... | |
| | Signals MLX0, MUX0, and WX1 go true | MLX0 = MUX0 = WX1 = PH12T0<br>M0-M15 = M0-M15 NMUX0 + ...<br>M16-M31 = M16-M31 NMLX0 + ...<br>W0-W3 = WX1 + ... | Signals MUX0 and MLX0 clear the M-register. Signals W0-W3 specify a full write operation |
| | Reset latch PH12PD1 | PH1PD1 = PH12PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 RESET | |

(Continued)

Table 3-14. AIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 | Signals AXFR and IXDA go true | AXFR = IXDA = CM010 + ...<br>CM010 = PH12D1 T1D1 F0<br>A0-A7 = FR0-FR7 AXFR + ...<br>I0-I7 = DA0-DA7 IXDA + ... | Signal AXFR gates the address into the A-register. Signal IXDA gates the status into the I-register |
| | If signal DOR is true, reset flip-flop CC1. If signal IOR is true, reset flip-flop CC2 | R/CC1 = PH12T1 DOR FS<br>R/CC2 = PH12T1 IOR FS<br>R/FS = AVO + ... | The device controller drives the DOR and IOR lines true to indicate I/O interrupt recognition and normal interrupt conditions, respectively |
| | Set SX20 | SX20 = PH12PD1<br>S26 = SX20 + ... | S-register is set to address core memory location X'20' |
| | Signal SX0 goes true | SX0 = PH12PD1 T1D1 F0 + ... | All S-register bits except S26 are reset |
| | Set flip-flop ORD | S/ORD = CM010 | Signal ORD gates the conclusion of an AIO |
| | Reset flip-flops MAR and MDR1 | R/MAR = R/MDR1 = MCD1 T1D1 + ...<br>MCD1 = PH12D1 + ... | |
| | Set latch PH12PD1 | PH12PD1 = PH12PD1 T1D1 + ... | |
| T2D1 | Reset flip-flop FS | R/FS = PH12D1 T2D1 + ... | |
| T3D1 | Reset flip-flop FNT | R/FNT = PH12T3 + ... | |
| | Set flip-flop LS1 | S/LS1 = CM001 + ...<br>CM001 = PH12T3 | Selects FS-register |
| | Signals MXA and MBOXI go true | MXA = PH12T3 ORD<br>MBOXI = MXA + ...<br>M21-M23 = MXA UN0-UN2 + ...<br>M24-M31 = MXA N0-A7 + ...<br>M0-M7 = MBOXI I0-I7 + ... | Address and status is gated into M-register by signals MXA and MBOXI |
| | Reset latch PH12D1 | PH12D1 = PH12D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ... | C-register is cleared |

(Continued)

Table 3-14. AIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 (Cont.) | Signal FX0 goes true | FX0 = PH14T4 + ...<br>R/F0 = FX0 | F-register is cleared. AIO function indicator line goes false |
| | Set latch PH13D1 | PH13D1 = PH12PD1 T4D1 + ... | Next phase in sequence |
| | Set latch PH1D2 | PH1D2 = PH12PD1 T4D1 NF1 + ... | A new I/O operation may proceed through PH1D2. When the current operation sets flip-flop FIN, the new operation can proceed past PH1D2 |
| PH13D1 | Delay line D1 is started | PLED1 = PH13D1 + ... | Delay line D1 is started unconditionally at the end of T4 of the preceding phase |
| T0D1 | Signal CXBCL goes true | CXBCL = NCXBCLI + ...<br>CXBCLI = NPH13D1 + NT0D1 + ... | Previously stored status information from bits FS8-FS14 is gated to C8-C14 of the C-register for subsequent storage in core memory location X'20' |
| | Bits A4-A7 are transferred to H4-H7 | S/H4-S/H7 = A5-A7 PH13T0 + ... | Device address is set into the H-register |
| | Signal MXIS goes true | MXIS = PH13D1 T0D1 CM015<br>CM015 = ORD NCC1 (NA0 + CM015-0)<br>CM015-0 = (NA4 + OF4) (A4 + NOF4) (NA5 + OF5) (NA5 + NOF5) (NA6 + OF6) (NA6 + NOF6) (NA7 + NOR7) (A7 + NOF7)<br>M8-M9 = BC8-BC9 MXIS + ...<br>M10-M12 = OF1-OF2 MXIS + ... | If a device controller with a single device has sent its address to the MIOP (NA0 true), or if the stored device address compares with the address offered by the device controller, signal MXIS will gate the status from FS8, FS9, IS0-IS2 to the M-register |
| | Set flip-flop CC2 | S/CC2 = CM015 PH13T0 OF2 | If the conditions exist that cause signal MXIS to go true in addition to signal OF2 true, an unusual condition interrupt is specified by condition code flip-flop CC2 being set. Signal OF2 is true if the unusual end interrupt status bit is true |
| | Set flip-flop TRA1 | S/TRA1 = PH13D1 CM015 | If signal TRA1 is true during PH13D1 T2D1, the 4-bit device address will be stored in the subchannel for the responding device controller |

(Continued)

Table 3-14. AIO Instruction Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 (Cont.) | Set flip-flop MS | S/MS = PH13T0 NPR2 ORD + ... | The contents of the M-register (status, MIOP address, and device controller address) are stored in core memory location X'20' |
| | Reset PH13PD1 | R/PH13PD1 = PH13PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set PH13PD1 | S/PH13PD1 = PH13D1 T1D1 | |
| T2D1 | Signal OFXH goes true | OFXH = OFXHI + ...<br>OFXHI = PH13D1 T2D1 TRA1 + ... | The 4-bit device address is set into the IS-register (bits 4-7), and the interrupt status is cleared from the IS-register (bits 0-2) |
| T3D1 | Reset latch PH13D1 | PH13D1 = PH13D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signals HUX0, HX0, and CX0 go true | HUX0 = HX0 = PH13PD1 T4D1<br>CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1 | The H- and C-registers are cleared |
| | Reset flip-flop LS2 | R/LS2 = PH13PD1 T4D1 + ... | Selects OF-register |
| | Set latch PH14D1 | PH14D1 = PH13PD1 T4D1 + ... | Next phase in sequence |
| PH14D1 | Delay line D1 started | PLED1 = PH14D1 + ... | Delay line D1 starts unconditionally at the end of T4 of the preceding phase |
| T0D1 | Reset latch PH14PD1 | PH14PD1 = PH14PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set latch PH14PD1 | PH14PD1 = PH14PD1 T1D1 + ... | |
| T2D1 | Set flip-flop FIN | S/FIN = PH14D1 T2D1 + ... | The current operation no longer needs the MIOP's fast access memory |
| T3D1 | Reset latch PH14D1 | PH14D1 = PH14D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ... | The C-register is cleared |

3-40 Order-Out Service Cycle

The order-out service cycle phases follow the sequence shown in figure 3-35. The primary purpose of the order-out service cycle is to fetch the command doubleword specified by the MIOP command address register, send the order bits of the command to the connected device controller, and store the remaining information encoded in the command in the subchannel for the connected device controller.

Operations performed during PH4D1 and PH5D1 depend on whether or not a transfer in channel command is encountered in the command list when the MIOP fetches the command doubleword. If the first word of the command fetched during PH4D1 is a transfer in channel command, TRA will go true. If TRA is true, at the end of PH5D1 the MIOP will repeat PH4D1. This time the command fetched during PH4D1 will be the one specified by the command address field of the transfer in channel command. Therefore, if a transfer in channel command is encountered by the MIOP, the command list will be accessed for two commands (the transfer in channel command and the command specified by its address field) during one order-out service cycle. Normally, only one command is fetched during one order-out service cycle. The portion of a command list involving a transfer in channel might appear as follows:

| Location in Command List | Command |
|---|---|
| C | Write |
| C + 2 | Transfer in channel (to C) |
| C + 4 | Stop |

Command C is executed in the normal manner, that is, PH4D1 and PH5D1 are not repeated during this service cycle. During the next order-out service cycle, the transfer in channel command (C + 2) is fetched during PH4D1. The command address encoded in the transfer in channel command is set into the CA-register, and TRA goes true. If TRA is true, during PH5D1 the MIOP repeats PH4D1. This time, the command fetched during PH4D1 (command C) is the command specified by the address field of the transfer in channel command. PH5D1 is also repeated, and the MIOP sequences through PH6D1 and PH7D1 in the normal manner. Command C is iterated until the device sets chaining modifier bit CM by means of an order-in service cycle following the data transfers. If CM is true during an order-out service cycle, the MIOP sequences through PH4D1 twice before it progresses to PH5D1. The first time no core memory access is made, but the command address in the CA-register is incremented by one. The second time, the first word of command C + 4 is fetched. Transfer in channel command C + 2 is skipped; therefore, the MIOP no longer branches back to command C.

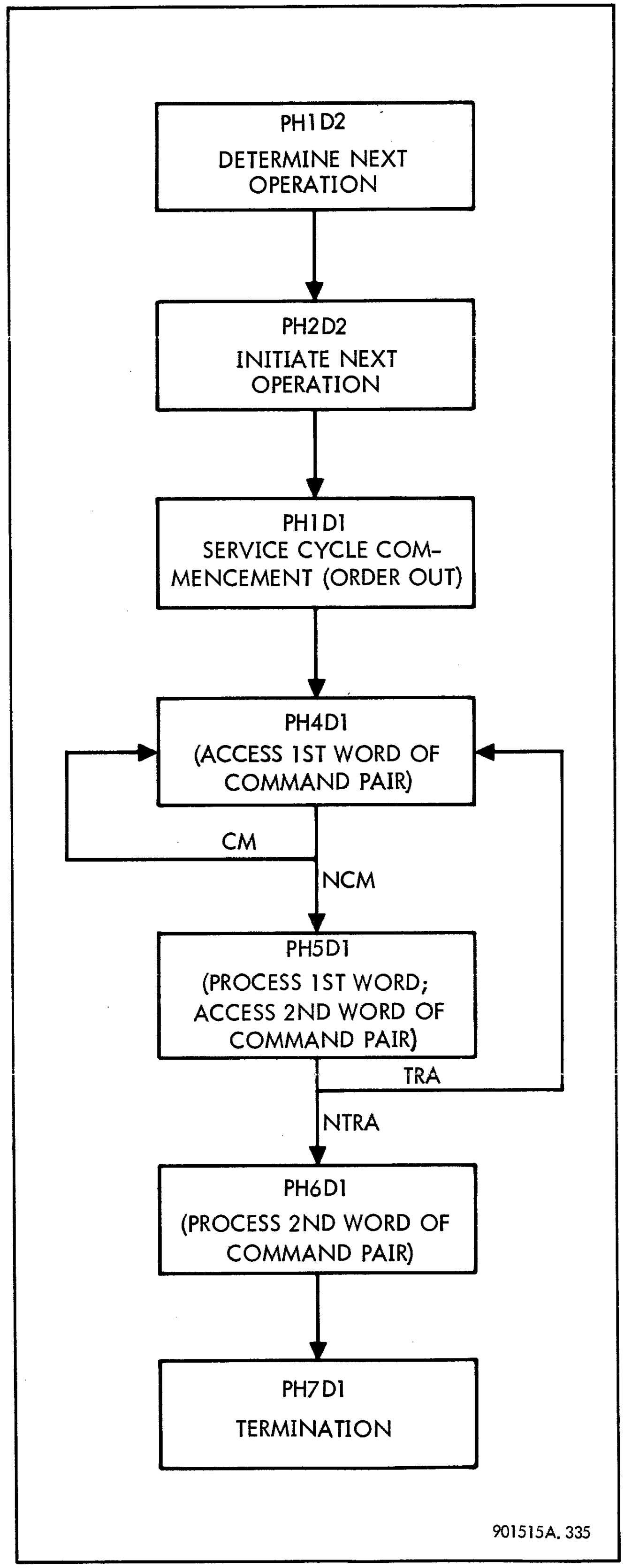



Figure 3-35. Phase Diagram for an Order-Out Service Cycle

PH1D2. During this phase the MIOP determines whether the operation is being initiated by a service call from a device controller or the CPU. Since the operation is an order-out service cycle, the service call from the device controller causes the MIOP to drive the acknowledge service call function indicator line and the function strobe line at the device controller interface.

PH2D2. The delay line starts after the device controller responds to the function strobe. The MIOP clears certain flip-flops and registers, and prepares other logic to receive the command doubleword from core memory. The address that the responding device controller has placed on the FR lines is set into the A-register for purposes of selecting the subchannel associated with that controller.

PH1D1. During this phase, the MIOP controls the write-byte lines to perform a read operation from core memory, drops the acknowledge service call line at the device controller interface, and controls the end data and end service flip-flops so that a terminal order will be included in the current service cycle.

PH4D1. The following functions are performed during this phase:

a. The command address in the CA-register is incremented by one.

b. If the chaining modifier (CM) is false, or if a transfer in channel command had been accessed the first time through PH4D1 and this is the second time through, the MIOP accesses the core memory location specified by the CA-register (before it is incremented).

c. If the chaining modifier is true, and this is the first time through PH4D1, no memory access is made. The command address, however, is incremented by one. At the end of PH4D1, the MIOP immediately repeats PH4D1. This time a memory access is made; one command doubleword will have been skipped, however, since the command address was incremented (the first time through) with no memory access.

PH5D1. The order bits of the command doubleword accessed from core memory are transferred into the O-register and are therefore present on the data lines. A test of the I-register is made to determine if the command doubleword specified a transfer in channel command. If it did, the address field of the transfer in channel command is set into the CA-register (see figure 3-36). The MIOP then sequences back to PH4D1 and makes a core memory access for the first word of the command doubleword specified by this address (see figure 3-37).

The MIOP then sequences through PH5D1 again (during the current service cycle) and accesses the second word of the command doubleword.

If during a test of the O-register it is determined that the first word accessed during PH4D1 was not a transfer in channel command, the MIOP makes a memory access for the second word of the command doubleword. Also, the MSB's of the byte address (from the first word of the command doubleword) are stored in the BA-register, the LSB's are set into the H-register for subsequent storage into the OF-register, and the order bits are set into the O-register.

PH6D1. The byte count obtained from the second word of the command doubleword is stored in the BC-register, the flags (from the second word), and the LSB's of the byte address (from the first word), are stored in the OF-register.

PH7D1. The termination phase (PH7D1) starts when the request strobe is received from the device controller. The following functions occur during this phase:

a. The order is cleared from the O-register in preparation for the terminal order (see figure 3-38).

b. The status information in the FS-register is updated.

c. The new flags encoded in the command doubleword are stored in the FS-register. If an error condition was detected when reading core memory for the command doubleword, the old flags in the FS-register are retained.

d. The terminal order is assembled in the O-register and sent to the device controller.

e. Request strobe acknowledge and end service signals are sent to the device controller, permitting it to disconnect from the interface lines.

