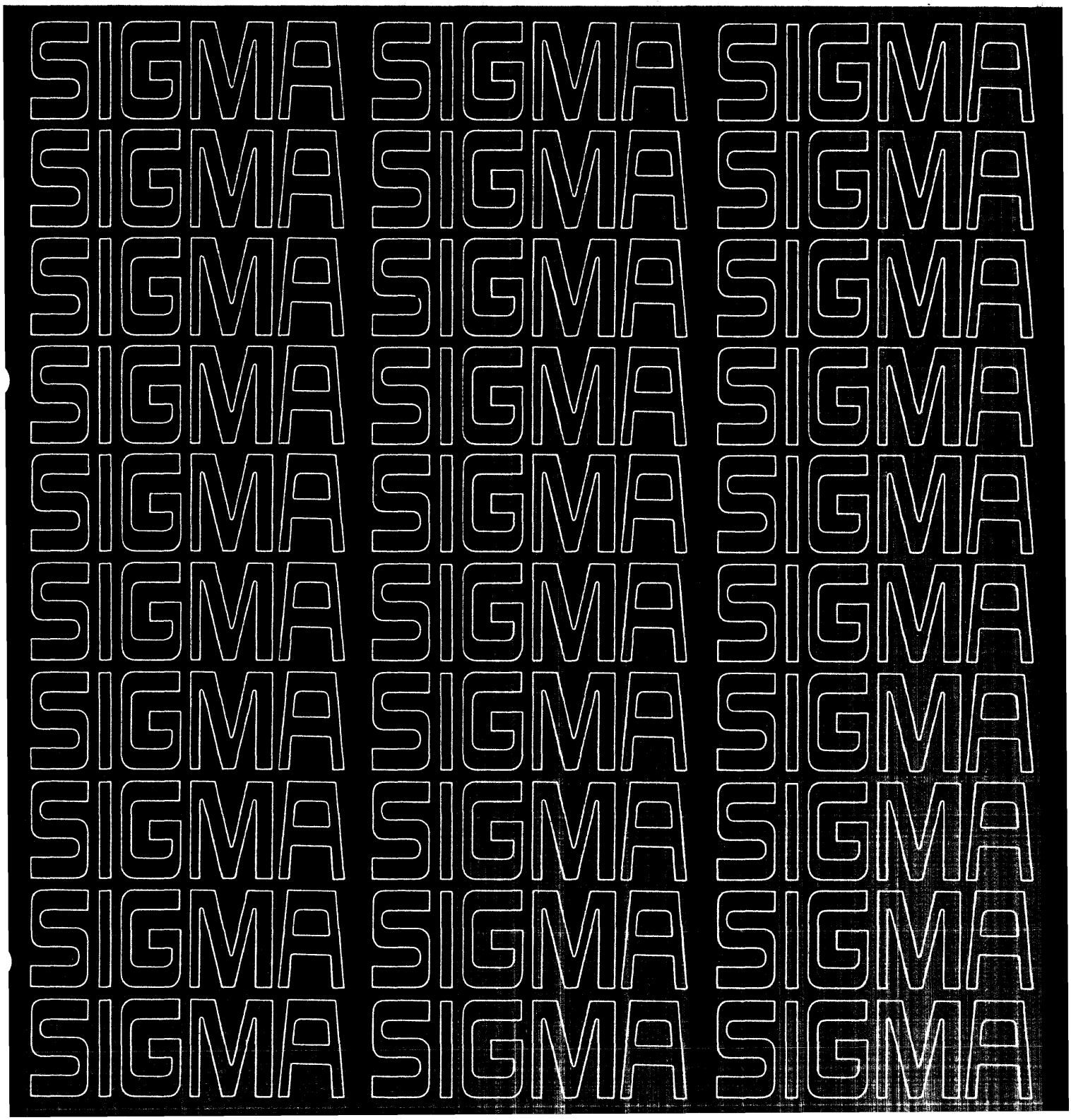
TECHNICAL MANUAL

SELECTOR INPUT/OUTPUT PROCESSOR (SIOP) MODELS 8285 AND 8485

SCIENTIFIC DATA SYSTEMS



SDS 901195A
$6.50

TECHNICAL MANUAL

SELECTOR INPUT/OUTPUT PROCESSOR (SIOP)

MODELS 8285 AND 8485

November 1968

SCIENTIFIC DATA SYSTEMS • 701 South Aviation Boulevard • El Segundo, Calif., 90245 • 213/772-4511

LIST OF EFFECTIVE PAGES

Total number of pages is 182, as follows:

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont.)