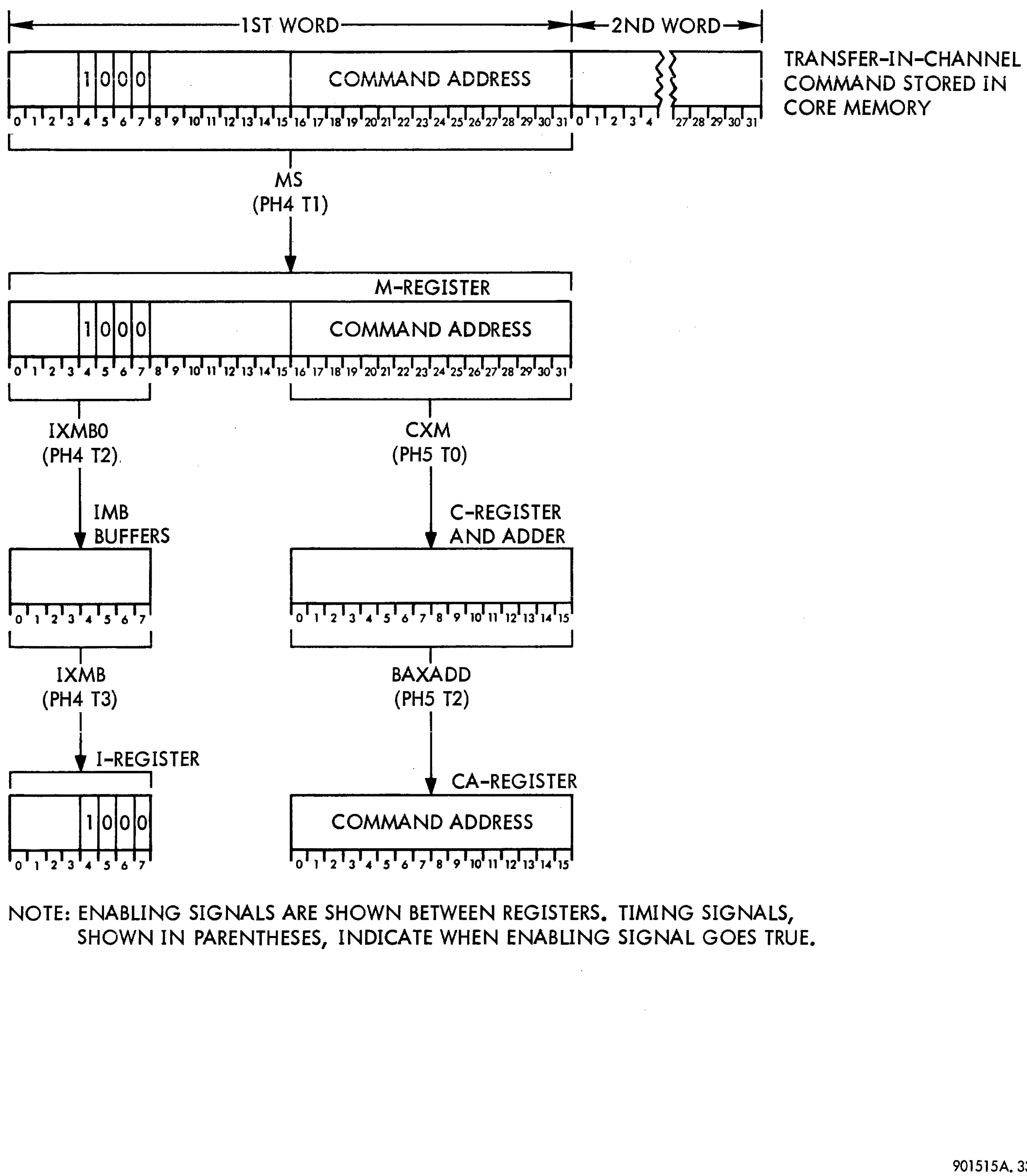



Figure 3-36. Processing a Transfer in Channel Command

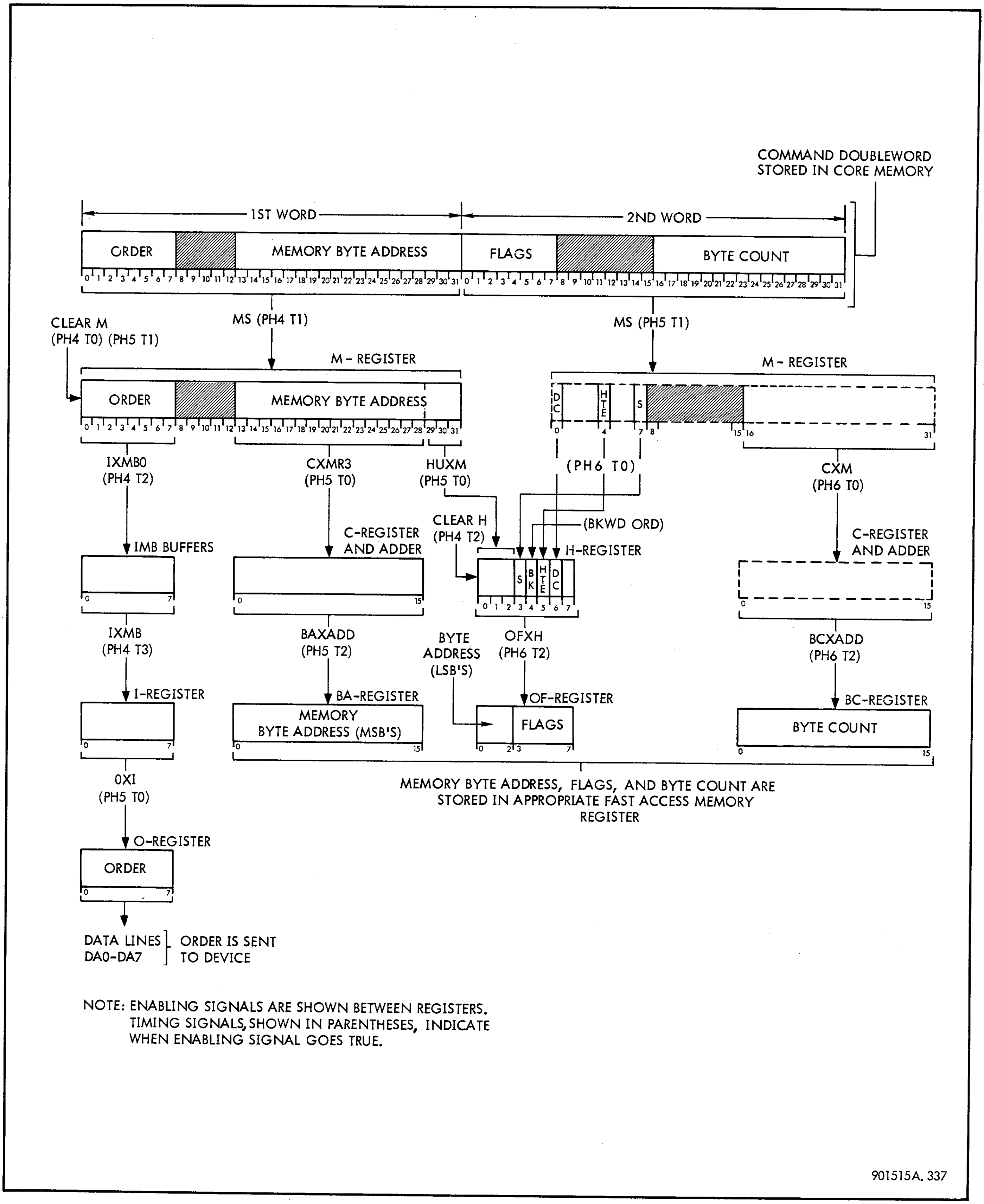



Figure 3-37. Processing a Command Doubleword

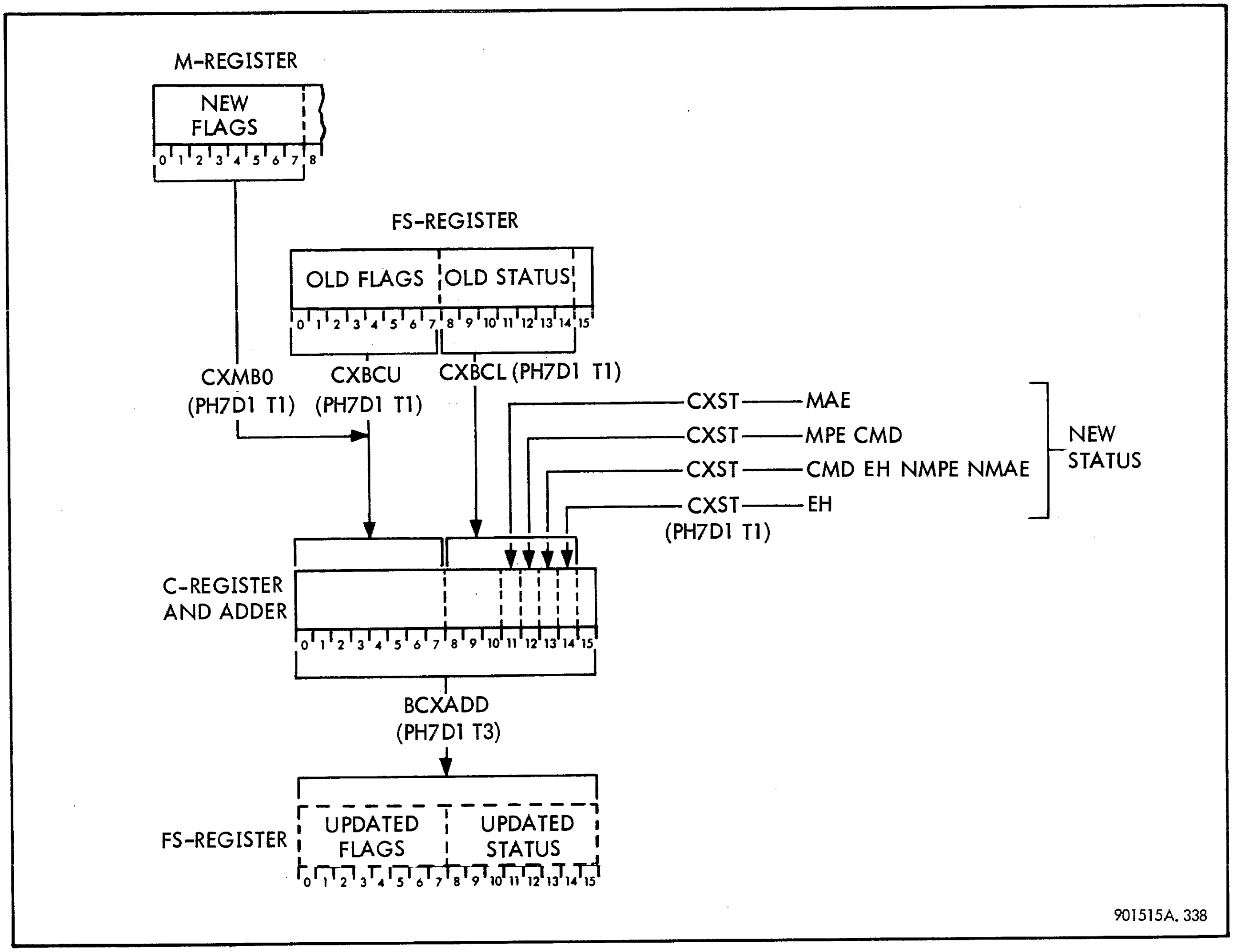



Figure 3-38. Termination Phase of an Order-Out Service Cycle

Table 3-15. Order-Out Service Cycle Phase Sequence

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D2 | Delay line D2 started | PLED2 | = | PH1D2 NFSL NAVO SC + ... | The delay line starts when service call SC is received from a device controller |
| T0D2 | Reset latch PH1PD2 | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Set latch PH1PD2 | PH1PD2 | = | PH1D2 T1D2 + ... | |
| T2D2 | Set flip-flop F1 | S/F1 | = | PH1T2D2 NFNT SC | Flip-flop F1 drives the ASC function indicator line |
| T3D2 | Reset latch PH1D2 | PH1D2 | = | PH1D2 NT3D2 + ... | |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T4D2 | Set PH2D2 | S/PH2D2 | = | PH1PD2 T4D2 | Next phase in sequence |
| | Set flip-flop FS | S/FS | = | PH1PD2 T4D2 F1 + ... | Primes the service connect flip-flop in the highest priority device controller with a service call pending. This flip-flop is set on the falling edge of signal FS |
| PH2D2 | Delay line D2 started | PLED2 | = | PH2D2 FIN (AVO FSL + ...) [(NFNT + ...) RSA2 + ...] | The delay line starts when signal FSL is received from the device controller |
| T0D2 | Signal AX0 goes true | AX0 | = | PHT0D2 | A-register is prepared to receive the device controller address |
| | | A0-A7 | = | A0-A7 NAX0 + ... | |
| | Reset flip-flops MAE, MPE, and latch PH1PD2 | R/MAE | = | R/MPE = PH2T0D2 | |
| | | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Reset flip-flops LS0, LS1, and LS2 | R/LS0 | = | R/LS1 = R/LS2 = CM002 + ... | Selects the BA-, BC-, and OF-registers of appropriate subchannel |
| | | CM002 | = | PH2D2 T1D2 NFNT | |
| | Signal AXFR goes true | AXFR | = | PH2T1D2 F1 + ... | Device controller address on FR lines is set into A-register to select appropriate subchannel |
| | Reset CMD, EH, EHE, IXMB, ORD, OUT, TPE, TRA1, and latches TORD and ZBC | R/CMD | = | R/EH = R/EHE = R/ORD = R/IXMB = R/OUT = R/TPE = R/TRA1 = PH2T1D2 | |
| | | TORD | = | TORD NPH2T1D2 + ... | |
| | | ZBC | = | ZBC NPH2T1D2 + ... | |
| | Set flip-flops FP and latch PH2PD2 | S/FP | = | PH2T1D2 | |
| | | PH2PD2 | = | PH2D2 T1D2 | |
| T2D2 | Clear C-, H-, and I-registers | CX0 | = | HX0 = HUX0 = PH2T2D2 | |
| | | IX0 | = | PH2D2 T2D2 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | | R/H3-R/H7[1] | = | HX0 + ... | |
| | | I0-I7 | = | I0-I7 NIX0 + ... | |
| | Reset J-register | R/J0 | = | R/J1 = R/J2 = PH2T2D2 | |
| | Clear O-register | OX0 | = | PH2T2D2 NCM044 + ... | The O-register is prepared to receive order bits of the command doubleword |
| | | NCM044 | = | NFNT + F0 | |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T2D2 (Cont.) | Set flip-flops SUB and K15 | S/SUB | = | S/K15 = PH2T2D2 | A one will be subtracted from all data transferred through the adder |
| | Reset flip-flop FS | R/FS | = | PH2D2 T2D2 F1 + ... | The service connect flip-flop in the device subcontroller is set when it senses the falling edge of FS. The device controller is, therefore, connected for service |
| | Reset flip-flops ED and ES | R/ED | = | R/ES = PH2D2 T2D2 F1 + ... | |
| | Reset flip-flop PH2D2 | R/PH2D2 | = | T3D2 + RESET | |
| T4D2 | Set latch PH1D1 | PH1D1 | = | PH1D1S0 + ... | Next phase in sequence |
| | | PH1D1S0 | = | PH2PD2 T4D2 F1 | |
| | Set latch PH1D2 | PH1D2 | = | PH2PD2 T4D2 NF1 NFNT | If the requesting device controller does not reply to the ASC, the MIOP receives signal AVO. The operation is thus aborted, and the MIOP returns to the beginning phase, PH1D2 |
| | | R/F1 | = | R/FS = AVO + ... | |
| PH1D1 | Delay line D1 started | PLED1 | = | PH1D1 RS CM023 + ... | The delay line starts when signal RS is received from the device controller |
| | | CM023 | = | NMS + MDR1 | |
| T0D1 | Reset flip-flop RSA1 | R/RSA1 | = | PH1O2D1 T0D1 | Prevents sending of RSA to the device controller prematurely |
| | | RSA | = | RSA1 NRSA2 | |
| | Set flip-flops ED, ES, and TRA1 | S/ED | = | S/ES = S/TRA1 = CM030 + ... | Flip-flops ED, ES, and TRA1 are set if a word boundary crossing is detected. During a data-in or data-out service cycle, signals ED and ES cause the service cycle to conclude without a terminal order. During an order-out service cycle, however, ES is unconditionally reset at T2D1 so that a terminal order will result |
| | | CM030 | = | CM004 CM029 | |
| | | CM004 | = | PH1O2D1 T0D1 | |
| | | CM029 | = | NJ1 NJ2 H4 + J1 J2 NH4 | |
| | | S/H4 | = | OF4 + ... | |
| | Clear M-register latches | MLX0 | = | MUX0 = CM011 + ... | The M-register is prepared to receive the first word of the command doubleword from core memory |
| | | CM011 | = | PH1O4D1 T0D1 | |
| | | M0-M15 | = | M0-M15 MUX0 + ... | |
| | | M16-M31 | = | M16-M31 MLX0 + ... | |
| | Clear W-register latches | WX0 | = | CM011 + ... | Signals W0-W3 define a read operation |
| | | W0-W3 | = | W0-W3 NWX0 | |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 (Cont.) | Reset flip-flops MS, MAE, and PRCH | R/MS = MCD1 T0D1 + ...<br>MCD1 = PH1O4D1 + ...<br>R/MAE = PH1D1 T0D1<br>R/PRCH = T0D1 | |
| | Reset latch PH1PD1 | PH1PD1 = PH1PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set flip-flops ORD and OUT | S/ORD = PH1D1 T1D1 DOR + ...<br>S/OUT = PH1D1 T1D1 IOR + ... | The device controller specifies the service cycle as an order out by driving the DOR and IOR lines |
| | Signal FX0 goes true | FX0 = FX0I + ...<br>FX0I = PH1D1 T1D1<br>R/F1 = FX0 + ... | The F-register is cleared; therefore, the ASC function indicator line goes false |
| | Reset flip-flop RSA2 | R/RSA2 = PH1O2D1 T1D1 + ... | Primes RSA |
| | Reset flip-flops MAR, MDR1, and MPE | R/MAR = R/MDR1 = MCD1 T1D1 + ...<br>MCD1 = PH1O4D1 + ...<br>R/MPE = PH1O4D1 T1D1 + ... | |
| | Set PH1PD1 | PH1PD1 = PH1D1 T1D1 + ... | |
| T2D1 | Reset flip-flop FIN | R/FIN = PH1T2D1 + ... | A new I/O operation cannot proceed past the first phase. Signal NFIN inhibits starting of delay line D2 during PH2D2 |
| | Set latch PH1D2 | PH1D2 = PH1T2D1 + ... | A new I/O operation may proceed through the first phase (PH1D2) |
| | Reset flip-flop ES | R/ES = ORD PH1T2D1 + ... | When the device controller senses ES false, it will generate another request strobe for the terminal order |
| | Set latch TORD | TORD = ORD PH1T2D1 + ... | Indicates that a terminal order will be sent to the device controller during the termination phase |
| | Set flip-flops LS0 and LS1 | S/LS0 = PH1T2D1 ORD + ...<br>S/LS1 = CM001 + ...<br>CM001 = PH1D1 T2D1 ORD + ... | Selects CA- and FS-registers |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T2D1 (Cont.) | Signal HUX0 goes true | HUX0 | = | HUX0I + ... | Upper three bits of H-register are cleared |
| | | HUX0I | = | PH1D1 T2D1 OUT | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| T3D1 | Reset latch PH1D1 | PH1D1 | = | PH1D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | NCX0 C0-C14 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Set flip-flop LS2 | S/LS2 | = | PH1O2PD1 T4D1 TORD + ... | Selects IS-register |
| | Set latch PH4D1 | PH4D1 | = | PH4D1S0 T4D1 + ... | Next phase in sequence |
| | | PH4D1S0 | = | PH1PD1 ORDOUT + ... | |
| PH4D1 | Delay line D1 started | PLED1 | = | PH4D1 CM023 | Since no core memory access has been made during this service cycle (signal NMS true), the delay line starts at the end of T4 of the preceding phase |
| | | CM023 | = | NMS + MDR1 | |
| T0D1 | Reset flip-flop SUB | R/SUB | = | PH4D1 T0D1 + ... | Data transferred through the adder will be incremented by one, since K15 is true |
| | Signal CXBA goes true | CXBA | = | PH4D1 T0D1 + ... | The command address is transferred from the CA-register to the C-register and adder. Since K15 is true, the address is incremented by one |
| | | C0-C14 | = | BA0-BA14 CXBA + ... | |
| | | S/C15 | = | BA15 CXBA + ... | |
| | Signal SX0 goes true | SX0 | = | CM011 + ... | The S-register is cleared (bits S15-S31) |
| | | CM011 | = | PH1O4D1 T0D1 | |
| | | S15-S31 | = | S15-S31 NSX0 + ... | |
| | Set flip-flop SXBA | S/SXBA | = | PH1O4D1 T0D1 | The command address is transferred from the CA-register to bits S15-S30 of the S-register. Bit S31 is zero. The first word of the command doubleword will, therefore, be fetched from core memory during the next memory access |
| | | S15-S30 | = | BA0-BA15 SXBA + ... | |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 (Cont.) | Signals MLX0 and MUX0 go true | MLX0 = MUX0 = CM011 + ...<br>M0-M15 = M0-M15 NMUX0 + ...<br>M16-M31 = M16-M31 NMLX0 + ... | The M-register is cleared |
| | Signal WX0 goes true | WX0 = CM011 + ...<br>W0-W3 = W0-W3 NWX0 + ... | A core memory read operation is specified |
| | Reset latch PH4PD1 | PH4PD1 = PH4PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Reset flip-flop SXBA | R/SXBA = T1D1D + ... | The transfer signal (SXBA) that transfers the command address into the S-register is no longer needed |
| | Reset flip-flops MAR, MPE, and MDR1 | R/MAR = R/MDR1 = MCD1 T1D1 + ...<br>MCD1 = PH1O4D1 + ...<br>R/MPE = PH1O4D1 T1D1 + ... | |
| | Set flip-flop MS | S/MS = MSSET + ...<br>MSSET = PH4T1D1 NEH NBC15 + ... | If there have been no error conditions detected thus far during the current service (NEH), and if the device controller has not set the chaining modifier CM during the previous order-in service cycle (NBC15), flip-flop MS will be set. Signal MS starts a core memory access |
| | Set latch PH4PD1 | PH4PD1 = PH4D1 T1D1 | |
| T2D1 | Signals HUX0 and HX0 go true | HUX0 = HX0 = CM020 + ...<br>CM020 = PH4D1 T2D1 (NEH + MS)<br>R/H0-R/H2 = HUX0 + ...<br>R/H3-R/H7 = HX0 + ... | The H-register is cleared |
| | Signal BAXADD goes true | BAXADD = CM020 + ... | The incremented command address is returned to the CA-register |
| T3D1 | Signal IX0 goes true | IX0 = PH4PD1 T3D1 + ...<br>I0-I7 = I0-I7 NIX0 + ... | The I-register is cleared |
| | Set flip-flop IXMB | S/IXMB = PH4PD1 T3D1 + ... | Signal IXMB transfers the information from the IMB buffers to the I-register. If a memory cycle was started (MS true), during PH5D1 the |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T3D1 (Cont.) | | | buffers will contain the order bits of the command doubleword accessed from core memory |
| | Set flip-flop CMD | S/CMD = PH4PD1 T3D1 (NEH + MS) + ... | This function applies when chaining modifier CM is true (signal BC15 true) and this is the first time through PH4D1 (no core memory access made). The current command doubleword is not accessed; however, when PH4D1 is repeated, the command doubleword in the next higher memory location will be accessed |
| | Reset latch PH4D1 | PH4D1 = PH4D1 NPHD1X0<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Reset flip-flops LS1 and LS2 | R/LS1 = R/LS2 = PH4PD1 T4D1 + ... | Selects BC– and OF–registers |
| | Reset flip-flop K15 | R/K15 = PH4PD1 T4D1 MS + ... | Data being transferred through the adder will not be altered |
| | Set flip-flops MAE and EH | S/MAE = PH4PD1 CM048 + ...<br>CM048 = T4D1 MS NAH<br>S/EH = MAE + ... | If a memory access was made, and memory did not respond with address here AH, flip-flops MAE and EH record the error condition until the termination phase |
| | Set latch PH4D1 | PH4D1 = PH4D1S0 T4D1<br>PH4D1S0 = PH4PD1 NEH NMS | This function applies if signal CM (BC15) is true and this is the first time through PH4D1. In this case, PH4D1 will be repeated and a memory access will be made the second time through |
| | Signal CX0 goes true | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1 | The C-register is cleared |
| | Set latch PH5D1 | PH5D1 = PH5D1S0 T4D1 + PH5D1S1 T4D1<br>PH5D1S0 = PH4PD1 EH<br>PH5D1S1 = PH4PD1 MS | If a memory access had been made (MS true) or any error conditions detected (EH true), the MIOP will advance to PH5D1 |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH5D1 | Delay line D1 started | PLED1 | = | PH5D1 CM023 + ... | The delay line starts when the reaction signals from core memory have been received (signal MDR1 true) |
| | | CM023 | = | NMS + MRD1 | |
| TOD1 | Signal OX0 goes true | OX0 | = | PH5D1 TOD1 ORD + ... | The O-register is cleared |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |
| | Set flip-flop OXI | S/OXI | = | PH5D1 TOD1 ORD + ... | The order bits of the command doubleword are transferred from the I-register to the O-register and, therefore, to the device controller |
| | | O0-O7 | = | I0-I7 OXI + ... | |
| | Reset flip-flop IXMB | R/IXMB | = | TOD1 + ... | |
| | Set latch S31 | S31 | = | PH5D1 TOD1 + ... | Since the address of the first word of the command doubleword was set into the S-register during PH4D1, bit S31 true specifies the address of the second word |
| | Set flip-flop EH | S/EH | = | CMD MPE + ... | If a parity error was detected while reading the first word of the command doubleword from core memory (signal MPE true), error halt flip-flop EH will be set |
| | | MPE | = | PE MS | |
| | Signal CXM goes true | CXM | = | PH5D1 TOD1 TRA | If the first word of the command doubleword (currently in the I-register) specifies a transfer in channel command (signal TRA true), the address bits of the transfer in channel command currently in the M-register will be transferred to the C-register |
| | | C0-C14 | = | M16-M30 CXM + ... | |
| | | S/C15 | = | M31 CXM + ... | |
| | | TRA | = | I4 NI5 NI6 NI7 | |
| | Set flip-flop EH | S/EH | = | PH5D1 TOD1 TRA TRA1 + ... | Signal TRA1 is true only if this is the second time through PH5D1, that is, if the first word accessed from core memory was a transfer in channel command. Signals TRA and TRA1 both true indicate that two transfers in channel command in succession have been accessed. This defines a control error |
| | | S/TRA1 | = | PH5PD1 T4D1 + ... | |
| | Reset flip-flop LS0 | R/LS0 | = | PH5D1 TOD1 NTRA + ... | Selects BA-register if not a transfer in channel command (NTRA) |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 (Cont.) | Signal CXMR3 goes true | CXMR3 | = | PH5D1 T0D1 NTRA | The 16 MSB's of the byte address are transferred from the M-register to the BA-register if the last word accessed from memory was not a transfer in channel command |
| | | C0-C14 | = | M13-M27 CXMR3 + ... | |
| | | S/C15 | = | M28 CXMR3 + ... | |
| | Signal HUXM goes true | HUXM | = | PH5D1 T0D1 NTRA | The three LSB's of the byte address are transferred to the H-register if the last word accessed from memory was not a transfer in channel command |
| | | H0-H2 | = | M29-M31 HUXM + ... | |
| | Reset flip-flop MS | R/MS | = | MCD1 T0D1 + ... | |
| | | MCD1 | = | PH5D1 + ... | |
| | Reset latch PH5PD1 | PH5PD1 | = | PH5PD1 NPHPD1X0 + ... | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Reset flip-flops MAR and MDR1 | R/MAR | = | R/MDR1 = MCD1 T1D1 + ... | |
| | Signals MLX0 and MUX0 go true | MLX0 | = | MUX0 = MLX0I + ... | The M-register is cleared |
| | | MLX0I | = | PH5D1 T1D1 + ... | |
| | | M0-M15 | = | M0-M15 NMUX0 + ... | |
| | | M16-M31 | = | M16-M31 NMLX0 + ... | |
| | Set flip-flop MS | S/MS | = | MSSET + ... | A core memory cycle is started to access the second word of the command doubleword if the last word was not a transfer in channel command |
| | | MSSET | = | PH5D1 T1D1 NEH NTRA | |
| | Set latch PH5PD1 | PH5PD1 | = | PH5D1 T1D1 | |
| T2D1 | Reset flip-flop OXI | R/OXI | = | T2D1 + ... | |
| | Signal BAXADD goes true | BAXADD | = | PH5D1 T2D1 NEH + ... | If a transfer in channel command is being processed, signal BXADD stores the command address encoded in the transfer in channel command in the CA-register (signal LS0 is true). If a transfer in channel is not being processed, signal BAXADD stores the byte address encoded in the first word of the command doubleword in the BA-register (signal LS0 is false) |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T2D1 (Cont.) | Set flip-flop RSA1 | S/RSA1 | = | PH5D1 T2D1 ORD CM007 + ... | If a transfer in channel command is not being processed (signal NTRA true) or if an error condition was detected (signal EH true), flip-flop RSA1 is set and request strobe acknowledge RSA is set to the device controller |
| | | CM007 | = | EH + NTRA | |
| | | RSA | = | RSA1 NRSA2 | |
| T3D1 | Reset latch PH5D1 | PH5D1 | = | PH5D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Set flip-flop TRA1 | S/TRA1 | = | PH5PD1 T4D1 + ... | Signal TRA1 is used in conjunction with TRA to detect two successive transfers in channel commands |
| | Set flip-flop K15 | S/K15 | = | PH5PD1 T4D1 TRA + ... | Flip-flop K15 is set if a transfer in channel command is being processed |
| | Set flip-flops MAE and EH | S/MAE | = | PH5PD1 CM048 + ... | If core memory has not responded to the last memory request with signal address here (AH), flip-flops MAE and EH record the error condition until the termination phase |
| | | CM048 | = | T4D1 MS NAH | |
| | | S/EH | = | MAE + ... | |
| | Set latch PH4D1 | PH4D1 | = | PH4D1S0 T4D1 + ... | If a transfer in channel command is being processed (signal TRA true) and no error conditions have been recorded (signal NEH true), the MIOP will return to PH4D1 |
| | | PH4D1S0 | = | PH5PD1 NCM007 + ... | |
| | | NCM007 | = | NEH TRA | |
| | Set latch PH6D1 | PH6D1 | = | PH6D1S0 T4D1 + ... | If a transfer in channel command is not being processed (signal NTRA true) or an error was recorded (signal EH true), the MIOP will progress to PH6D1 |
| | | PH6D1S0 | = | PH5PD1 CM007 | |
| | | CM007 | = | EH + NTRA | |
| PH6D1 | Delay line D1 started | PLED1 | = | PH6D1 CM023 + ... | The delay line starts when the reaction signals are received from core memory (signal MDR1 true). The second word of the command |
| | | CM023 | = | NMS + MDR1 | |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH6D1 (Cont.) | | | | | doubleword is now in the M-register |
| T0D1 | Signal CXM goes true | CXM | = | PH6D1 T0D1 + ... | Signal CXM transfers the byte count from the M-register to the C-register |
| | | C0-C14 | = | M16-M30 CXM + ... | |
| | | S/C15 | = | M31 CXM + ... | |
| | Set bits H3-H6 of the H-register | S/H3 | = | PH6T0D1 M7 + ... | The flags are transferred from the M-register to the H-register. Bit H-4 is set if the order in the I-register specifies a read backward |
| | | S/H4 | = | PH6T0D1 CM003 + ... | |
| | | S/H5 | = | PH6T0D1 M4 + ... | |
| | | S/H6 | = | PH6T0D1 M0 + ... | |
| | | CM003 | = | (ORD BKWD + ...) | |
| | | BKWD | = | I4 I5 NI6 NI7 | |
| | Set flip-flop EH | S/EH | = | CMD MPE + ... | The error halt flip-flop is set if a parity error is detected when reading core memory (signal PE true) |
| | | MPE | = | MS PE | |
| | Reset latch PH6PD1 | PH6PD1 | = | PH6PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Set latch PH6PD1 | PH6PD1 | = | PH6D1 T1D1 | |
| T2D1 | Signal BCXADD goes true | BCXADD | = | CM022 + ... | If no error conditions have been detected (signal NEH true), the byte count MSB's are stored in the BC-register |
| | | CM022 | = | PH6D1 T2D1 NEH + ... | |
| | Signal OFXH goes true | OFXH | = | CM022 + ... | If no error conditions have been detected, the byte count LSB's and the flags are stored in the OF-register |
| T3D1 | Reset latch PH6D1 | PH6D1 | = | PH6D1 NPHD1X0 + ... | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Set flip-flops LS1 and LS2 | S/LS1 | = | S/LS2 = PH6PD1 T4D1 + ... | Selects FS- and IS-registers |
| | Set latch PH7D1 | PH7D1 | = | PH6PD1 T4D1 + ... | Next phase in sequence |

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH7D1 | Delay line D1 started | PLED1 | = | PH7D1 CM025 + ... | The delay line starts after the response signals have been received from core memory and request strobe RS has been received from the device controller |
| | | CM025 | = | CM023 CM025I1 | |
| | | CM023 | = | MDR1 + ... | |
| | | CM025I1 | = | NCM025I1I + ... | |
| | | NCM025I1I | = | RS RSA2 | |
| T0D1 | Signals HUX0 and HX0 go true | HUX0 | = | HX0 = HX0I + ... | The H-register is cleared |
| | | HX0I | = | PH7D1 T0D1 + ... | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| | | R/H3-R/H7 | = | HX0 + ... | |
| | Reset flip-flop K15 | R/K15 | = | PH7D1 T0D1 + ... | Data transferring through the adder will not be altered |
| | Signal OX0 goes true | OX0 | = | PH7D1 T0D1 TORD + ... | The O-register is cleared |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |
| | Reset flip-flop RSA1 | R/RSA1 | = | PH7D1 T0D1 TORD + ... | Prevents request strobe acknowledge RSA from being sent to the device controller prematurely |
| | Reset latch PH7PD1 | PH7PD1 | = | PH7PD1 NPHPD1X0 + ... | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Signal CXBCL goes true | CXBCL | = | PH7T1D1 + ... | Status information from bits 8-14 of the FS-register is transferred to bits 8-14 of the C-register |
| | | C8-C14 | = | BC8-BC14 CXBCL + ... | |
| | Signal CXBCU goes true | CXBCU | = | CXBCUI + ... | If error conditions were detected during the current service cycle (signal EH true), the flags stored in the FS-register during a previous order-out service cycle are transferred to the C-register for subsequent storage back into the FS-register |
| | | CXBCUI | = | PH7D1 T1D1 CM045 | |
| | | CM045 | = | EH + NCMD | |
| | | C0-C7 | = | BC0-BC7 CXBCU + ... | |
| | Signal CXMB0 goes true | CXMB0 | = | PH7T1D1 NCM045 | If no error conditions were detected, the new flags in the M-register are transferred to the C-register for subsequent storage in the FS-register |
| | | NCM045 | = | CMD NEH | |
| | | C0-C7 | = | M0-M7 CXMB0 + ... | |

(Continued)

Table 3-15. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 (Cont.) | Signal CXST goes true | CXST = PH7T1D1 | Signal CXST updates the status information currently in the C-register |
| | | C11 = CXST MAE + ... | |
| | | C12 = CXST C12ST + ... | |
| | | C12ST = CMD MPE | |
| | | C13 = CXST C13ST + ... | |
| | | C13ST = CMD EH NMPE NMAE | |
| | | C14 = CXST EH + ... | |
| | Signal OXTORD goes true | OXTORD = PH7T1D1 TORD | Signal OXTORD gates a terminal order to the O-register and, therefore, to the device controller |
| | | O3 = OXTORD EH + ... | |
| | | O4 = OXTORD O4TOS + ... | |
| | | O4TOS = ORDOUT EH + ... | |
| | Set flip-flop ES | S/ES = PH7T1D1 + ... | Signal ES is applied to the reset input of the service connect flip-flop in the device controller. This flip-flop resets when it receives signal RSA from the MIOP |
| | Reset flip-flop RSA2 | R/RSA2 = PH7T1D1 TORD + ... | Primes RSA |
| | Set latch PH7PD1 | PH7PD1 = PH7D1 T1D1 + ... | |
| T3D1 | Signal BCXADD goes true | BCXADD = CM021 + ... | The new flags (if no error was detected), or old flags (if an error was detected), and updated status are stored in the FS-register |
| | | CM021 = PH7PD1 T3D1 NFIN | |
| | Set flip-flop RSA1 | S/RSA1 = PH7PD1 T3D1 | The service connect flip-flop in the device controller is reset. The device controller is, therefore, disconnected electrically from the MIOP interface |
| | | RSA = RSA1 NRSA2 | |
| | Reset latch PH7D1 | PH7D1 = PH7D1 NPHD1X0 + ... | |
| | | NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Set flip-flop FIN | S/FIN = PH7PD1 T4D1 + ... | The current operation no longer needs the MIOP's fast access memory |
| | Signal CX0 goes true | CX0 = CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 = NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 = C0-C14 NCX0 + ... | |
| | | R/C15 = CX0 + ... | |

## 3-41 Order-In Service Cycle

The order-in service cycle phases follow the sequence shown in figure 3-39. At the conclusion of a data exchange, the device controller specifies an order-in service cycle during which time it sends the operational status byte to the MIOP. The status is stored in the appropriate MIOP subchannel during the termination phase of the service cycle.

PH1D2. The MIOP determines that the request was initiated by a device controller and therefore drives the acknowledge service call line and the function strobe line at the device controller interface.

PH2D2. The delay line starts after the device controller responds to the function strobe. The MIOP clears certain flip-flops and registers. The address that the responding device controller has placed on the FR lines is set into the A-register to select the subchannel associated with that device controller.

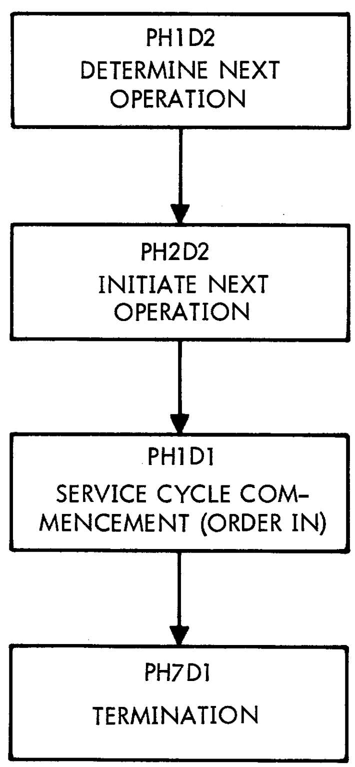



Figure 3-39. Phase Sequences For an Order-In Service Cycle

PH1D1. The MIOP inspects the operational status byte and sets up the logic to ensure that a terminal order will be sent to the device controller during the current service cycle.

PH7D1. The termination phase starts when the request strobe is received from the device controller. The following functions occur during this phase:

a. The status in the FS-register is updated (see figure 3-40).

b. The interrupt status in the IS-register is updated (see figure 3-41).

c. The terminal order is assembled in the O-register and sent to the device controller.

d. Request strobe acknowledge and end service signals are sent to the device controller, permitting it to disconnect from the interface lines.

## 3-42 Data-Out Service Cycle

The data-out service cycle phases follow the sequence shown in figure 3-42. During a data-out service cycle, the MIOP accesses the core memory location specified by the byte address register for one word of data, and sends the data to the device controller one byte at a time. For each byte transmitted, the MIOP cycles through PH2D1. The MIOP includes PH3D1 in the sequence when a carry must be made from the three LSB's to the 16 MSB's of the byte address. When the byte count has been reduced to zero and data chaining is specified, phases PH4D1, PH5D1, and PH6D1 are included, during which time the MIOP accesses the new command doubleword. If the first word of the command doubleword specifies a transfer in channel command (TRA true), phases PH4D1 and PH5D1 are repeated so the MIOP can branch to the new memory location for the data. If after completion of PH2D1 the byte count has decreased to zero and data chaining is not specified, or if the byte count does not equal zero, the MSB's of the byte address do not need updating, and either the MIOP or the device controller has specified end data ED, the MIOP advances directly from PH2D1 to the termination phase.

PH1D2. The MIOP determines that the request was initiated by a device controller and, therefore, drives the acknowledge service call line and the function strobe line at the device controller interface.

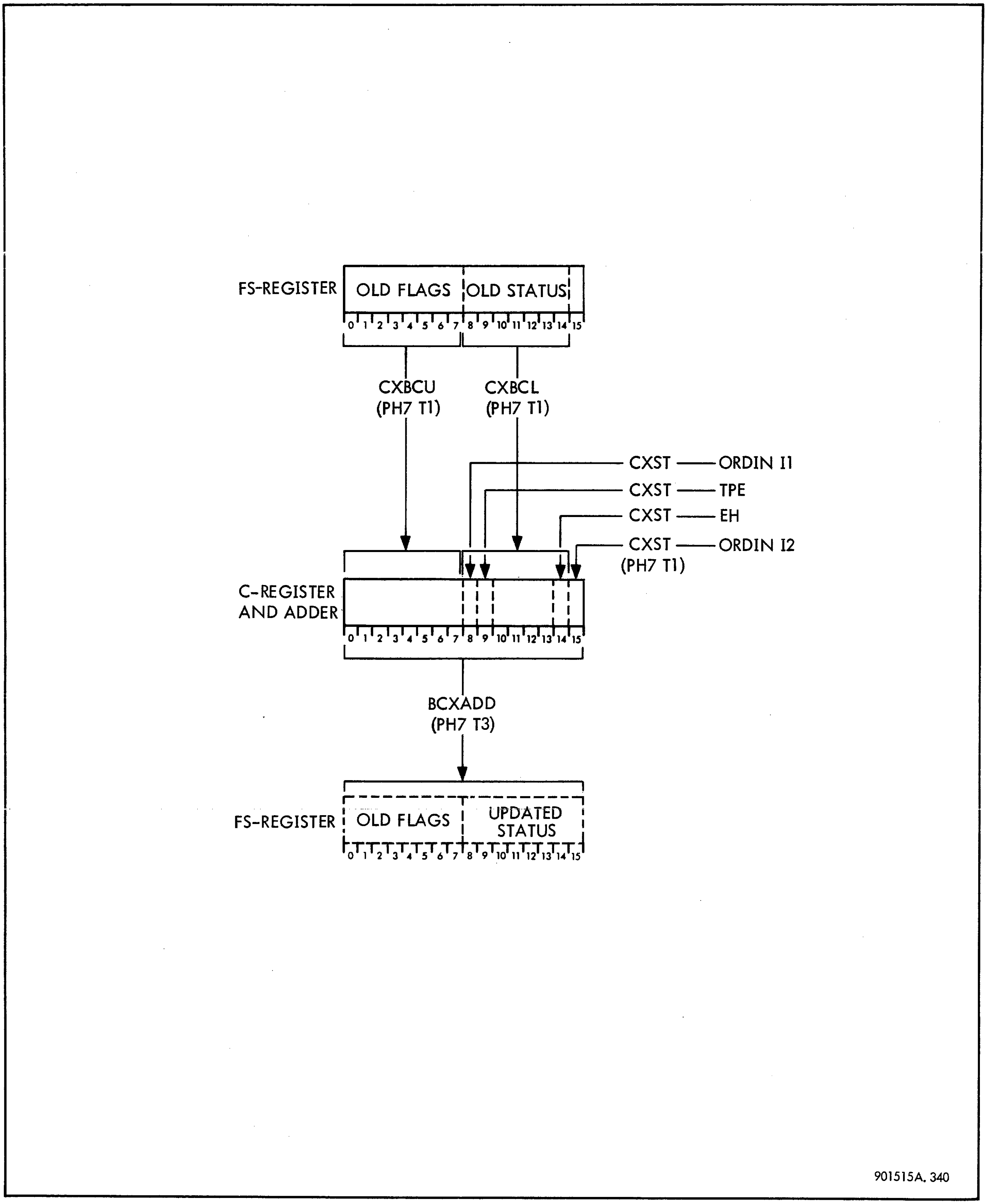



Figure 3-40. Updating Flags and Status During an Order-In Service Cycle

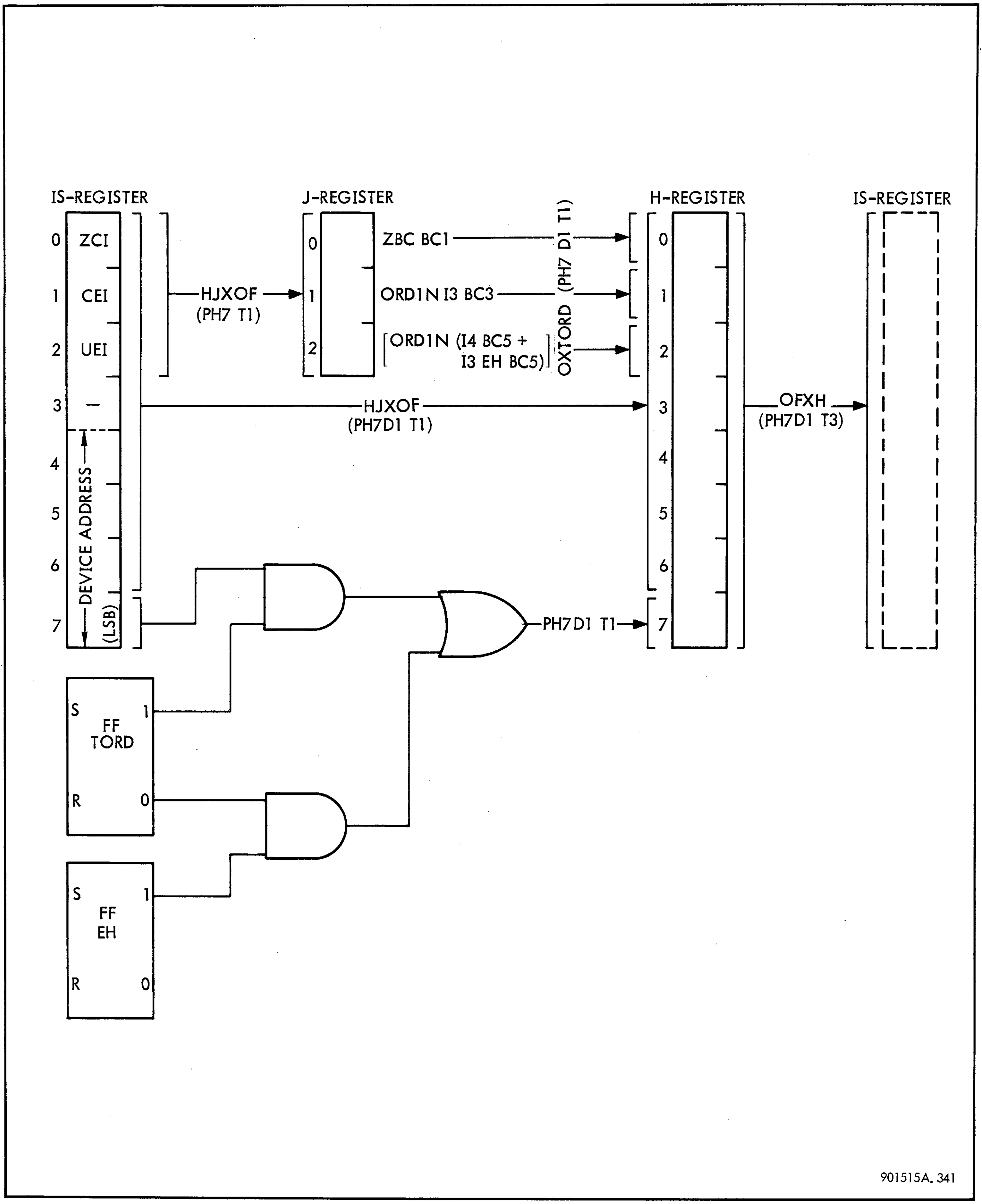



Figure 3-41. Updating Interrupt Status During an Order-In Service Cycle

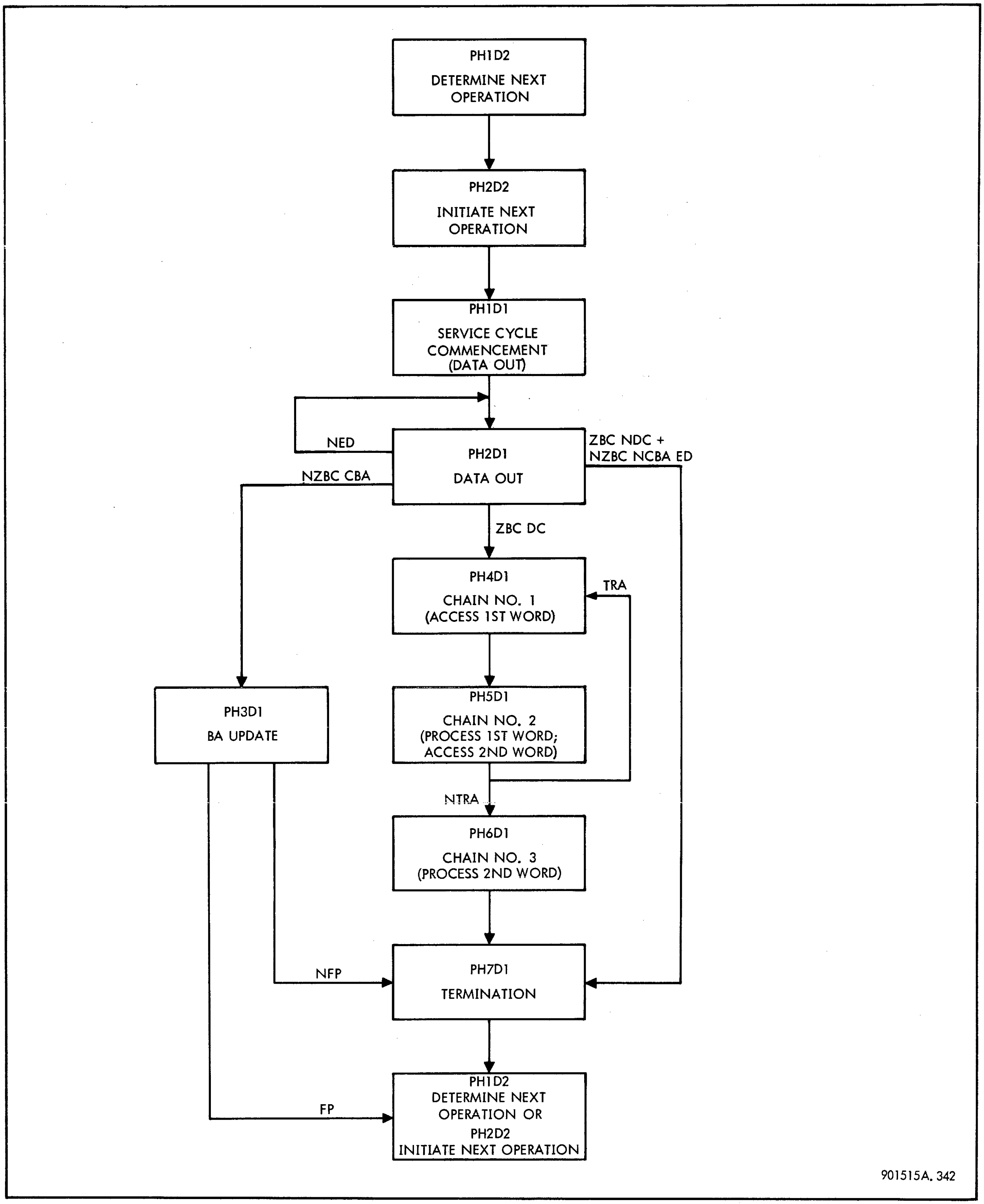



Figure 3-42. Phase Sequence Diagram for a Data-Out Service Cycle

Table 3-16. Order-In Service Cycle Phase Sequence

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D2 | Delay line D2 started | PLED2 | = | PH1D2 NFSL NAV0 SC + ... | The delay line starts when service call SC is received from a device controller |
| T0D2 | Reset latch PH1PD2 | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Set latch PH1PD2 | PH1PD2 | = | PH1D2 T1D2 + ... | |
| T2D2 | Set flip-flop F1 | S/F1 | = | PH1T2D2 NFNT SC | Flip-flop F1 drives the ASC function indicator line |
| T3D2 | Reset latch PH1D2 | PH1D2 | = | PH1D2 NT3D2 + ... | |
| T4D2 | Set PH2D2 | S/PH2D2 | = | PH1PD2 T4D2 | Next phase in sequence |
| | Set flip-flop FS | S/FS | = | PH1PD2 T4D2 F1 + ... | Primes the service connect flip-flop in the highest priority device controller with a service call pending. This flip-flop is set on the falling edge of signal FS |
| PH2D2 | Delay line D2 started | PLED2 | = | PH2D2 FIN (AVO FSL + ...) [(NFNT + ...) RSA2 + ...] | The delay line starts when signal FSL is received at the device controller interface |
| T0D2 | A-register cleared | AX0 | = | PHT0D2 | The A-register is prepared to receive the device controller address |
| | | A0-A7 | = | A0-A7 NAX0 + ... | |
| | Reset flip-flops MAE and MPE | R/MAE | = | R/MPE = PH2T0D2 | |
| | Reset latch PH1PD2 | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Reset flip-flops LS0, LS1, and LS2 | R/LS0 | = | R/LS1 = R/LS2 = CM002 + ... | Selects the BA-, BC-, and OF-registers of appropriate subchannel |
| | | CM002 | = | PH2D2 T1D2 NFNT | |
| | Signal AXFR goes true | AXFR | = | PH2T1D2 F1 + ... | Device controller address on the FR lines is set into the A-register to select appropriate subchannel |
| | Reset flip-flops CMD, EHE, EH, IXMB, ORD, OUT, TPE, TRA1, and latches TORD and ZBC | R/CMD | = | R/EHE = R/EH = R/IXMB = R/ORD = R/OUT = R/TPE = R/TRA1 = PH2T1D2 | |
| | | TORD | = | TORD NPH2T1D2 + ... | |
| | | ZBC | = | ZBC NPH2T1D2 + ... | |
| | Set flip-flop FP | S/FP | = | PH2T1D2 | |
| | Set latch PH2PD2 | PH2PD2 | = | PH2D2 T1D2 | |

(Continued)

Table 3-16. Order-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T2D2 | Clear C-, H-, and I-registers | CX0 = HUX0 = PH2T2D2<br>IX0 = PH2D2 T2D2<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ...<br>R/H0-R/H2 = HUX0 + ...<br>R/H3-R/H7 = HX0 + ...<br>I0-I7 = I0-I7 NIX0 + ... | |
| | Clear J-register | R/J0-R/J2 = PH2T2D2 + ... | |
| | Set flip-flops SUB and K15 | S/SUB = S/K15 = PH2T2D2 | A one will be subtracted from all data transferred through the adder |
| | Reset flip-flop FS | R/FS = PH2D2 T2D2 F1 + ... | The service connect flip-flop in the device controller is set when it senses the falling edge of FS. The device controller is therefore connected for service |
| | Reset flip-flops ED and ES | R/ED = R/ES = PH2D2 T2D2 F1 + ... | |
| T3D2 | Set flip-flop EHE | S/EHE = PH2PD2 T3D2 OF5 + ... | Error halt enable flip-flop EHE is set if the halt on transmission error (HTE) flag stored in the OF-register is true |
| | Set flip-flop EH | S/EH = PH2PD2 T3D2 OF7 + ... | If an error is detected during a core memory cycle of a data-in service cycle that is not accompanied by a terminal order, the transmission error halt (TEH) flag (OF7) is set. Thus, the error is reported to the device during the terminal order of the current service cycle |
| | Reset PH2D2 | R/PH2D2 = T3D2 + RESET | |
| T4D2 | Set latch PH1D1 | PH1D1 = PH1D1S0 + ...<br>PH1D1S0 = PH2PD2 T4D2 F1 | Next phase in sequence |
| | Set latch PH1D2 | PH1D2 = PH2PD2 T4D2 NF1 NFNT + ...<br>R/F1 = R/FS = AVO + ... | If the requesting device controller does not reply to the ASC, the MIOP receives signal AVO. Therefore, the operation is aborted, and the MIOP returns to the beginning phase, PH1D2 |

(Continued)

Table 3-16. Order-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D1 | Delay line D1 started | PLED1 | = | PH1D1 RS CM023 + ... | The delay line starts when signal RS is received from the device controller |
| | | CM023 | = | NMS + MDR1 | |
| T0D1 | Reset flip-flop RSA1 | R/RSA1 | = | PH1O2D1 T0D1 | Prevents sending of RSA to the device controller prematurely |
| | | RSA | = | RSA1 NRSA2 | |
| | Set flip-flops ED, ES, and TRA1 | S/ED | = | S/ES = S/TRA1 = CM030 + ... | Flip-flops ED, ES, and TRA1 are set if a word boundary crossing is detected. During a data-in or data-out service cycle, signals ED and ES cause the service cycle to conclude without a terminal order. During an order-in service cycle, however, ES is unconditionally reset at T2D1 so that a terminal order will result |
| | | CM030 | = | CM004 CM029 | |
| | | CM004 | = | PH1O2D1 T0D1 | |
| | | CM029 | = | NJ1 NJ2 H4 + J2 J2 NH4 | |
| | | S/H4 | = | OF4 + ... | |
| | Reset flip-flops MS, MAE, and PRCH | R/MS | = | MCD1 T0D1 + ... | |
| | | MCD1 | = | PH1O4D1 + ... | |
| | | R/MAE | = | PH1D1 T0D1 | |
| | | R/PRCH | = | T0D1 | |
| | Reset latch PH1PD1 | PH1PD1 | = | PH1PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Set flip-flop ORD | S/ORD | = | PH1D1 T1D1 DOR + ... | The device controller drives signal DOR true and holds IOR false to define an order-in service cycle |
| | Signal FX0 goes true | FX0 | = | FX0I + ... | The F-register is cleared; therefore, the ASC function indicator line goes false |
| | | FX0I | = | PH1D1 T1D1 | |
| | | R/F1 | = | FX0 + ... | |
| | Reset flip-flop RSA2 | R/RSA2 | = | PH1O2D1 T1D1 + ... | Primes RSA |
| | Reset flip-flops MAR and MDR1 | R/MAR | = | R/MDR1 = MCD1 T1D1 + ... | |
| | Set latch PH1PD1 | PH1PD1 | = | PH1D1 T1D1 + ... | |
| | Reset flip-flop FIN | R/FIN | = | PH1T2D1 + ... | A new operation cannot proceed past the first phase when signal FIN is false |
| | Set latch PH1D2 | PH1D2 | = | PH1T2D1 + ... | A new operation may proceed through the first phase (PH1D2) |

(Continued)

Table 3-16. Order-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T1D1 (Cont.) | Reset flip-flop ES | R/ES | = | ORD PH1T2D1 + ... | Signal ES false will cause the device controller to generate another request strobe (RS) for the terminal order |
| | Set latch TORD | TORD | = | ORD PH1T2D1 + ... | Indicates that a terminal order will be sent to the device controller during the termination phase |
| | Set flip-flop LS1 | S/LS1 | = | CM001 + ... | Selects FS-register |
| | | CM001 | = | PH1D1 T2D1 ORD + ... | |
| | Set flip-flop RSA1 | S/RSA1 | = | PH1T2D1 ORD NOUT NTRA1 + ... | Since RSA2 is false, request strobe acknowledge RSA is sent to the device controller. If ES had been set during T0D1 because of a word boundary crossing (signal TRA1 true) and reset during T2D1, RSA1 will not be set until T4D1, thus providing the required time lapse between ES going false and RSA going true |
| | | RSA | = | RSA1 NRSA2 | |
| T3D1 | Reset latch PH1D1 | PH1D1 | = | PH1D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Set flip-flop TPE | S/TPE | = | PH1PD1 T4D1 I0 ORDIN + ... | Transmission error flip-flop TPE is set if a transmission error is reported by means of the operational status byte (bit 0 of the I-register true) |
| | Set flip-flop EH | S/EH | = | EHE TPE + CM031 T4D1 NBC6 + ... | Bit 1 of the I-register is true if an incorrect length condition is reported by means of the operational status byte. Error halt flip-flop EH is set if either of the two following conditions exist: a. Signal EHE is true and a transmission error is reported (signal TPE true) |
| | | CM031 | = | PH1PD1 ORDIN EHE I1 | |

(Continued)

Table 3-16. Order-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T4D1 (Cont.) | | | | | b. Signal EHE is true, incorrect length is reported (signal I1 true), and the suppress incorrect length flag SIL stored in bit 6 of the FS-register (BC6) is false |
| | Set flip-flop LS2 | S/LS2 | = | PH1O2PD1 T4D1 TORD + ... | Selects IS-register |
| | Set latch PH7D1 | PH7D1 | = | PH7D1S0 T4D1 + ... | Next phase in sequence |
| | | PH7D1S0 | = | CM008 PH1O2PD1 | |
| | | CM008 | = | ORDIN + ... | |
| PH7D1 | Delay line D1 started | PLED1 | = | PH7D1 CM025 + ... | The delay line starts when request strobe RS has been received from the device controller |
| | | CM025 | = | CM023 CM025I1 | |
| | | CM023 | = | NMS + ... | |
| | | CM025I1 | = | NCM025I1I + ... | |
| | | NCM025I1I | = | RS RSA2 | |
| T0D1 | Signals HUX0 and HX0 go true | HUX0 | = | HX0 = HX0I + ... | The H-register is cleared |
| | | HX0I | = | PH7D1 T0D1 + ... | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| | | R/H3-R/H7 | = | HX0 + ... | |
| | Reset flip-flop K15 | R/K15 | = | PH7D1 T0D1 + ... | Data transferred through the adder will not be altered |
| | Signal OX0 goes true | OX0 | = | PH7D1 T0D1 TORD + ... | The O-register is cleared |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |
| | Reset flip-flop RSA1 | R/RSA1 | = | PH7D1 T0D1 TORD + ... | Prevents request strobe acknowledge RSA from being sent to the device controller prematurely |
| | Reset latch PH7PD1 | PH7PD1 | = | PH7PD1 NPHPD1X0 + ... | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Signal CXBCL goes true | CXBCL | = | PH7T1D1 + ... | Status information from bits 8-14 of the FS-register is transferred to bits 8-14 of the C-register |
| | | C8-C14 | = | BC8-BC14 CXBCL + ... | |
| | Signal CXBCU | CXBCU | = | CXBCUI + ... | If error conditions were detected during the current service cycle (signal EH true), the flags stored in the FS-register during a previous order-out service cycle are transferred to the C-register |
| | | CXBCUI | = | PH7D1 T1D1 CM045 | |
| | | CM045 | = | EH + NCMD | |
| | | C0-C7 | = | BC0-BC7 CXBCU + ... | |

(Continued)

Table 3-16. Order-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 (Cont.) | | | for subsequent storage back into the FS-register |
| | Signal CXST goes true | CXST = PH7T1D1<br>C8 = CXST C8ST + ...<br>C8ST = ORDIN I1<br>C9 = CXST TPE + ...<br>C14 = CXST EH + ...<br>S/C15 = CXST ORDIN I2 + ... | Signal CXST updates the status information currently in the C-register |
| | Signal OXTORD goes true | OXTORD = PH7T1D1 TORD<br>O0 = OXTORD O0TOS + ...<br>O0TOS = CM017 + CM018 + CM019<br>CM017 = ORDIN I3 BC3<br>CM018 = ORDIN I4 BC5<br>CM019 = ORDIN I3 BC5<br>O3 = OXTORD EH + ... | Signal OXTORD gates a terminal order to the O-register and, therefore, to the device controller |
| | Signal HJXOF goes true | HJXOF = PH7T1D1 + ...<br>S/H0-S/H6 = OF0-OF6 HJXOF + ...<br>S/J0-S/J2 = OF0-OF2 HJXOF + ... | Interrupt status bits IS0-IS2 (OF0-OF6) are transferred to the H-register for updating |
| | Set bit H7 of the H-register | S/H7 = PH7T1D1 CM005 + ...<br>CM005 = TORD OF7 + NTORD EH | The LSB of the device address (OF7) is set into bit H7 of the H-register |
| | Signal OXTORD goes true | OXTORD = PH7T1D1 TORD<br>S/H1 = OXTORD CM017 + ...<br>S/H2 = OXTORD (CM018 + CM019) | The interrupt status in the H-register is updated |
| | Set flip-flop ES | S/ES = PH7T1D1 + ... | Signal ES is applied to the reset input of the service connect flip-flop in the device controller. This flip-flop resets when it receives signal RSA |
| | Reset flip-flop RSA2 | R/RSA2 = PH7T1D1 TORD + ... | Primes RSA |
| | Set latch PH7PD1 | PH7PD1 = PH7D1 T1D1 + ... | |
| T3D1 | Signal BCXADD goes true | BCXADD = CM021 + ...<br>CM021 = PH7PD1 T3D1 NFIN | The flags and updated status are transferred from the adder to the FS-register |

(Continued)

Table 3-16. Order-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T3D1 (Cont.) | Signal OFXH goes true | OFXH = CM021 + ... | The device address and interrupt status are transferred from the H-register to the IS-register |
| | Set flip-flop RSA1 | S/RSA1 = PH7PD1 T3D1<br>RSA = RSA1 NRSA2 | The service connect flip-flop in the device controller is reset. The device controller is, therefore, electrically disconnected from the MIOP |
| | Reset latch PH7D1 | PH7D1 = PH7D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Set flip-flop FIN | S/FIN = PH7PD1 T4D1 + ... | The current operation no longer needs the MIOP fast access memory |
| | Signal CX0 goes true | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C14 = CX0 + ... | The C-register is cleared |

PH1D1. (See figure 3-43.) The following functions are performed during PH1D1:

a. Core memory is accessed for one data word.

b. The byte address LSB's are checked for a word boundary crossing.

c. The end service line is controlled as required.

d. The byte address LSB's are decoded to select the specific byte of the data word to send to the device controller.

PH2D1. The following functions are performed during PH2D1:

a. The selected byte of data currently in the I-register is transferred to the O-register and placed on the data lines.

b. The byte count is decremented by one.

c. The byte address LSB's are checked for a word boundary crossing.

d. A check is made for a zero byte count condition, and the end data and end service lines are controlled accordingly.

e. The byte address LSB's are checked for a carry to the MSB's.

f. The selected byte of data is sent to the device controller each time PH2D1 is performed.

PH3D1. The MSB's of the byte address in the BA-register are incremented by one, thus effecting the carry from the three LSB's in the OF-register.

PH4D1. The MIOP enters PH4D1 during a data-out service cycle only if data chaining is specified and the byte count has decreased to zero. During this phase the write byte lines are controlled to specify a read operation, and core memory is accessed for the first word of the command doubleword specified by the CA-register. Also, the command doubleword address in the CA-register is incremented by one.

PH5D1. The order bits of the first word of the command doubleword, currently in the I-register, are inspected to determine if a transfer in channel command is specified.

If it is, the address field of the transfer in channel command is set into the CA-register, and the MIOP repeats PH4D1 and PH5D1. Upon repeating PH4D1, the first word of the command doubleword specified by the transfer in channel command will be accessed by the MIOP. The second word will be accessed during PH5D1. When the first word (accessed during PH4D1) is a transfer in channel, the MIOP returns to PH4D1 without accessing memory during PH5D1. When the first word in not a transfer in channel, the MIOP accesses the second word during PH5D1.

PH6D1. The byte count obtained from the second word of the command doubleword is stored in the BC-register and the flags (from the second word) and the LSB's of the byte address (from the first word) are stored in the OF-register.

PH7D1. The following functions are performed during PH7D1:

a. The status in the FS-register is updated.

b. If data chaining occurred without error, the new flags are also set into the FS-register.

c. If a terminal order is required, the interrupt status in the IS-register is updated.

d. If a terminal order is required, the terminal order is assembled in the O-register and sent to the device controller.

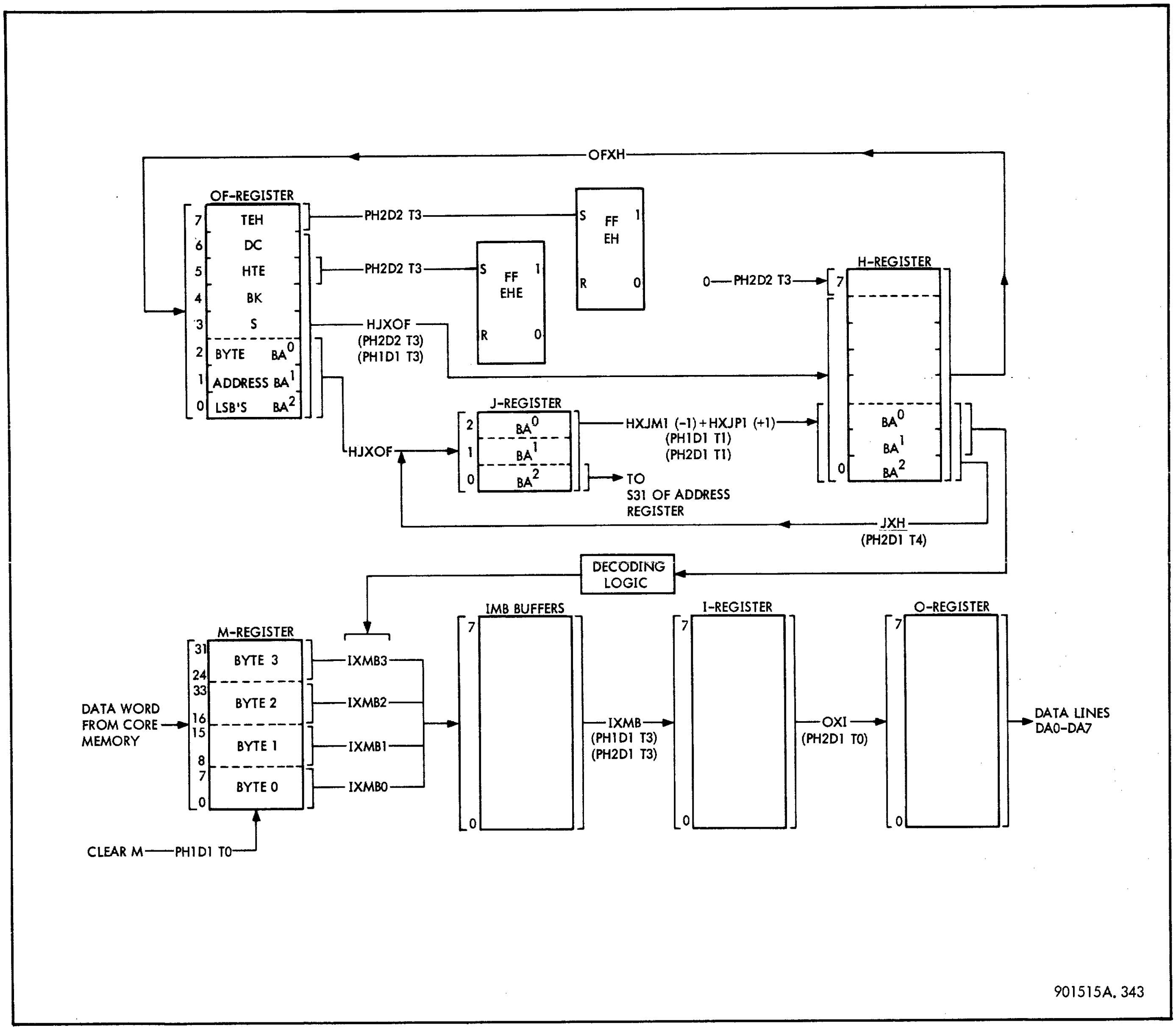



Figure 3-43. Processing a Data-Out Service Cycle

Table 3-17. Data-Out Service Cycle Phase Sequence

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D2 | Delay line D2 started | PLED2 | = | PH1D2 NFSL NAVO SC + ... | The delay line starts when service call SC is received from a device controller |
| T0D2 | Reset latch PH1PD2 | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Set latch PH1PD2 | PH1PD2 | = | PH1D2 T1D2 + ... | |
| T2D2 | Set flip-flop F1 | S/F1 | = | PH1T2D2 T1D2 + ... | Flip-flop F1 drives the ASC function indicator line |
| T3D2 | Reset latch PH1D2 | PH1D2 | = | PH1D2 NT3D2 + ... | |
| T4D2 | Set flip-flop PH2D2 | S/PH2D2 | = | PH1PD2 T4D2 | Next phase in sequence |
| | Set flip-flop FS | S/FS | = | PH1PD2 T4D2 F1 + ... | Primes the service connect flip-flop in the highest priority device controller with a service call pending. This flip-flop will set on the falling edge of signal FS |
| PH2D2 | Delay line D2 started | PLED2 | = | PH2D2 FIN (AVO FSL + ...) (NFNT + ...) RSA2 + ... | The delay line starts when signal FSL is received from the device controller |
| T0D2 | Signal AX0 goes true | AX0 | = | PHT0D2 | The A-register is prepared to receive the device controller address |
| | | A0-A7 | = | A0-A7 NAX0 + ... | |
| | Reset flip-flops MAE, MPE, and latch PH1PD2 | R/MAE | = | R/MPE = PH2T0D2 | |
| | | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Reset flip-flops LS0, LS1, and LS2 | R/LS0 | = | R/LS1 = R/LS2 = CM002 + ... | Selects the BA-, BC-, and OF-registers of appropriate subchannels |
| | | CM002 | = | PH2D2 T1D2 NFNT | |
| | Reset flip-flops CMD, EHE, EH, IXMB, ORD, OUT, TPE, TRA1, and latches TORD and ZBC | R/CMD | = | R/EHE = R/EH = R/IXMB = R/ORD = R/OUT = R/TPE = R/TRA1 = PH2T1D2 | |
| | | TORD | = | TORD NPH2T1D2 + ... | |
| | | ZBC | = | ZBC NPH2T1D2 + ... | |
| | Set flip-flop FP | S/FP | = | PH2T1D2 | |
| | Set latch PH2PD2 | PH2PD2 | = | PH2D2 T1D2 | |
| T2D2 | Clear C- and I-register latches and H-register flip-flops | CX0 | = | HX0 = HUX0 = PH2T2D2 | |
| | | IX0 | = | PH2D2 T2D2 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T2D2 (Cont.) | | R/C15 | = | CX0 + ... | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| | | R/H3-R/H7 | = | HX0 + ... | |
| | | I0-I7 | = | I0-I7 NIX0 + ... | |
| | Clear J-register flip-flops | R/J0-R/J2 | = | PH2T2D2 + ... | |
| | Clear O-register latches | OX0 | = | PH2T2D2 NCM044 + ... | The O-register is prepared to receive the selected byte of the data word from the M- and I-registers |
| | | NCM044 | = | NFNT + F0 | |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |
| | Reset flip-flops SUB and K15 | R/SUB | = | R/K15 = PH2T2D2 | A one will be subtracted from all data transferred through the adder |
| | Reset flip-flop FS | R/FS | = | PH2D2 T2D2 F1 + ... | The service connect flip-flop in the device subcontroller is set when it senses the falling edge of FS. The device controller is, therefore, connected for service |
| | Reset flip-flops ED and ES | R/ED | = | R/ES = PH2D2 T2D2 F1 + ... | |
| T3D2 | Signal HJXOF goes true | HJXOF | = | PH2PD2 T3D2 F1 + ... | The LSB of the word address $BA^2$, stored in OF0 of the OF-register, is set in J0 of the J-register so that it can control the LSB of the core memory address register S31 during the next phase |
| | | S/H0-S/H6 | = | OF0-OF6 HJXOF + ... | |
| | | S/J0-S/J2 | = | OF0-OF2 HJXOF + ... | |
| | Set flip-flop EHE | S/EHE | = | PH2PD2 T3D2 OF5 + ... | Error halt enable flip-flop EHE is set if the halt on transmission error (HTE) flag stored in the OF-register is true |
| | Set flip-flop EH | S/EH | = | PH2PD2 T3D2 OF7 + ... | If the transmission error halt (TEH) flag is true, error halt flip-flop EH will be set |
| | Reset PH2D2 | R/PH2D2 | = | T3D2 + RESET | |
| T4D2 | Set latch PH1D1 | PH1D1 | = | PH1D1S0 + ... | Next phase in sequence |
| | | PH1D1S0 | = | PH2PD2 T4D2 F1 | |
| | Set latch PH1D2 | PH1D2 | = | PH2PD2 T4D2 NF1 NFNT + ... | If the requesting device controller does not reply to the ASC, the MIOP receives signal AVO. The operation is, therefore, aborted and the |
| | | R/F1 | = | R/FS = AVO + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T4D2 (Cont.) | | | | | MIOP returns to the beginning phase PH1D2 |
| PH1D1 | Delay line D1 started | PLED1 | = | PH1D1 RS CM023 + ... | The delay line starts when signal RS is received from the device controller |
| | | CM023 | = | NMS + MDR1 | |
| T0D1 | Clear S-register latches | SX0 | = | CM011 + ... | The S-register is prepared to receive the byte address |
| | | CM011 | = | PH1O4D1 T0D1 | |
| | | S15-S31 | = | S15-S31 NSX0 + ... | |
| | Set flip-flop SXBA | S/SXBA | = | PH1O4D1 T0D1 | The 16 MSB's of the byte address are transferred from the BA-register to S15-S30 of the S-register. The LSB of the word portion of the byte address is controlled by J0 during PH1D1 T1D1 |
| | | S15-S30 | = | BA0-BA15 SXBA + ... | |
| | Signals CXBCU and CXBCL go true | CXBCU | = | CXBCL = CM004 + ... | The byte count is transferred from the BC-register to the C-register and adder. Since signals SUB and K15 are both true, the byte count is decremented by one |
| | | CM004 | = | PH1O2D1 T0D1 | |
| | | C0-C7 | = | CXBCU BC0-BC7 + ... | |
| | | C8-C14 | = | CXBCL BC8-BC14 + ... | |
| | | S/C15 | = | CXBCU BC15 + ... | |
| | Reset flip-flop RSA1 | R/RSA1 | = | PH1O2D1 T0D1 | Prevents sending of RSA to the device controller prematurely |
| | | RSA | = | RSA1 NRSA2 | |
| | Set flip-flops ED, ES, and TRA1 | S/ED | = | S/ES = S/TRA1 = CM030 + ... | Flip-flops ED, ES, and TRA1 are set if a word boundary crossing is detected. Signal ED prevents the MIOP from sending another byte of data to the device controller during the current service cycle (OF4 is true if a read backward order is being executed by the device) |
| | | CM030 | = | CM004 CM029 | |
| | | CM004 | = | PH1O2D1 T0D1 | |
| | | CM029 | = | NJ1 NJ2 H4 + J1 J2 NH4 | |
| | | S/H4 | = | OF4 + ... | |
| | Reset flip-flops MS, MAE, and PRCH | R/MS | = | MCD1 T0D1 + ... | |
| | | MCD1 | = | PH1O4D1 + ... | |
| | | R/MAE | = | PH1D1 T0D1 | |
| | | R/PRCH | = | T0D1 | |
| | Clear M-register | MLX0 | = | MUX0 = CM011 + ... | The M-register is prepared to receive the word of data from core memory |
| | | CM011 | = | PH1O4D1 T0D1 | |
| | | M0-M15 | = | M0-M15 MUX0 + ... | |
| | | M16-M31 | = | M16-M31 MLX0 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 (Cont.) | Clear W-register | WX0 | = | CM011 + ... | Signals W0-W3 define a read operation |
| | | W0-W3 | = | W0-W3 NWX0 | |
| | Reset latch PH1PD1 | PH1PD1 | = | PH1PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Set flip-flop OUT | S/OUT | = | PH1D1 T1D1 IOR + ... | The device controller specifies a data-out service cycle by driving the IOR line and holding the DOR line false. Flip-flop ORD is controlled by signal DOR |
| | Set flip-flop ED | S/ED | = | PH1O2D1 T1D1 EDI + ... | When the device controller can receive no more bytes of data during the current service cycle, it sets flip-flop ED by generating signal EDI, received on the end data line |
| | Set latch S31 | S31 | = | PH1D1 T1D1 J0 NZBC + ... | The LSB of the word portion of the byte count currently in J0 provides the balance of the byte address in the S-register |
| | Reset flip-flop SXBA | R/SXBA | = | T1D1D + ... | Since the byte address is in the S-register, transfer signal SXBA is no longer needed |
| | Clear F-register | FX0 | = | FX0I + ... | The ASC function indicator line, controlled by F1, goes false |
| | | FX0I | = | PH1D1 T1D1 | |
| | | R/F1 | = | FX0 + ... | |
| | Signal HXJP1 goes true | HXJP1 | = | PH1O2D1 T1D1 NH4 | A one is added to the byte address LSB's in the H-register |
| | Reset flip-flop RSA2 | R/RSA2 | = | PH1O2D1 T1D1 + ... | Primes RSA |
| | Reset flip-flops MAR, MDR1, and MPE | R/MAR | = | R/MDR1 = MCD1 T1D1 + ... | |
| | | MCD1 | = | PH1O4D1 + ... | |
| | | R/MPE | = | PH1O4D1 T1D1 + ... | |
| | Set latch PH1PD1 | PH1PD1 | = | PH1D1 T1D1 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T2D1 | Reset flip-flop FIN | R/FIN = PH1T2D1 + ... | A new I/O operation cannot proceed past the first phase if signal FIN is false |
| | Set latch PH1D2 | PH1D2 = PH1T2D1 + ... | A new I/O operation may proceed through the first phase (PH1D2) |
| | Signal HUX0 goes true | HUX0 = HUX0I + ...<br>HUX0I = PH1D1 T2D1 OUT<br>R/H0-R/H2 = HUX0 + ... | Byte address LSB's are cleared from the H-register |
| | Set flip-flop ES | S/ES = PH1O2D1 T2D1 NORD ED NCM047 + ...<br>NCM047 = NZBS NEH | The current service cycle will be concluded without a terminal order (ES true) if this is the last byte of the current service cycle (ED true), if the byte count does not equal zero (NZBS true), and if no errors have been detected (NEH true) |
| | Set flip-flop MS | S/MS = PH1T2D1 CM042 + ...<br>CM042 = NORD OUT NH3 NEH | Signal MS starts a core memory cycle that reads the memory location specified by the S-register |
| T3D1 | Signal HJXOF goes true | HJXOF = PH1PD1 T3D1 OUT + ...<br>S/H0-S/H6 = OF0-OF6 HJXOF + ...<br>S/J0-S/J2 = OF0-OF2 HJXOF + ...<br>IXMB0 = NH1 NH2<br>IXMB1 = NH1 H2<br>IXMB2 = H1 NH2<br>IXMB3 = H1 H2<br>IMB0-IMB7 = IXMB0 M0-M7 + IXMB1 M8-M15 + IXMB2 M16-M23 + IXMB3 M24-M31 | The three LSB's of the byte address are transferred from the OF-register to the H- and J-registers to update the byte address, check for a word boundary crossing, and select from the M-register the specified byte of data for transfer to the device controller |
| | Clear I-register | IX0 = PH1O2PD1 T3D1 OUT + ...<br>I0-I7 = I0-I7 NIX0 + ... | |
| | Set flip-flop IXMB | S/IXMB = PH1O2PD1 OUT + ...<br>I0-I7 = IMB0-IMB7 IXMB + ... | Signal IXMB gates the information in the IMB buffers into the I-register. The buffers contain all zeros until the data word from core memory enters the M-register and the selected byte of that word |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T3D1 (Cont.) | | | enters the buffers (see figure 3-44) |
| | Reset latch PH1D1 | PH1D1 = PH1D1 NPHD1X0<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Set flip-flops MAE and EH | S/MAE = PH1PD1 CM048 + ...<br>CM048 = T4D1 MS NAH<br>S/EH = MAE + ... | If address here AH is not received from core memory by T4D1, memory address error flip-flop MAE is set. Signal MAE sets error halt flip-flop EH |
| | Set latch PH2D1 | PH2D1 = PH2D1S1 + ...<br>PH2D1S1 = PH1PD1 NORD OUT | Next phase in sequence |
| PH2D1 | Delay line D1 started | PLED1 = PH2D1 NFP RS RSA2 + PH2D1 FP CM023<br>CM023 = NMS + MDR1 | If this is the first time through PH2D1 (signal FP true), the delay line starts when the memory response signals have been received (signal MDR1 true). If this is the second, third, or fourth time through PH2D1 (signal FP false), the delay line starts when request strobe RS has been received from the device controller for a byte of data |
| T0D1 | Signals CXBCL and CXBCU go true | CXBCL = CXBCU = CM004 + ...<br>CM004 = PH1O2D1 T0D1<br>C0-C7 = BC0-BC7 CXBCU + ...<br>C8-C14 = BC8-BC14 CXBCL + ...<br>S/C15 = BC15 CXBCU | The byte count is transferred from the BC-register to the C-register and adder. Since signals SUB and K15 are both true, the byte count is decremented by one |
| | Clear O-register | OX0 = PH2T0D1 + ...<br>O0-O7 = O0-O7 NOX0 + ... | |
| | Set flip-flop OXI | S/OXI = PH2T0D1 OUT + ...<br>O0-O7 = I0-I7 OXI + ... | The selected byte of data in the I-register is transferred to the O-register and the device controller |
| | Reset flip-flop IXMB | R/IXMB = T0D1 | |
| | Set flip-flops ED, ES, and TRA1 | S/ED = S/ES = S/TRA1 = CM030 + ...<br>CM030 = CM004 CM029<br>CM004 = PH1O2D1 T0D1<br>CM029 = NJ1 NJ2 H4 + J1 J2 NH4<br>S/H4 = OF4 + ... | The two byte address LSB's in the J-register are checked for a word boundary crossing. If a word boundary crossing is detected, flip-flops ED, ES, and TRA1 are set |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 (Cont.) | Reset RSA1 | R/RSA1 | = | PH1O2D1 T0D1 | Prevents sending of RSA to the device controller prematurely |
| | | RSA | = | RSA1 NRSA2 | |
| | Set flip-flop EH | S/EH | = | EHE MPE + ... | The error halt flip-flop EH is set if a parity error was detected when the data word was accessed from core memory and the error halt enable flip-flop EHE is true. Flip-flop EHE is true if the HTE flag is true |
| | | S/MPE | = | MS PE | |
| | Reset latch PH2PD1 | PH2PD1 | = | PH2PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Set flip-flop ED | S/ED | = | PH1O2D1 T1D1 EDI + ... | When the device controller can receive no more bytes of data during the current service cycle, it drives the ED line |
| | | EDI | = | CR /ED/ | |
| | Signal HXJP1 goes true | HXJP1 | = | PH1O2D1 T1D1 NH4 | Signal HXJP1 increments the byte address by one by transferring the contents of the J-register plus one to the H-register |
| | | S/H0 | = | HXJP1 J1AJ2 NJ0 + ... | |
| | | R/H0 | = | HXJP1 J1AJ2 J0 + ... | |
| | | S/H1 | = | HXJP1 J2 NJ1 + ... | |
| | | R/H1 | = | HXJP1 J1AJ2 + ... | |
| | | S/H2 | = | HXJP1 NJ2 + ... | |
| | | R/H2 | = | HXJP1 J2 + ... | |
| | | J1AJ2 | = | J1 J2 | |
| | Reset flip-flop RSA2 | R/RSA2 | = | PH1O2D1 T1D1 + ... | Primes RSA |
| | Set latch PH2PD1 | PH2PD1 | = | PH2D1 T1D1 | |
| T2D1 | Signal BCXADD goes true | BCXADD | = | PH2T2D1 + ... | The incremented byte count in the C-register is transferred back to the BC-register |
| | Signal OFXH goes true | OFXH | = | PH2T2D1 + ... | The updated byte address (three LSB's) is transferred from the H-register to the OF-register |
| | Reset flip-flop OXI | R/OXI | = | T2D1 + ... | |
| | Set flip-flop ZBC | S/ZBC | = | ZBS PH2D1 T2D1 + ... | Signal ZBS indicates that the next count of the adder will be zero. Since the byte count is currently in the adder, a zero byte count condition sets flip-flop ZBC |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T2D1 (Cont.) | Set flip-flop ED, set latch TORD, and reset flip-flop ES | S/ED | = | PH1O2D1 T2D1 NORD CM047 + ... | Either a zero byte count (ZBC) or error halt (EH) condition will stop the MIOP from sending another byte of data during the current service cycle and will also ensure the sending of a terminal order to report the condition to the device controller |
| | | CM047 | = | EH + ZBC | |
| | | TORD | = | PH2D1 T2D1 CM047 + ... | |
| | | R/ES | = | PH1O2D1 T2D1 NORD CM047 + ... | |
| | Set flip-flop ES | S/ES | = | PH1O2D1 T2D1 NORD ED NCM047 | The current service cycle will be concluded without a terminal order (ES true) if this is the last byte of the current service cycle (ED true), if the byte count does not equal zero (NZBS true), and if no errors have been detected (NEH true) |
| | | NCM047 | = | NZBS NEH | |
| | Set flip-flop RSA1 | S/RSA1 | = | CM036 (NED + FP + TRA1) | Request strobe acknowledge RSA is sent to the device controller if error halt EH is not true and a zero byte count condition has not been detected (ZBC true), and if one or more of the following conditions exist:<br>a. If this is not the last byte of the current service cycle (NED)<br>b. If this is the first time through PH2D1 (FP)<br>c. If a word boundary will be crossed if another byte is transmitted (TRA1) |
| | | CM036 | = | PH2D1 T2D1 NEH NZBS | |
| | | RSA | = | RSA1 NRSA2 | |
| | Signal CBA goes true | CBA | = | NH4NH0 TRA1 + ... | Signal TRA1 is true if the two LSB's of the byte address ($BA^0$, $BA^1$) in the J-register were true before a one was added and transferred to the H-register. Signal H0 ($BA^2$) is false if $BA^2$ was true before a one was added when it was in the J-register. Since the three LSB's were true before a one was added, a carry must be made to the next higher bit of the byte address ($BA^3$) in the BA-register. Signal CBA provides for this carry |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T3D1 | Reset flip-flop FP | R/FP | = | PH2PD1 T3D1 + ... | Permits the delay line to start for the second, third, and fourth bytes of data to be sent to the device controller |
| | Clear I-register latches | IX0 | = | PH1O2PD1 T3D1 OUT + ... | |
| | | I0-I7 | = | I0-I7 NIX0 + ... | |
| | Set flip-flop IXMB | S/IXMB | = | PH1O2PD1 T3D1 OUT + ... | The selected byte of data in the IMB buffers is transferred to the I-register |
| | | I0-I7 | = | IMB0-IMB7 IXMB + ... | |
| | Set flip-flop LS0 | S/LS0 | = | PH1O2PD1 T3D1 ZBC + ... | Selects CA-register if the byte count equals zero |
| | Reset latch PH2D1 | PH2D1 | = | PH2D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Clear C-register latches | CX0 | = | CM013 T4D1 + ... | |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 | |
| | | R/C15 | = | CX0 + ... | |
| | Signal JXH goes true | JXH | = | PH1O2PD1 T4D1 | The incremented byte address (three LSB's) are returned to the J-register from the H-register |
| | | S/J0-S/J2 | = | H0-H2 JXH + ... | |
| | Set flip-flop RSA1 | S/RSA1 | = | PH2PD1 T4D1 | Request strobe acknowledge RSA is unconditionally sent to the device controller |
| | | RSA | = | RSA1 NRSA2 | |
| | Set flip-flop LS1 | S/LS1 | = | PH2PD1 T4D1 ED + ... | Selects FS-register |
| | Set flip-flop LS2 | S/LS2 | = | PH1O2PD1 T4D1 TORD + ... | Selects IS-register |
| | Set latch PH2D1 | PH2D1 | = | PH2D1S0 T4D1 + ... | If not end data, the MIOP repeats PH2D1 to send another byte of data to the device controller |
| | | PH2D1S0 | = | PH1PD1 NORD OUT | |
| | Set latch PH3D1 | PH3D1 | = | PH3D1S0 T4D1 + ... | If a carry to the 16 MSB's of the byte count is required (CBA true), and the byte count does not equal zero, the MIOP will advance to PH3D1 |
| | | PH3D1S0 | = | PH1O2PD1 NORD NFP NZBC CBA | |
| | Set latch PH4D1 | PH4D1 | = | PH4D1S0 T4D1 + ... | If the byte count equals zero (ZBC), and the data chaining flag is true (H6), the MIOP will advance to PH4D1 to start the chaining operation |
| | | PH4D1S0 | = | PH1O2PD1 ZBC H6 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 (Cont.) | Set latch PH7D1 | PH7D1 = PH7D1S0 T4D1 + ... | The MIOP advances to PH7D1 if the byte count equals zero (ZBC) and data chaining is not specified (NH6), or if the byte count does not equal zero (NZBC), the byte address MSB's do not need updating (NCBA), and end data has been specified (ED) |
| | | PH7D1S0 = PH1O2PD1 CM008 + ... | |
| | | CM008 = NBC NH6 + ZBC NCBA ED NFP + ... | |
| PH3D1 | Delay line D1 started | PLED1 = PH3D1 + ... | The delay line starts unconditionally at the end of T4D1 of the preceding phase |
| T0D1 | Reset flip-flop SUB | R/SUB = PH3D1 T0D1 NH4 + ... | Since K15 is true, all data transferred through the adder is incremented by one |
| | Signal CXBA goes true | CXBA = PH3D1 T0D1 + ... | The byte address MSB's are transferred to the C-register and adder, and are thus incremented by one. This operation effectively provides the carry to the byte address MSB's from the LSB's |
| | Reset latch PH3PD1 | PH3PD1 = PH3PD1 NPHPD1X0 | |
| | | NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Set flip-flop FP | S/FP = PH3D1 T1D1 OUT CM032 + ... | Flip-flop FP is set if there have been no memory parity or address errors recorded and if there will be no terminal order during the current service cycle. Logic decisions are made (directly and indirectly) during this phase and PH7D1 based on the state of signal FP |
| | | CM032 = NMPE NMAE NTPE NTORD | |
| | Set latch PH3PD1 | PH3PD1 = PH3D1 T1D1 + ... | |
| T2D1 | Signal BAXADD goes true | BAXADD = PH3D1 T2D1 + ... | The incremented byte address is transferred back to the BA-register |
| T3D1 | Set flip-flop FIN | S/FIN = PH3PD1 T3D1 FP + ... | Permits the next I/O operation to proceed through PH2D2 |
| | Reset latch PH3D1 | PH3D1 = PH3D1 NPHD1X0 | |
| | | NPHD1X0 = NT3D1 NRESET | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 | Clear C-register latches | CX0 = CM013 T4D1 + ...<br>CM013 = NPH7PD1 NPH10PD1 NPH11PD1<br>C0-C14 = C0-C14 NCX0 + ...<br>R/C15 = CX0 + ... | |
| | Set latch PH7D1 | PH7D1 = PH7D1S0 T4D1 + ...<br>PH7D1S0 = PH3PD1 NFP | The MIOP advances to PH7D1 from PH3D1 if there was a memory error recorded or if a terminal order was specified |
| PH4D1 | Delay line D1 started | PLED1 = PH4D1 CM023 + ...<br>CM023 = NMS + MDR1 | Since signal MDR1 was true before the preceding phase (PH2D1) started, the delay line starts for PH4D1 at the end of T4D1 of the preceding phase |
| T0D1 | Reset flip-flop SUB | R/SUB = PH4D1 T0D1 + ... | Since K15 is true, a one will be added to any data transferred through the adder |
| | Signal CXBA goes true | CXBA = PH4D1 T0D1 + ...<br>C0-C14 = CXBA BA0-BA14 + ...<br>S/C15 = CXBA BA15 + ... | The command address is transferred from the CA-register to the C-register and adder |
| | Clear S-register | SX0 = CM011 + ...<br>CM011 = PH1O4D1 T0D1<br>S15-S31 = S15-S31 NSX0 + ... | |
| | Set flip-flop SXBA | S/SXBA = PH1O4D1 T0D1<br>S15-S30 = BA0-BA15 SXBA + ... | The command address is transferred from the CA-register to bits S15-S30 of the S-register. Bit S31 is a zero. The first word of the command doubleword will, therefore, be fetched from core memory |
| | Set flip-flop FP | S/FP = PH4T0D1 MPE + ... | Flip-flop FP records, until PH7D1, that a memory parity error occurred (MPE) during the data portion of the data-out service cycle |
| | Set flip-flop EH | S/EH = MPE EHE + ... | Error halt flip-flop EH is set if a memory parity error occurred and the halt on transmission error flag HTE is true. Signal OF5 sets flip-flop EHE during PH2D2 if the HTE flag is true |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 (Cont.) | Reset flip-flop MS | R/MS | = | MCD1 T0D1 | |
| | | MCD1 | = | PH1O4D1 + ... | |
| | Signals MLX0 and MUX0 go true | MLX0 | = | MUX0 = CM011 + ... | |
| | | M0-M15 | = | M0-M15 NMUX0 + ... | |
| | | M16-M31 | = | M16-M31 NMLX0 + ... | |
| | Signal WX0 goes true | WX0 | = | CM011 + ... | A core memory read operation is specified |
| | | W0-W3 | = | W0-W3 NW0X + ... | |
| | Reset latch PH4PD1 | PH4PD1 | = | PH4PD1 NPHPD1X0 + ... | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Reset flip-flop SXBA | R/SXBA | = | T1D1D + ... | The transfer signal (SXBA) that transfers the command address into the S-register is no longer needed |
| | Reset flip-flops MAR, MPE, and MDR1 | R/MAR | = | R/MDR1 = MCD1 T1D1 + ... | |
| | | MCD1 | = | PH1O4D1 + ... | |
| | | R/MPE | = | PH1O4D1 T1D1 + ... | |
| | Set flip-flop MS | S/MS | = | MSSET + ... | If there have been no error conditions detected thus far during the current service (NEH), and if the device controller has not set the chaining modifier CM during the previous order-in service cycle (NBC15), flip-flop MS will be set. Signal MS starts a core memory access |
| | | MSSET | = | PH4T1D1 NEH NBC15 + ... | |
| | Set latch PH4PD1 | PH4PD1 | = | PH4D1 T1D1 | |
| T2D1 | Signals HUX0 and HX0 go true | HUX0 | = | HX0 = CM020 + ... | The H-register is cleared |
| | | CM020 | = | PH4D1 T2D1 (NEH + MS) | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| | | R/H3-R/H7 | = | HX0 + ... | |
| | Signal BAXADD goes true | BAXADD | = | CM020 + ... | The incremented command address is returned to the CA-register |
| T3D1 | Signal IX0 goes true | IX0 | = | PH4PD1 T3D1 + ... | The I-register is cleared |
| | | I0-I7 | = | I0-I7 NIX0 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T3D1 (Cont.) | Set flip-flop IXMB | S/IXMB | = | PH4PD1 T3D1 + ... | Signal IXMB transfers the information from the IMB buffers to the I-register. If a memory cycle was started (MS true), during PH5D1 the buffers will contain the order bits of the command doubleword accessed from core memory |
| | Set flip-flop CMD | S/CMD | = | PH4PD1 T3D1 (NEH + MS) + ... | This function applies when chaining modifier CM is true (signal BC15 true) and this is the first time through PH4D1 (no core memory access made). The current command doubleword is not accessed; however, when PH4D1 is repeated, the command doubleword in the next higher memory location will be accessed |
| | Reset latch PH4D1 | PH4D1 | = | PH4D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Reset flip-flops LS1 and LS2 | R/LS1 | = | R/LS2 = PH4PD1 T4D1 + ... | Selects BC- and OF-registers |
| | Reset flip-flop K15 | R/K15 | = | PH4PD1 T4D1 MS + ... | Data being transferred through the adder will not be altered |
| | Set flip-flops MAE and EH | S/MAE | = | PH4PD1 CM048 + ... | If a memory access was made, and memory did not respond with address here AH, flip-flops MAE and EH record the error condition until the termination phase |
| | | CM048 | = | T4D1 MS NAH | |
| | | S/EH | = | MAE + ... | |
| | Set latch PH4D1 | PH4D1 | = | PH4D1S0 T4D1 | This function applies if signal CM (BC15) is true and this is the first time through PH4D1. In this case, PH4D1 will be repeated and a memory access will be made the second time through |
| | | PH4D1S0 | = | PH4PD1 NEH NMS | |
| | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | Set latch PH5D1 | PH5D1 | = | PH5D1S0 T4D1 + PH5D1S1 T4D1 | If a memory access had been made (MS true), or any error conditions detected (EH true), the MIOP will advance to PH5D1 |
| | | PH5D1S0 | = | PH4PD1 EH | |
| | | PH5D1S1 | = | PH4PD1 MS | |

(Continued)

Table 3-17. Order-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| PH5D1 | Delay line D1 started | PLED1 = PH5D1 CM023 + ...<br>CM023 = NMS + MDR1 | The delay line starts when the reaction signals have been received from core memory (signal MDR1 true) |
| T0D1 | Reset flip-flop IXMB | R/IXMB = T0D1 + ... | |
| | Set latch S31 | S31 = PH5D1 T0D1 + ... | Since the address of the first word of the command doubleword was set into the S-register during PH4D1, bit S31 true specifies the address of the second word |
| | Set flip-flop EH | S/EH = CMD MPE + ...<br>S/MPE = PE MS | If a parity error was detected while reading the first word of the command doubleword from core memory (signal MPE true), error halt flip-flop EH will be set |
| | Signal CXM goes true | CXM = PH5D1 T0D1 TRA<br>C0-C14 = M16-M30 CXM + ...<br>S/C15 = M31 CXM + ...<br>TRA = I4 NI5 NI6 NI7 | If the first word of the command doubleword (currently in the I-register) specifies a transfer in channel command (signal TRA true), the address bits of the transfer in channel command currently in the M-register will be transferred to the C-register |
| | Set flip-flop EH | S/EH = PH5D1 T0D1 TRA TRA1 + ...<br>S/TRA1 = PH5PD1 T4D1 + ... | Signals TRA and TRA1 both true at the same time indicates that two transfers in channel command have been accessed from core memory in succession, indicating a control error |
| | Reset flip-flop LS0 | R/LS0 = PH5D1 T0D1 NTRA + ... | Selects BA-register if not a transfer in channel command (NTRA) |
| | Signal CXMR3 goes true | CXMR3 = PH5D1 T0D1 NTRA<br>C0-C14 = M13-M27 CXMR3 + ...<br>S/C15 = M28 CXMR3 + ... | The 16 MSB's of the byte address are transferred from the M-register to the BA-register if the last word accessed from core memory was not a transfer in channel command |
| | Signal HUXM goes true | HUXM = PH5D1 T0D1 NTRA<br>H0-H2 = M29-M31 HUXM + ... | The three LSB's of the byte address are transferred to the H-register if the last word accessed from core memory was not a transfer in channel command |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 (Cont.) | Reset flip-flop MS | R/MS | = | MCD1 T0D1 + ... | |
| | | MCD1 | = | PH5D1 + ... | |
| | Reset latch PH5PD1 | PH5PD1 | = | PH5PD1 NPHPD1X0 + ... | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Reset flip-flops MAR and MDR1 | R/MAR | = | R/MDR1 = MCD1 T1D1 + ... | |
| | Clear M-register latches | MLX0 | = | MUX0 = MLX0I + ... | |
| | | MLX0I | = | PH5D1 T1D1 + ... | |
| | | M0-M15 | = | M0-M15 NMUX0 + ... | |
| | | M16-M31 | = | M16-M31 NMLX0 + ... | |
| | Set flip-flop MS | S/MS | = | MSSET + ... | If the last word accessed from core memory was not a transfer in channel command, MS goes true to start another core memory access for the second word of the command doubleword |
| | | MSSET | = | PH5D1 T1D1 NEH NTRA | |
| | Set latch PH5PD1 | PH5PD1 | = | PH5D1 T1D1 | |
| T2D1 | Reset flip-flop OXI | R/OXI | = | T2D1 + ... | |
| | Signal BAXADD goes true | BAXADD | = | PH5D1 T2D1 NEH + ... | If a transfer in channel command is being processed, signal BAXADD stores the command address encoded in the transfer in channel command in the CA-register (signal LS0 is true). If a transfer in channel command is not being processed, signal BAXADD stores the byte address encoded in the first word of the command doubleword in the BA-register (signal LS0 is false) |
| T3D1 | Reset latch PH5D1 | PH5D1 | = | NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Clear C-register latches | CX0 | = | CM013 T4D1 + ... | |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T4D1 (Cont.) | Set flip-flop TRA1 | S/TRA1 | = | PH5PD1 T4D1 + ... | Signal TRA1 is used with TRA to detect two successive transfers in channel command |
| | Set flip-flop K15 | S/K15 | = | PH5PD1 T4D1 TRA + ... | Flip-flop K15 is set if a transfer in channel command is being processed |
| | Set flip-flops MAE and EH | S/MAE | = | PH5PD1 CM048 + ... | If core memory has not responded to the last memory request with signal address here (AH), flip-flops MAE and EH are set to record the error condition until the termination phase |
| | | CM048 | = | T4D1 MS NAH | |
| | | S/EH | = | MAE + ... | |
| | Set latch PH4D1 | PH4D1 | = | PH4D1S0 T4D1 + ... | If a transfer in channel command is being processed (signal TRA true) and no error conditions have been recorded (signal NEH true), the MIOP will return to PH4D1 |
| | | PH4D1S0 | = | PH5PD1 NCM007 + ... | |
| | | NCM007 | = | NEH TRA | |
| | Set latch PH6D1 | PH6D1 | = | PH6D1S0 T4D1 + ... | PH6D1 is the next phase in sequence if a transfer in channel command is not being processed or if an error condition was detected |
| | | PH6D1S0 | = | PH5PD1 CM007 | |
| | | CM007 | = | EH + NTRA | |
| PH6D1 | Delay line D1 started | PLED1 | = | PH6D1 CM023 + ... | The delay line starts when the reaction signals are received from core memory (signal MDR1 true). The second word of the command doubleword is now in the M-register |
| | | CM023 | = | NMS + MDR1 | |
| T0D1 | Signal CXM goes true | CXM | = | PH6D1 T0D1 + ... | Signal CXM transfers the byte count from the M-register to the C-register |
| | | C0-C14 | = | M16-M31 CXM + ... | |
| | | S/C15 | = | M31 CXM + ... | |
| | Set bits H3-H6 of the H-register | S/H3 | = | PH6T0D1 M7 + ... | The flags are transferred from the M-register to the H-register |
| | | S/H4 | = | PH6T0D1 CM003 + ... | |
| | | S/H5 | = | PH6T0D1 M4 + ... | |
| | | S/H6 | = | PH6T0D1 M0 + ... | |
| | | CM003 | = | NORD OF4 + ... | |
| | Set flip-flop EH | S/EH | = | CMD MPE + ... | The error halt flip-flop is set if a parity error is detected when reading core memory (signal PE true) |
| | | MPE | = | MS PE | |
| | Reset latch PH6PD1 | PH6PD1 | = | PH6PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T1D1 | Set latch PH6PD1 | PH6PD1 | = | PH6D1 T1D1 | |
| T2D1 | Signal BCXADD goes true | BCXADD | = | CM022 + ... | If no error conditions have been detected (signal NEH true), the byte count MSB's are stored in the BC-register |
| | | CM022 | = | PH6D1 T2D1 NEH + ... | |
| | Signal OFXH goes true | OFXH | = | CM022 + ... | If no error conditions have been detected, the byte count LSB's and the flags are stored in the OF-register |
| T3D1 | Reset latch PH6D1 | PH6D1 | = | PH6D1 NPHD1X0 + ... | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Signal CX0 goes true | CX0 | = | CM013 T4D1 + ... | The C-register is cleared |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Set flip-flops LS1 and LS2 | S/LS1 | = | S/LS2 = PH6PD1 T4D1 + ... | Selects FS- and IS-registers |
| | Set latch PH7D1 | PH7D1 | = | PH6PD1 T4D1 + ... | Next phase in sequence |
| PH7D1 | Delay line D1 started | PLED1 | = | PH7D1 CM025 + ... | If a terminal order is to be sent to the device controller, the delay line starts when signal RS is received from the device controller and MDR1 or NMS are true. If no terminal order is to be sent, the delay line starts when T4D1 of the previous phase is completed and MDR1 or NMS are true |
| | | CM025 | = | CM023 CM025I1 | |
| | | CM023 | = | MDR1 + ... | |
| | | NCM025I1 | = | NCM025I1I + ... | |
| | | NCM025I1I | = | RS RSA2 | |
| T0D1 | Clear H-register | HUX0 | = | HX0 = HX0I + ... | |
| | | HX0I | = | PH7D1 T0D1 + ... | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| | | R/H3-R/H7 | = | HX0 + ... | |
| | Reset flip-flop K15 | R/K15 | = | PH7D1 T0D1 + ... | Data that is transferred through the adder will not be altered |
| | Clear O-register latches | OX0 | = | PH7D1 T0D1 TORD + ... | If a terminal order is specified (signal TORD true), the O-register is cleared |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 (Cont.) | Reset flip-flop RSA1 | R/RSA1 = PH7D1 T0D1 TORD + ... | Flip-flop RSA1 is set again during T3D1 to generate RSA. Signal RSA is generated only if a terminal order is specified |
| | Reset latch PH7PD1 | PH7PD1 = PH7PD1 NPHPD1X0 + ...<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Signal CXBCL goes true | CXBCL = PH7T1D1 + ...<br>C8-C14 = BC8-BC14 CXBCL + ... | Status information from bits 8-14 of the FS-register is transferred to bits 8-14 of the C-register |
| | Signal CXBCU goes true | CXBCU = CXBCUI + ...<br>CXBCUI = PH7D1 T1D1 CM045<br>CM045 = EH + NCMD<br>C0-C7 = BC0-BC7 CXBCU + ... | The flags stored in the FS-register are transferred to the C-register if an error was detected (signal EH true) or if data chaining did not occur (signal NCMD true) |
| | Signal CXMB0 goes true | CXMB0 = PH7D1 T1D1 NCM045<br>NCM045 = CMD NEH<br>C0-C7 = M0-M7 CXMB0 + ... | If data chaining occurred (signal CMD true) and no error conditions were detected (signal NEH true), the new flags in the M-register are transferred to the C-register for subsequent storage in the FS-register |
| | Signal CXST goes true | CXST = PH7T1D1<br>C10 = CXST C10ST<br>C10ST = MPE NCMD + FP NORD<br>C11 = CXST MAE + ...<br>C12 = CXST C12ST + ...<br>C12ST = CMD MPE<br>C13 = CXST C13ST + ...<br>C13ST = CMD EH NMPE NMAE<br>C14 = CXST EH + ... | Signal CXST updates the status information currently in the C-register |
| | Signal OXTORD goes true | OXTORD = PH7T1D1 TORD<br>O0 = OXTORD O0TOS + ...<br>O0TOS = CM016 + ...<br>CM016 = ZBC BC1<br>O1 = OXTORD O1TOS + ...<br>O1TOS = ZBC NBC0<br>O2 = OXTORD BC2 + ...<br>O3 = OXTORD EH + ...<br>O4 = OXTORD O4TOS + ...<br>O4TOS = OUT EH NCMD + ... | Signal OXTORD gates a terminal order to the O-register and, therefore, to the device controller if a terminal order was specified |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T1D1 (Cont.) | Signal HJXOF goes true | HJXOF<br>S/H0-S/H6<br>S/J0-S/J2 | =<br>=<br>= | PH7T1D1 + ...<br>HJXOF OF0-OF6 + ...<br>HJXOF OF0-OF2 + ... | If a terminal order is specified, the interrupt status and the three MSB's of the device address (from IS4-IS6) are set into the H- and J-registers. If a terminal order is not specified, the operating flags (from OF3-OF6) and the three byte address LSB's are set into the H- and J-registers. If a terminal order is specified, the interrupt status is updated in the H-register and returned to the IS-register. If a terminal order is not specified, the information is returned to the OF-register unaltered |
| | Set flip-flop H7 | S/H7<br>CM005 | =<br>= | PH7T1D1 CM005 + ...<br>TORD OF7 + NTORD EH | If a terminal order is specified, the LSB of the device address (from IS7) is set into the H-register to complete the device address transfer. If a terminal order is not specified and an error halt condition occurred (signal EH true), bit H7 will be set, thus retaining a record of the error when the H-register contents are transferred back to the OF-register |
| | Signal OXTORD goes true | OXTORD<br>S/H0<br>CM016 | =<br>=<br>= | TORD PH7T1D1<br>OXTORD CM016 + ...<br>ZBC BC1 | If a terminal order is specified, signal OXTORD permits the interrupt status in the H-register to be updated. Bit H0 is set if the byte count equals zero and the interrupt at zero byte count flag (BC1) is true |
| | Set flip-flop ES | S/ES | = | PH7T1D1 + ... | Signal ES causes the device controller to disconnect from the device controller interface lines after it also receives signal RSA |
| | Set flip-flop FIN | S/FIN<br>CM032 | =<br>= | PH7T1D1 CM032 + ...<br>NMPE NMAE NTPE NTORD | Flip-flop FIN is set if a terminal order is not specified and no memory address or parity errors were detected during the current service cycle. Signal FIN permits the next I/O operation to continue. Signal NFIN gates the information in the adder and H-register back into their respective fast access registers during T3D1 of this phase |

(Continued)

Table 3-17. Data-Out Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T1D1 (Cont.) | Reset flip-flop RSA2 | R/RSA2 | = | PH7T1D1 TORD + ... | Primes RSA |
| | Set latch PH7PD1 | PH7PD1 | = | PH7D1 T1D1 + ... | |
| T3D1 | Signal BCXADD goes true | BCXADD | = | CM021 + ... | The flags and status currently in the C-register are set into the FS-register |
| | | CM021 | = | PH7PD1 T3D1 NFIN | |
| | Signal OFXH goes true | OFXH | = | CM021 + ... | If a terminal order is specified, the interrupt status and device address in the H-register are transferred to the IS-register. If a terminal order is not specified, the operating flags and byte address LSB's in the H-register are transferred to the OF-register |
| | Set flip-flop RSA1 | S/RSA1 | = | PH7PD1 T3D1 | The service connect flip-flop in the device controller is reset when it receives signal RSA. The device controller is, therefore, electrically disconnected from the interface |
| | | RSA | = | RSA1 NRSA2 | |
| | Reset latch PH7D1 | PH7D1 | = | PH7D1 NPHD1X0 + ... | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Set flip-flop FIN | S/FIN | = | PH7PD1 T4D1 + ... | The current operation no longer needs the MIOP's fast access memory |

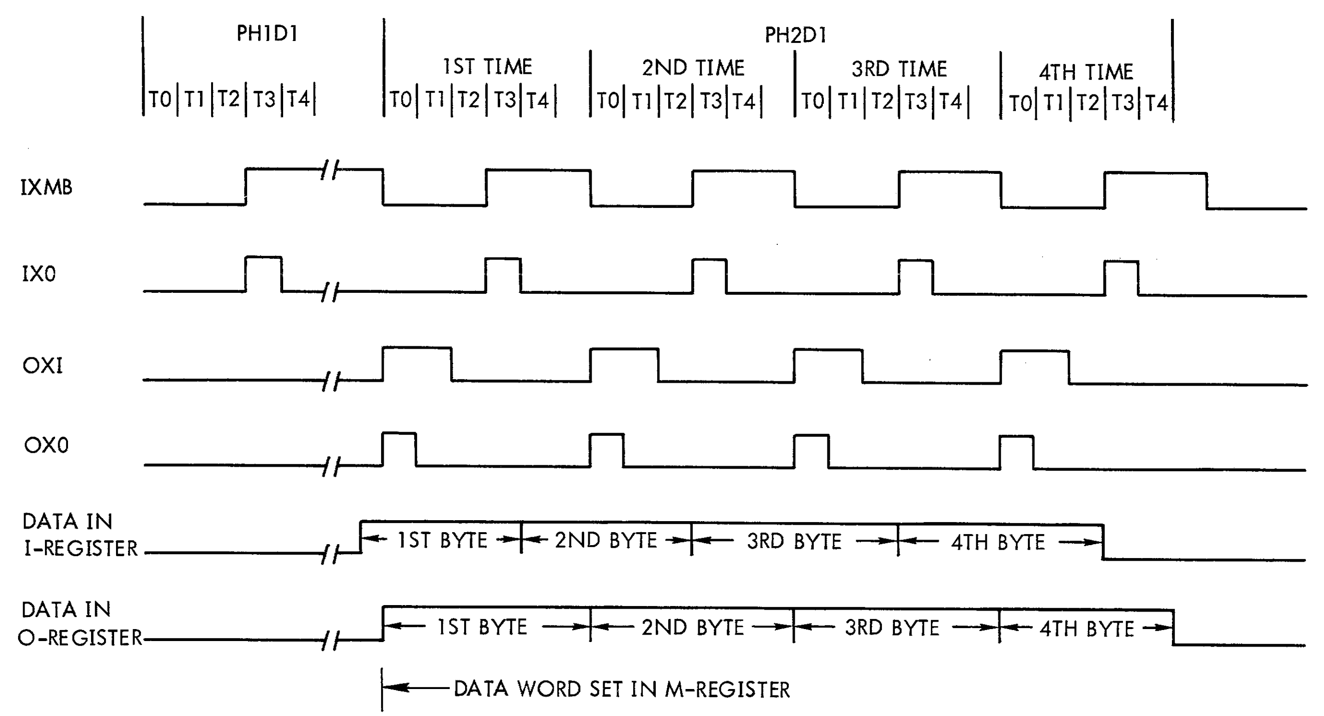



Figure 3-44. Processing Data During a Data-Out Service Cycle, Timing Diagram

3-43 Data-In Service Cycle

The data-in service cycle phases follow the sequence shown in figure 3-45. During a data-in service cycle, the MIOP accepts up to four bytes of data from the device controller and stores the word (or part of word) into the core memory location specified by the byte address register. The MIOP accepts the first byte of data during PH1D1. If more bytes are to be accepted, the MIOP cycles through PH2D1 for each additional byte. The MIOP includes PH3D1 in the sequence when a carry from the three LSB's to the 16 MSB's of the byte address is required. (When the device is performing a read backward order, a borrow from the MSB's is required.) When the byte count has been reduced to zero and data chaining is specified, phases PH4D1, PH5D1, and PH6D1 are included in the sequence, during which time the MIOP accesses the next command doubleword from core memory. If the first word of the command doubleword specifies a transfer in channel command (signal TRA true), phases PH4D1 and PH5D1 are repeated so that the MIOP can branch to the new memory location for the data. The MIOP enters the termination phase (PH7D1) either immediately after the first byte of data is accepted, after the byte address has been updated, or after the chaining operations have been completed.

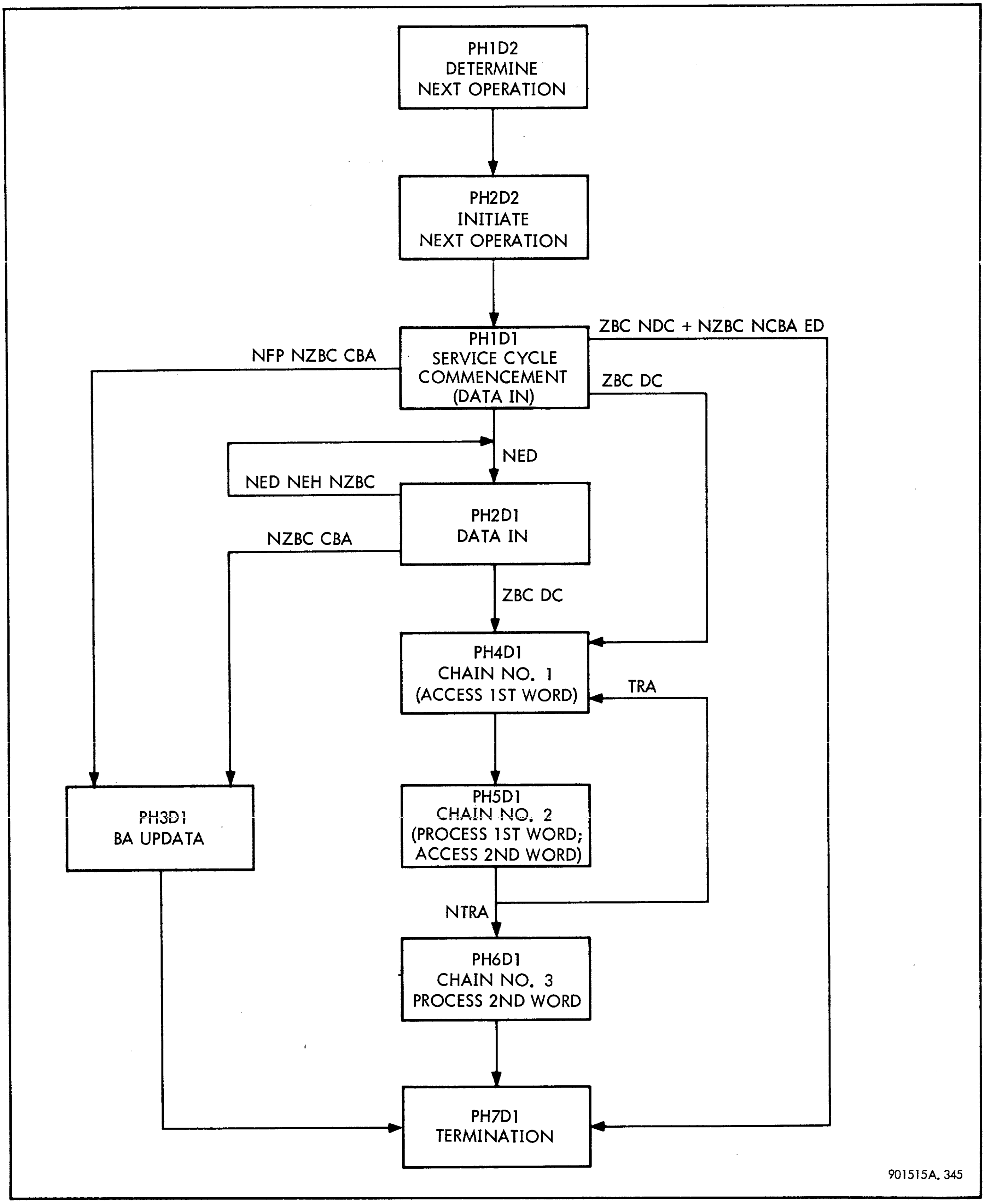



Figure 3-45. Phase Sequence Diagram for a Data-In Service Cycle

PH1D2. The MIOP determines that the request was initiated by a device controller and, therefore, drives the acknowledge service line and function strobe line at the device controller interface.

PH2D2. The delay line starts after the device controller responds to the function strobe. The address the responding device controller has placed on the FR lines is set into the A-register to select the subchannel associated with that device controller. Also certain MIOP flip-flops and registers are cleared.

PH1D1. The following functions are performed during PH1D1:

a. The MIOP sets into its I-register the byte of data the device controller has placed on the data lines (see figure 3-46).

b. The data in the I-register is transferred into the byte position of the M-register specified by the two byte address LSB's currently in the J-register.

c. The write byte lines are controlled by the two byte address LSB'S.

d. The byte address is updated and checked for a word boundary crossing, and the byte count is decremented by one.

e. The end data and end service lines are controlled as required.

f. If the first byte is the last to be accepted from the device controller during the current service cycle, the MIOP stores this byte into the core memory location specified by the byte address.

PH2D1. The following functions are performed during PH2D1:

a. The MIOP sets into its I-register the byte of data the device controller has placed on the data lines. (This is the second, third, or fourth byte.)

b. The data in the I-register is transferred to the byte position specified by the two byte address LSB's currently in the J-register. (The byte address LSB's are altered with every byte accepted from the device controller.)

c. The write byte lines are controlled (indirectly) by the two byte address LSB's.

d. The byte count is decremented by one and checked for zero.

e. The byte address is updated and checked for a word boundary crossing.

f. The end data and end service lines are controlled as required.

g. If the current byte is the last to be accepted from the device controller during the current service cycle, the MIOP stores the contents of the M-register into the core memory location specified by the byte address.

PH3D1. The MSB's of the byte address in the BA-register are incremented by one if the device is executing a read order or decremented by one if the device is executing a read backward order. The carry or borrow is effectively made from the three LSB's of the byte address as required.

Note

Phases PH4D1, PH5D1, PH6D1, and PH7D1 for a data-in service cycle are identical to the same phases of a data-out service cycle. (See paragraph 3-42.)

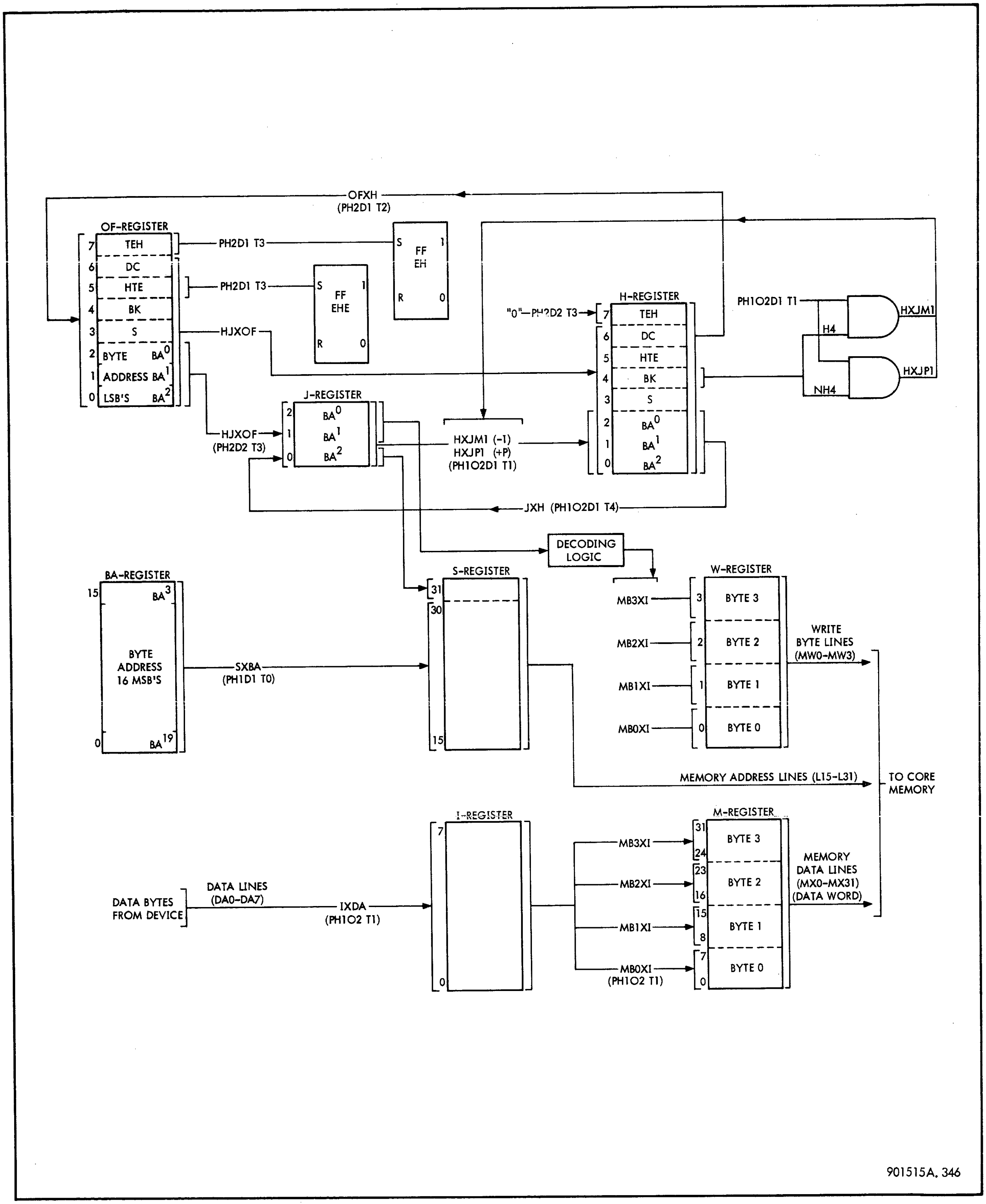



Figure 3-46. Processing Data During a Data-In Service Cycle

Table 3-18. Data-In Service Cycle Phase Sequence

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH1D2 | Delay line D2 started | PLED2 | = | PH1D2 NFSL NAVO SC + ... | The delay line starts when service call SC is received from a device controller |
| T0D2 | Reset latch PH1PD2 | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D2 | Set latch PH1PD2 | PH1PD2 | = | PH1D2 T1D2 + ... | |
| T2D2 | Set flip-flop F1 | S/F1 | = | PH1T2D2 T1D2 NFNT SC + ... | Flip-flop F1 drives the ASC function indicator line |
| T3D2 | Reset latch PH1D2 | PH1D2 | = | PH1D2 NT3D2 + ... | |
| T4D2 | Set PH2D2 | S/PH2D2 | = | PH1PD2 T4D2 | Next phase in sequence |
| | Set flip-flop FS | S/FS | = | PH1PD2 T4D2 F1 + ... | Primes the service connect flip-flop in the highest priority device controller with a service call pending. This flip-flop will set on the falling edge of signal FS |
| PH2D2 | Delay line D2 started | PLED2 | = | PH2D2 FIN (AVO FSL + ...) (NFNT + ...) RSA2 + ... | The delay line starts when signal FSL is received from the device controller |
| T0D2 | Signal AX0 goes true | AX0 | = | PH2T0D2 | The A-register is prepared to receive the device controller address |
| | | A0-A7 | = | A0-A7 NAX0 + ... | |
| | Reset flip-flops MAE, MPE, and latch PH1PD2 | R/MAE | = | R/MPE = PH2T0D2 | |
| | | PH1PD2 | = | PH1PD2 NT0D2 + ... | |
| T1D1 | Reset flip-flops LS0, LS1, and LS2 | R/LS0 | = | R/LS1 = R/LS2 = CM002 + ... | Selects the BA-, BC-, and OF-registers of appropriate subchannels |
| | | CM002 | = | PH2D2 T1D2 NFNT | |
| | Reset CMD, EHE, EH, IXMB, ORD, OUT, TPE, TRA1, and latches TORD and ZBC | R/CMD | = | R/EHE = R/EH = R/IXMB = R/ORD = R/OUT = R/TPE = R/TRA1 = PH2T1D2 | |
| | | TORD | = | TORD NPH2T1D2 + ... | |
| | | ZBC | = | ZBC NPH2T1D2 + ... | |
| | Set flip-flop FP | S/FP | = | PH2T1D2 | |
| | Set latch PH2PD2 | PH2PD2 | = | PH2D2 T1D2 | |
| T2D2 | Clear C-, H-, and I-registers | CX0 | = | HX0 = HUX0 = PH2T2D2 | |
| | | IX0 | = | PH2D2 T2D2 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T2D2 (Cont.) | | R/H0-R/H2 = HUX0 + ...<br>R/H3-R/H7 = HX0 + ...<br>I0-I7 = I0-I7 NIX0 + ... | |
| | Clear J-register | R/J0-R/J2 = PH2T2D2 + ... | |
| | Reset flip-flops SUB and K15 | R/SUB = R/K15 = PH2T2D2 | A one will be subtracted from all data transferred through the adder |
| | Reset flip-flop FS | R/FS = PH2D2 T2D2 F1 + ... | The service connect flip-flop in the device subcontroller is set when it senses the falling edge of FS. The device controller is, therefore, connected for service |
| | Reset flip-flops ED and ES | R/ED = R/ES = PH2D2 T2D2 F1 + ... | |
| T3D2 | Signal HJXOF | HJXOF = PH2PD2 T3D2 F1 + ...<br>S/H0-S/H6 = OF0-OF6 HJXOF + ...<br>S/J0-S/J2 = OF0-OF2 HJXOF + ... | The LSB of the word address BA2, stored in OF0 of the OF-register, is set in J0 of the J-register so that it may control the LSB of the core memory address register S31 during the next phase |
| | Set flip-flop EHE | S/EHE = PH2PD2 T3D2 OF5 + ... | Error halt enable flip-flop EHE is set if the halt on transmission error (HTE) flag stored in the OF-register is true |
| | Set flip-flop EH | S/EH = PH2PD2 T3D2 OF7 + ... | If the transmission error halt (TEH) flag is true, error halt flip-flop EH will be set |
| | Reset PH2D2 | R/PH2D2 = T3D2 + RESET | |
| T4D2 | Set latch PH1D1 | PH1D1 = PH1D1S0 + ...<br>PH1D1S0 = PH2PD2 T4D2 F1 | Next phase in sequence |
| | Set latch PH1D2 | PH1D2 = PH2PD2 T4D2 NF1 NFNT + ...<br>R/F1 = R/FS = AVO + ... | If the requesting device controller does not reply to the ASC, the MIOP receives signal AVO. The operation is, therefore, aborted and the MIOP returns to the beginning phase PH1D2 |
| PH1D1 | Delay line D1 started | PLED1 = PH1D1 RS CM023 + ...<br>CM023 = NMS + MDR1 | The delay line starts when signal RS is received from the device controller |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T0D1 | Clear S-register latches | SX0 = CM011 + ...<br>CM011 = PH1O4D1 T0D1<br>S15-S31 = S15-S31 NSX0 + ... | The S-register is prepared to receive the byte address |
| | Set flip-flop SXBA | S/SXBA = PH1O4D1 T0D1<br>S15-S30 = BA0-BA15 SXBA + ... | The 16 MSB's of the byte address are transferred from the BA-register to S15-S30 of the S-register. The LSB of the word portion of the byte address is controlled by J0 during PH1D1 T1D1 |
| | Signals CXBCU and CXBCL go true | CXBCU = CXBCL = CM004 + ...<br>CM004 = PH1O2D1 T0D1<br>C0-C7 = CXBCU BC0-BC7 + ...<br>C8-C14 = CXBCL BC8-BC14 + ...<br>S/C15 = CXBCU BC15 + ... | The byte count is transferred from the BC-register to the C-register and adder. Since signals SUB and K15 are true, the byte count is decremented by one |
| | Reset flip-flop RSA1 | R/RSA1 = PH1O2D1 T0D1<br>RSA = RSA1 NRSA2 | Prevents sending of RSA to the device controller prematurely |
| | Set flip-flops ED, ES, and TRA1 | S/ED = S/ES = S/TRA1 = CM030 + ...<br>CM030 = CM004 CM029<br>CM004 = PH1O2D1 T0D1<br>CM029 = NJ1 NJ2 H4 + J1 J2 NH4<br>S/H4 = OF4 + ... | Flip-flops ED, ES, and TRA1 are set if a word boundary crossing is detected. Signal ED prevents the MIOP from sending another byte of data to the device controller during the current service cycle (OF4 is true if a read backward order is being executed by the device) |
| | Reset flip-flops MS, MAE, and PRCH | R/MS = MCD1 T0D1 + ...<br>MCD1 = PH1O4D1 + ...<br>R/MAE = PH1D1 T0D1<br>R/PRCH = T0D1 | |
| | Clear M-register latches | MLX0 = MUX0 = CM011 + ...<br>CM011 = PH1O4D1 T0D1<br>M0-M15 = M0-M15 MUX0 + ...<br>M16-M31 = M16-M31 MLX0 + ... | The M-register is prepared to receive the data bytes after they are received from the device controller |
| | Clear W-register latches | WX0 = CM011 + ...<br>W0-W3 = W0-W3 NWX0 | |
| | Reset latch PH1PD1 | PH1PD1 = PH1PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 | Define service cycle type | S/OUT = PH1D1 T1D1 IOR + ...<br>S/ORD = PH1D1 T1D1 DOR + ... | The device controller specifies a data-in service cycle by holding the DOR and IOR lines false |
| | Set flip-flop ED | S/ED = PH1O2D1 T1D1 EDI + ... | When the device controller can transmit no more bytes of data during the current service cycle, it sets flip-flop ED by generating signal EDI, received on the end data line |
| | Signal IXDA goes true | IXDA = PH1O2D1 T1D1 NOUT + ...<br>I0-I7 = IXDA DA0-DA7 + ...<br>IP = IXDA DAP + ... | Signal IXDA gates the data present on the data lines (from the device controller) into the I-register and the parity bit DAP to flip-flop IP |
| | Set flip-flop PRCH | S/PRCH = PH1O2D1 T1D1 PC | The device controller drives the parity check line (PC) to indicate to the MIOP that parity should be checked |
| | Set latch S31 | S/S31 = PH1D1 T1D1 J0 NZBC + ... | The LSB of the word portion of the byte count currently in J0 provides the balance of the byte address in the S-register |
| | Reset flip-flop SXBA | R/SXBA = T1D1D + ... | Since the byte address is in the S-register, transfer signal SXBA is no longer needed |
| | Clear F-register | FX0 = FX0I + ...<br>FX0I = PH1D1 T1D1<br>R/F1 = FX0 + ... | The ASC function indicator line, controlled by F1, goes false |
| | Signal HXJP1 or HXJM1 goes true | HXJP1 = PH1O2D1 T1D1 NH4<br>HXJM1 = PH1O2D1 T1D1 H4 | If the device is executing a read order, signal NH4 is true. Signal HXJP1 thus goes true and increments the byte address by one as it is transferred from the J-register to the H-register. If the device is executing a read backward order, signal H4 is true. Signal HXJM1 thus goes true and decrements the byte address by one as it is transferred from the J-register to the H-register |
| | Reset flip-flop RSA2 | R/RSA2 = PH1O2D1 T1D1 + ... | Primes RSA |
| | Reset flip-flops MAR, MDR1, and MPE | R/MAR = R/MDR1 = MCD1 T1D1 + ...<br>MCD1 = PH1O4D1 + ...<br>R/MPE = PH1O4D1 T1D1 + ... | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 (Cont.) | Set latch PH1PD1 | PH1PD1 = PH1D1 T1D1 + ... | |
| T2D1 | Reset flip-flop FIN | R/FIN = PH1T2D1 + ... | A new I/O operation cannot proceed past the first phase if signal FIN is false |
| | Set latch PH1D2 | PH1D2 = PH1T2D1 + ... | A new I/O operation may proceed through the first phase (PH1D2) |
| PH1O2D1 T2D1 | Signal BCXADD goes true | BCXADD = CM022 + ...<br>CM022 = PH1D1 T2D1 NORD NOUT + ... | The decremented byte count is returned to the BC-register from the adder |
| | Signal OFXH goes true | OFXH = CM022 + ... | The three adjusted LSB's of the byte address and the operating flags are transferred back to the OF-register |
| | One of signals MB0XI-MB3XI go true | MB0XI = CM014 NJ1 NJ2 + ...<br>MB1XI = CM014 NJ1 J2 + ...<br>MB2XI = CM014 J1 NJ2 + ...<br>MB3XI = CM014 J1 J2 + ...<br>CM014 = PH1O2D1 T2D1 NOUT | The two byte address LSB's in the J-register are decoded to produce signals MB0XI-MB3XI |
| | Data byte in I-register is gated into selected byte position of M-register | M0-M7 = MB0XI I0-I7 + ...<br>M8-M15 = MB1XI I0-I7 + ...<br>M16-M23 = MB2XI I0-I7 + ...<br>M24-M31 = MB3XI I0-I7 + ... | The byte of data currently in the I-register is gated into the byte position of the M-register specified by the decoded byte address signals (MB0XI-MB3XI) |
| | Set latches W0-W3 | W0 = MB0XI + ...<br>W1 = MB1XI + ...<br>W2 = MB2XI + ...<br>W3 = MB3XI + ... | Signals MB0XI-MB3XI control the W-register flip-flops to define which bytes of data are to be written into core memory |
| | Set latch ZBC | ZBC = PH1T2D1 ZBS NORD NOUT + ... | Signal ZBS indicates that the next time the byte count is decremented it will equal zero |
| | Set flip-flop ED, set latch TORD, and reset flip-flop ES | S/ED = R/ES = PH1O2D1 T2D1 NORD CM047 + ...<br>TORD = PH2D1 T2D1 CM047 + ...<br>CM047 = ZBS + EH | If an error has been recorded during a previous service cycle (signal EH true) or if the byte count equals zero (signal ZBS true), flip-flop ED will be set to prevent the MIOP from accepting another byte of data during the |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T2D1 (Cont.) | | | current service cycle. TORD will also be set so that the current service cycle will be concluded with a terminal order, and flip-flop ES will be reset to cause the device controller to send another request strobe for the terminal order |
| | Set flip-flop ES | S/ES = PH1O2D1 T2D1 NORD ED NCM047<br>NCM047 = NZBS NEH | If the current byte of data is the last to be accepted from the device controller, if no errors were recorded, and if the byte count does not equal zero, flip-flop ES will be set. The device controller will, therefore, electrically disconnect from the MIOP upon receipt of signal RSA |
| | Set flip-flop RSA1 | S/RSA1 = CM037 NORD NOUT (TRA1 + NED) + ...<br>CM037 = PH1T2D1 NEH NZBS<br>RSA = RSA1 NRSA2 | Signal RSA will be sent to the device controller if error halt and zero byte count conditions do not exist, and if either of the following conditions exist:<br>a. If signal TRA1 is true, indicating a word boundary will be crossed if another byte of data is accepted<br>b. If the device has not specified end data |
| | Set flip-flop MS | S/MS = CM014 CM040 NORD NH3 + ...<br>CM014 = PH1O2D1 NOUT T2D1<br>CM040 = EH + CM047<br>CM047 = ZBS + EH | Signal MS starts a core memory cycle that writes the contents of the M-register into the byte location specified by the write byte lines and the word location specified by the byte address (word location) in the S-register. Flip-flop MS is set if the skip flag in bit 3 of the H-register is false and if one or more of the following conditions exist:<br>a. If end data ED had been specified by either the MIOP or the device controller<br>b. If error halt flip-flop EH had been set<br>c. If the byte count equals zero |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T2D1 (Cont.) | Signal CBA goes true | CBA | = | H0 H4 TRA1 + NH0 NH4 TRA1 | If the device is performing a read forward operation, and all three LSB's of the byte address ($BA^0$ - $BA^2$) were true before they were incremented, signals NH4, TRA1, and NH0 would now be true. Signal CBA would, therefore, go true to provide for the required carry to the BA-register (during PH3D1). If the device is performing a read backward operation, and all three LSB's of the byte address were false before they were decremented, signals H4, TRA1, and H0 would now be true. Signal CBA would, therefore, go true to provide for the borrow from the BA-register |
| T3D1 | Reset flip-flop FP | R/FP | = | PH1PD1 T3D1 NOUT + ... | Signal NFP is required for the delay line to start during PH2D1 |
| | Set flip-flop LS0 | S/LS0 | = | PH1O2PD1 T3D1 ZBC + ... | Selects CA-register |
| | Reset latch PH1D1 | PH1D1 | = | PH1D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Clear C-register latches | CX0 | = | CM013 T4D1 + ... | |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Signal JXH goes true | JXH | = | PH1O2PD1 T4D1 | The three updated byte address LSB's are transferred from the H-register back to the J-register |
| | | S/J0-S/J2 | = | H0-H2 JXH + ... | |
| | Set flip-flop RSA1 | S/RSA1 | = | PH1PD1 T4D1 NOUT + ... | Request strobe acknowledge RSA is sent to the device controller |
| | | RSA | = | RSA1 NRSA2 | |
| | Set flip-flops PRCH and TPE | S/PRCH | = | PH1O2D1 T1D1 PC | Flip-flop PRCH is set if the device controller drives the PC line. Signal IEVEN indicates there is an even number of bits in the I-register and IP flip-flop. Since parity is odd, signal IEVEN indicates |
| | | IEVEN | = | I02EVEN I35EVEN I68EVEN + NI02EVEN NI35EVEN I68EVEN + NI02EVEN I35EVEN NI68EVEN + I02EVEN NI35EVEN NI68EVEN | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 (Cont.) | | I02EVEN = NI0 NI1 NI2 + I0 I1 NI2 + I0 NI1 I2 + NI0 I1 I2<br>I35EVEN = NI3 NI4 NI5 + I3 I4 NI5 + I3 NI4 I5 + NI3 I4 I5<br>I68EVEN = NI6 NI7 NIP + I6 I7 NIP + I6 NI7 IP + NI6 I7 IP<br>S/TPE = CM038 PRCH IEVEN + ...<br>CM038 = PH1O2PD1 T4D1 NORD NOUT | that a parity error occurred, in which case flip-flop TPE is set |
| | Set flip-flop EH | S/EH = EHE TPE + ... | Error halt flip-flop EH is set if signal EHE is true (flip-flop EHE is set by the HTE flag) and signal TPE is also true |
| | Set flip-flop LS1 | S/LS1 = PH1PD1 T4D1 EH NOUT + ... | Selects FS-register |
| | Set flip-flop LS2 | S/LS2 = PH1O2PD1 T4D1 TORD + ... | Selects IS-register |
| | Set flip-flops MAE and EH | S/MAE = PH1PD1 CM048 + ...<br>CM048 = T4D1 MS NAH<br>S/EH = MAE + ... | If address here AH is not received from core memory by T4D1, memory address error flip-flop MAE is set. Signal MAE sets error halt flip-flop EH |
| | Set latch PH2D1 | PH2D1 = PH2D1S0 T4D1 + ...<br>PH2D1S0 = PH1O2PD1 NORD NED | If the current byte of data is not the last to be accepted from the device controller (signal NED true), the MIOP will accept additional bytes during PH2D1 |
| | Set latch PH3D1 | PH3D1 = PH3D1S0 T4D1 + ...<br>PH3D1S0 = PH1O2PD1 NORD NFP NZBC CBA | The MIOP will advance to PH3D1 if the 16 MSB's of the byte address require updating (carry to, or borrow from) |
| | Set latch PH4D1 | PH4D1 = PH4D1S0 T4D1 + ...<br>PH4D1S0 = PH1O2PD1 ZBC H6 | If the byte count equals zero (signal ZBC true) and the data chaining flag is true (signal H6 true), the MIOP will advance to PH4D1 to start the chaining operation |
| | Set latch PH7D1 | PH7D1 = PH7D1S0 T4D1 + ...<br>PH7D1 = PH1O2PD1 CM008 + ...<br>CM008 = ZBC NH6 + NFP NZBC NCBA ED + ... | The MIOP will advance directly to the termination phase (PH7D1) if the byte count equals zero (ZBC) and the data chaining flag (H6) is false, or signal FP is false, |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T4D1 (Cont.) | | | the byte count does not equal zero (NZBC), the byte address MSB's do not need updating (NCBA), and if the current byte of data is the last to be accepted during the current service cycle (ED) |
| PH2D1 | Delay line D1 started | PLED1 = PH2D1 NFP RS RSA2 | The delay line starts when request strobe RS is received from the device controller |
| T0D1 | Signals CXBCL and CXBCU go true | CXBCU = CXBCL = CM004 + ...<br>CM004 = PH1O2PD1 T0D1<br>C0-C7 = BC0-BC7 CXBCU + ...<br>C8-C14 = BC8-BC14 CXBCL + ...<br>S/C15 = BC15 CXBCU | The byte count is transferred from the BC-register to the C-register and adder. Since signals SUB and K15 are both true, the byte count is decremented by one |
| | Set flip-flop PRCH | S/PRCH = PH1O2PD1 T1D1 PC | The device controller drives the PC line if parity is to be checked |
| | Clear I-register and IP flip-flop | IX0 = PH2T0D1 NOUT + ...<br>I0-I7 = I0-I7 NIX0 + ...<br>IP = IP NIX0 + ... | |
| | Set flip-flops ED, ES, and TRA1 | S/ED = S/ES = S/TRA1 = CM030 + ...<br>CM030 = CM004 CM029<br>CM004 = PH1O2D1 T0D1<br>CM029 = NJ1 NJ2 H4 + J1 J2 NH4<br>S/H4 = OF4 + ... | The two LSB's of the byte address are checked for a word boundary crossing. If a word boundary crossing is detected, flip-flops ED, ES, and TRA1 are set |
| | Reset flip-flop RSA1 | R/RSA1 = PH1O2PD1 T0D1<br>RSA = RSA1 NRSA2 | Prevents sending of RSA to the device controller prematurely |
| | Reset latch PH2PD1 | PH2PD1 = PH2PD1 NPHPD1X0<br>NPHPD1X0 = NT0D1 NRESET | |
| T1D1 | Signal IXDA goes true | IXDA = PH1O2D1 T1D1 NOUT + ...<br>I0-I7 = IXDA DA0-DA7 + ...<br>IP = IXDA DAP + ... | Signal IXDA gates the data present on the data lines (from the device controller) into the I-register, and the parity bit DAP to flip-flop IP |
| | Set flip-flop PRCH | S/PRCH = PH1O2D1 T1D1 PC | The device controller drives the parity check line (PC) to indicate to the MIOP that parity should be checked |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T1D1 (Cont.) | Set flip-flop ED | S/ED | = | PH1O2D1 T1D1 EDI + ... | When the device controller can transmit no more bytes of data during the current service cycle, it sets flip-flop ED by generating signal EDI, received by the MIOP on the end data line |
| | Reset flip-flop RSA2 | R/RSA2 | = | PH1O2D1 T1D1 + ... | Primes RSA |
| | Signal HXJP1 or HXJM1 goes true | HXJP1<br>HJXM1 | =<br>= | PH1O2D1 T1D1 NH4<br>PH1O2D1 T1D1 H4 | If the device is executing a read order, signal NH4 is true. Thus, signal HXJP1 goes true and increments the byte address by one as it is transferred from the J-register to the H-register. If the device is executing a read backward order, signal H4 is true. Thus, signal HXJM1 goes true and decrements the byte address by one as it is transferred from the J-register to the H-register |
| | Set latch PH1PD1 | PH1PD1 | = | PH1D1 T1D1 + ... | |
| | For the next function, return to PH1O2D1, T2D1 of this table, and proceed through T2D1, T3D1, and T4D1. During T4D1, the MIOP will either cycle through PH2D1 for another byte of data or will progress to another phase as required | | | | |
| PH3D1 | Delay line D1 started | PLED1 | = | PH3D1 + ... | The delay line starts unconditionally at the end of T4D1 of the preceding phase |
| T0D1 | Reset flip-flop SUB | R/SUB | = | PH3D1 T0D1 NH4 + ... | Flip-flop SUB is reset only if the device is executing a read order (NH4). If flip-flop SUB is reset all data transferring through the adder will be incremented by one, since K15 is true. If SUB remains true, all data transferring through the adder will be decremented by one |
| | Signal CXBA goes true | CXBA | = | PH3D1 T0D1 + ... | The byte address MSB's are transferred to the C-register and adder, and are thus incremented or decremented by |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T0D1 (Cont.) | | | | | one, depending upon the state of signal SUB. This operation effectively provides the carry or borrow to or from the three LSB's of the byte address |
| | Reset latch PH3PD1 | PH3PD1 | = | PH3PD1 NPHPD1X0 | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Set latch PH3PD1 | PH3PD1 | = | PH3D1 T1D1 + ... | |
| T2D1 | Signal BAXADD goes true | BAXADD | = | PH3D1 T2D1 + ... | The incremented or decremented byte address is transferred back to the BA-register from the C-register and adder |
| T3D1 | Reset latch PH3D1 | PH3D1 | = | PH3D1 NPHD1X0 | |
| | | NPHD1X0 | = | NT3D1 NRESET | |
| T4D1 | Clear C-register | CX0 | = | CM013 T4D1 + ... | |
| | | CM013 | = | NPH7PD1 NPH10PD1 NPH11PD1 | |
| | | C0-C14 | = | C0-C14 NCX0 + ... | |
| | | R/C15 | = | CX0 + ... | |
| | Set latch PH7D1 | PH7D1 | = | PH7D1S0 T4D1 + ... | Next phase in sequence from PH3D1 |
| | | PH7D1S0 | = | PH3PD1 NFP | |
| | The operations performed during phases PH4D1, PH5D1, and PH6D1 (the sequences performed when data chaining is specified) of a data-in service cycle are identical to phases PH4D1, PH5D1, and PH6D1 of a data-out service cycle (see table 3-17) | | | | |
| PH7D1 | Delay line D1 started | PLED1 | = | PH7D1 CM025 + ... | If a terminal order is to be sent to the device controller, the delay line starts when signal RS is received from the device controller and signals MDR1 and NMS are true. If no terminal order is to be sent, the delay line starts when T4D1 of the previous phase is completed and signals MDR1 or NMS are true |
| | | CM025 | = | CM023 CM025I1 | |
| | | CM023 | = | MDR1 + ... | |
| | | NCM025I1 | = | NCM025I1I + ... | |
| | | NCM025I1I | = | RS RSA2 | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| PH7D1 (Cont.) | Delay line D1 started | PLED1 | = | PH7D1 CM025 + ... | If a terminal order is to be sent to the device controller, the delay line starts when signal RS is received from the device controller and MDR1 or NMS are true. If no terminal order is to be sent, the delay line starts when T4D1 of the previous phase is completed and MDR1 or NMS are true |
| | | CM025 | = | CM023 CM025I1 | |
| | | CM023 | = | MDR1 + ... | |
| | | NCM025I1 | = | NCM025I1I + ... | |
| | | NCM025I1I | = | RS RSA2 | |
| T0D1 | Clear H-register flip-flops | HUX0 | = | HX0 = HX0I + ... | |
| | | HX0I | = | PH7D1 T0D1 + ... | |
| | | R/H0-R/H2 | = | HUX0 + ... | |
| | | R/H3-R/H7 | = | HX0 + ... | |
| | Reset flip-flop K15 | R/K15 | = | PH7D1 T0D1 + ... | Data transferred through the adder will not be altered |
| | Clear O-register latches | OX0 | = | PH7D1 T0D1 TORD + ... | If a terminal order is specified (signal TORD true), the O-register is cleared |
| | | O0-O7 | = | O0-O7 NOX0 + ... | |
| | Reset flip-flop RSA1 | R/RSA1 | = | PH7D1 T0D1 TORD + ... | Flip-flop RSA1 is set again during T3D1 to generate RSA. Signal RSA is generated only if a terminal order is specified |
| | Reset latch PH7PD1 | PH7PD1 | = | PH7PD1 NPHPD1X0 + ... | |
| | | NPHPD1X0 | = | NT0D1 NRESET | |
| T1D1 | Signal CXBCL goes true | CXBCL | = | PH7T1D1 + ... | Status information from bits 8-14 of the FS-register is transferred to bits 8-14 of the C-register |
| | | C8-C14 | = | BC8-BC14 CXBCL + ... | |
| | Signal CXBCU goes true | CXBCU | = | CXBCUI + ... | The flags stored in the FS-register are transferred to the C-register if an error was detected (signal EH true), or if data chaining did not occur (signal NCMD true) |
| | | CXBCUI | = | PH7D1 T1D1 CM045 | |
| | | CM045 | = | EH + NCMD | |
| | | C0-C7 | = | BC0-BC7 CXBCU + ... | |
| | Signal CXMB0 goes true | CXMB0 | = | PH7D1 T1D1 NCM045 | If data chaining occurred (signal CMD true) and no error conditions were detected (signal NEH true), the new flags in the M-register are transferred to the C-register for subsequent storage in the FS-register |
| | | NCM045 | = | CMD NEH | |
| | | C0-C7 | = | M0-M7 CXMB0 + ... | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | | | Comments |
|---|---|---|---|---|---|
| T1D1 (Cont.) | Signal CXST goes true | CXST | = | PH7T1D1 | Signal CXST updates the status information currently in the C-register |
| | | C10 | = | CXST C10ST | |
| | | C10ST | = | MPE NCMD + FP NORD | |
| | | C11 | = | CXST MAE + ... | |
| | | C12 | = | CXST C13ST + ... | |
| | | C12ST | = | CMD MPE | |
| | | C13 | = | CXST C13ST + ... | |
| | | C13ST | = | CMD EH NMPE NMAE | |
| | | C14 | = | CXST EH + ... | |
| | Signal OXTORD goes true | OXTORD | = | PH7T1D1 TORD | Signal OXTORD gates a terminal order to the O-register and, therefore, to the device controller if a terminal order was specified |
| | | O0 | = | OXTORD O0TOS + ... | |
| | | O0TOS | = | CM016 + ... | |
| | | CM016 | = | ZBC BC1 | |
| | | O1 | = | OXTORD O1TOS + ... | |
| | | O1TOS | = | ZBC NBC0 | |
| | | O2 | = | OXTORD BC2 + ... | |
| | | O3 | = | OXTORD EH + ... | |
| | | O4 | = | OXTORD O4TOS + ... | |
| | | O4TOS | = | OUT EH NCMD + ... | |
| | Signal HJXOF goes true | HJXOF | = | PH7T1D1 + ... | If a terminal order is specified, the interrupt status and the three MSB's of the device address (from IS4-IS6) are set into the H- and J-registers. If a terminal order is not specified, the operating flags (from OF3-OF6) and the three byte address LSB's are set into the H- and J-registers. If a terminal order is specified, the interrupt status is updated in the H-register and returned to the IS-register. If a terminal order is not specified, the information is returned to the OF-register unaltered |
| | | S/H0-S/H6 | = | HJXOF OF0-OF6 + ... | |
| | | S/J0-S/J2 | = | HJXOF OF0-OF2 + ... | |
| | Set flip-flop H7 | S/H7 | = | PH7T1D1 CM005 + ... | If a terminal order is specified, the LSB of the device address (from IS7) is set into the H-register to complete the device address transfer. If a terminal order is not specified |
| | | CM005 | = | TORD OF7 + NTORD EH | |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T1D1 (Cont.) | | | and an error halt condition occurred (signal EH true), bit H7 will be set, thus retaining a record of the error when the H-register contents are transferred back to the OF-register |
| | Signal OXTORD goes true | OXTORD = TORD PH7T1D1<br>S/H0 = OXTORD CM016 + ...<br>CM016 = ZBC BC1 | If a terminal order is specified, signal OXTORD permits the interrupt status in the H-register to be updated. Bit H0 is set if the byte count equals zero and the interrupt at zero byte count flag (BC1) is true |
| | Set flip-flop ES | S/ES = PH7T1D1 | Signal ES causes the device controller to disconnect from the device controller interface lines after it also receives signal RSA |
| | Set flip-flop FIN | S/FIN = PH7T1D1 CM032 + ...<br>CM032 = NMPE NMAE NTPE NTORD | Flip-flop FIN is set if a terminal order is not specified, no transmission errors, and no memory address or parity errors were detected during the current service cycle. Signal FIN permits the next I/O operation to continue. Signal NFIN gates the information in the adder and H-register back into their respective fast access registers during T3D1 of this phase |
| | Reset flip-flop RSA2 | R/RSA2 = PH7T1D1 TORD + ... | Primes RSA |
| | Set latch PH7PD1 | PH7PD1 = PH7D1 T1D1 + ... | |
| T3D1 | Signal BCXADD goes true | BCXADD = CM021 + ...<br>CM021 = PH7PD1 T3D1 NFIN | The flags and status currently in the C-register are set into the FS-register |
| | Signal OFXH goes true | OFXH = CM021 + ... | If a terminal order is specified, the interrupt status and device address in the H-register are transferred to the IS-register. If a terminal order is not specified, the operating flags and byte address LSB's in the H-register are transferred to the OF-register |

(Continued)

Table 3-18. Data-In Service Cycle Phase Sequence (Cont.)

| Phase | Function Performed | Signals Involved | Comments |
|---|---|---|---|
| T3D1 (Cont.) | Set flip-flop RSA1 | S/RSA1 = PH7PD1 T3D1<br>RSA = RSA1 NRSA2 | The service connect flip-flop in the device controller is reset when it receives signal RSA. The device controller is, therefore, electrically disconnected from the interface |
| | Reset latch PH7D1 | PH7D1 = PH7D1 NPHD1X0 + ...<br>NPHD1X0 = NT3D1 NRESET | |
| T4D1 | Set flip-flop FIN | S/FIN = PH7PD1 T4D1 + ... | The current operation no longer needs the MIOP's fast access memory |

## 3-44 GLOSSARY

A glossary of MIOP signals appears in table 3-19.

Table 3-19. Glossary of MIOP Signals

| Signal | Definition |
|---|---|
| A0-A7 | An 8-bit register. The A-register contains the current fast access memory address |
| ADD0-ADD15 | The 16 outputs of the adder. ADD is equal to either C + K15 (if NSUB) or C - K15 (if SUB) |
| AH | The address here line from core memory |
| AIO | Acknowledge I/O to peripherals |
| AIOFN | Logic signal. Indicates that an AIO function is being signaled by the CPU |
| AR | The address release line from core memory |
| ASC | Acknowledge service call to device controllers |
| AVO | Device controller available from priority cable |
| AX0 | Logic signal. Clears the A-register |
| AXFR | Logic signal. Transfers FR to the A-register |
| AXM | Logic signal. Transfers M0-M7 to the A-register |
| BA | Sixteen bits of fast access memory. Contains the most significant 16 bits of the byte address |
| BAXADD | Logic signal. Transfers ADD to BA/CA |
| BC | Sixteen bits of fast access memory. Contains the byte count |
| BCXADD | Logic signal. Transfers ADD to BC/FS |
| BKWD | Logic signal. Indicates that a read backward order is being processed |
| C | A 16-bit register. C is the input to the adder |
| CA | Sixteen bits of fast access memory. Contains the command address. (Since CA and BA have the same source, as shown by the LS0 definition, CA will not appear in the equations) |

(Continued)

Table 3-19. Glossary of MIOP Signals (Cont.)

| Signal | Definition |
|---|---|
| CBA | Logic signal. Indicates that a carry or borrow into the most significant 16 bits of the byte address (BA) has been generated |
| CC1 | Logic flip-flop. NCC1 drives the NCOND1 response to the CPU |
| CC2 | Logic flip-flop. NCC2 drives the NCOND2 response to the CPU |
| CL1 | The 1-megacycle clock line to peripherals |
| CL1S | The 1-megacycle clock source line from the CPU |
| CM001-CM050 | Fifty miscellaneous common terms |
| CMD | Logic flip-flop. Disables the chaining modifier bit (FS15). During data chaining, also indicates that the data transfer has been successfully concluded (i.e., without an error halt – see EH) and the chaining is at least started. During most CPU-initiated functions (HIO, SIO, TDV, TIO), CMD also gates the storing of the first status word in core memory |
| CNST | The control strobe line from the CPU |
| CNSTNME | Logic signal. Causes the incoming strobe to be passed on to the next MIOP (because the function is not to be executed by this MIOP) |
| COND1 | Line to the CPU. Causes condition code 1 (within the CPU) to set |
| COND2 | Line to the CPU. Causes condition code 2 (within the CPU) to set |
| CX0 | Logic signal. Clears the C-register |
| CXBA | Logic signal. Transfers BA to the C-register |
| CXBCL | Logic signal. Transfers BC8-BC14 to C8-C14 |
| CXBCU | Logic signal. Transfers BC0-BC7 to C0-C7 and BC15 to C15 |
| CXM | Logic signal. Transfers M16-M31 to C |
| CXMB0 | Logic signal. Transfers M0-M7 to C0-C7 |
| CXMR3 | Logic signal. Transfers M13-M28 to the C-register |
| CXST | Logic signal. Transfers status update information to C8-C14 |
| DA0-DA7 | The eight input-output data lines from and to peripherals |
| DAP | The data parity lines associated with DA0-DA7 |
| DG | Data gate signal from core memory. Implies data from memory is stable |
| DOR | The data order request line from peripherals |
| DR | The data release line from core memory |
| ED | Logic flip-flop. Receives an end data signal (EDI) from a peripheral. Also drives the end data line to peripherals |
| EDI | The end data signal from a peripheral |
| EH | Logic flip-flop. Gates an error halt of the peripheral currently being serviced. (An error halt causes the current operation of the MIOP to be terminated with unusual end being reported to the peripheral) |
| EHE | Logic flip-flop. Enables EH to be set under certain conditions. (EHE is actually the halt on error flag) |
| ES | Logic flip-flop. Drives the end service line to peripherals |
| F | A 6-bit register. Drives the function lines to peripherals. (F0 drives AIO; F1 drives ASC; F2 drives HIO; F3 drives SIO; F4 drives TIO; F5 drives TDV) |

(Continued)

Table 3-19. Glossary of MIOP Signals (Cont.)

| Signal | Definition |
|---|---|
| FIN | Logic flip-flop. Signals the (independently running) priority determination logic (delay line 2) that the current operation is nearing completion. Specifically, FIN implies that the current operation no longer needs fast access memory |
| FNT | Logic flip-flop. Gates the execution of a CPU-initiated function (AIO, HIO, SIO, TDV, TIO) |
| FNC | The three function code lines from the CPU |
| FP | Logic flip-flop. Indicates (during phase 2) that the next output data byte is the first to be transmitted during the current service cycle. May also gate (during phase 3) the termination of a service cycle. Finally, FP indicates (during phase 7) a memory parity error during the data portion of a data in/out operation with chaining |
| FR0-FR7 | The eight function response lines from peripherals |
| FS | Logic flip-flop. Drives the function strobe line to peripherals |
| FS0-FS15 | Sixteen bits of fast access memory. FS contains the flags and status. (Since FS and BC have the same source see LS1, FS will not appear in the equations, eliminating confusion with the function strobe flip-flop above) |
| FSL | The leading function strobe acknowledge line from peripherals |
| FX0 | Logic signal. Clears the F-register |
| FXFN | Logic signal. Transfers the decoded FNC lines to the F-register |
| H0-H7 | An 8-bit register. H is used to operate on OF/IS |
| HIO | Halt I/O to peripherals |
| HJXOF | Logic signal. Transfers OF0-OF6 to H0-H6 and transfers OF0-OF2 to the J-register |
| HUX0 | Logic signal. Clears H0-H2 |
| HUXM | Logic signal. Transfers M29-M31 to H0-H2 |
| HX0 | Logic signal. Clears H3-H7 |
| HXJM1 | Logic signal. Transfers J minus 1 to H0-H2 |
| HXJP1 | Logic signal. Transfers J plus 1 to H0-H2 |
| I0-I7 | An 8-bit register. I receives input data bytes (DA) from peripherals and holds output data bytes (before transferring them to the O-register) for parity generation |
| I02EVEN | Logic signal. Indicates that there are an even number of bits in I0-I2 |
| I35EVEN | Logic signal. Indicates that there are an even number of bits in I3-I5 |
| I68EVEN | Logic signal. Indicates that there are an even number of bits in I6, I7, and IP |
| IC | The interrupt call line from peripherals |
| IEVEN | Logic signal. Indicates that there are an even number of bits in I and IP |
| IMB0-IMB7 | Eight logic signals. IMB0-IMB7 equals the byte of M (i.e., M0-M7, M8-M15, M16-M23, or M24-M31) being transferred to the I-register |
| IOPA | The three MIOP address lines from the CPU |
| IOR | The input-output request line from peripherals |
| IP | Logic flip-flop. Receives the data parity bit (DAP) |
| IR | Logic flip-flop. Drives the interrupt request line to the CPU |

(Continued)

Table 3-19. Glossary of MIOP Signals (Cont.)

| Signal | Definition |
|---|---|
| IS0-IS7 | Eight bits of fast access memory. IS contains the interrupt status and the number of the last successfully started (via SIO) device on the device controller. (Since IS and OF have the same source see[LS2], IS will not appear in the equations) |
| IX0 | Logic signal. Clears I and IP |
| IXDA | Logic signal. Transfers DA to I and transfers DAP to IP |
| IXMB | Logic flip-flop. Transfers IMB to I |
| IXMB0 | Logic signal. Gates M0-M7 to IMB |
| IXMB1 | Logic signal. Gates M8-M15 to IMB |
| IXMB2 | Logic signal. Gates M16-M23 to IMB |
| IXMB3 | Logic signal. Gates M24-M31 to IMB |
| J0-J2 | Three-bit register. J is used to increase or decrease H0-H2 (see HXJM1 and HXJP1) |
| JXH | Logic signal. Transfers H0-H2 to the J-register |
| K0-K15 | Fifteen logic signals (K0-K14) and one logic flip-flop (K15). K indicates that there is a carry into a given stage of the adder |
| KP0T2 | Logic signal. Indicates that stages 0-2 of the adder will propagate a carry |
| KP3T5 | Logic signal. Indicates that stages 3-5 of the adder will propagate a carry |
| KP6T8 | Logic signal. Indicates that stages 6-8 of the adder will propagate a carry |
| KP9T11 | Logic signal. Indicates that stages 9-11 of the adder will propagate a carry |
| KP12T14 | Logic signal. Indicates that stages 12-14 of the adder will propagate a carry |
| KP15 | Logic signal. Indicates that stage 15 of the adder will propagate a carry |
| LASTONE | Logic signal. Indicates that this MIOP is (physically) the last one in the MIOP priority string |
| LS0 | Logic flip-flop. Controls whether BA (NLS0) or CA (LS0) is being accessed in fast access memory |
| LS1 | Logic flip-flop. Controls whether BC (NLS1) or FS (LS1) is being accessed in fast access memory |
| LS2 | Logic flip-flop. Controls whether OF (NLS2) or IS (LS2) is being accessed in fast access memory |
| L15-L31 | Address lines to core memory |
| M0-M31 | A 32-bit register. M receives the data (MD) from core memory and also drives the data lines to core memory |
| MAE | Logic flip-flop. Indicates that a memory address error has occurred (i.e., that nonexistent core memory has been addressed – see AH) |
| MAR | Logic flip-flop. Receives the address release (AR) signal from core memory |
| MB0XI | Logic signal. Transfers the I-register to M0-M7 |
| MB1XI | Logic signal. Transfers the I-register to M8-M15 |
| MB2XI | Logic signal. Transfers the I-register to M16-M23 |
| MB3XI | Logic signal. Transfers the I-register to M24-M31 |
| MCD1 | Logic signal. Causes the memory flip-flops (MAR, MDR1, and MS) to clear during the appropriate phases and times of delay line 1 |

(Continued)

Table 3-19. Glossary of MIOP Signals (Cont.)

| Signal | Definition |
|---|---|
| MD0-MD31 | The 32 memory data signals from core memory |
| MDR1 | Logic flip-flop. During a full write memory cycle, indicates that data release (DR) has risen. During a read cycle, indicates that parity OK (POK) or parity error (PE) have risen. MDR1 also indicates that the memory address error (MAE) flip-flop is set |
| ME | Logic signal. Indicates that the function currently being performed by the CPU is to be executed by this MIOP |
| MLX0 | Logic signal. Clears M16-M31 |
| MPE | Logic flip-flop. Receives the parity error (PE) signal from core memory |
| MQ | The memory request line to core memory |
| MS | Logic flip-flop. Gates the MIOP's entire execution of a core memory cycle (from MQ to the setting of MDR1) |
| MUX0 | Logic signal. Clears M0-M15 |
| MW0-MW3 | Write byte lines to core memory |
| MXA | Logic signal. Transfers IOPA to M21-M23 and gransfers the A-register to M24-M31 |
| MXADD | Logic signal. Transfers ADD to M0-M15 |
| MXBC | Logic signal. Transfers BC to M16-M31 |
| MXFR | Logic signal. Transfers FR to M0-M7 |
| MXIS | Logic signal. Transfers BC8-BC9 to M8-M9 and transfers OF0-OF2 to M10-M12 |
| MXM | Logic signal. Transfers MD to the M-register |
| MYNUM | Logic signal. Indicates that the CPU is currently addressing this MIOP (i.e., IOPA equals NUM) |
| NUM0-NUM2 | The three bits of the MIOP number. Derived from three toggle switches |
| O0-O7 | An 8-bit register. O drives the data lines (DA) |
| OF0-OF7 | Eight bits of fast access memory. OF contains the least significant three bits of the byte address. OF also contains a duplication of some of the flags (see FS) |
| OFXH | Logic signal. Transfers H to OF/IS |
| OP | Logic flip-flop. Drives the data parity line (DAP) |
| ORD | Logic flip-flop. Receives the data/order request (DOR) signal. Also used to gate the conclusion of an AIO function |
| ORDIN | Logic signal. Indicates that an order-in operation is currently being performed |
| ORDOUT | Logic signal. Indicates that an order-out operation is currently being performed |
| OUT | Logic flip-flop. Receives the input-output request (IOR) signal. Also used to gate the conclusion of an SIO function |
| OX0 | Logic signal. Clears O and OP |
| OXI | Logic flip-flop. Transfers the I-register to O and transfers IP to OP |
| OXTORD | Logic signal. Gates a terminal order to O |
| PC | The parity check line from peripherals |
| PE | The parity error line from core memory |
| PER | The parity error release line from core memory |

(Continued)

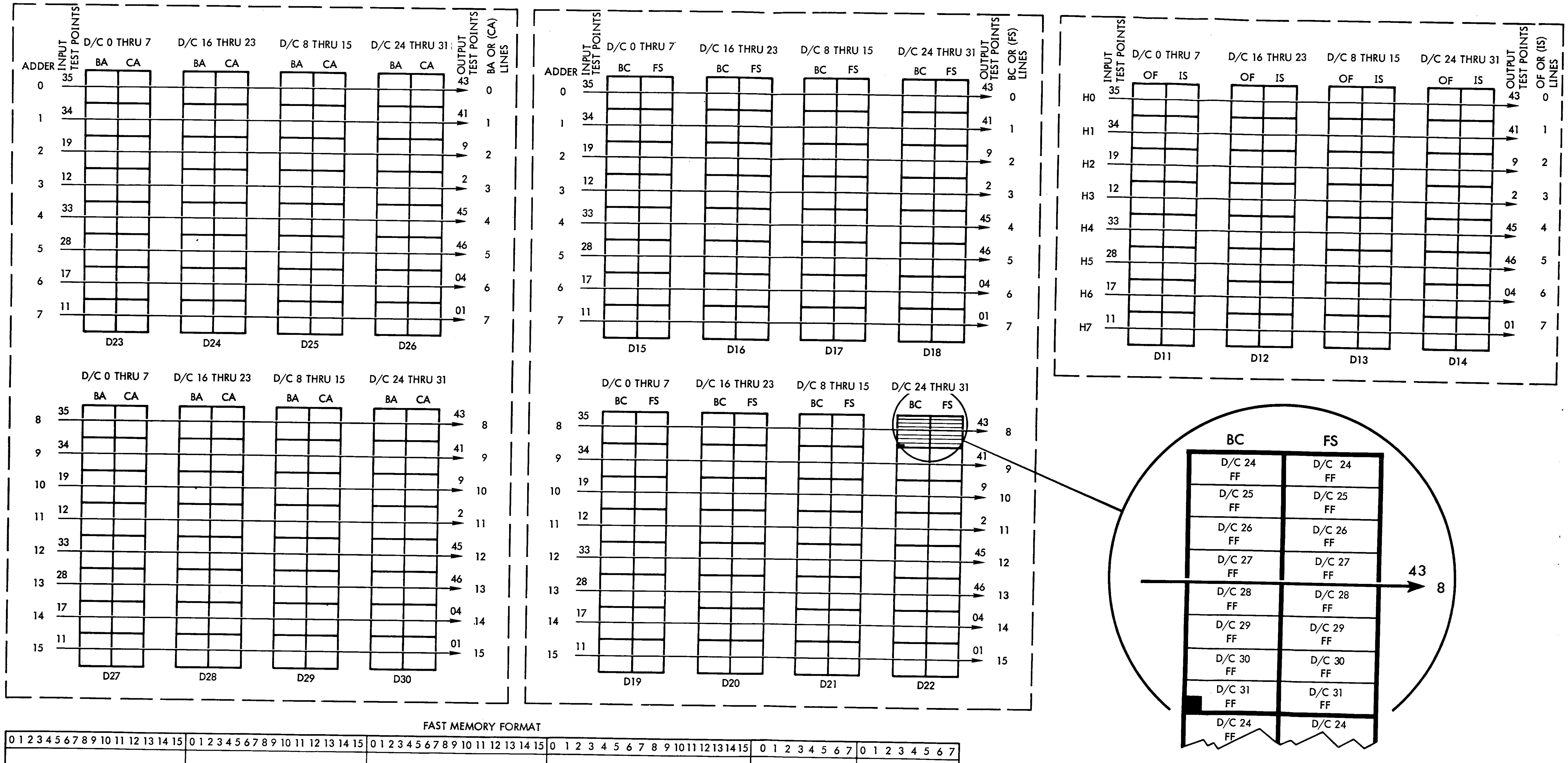
Table 3-19. Glossary of MIOP Signals (Cont.)

| Signal | Definition |
|---|---|
| PH1D1-PH14D1 | The 12 octally numbered phases associated with delay line 1. PH1D1-PH14D1 are only used during time T0D1, T1D1, and T2D1, and when the delay line is inactive |
| PH1PD1-PH14PD1 | The 12 octally numbered phases associated with delay line 1. PH1PD1-PH14PD1 are only used during times T3D1 and T4D1 |
| PH1D2, PH2D2 | The two phases associated with delay line 2. PH1D2 and PH2D2 are only used during times T0D2, T1D2, and T2D2, and when the delay line is inactive |
| PH1PD2, PH2PD2 | The two phases associated with delay line 2. PH1PD2 and PH2PD2 are only used during times T3D2 and T4D2 |
| PHD1X0 | Logic signal. Clears PH1D1-PH14D1 |
| PHPD1X0 | Logic signal. Clears PH1PD1-PH14PD1 |
| PLD1 | Logic flip-flop. Pulses delay line 1 (causing one sequence of timing pulses T0D1 through T4D1) |
| PLD2 | Logic flip-flop. Pulses delay line 2 (causing one sequence of timing pulses T0D2 through T4D2) |
| PLED1 | Logic signal. Enables PLD1 to set |
| PLED2 | Logic signal. Enables PLD2 to set |
| POK | A memory-generated signal indicating that there is no parity error (parity OK) |
| PR | The proceed line to the CPU |
| PR1 | Logic flip-flop. Gates the proceed (PR) signal to the CPU |
| PR2 | Logic flip-flop. NPR2 gates the proceed (PR) signal to the CPU. PR2 is set when the CPU responds to the PR signal by dropping the control strobe (CNST) signal |
| PRCH | Logic flip-flop. Receives the parity check (PC) signal |
| PWD1 | Prevents starting delay line 1 until the delay line is clear |
| PWD2 | Prevents starting delay line 2 until the delay line is clear |
| READ | Logic signal. Indicates that the current core memory cycle is a read cycle |
| RESET | Logic signal. Causes a master reset of the MIOP |
| RIO | The reset I/O line from the CPU |
| RS | The request strobe line from peripherals |
| RSA | The request strobe acknowledge line to peripherals |
| RSA1 | Logic flip-flop. Gates the request strobe acknowledge (RSA) signal to peripherals |
| RSA2 | Logic flip-flop. NRSA2 gates the request strobe acknowledge (RSA) signal to peripherals. RSA2 is set when the peripheral responds to the RSA signal by dropping the request strobe (RS) signal |
| RST | Reset line to peripherals |
| S15-S31 | A 17-bit register. S drives the address lines to core memory |
| SC | The service call line from peripherals |
| I/SENSED1 | Start pulse to delay line 1 |
| I/SENSED2 | Start pulse to delay line 2 |
| SENSEnD1 | Output taps from delay line 1 |
| SENSEDnD2 | Output taps from delay line 2 |
| SIO | Start I/O to peripherals |

(Continued)

Table 3-19. Glossary of MIOP Signals (Cont.)

| Signal | Definition |
|---|---|
| SPA3-SPA7 | The address lines to fast access memory |
| SUB | Logic flip-flop. If SUB is set, ADD = C -K15. If SUB is reset, ADD = C + K15 |
| SX0 | Logic signal. Clears the S-register |
| SX20 | Logic signal. Forces S26 to set. This, in conjunction with SX0, forces a $00020_{16}$ address into the S-register |
| SXBA | Logic flip-flop. Transfers BA to S |
| T0D1-T4D1 | The five timing signals generated by delay line 1 |
| T1D1D | Logic signal. T1D1D is T1D1 delayed |
| T0D2-T4D2 | The five timing signals generated by delay line 2 |
| TDV | Test device to peripherals |
| TIO | Test I/O to peripherals |
| TORD | Logic flip-flop. Indicates that a terminal order is to be sent to the peripheral currently being serviced |
| TPE | Logic flip-flop. Indicates that a transmission error has occurred. During most CPU-initiated functions (HIO, SIO, TDV, TIO), TPE also gates the storing of the second status word in core memory |
| TRA | Logic signal. Indicates that a transfer in channel order is being processed |
| TRA1 | Logic flip-flop. Used to count the number of successive transfer in channel orders (two successive transfer in channel orders will cause an error halt – see EH) and to indicate that a word boundary has been crossed during a data-in or data-out operation. TRA1 is also used during an AIO function to indicate that the device controller interrupt status (IS) within the MIOP is associated with the responding device |
| TRR | MIOP tester (JX58) reset |
| UN0-UN2 | MIOP address |
| W | Four-bit register. Drives the write byte indicator lines to core memory |
| WX0 | Logic signal. Clears the W-register |
| WX1 | Logic signal. Sets the W-register |
| ZBC | Logic flip-flop. Indicates that the byte count has gone to zero during a data-in or data-out operation |
| ZBS | Logic signal. Indicates that ADD = 0 (when SUB = 1 and K15 = 1) |




FAST MEMORY FORMAT

| BA | CA | BC | FS | OF | IS |
|---|---|---|---|---|---|
| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 | 0 1 2 3 4 5 6 7 | 0 1 2 3 4 5 6 7 |
| BYTE ADDRESS | COMMAND ADDRESS | BYTE COUNT | DC IZC CC ICE HTE IUE SIL S IL TE TMP MAE CMP CE IOP UE CM | BA2 2 BA2 1 BA2 0 S BK HTE DC TEH | ZCI CEI UEI SPARE D#23 D#22 D#21 D#20 |

LEGEND

SELECT

| | |
|---|---|
| BA = NLS0 | CA = LS0 |
| BC = NLS1 | FS = LS1 |
| OF = NLS2 | IS = LS2 |

Figure 3-23. MIOP Fast Access Memory Organization

901515A. 320

# SECTION IV
# MAINTENANCE AND PARTS LIST

## 4-1 GENERAL

This section includes preventive maintenance procedures and a parts list for the Sigma 5 and Sigma 7 MIOP, assembly No. 117610. The applicable systems test monitor, peripheral equipment systems test, and MIOP test programs are listed in paragraph 4-5.

## 4-2 PREVENTIVE MAINTENANCE

All the documents listed on the Assembly of Maintenance Documents should be available at the site. They should be complete and should accurately reflect the change level of the equipment. Field change record stickers should be applied to the equipment according to Tek-Tip 65-50-32 and should reflect the change level of the equipment.

## 4-3 EXTERNAL VISUAL INSPECTION

External surfaces of the equipment must be kept clean and dust-free. Doors and panels must close completely and be in reasonable alignment. The tops of cabinets must remain cleared to allow free intake and exhaust of air.

## 4-4 INTERNAL VISUAL INSPECTION

The interiors of equipment must be free of wire cuttings, dust, spare parts and other foreign matter. No clip leads or push-on jumpers should be in use during normal operation and all cables must be neatly dressed by clamps or routing. All chassis and frames must be properly bolted down, with all hardware in place. Air filters should be checked for cleanliness and replaced periodically.

## 4-5 MIOP TEST PROGRAMS

The applicable test programs are listed in table 4-1. The Sigma 5 and 7 System Test Monitor and peripheral system test programs should be run to test the system. If it is determined from these programs that a malfunction exists in the MIOP, the Sigma 7 Multiplexing Test program should then be run to locate the malfunction.

## 4-6 PARTS LIST TABLE

The MIOP consists of the modules listed in table 4-2. The table is arranged in six columns as follows:

a. Figure number that shows the location of the module.

b. Brief description of the module.

c. Reference designator (slot and chassis number) of the module.

d. Name of the company that manufactures the module.

e. Assembly part number of each module.

f. The quantity of each module required for each MIOP.

The location of each module is shown in figure 4-1.

Table 4-1. Checkout Programs for MIOP

| Publication Number | Title |
|---|---|
| 901076 | Sigma 5 and 7 Systems Test Monitor |
| 901085 | Sigma 5 and 7 Buffered Line Printer System Test |
| 901086 | Sigma 5 and 7 Keyboard-Printer System Test |
| 901090 | Sigma 5 and 7 Medium-Speed RAD File System Test |
| 901110 | Sigma 5 and 7 9-Channel Magnetic Tape System Test |
| 901120 | Sigma 5 and 7 Card Punch System Test |
| 901121 | Sigma 5 and 7 Card Reader System Test |
| 901122 | Sigma 5 and 7 Paper Tape Reader/ Punch System Test |
| 901126 | Sigma 7 Multiplexing Test |

| | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | AT11 | FT27 | LT13 | IT25 | LT12 | IT16 | FT27 | FT26 | FT26 | FT26 | FT26 | FT26 | XT10 | | AT11 | IT16 | AT11 | BT11 | AT11 | BT11 | AT12 | IT25 | AT11 | IT25 | LT13 | XT10 | FT24 | FT24 | IT25 | | FT24 | FT24 |
| B | XT10 | FT23 | FT23 | FT23 | BT11 | FT23 | FT23 | FT23 | FT24 | FT24 | FT27 | FT27 | IT16 | IT25 | BT11 | IT16 | IT25 | BT11 | AT10 | | FT24 | FT24 | BT11 | AT24 | FT27 | FT27 | LT21 | BT11 | XT10 | LT21 | BT11 | |
| C | | | FT24 | FT24 | FT24 | FT24 | FT24 | FT24 | FT24 | FT24 | | BT11 | LT13 | IT25 | IT25 | IT25 | IT25 | XT10 | AT12 | LT26 | AT13 | IT25 | FT24 | | XT10 | | FT24 | FT24 | LT21 | LT13 | IT16 | IT25 |
| D | | | ④ FT25 | ③ FT25 | ② FT25 | ① FT25 | ④ FT25 | ③ FT25 | ② FT25 | ① FT25 | ④ FT25 | ③ FT25 | ② FT25 | ① FT25 | ④ FT25 | ③ FT25 | ② FT25 | ① FT25 | ④ FT25 | ③ FT25 | ② FT25 | ① FT25 | XT10 | | HT15 | DT11 | HT15 | DT11 | | | | AT11 |

① BASIC IOP SUBCHANNELS 0 THROUGH 7

② OPTIONAL SUBCHANNELS 16 THROUGH 23

③ OPTIONAL SUBCHANNELS 8 THROUGH 15

④ OPTIONAL SUBCHANNELS 24 THROUGH 31

901515A. 401

Figure 4-1. MIOP Module Location Chart

Table 4-2. Multiplexing Input/Output Processor, Replaceable Parts

| Fig. & Index No. | Description | Reference Designator | Manufacturer | Part No. | Qty |
|---|---|---|---|---|---|
| 4-1 | Multiplexing Input/Output Processor | | SDS | 117610 | Ref. |
| | Model 8471 (Sigma 7, basic) | | | | |
| | Model 8472 (Sigma 7, additional sub-channels) | | | | |
| | Model 8271 (Sigma 5, basic) | | | | |
| | Model 8272 (Sigma 5, additional sub-channels) | | | | |
| | . Cable Receiver AT10 | 14B | SDS | 123018 | 1 |
| | . Cable Driver-Receiver AT11 | 14A, 16A, 10A, 18A, 32A, 1D | SDS | 123019 | 6 |
| | . Cable Driver AT12 | 12A, 14C | SDS | 124629 | 2 |
| | . Cable Driver AT13 | 12C | SDS | 125260 | 1 |
| | . Clock Driver No. 2 AT24 | 12C | SDS | 128168 | 1 |
| | . Band Gate BT11 | 13A, 15A, 2B, 5B, 10B, 15B, 28B, 21C | SDS | 116029 | 9 |
| | . Delay Line DT11 | 5D, 11D | SDS | 126963 | 2 |
| | . Increment-Decrement Register FT23 | 26B, 27B, 29B, 30B, 31B | SDS | 126749 | 5 |

(Continued)

Table 4-2. Multiplexing Input/Output Processor, Replaceable Parts (Cont.)

| Fig. & Index No. | Description | Reference Designator | Manufacturer | Part No. | Qty |
|---|---|---|---|---|---|
| 4-1 | . Buffered Latch No. 1 FT24 | 1A, 2A, 5A, 6A, 11B, 12B, 23B, 24B, 5C, 6C, 10C, 23C, 24C - 30C | SDS | 126745 | 19 |
| | . Fast Access Memory FT25 | 11D-30D | SDS | 126743 | 20 |
| | . Buffered Latch No. 3 FT26 | 21A-25A | SDS | 126856 | 5 |
| | . Buffered Latch No. 2 FT27 | 26A, 31A, 7B, 8B, 21B, 22B | SDS | 126986 | 6 |
| | . Delay Line Sensors HT15 | 6D, 8D | SDS | 127391 | 2 |
| | . Gated Inverter IT16 | 17A, 27A, 17B, 20B, 2C | SDS | 125264 | 5 |
| | . NAND Gate IT25 | 4A, 9A, 11A, 29A, 16B, 19B, 1C, 11C, 16C-19C | SDS | 128190 | 12 |
| | . Parity Generator LT12 | 28A | SDS | 123185 | 1 |
| | . Buffer Inverter No. 1 LT13 | 8A, 30A, 3C, 20C | SDS | 123016 | 4 |
| | . Logic Element LT21 | 3B, 6B, 4C | SDS | 126615 | 3 |
| | . Switch Comparator LT26 | 13C | SDS | 126982 | 1 |
| | . Terminator Module XT10 | 7A, 20A, 4B, 25B, 32B, 8C, 15C, 10D | SDS | 116257 | 8 |

SDS

# READER SURVEY

PUBLICATION NO. ____________ TITLE: ____________

**IS MATERIAL PRESENTED PROPERLY:**

- FULLY COVERED? ☐
- CLEARLY EXPLAINED? ☐
- WELL ILLUSTRATED? ☐
- WELL ORGANIZED? ☐
- OTHER ____________ ☐

**HOW DID YOU USE THIS PUBLICATION?**

- FOR TROUBLESHOOTING AND REPAIR ☐
- FOR PROGRAMMING INFORMATION ☐
- FOR OPERATING INFORMATION ☐
- AS A STUDENT ☐
- AS AN INSTRUCTOR ☐
- OTHER ____________ ☐

**WHAT IS YOUR POSITION?**

CUSTOMER PERSONNEL

CUSTOMER ORGANIZATION ____________

____________

- TECHNICIAN ☐
- ANALYST ☐
- MANAGER ☐
- OPERATOR ☐
- PROGRAMMER ☐
- STUDENT ☐
- OTHER ____________ ☐

SDS PERSONNEL

- CUSTOMER ENGINEER ☐
- SALES REPRESENTATIVE ☐
- SYSTEMS ENGINEER ☐
- INSTRUCTOR ☐
- STUDENT ☐
- OTHER ____________ ☐

COMMENTS: ____________

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 1026
SANTA MONICA, CALIF.

BUSINESS REPLY MAIL

NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

SCIENTIFIC DATA SYSTEMS
1649 Seventeenth Street
Santa Monica, California 90404

ATTN: TECHNICAL PUBLICATIONS DEPT.

FOLD

CUT ALONG LINE