LIST OF TABLES (Cont.)

LIST OF RELATED PUBLICATIONS

Publication Title	Publication No.
Sigma 5 Computer Reference Manual	900959
Sigma 7 Computer Reference Manual	900950
Sigma Computer Systems Interface Design Manual	900973
Sigma 7 Computer Technical Manual	901060
Sigma 5 Computer Technical Manual	901172
Sigma 5 and 7 Systems Test Monitor	901076
Sigma 5 and 7 Buffered Line Printer System Test, Diagnostic Program Manual	901085
Sigma 5 and 7 Keyboard/Printer System Test, Diagnostic Program Manual	901086
Sigma 5 and 7 Medium Speed Rapid Access Data (RAD) System Test, Diagnostic Program Manual	901090
Sigma 5 and 7 9-Channel Magnetic Tape System Test	901110
Sigma 5 and 7 9-Channel Magnetic Tape System Test, Diagnostic Program Manual	901119
Sigma 5 and 7 Card Punch System Test, Diagnostic Program Manual	901120
Sigma 5 and 7 Card Reader System Test, Diagnostic Program Manual	901121
Sigma 5 and 7 Paper Tape Reader/Punch System Test, Diagnostic Program Manual	901122
Sigma 5 and 7 9-Channel Magnetic Tape Test, Diagnostic Program Manual	901129
Sigma 5 and 7 Magnetic Tape Test (7-Channel) Diagnostic Program Manual	901165

SECTION I
GENERAL DESCRIPTION

1-1 INTRODUCTION

This manual describes SDS Selector Input/Output Processor (SIOP) Models 8285 and 8485. Model 8285 is used with the Sigma 5 computer, and Model 8485 is used with the Sigma 7 computer. The bus-sharing feature (modification kit, SDS part No. 137304) may be used on either model.

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

Technical manuals describing equipment associated with the SIOP are referred to in the list of related publications in the front matter of this manual.

1-2 PHYSICAL DESCRIPTION

The basic SIOP consists of 169 modules installed in chassis A, B, C, D, E, and F. The optional bus-sharing feature consists of five additional modules installed in slots 11 through 15 of chassis A.

1-3 FUNCTIONAL DESCRIPTION

The primary purpose of the SIOP is to connect high speed peripheral devices to the CPU and memory units of the computer and to control the exchange of data between the device and core memory. However, the SIOP may be used with slow and medium speed devices. Once an I/O operation has been started by the CPU, the operation is performed to completion by the SIOP, the device controller, and memory without intervention by the CPU. The SIOP processes the I/O operation while the CPU is performing functions possibly unrelated to the I/O operation. The SIOP controls the I/O operations by executing a command list prepared by the CPU which is stored in core memory (see figure 1-1 for a simplified block diagram of the I/O system).

An I/O operation starts when the CPU issues a start input/output (SIO) instruction addressed to a particular SIOP and device controller (or device-device controller combination). The addressed SIOP, after receiving the address information, places the device address on the SIOP-device controller interface lines and waits for a response. The addressed device responds by returning condition code and status information to the SIOP. The SIOP sends the condition code information to the CPU, and depending upon the coding of the SIO instruction, may or may not send the status and other information related to the SIOP, device, and device controller to the CPU (via core memory). The SIO instruction causes the addressed device controller to go to the busy state and make service requests from the SIOP. As a result of the service requests (after the SIO instruction has been concluded), the device controller connects itself to the SIOP-device controller interface lines. The time during which the device controller is connected is called a service cycle. It is during the service cycles that follow an SIO instruction that data is exchanged between the device and core memory. The SIOP performs core memory accesses as required by the device, continually updates information such as word and byte address information related to the data, and controls the I/O operation until it is completed, aborted, or halted by a halt input/output (HIO) instruction issued by the CPU.

During execution of the I/O instructions, two unique memory locations, X'20' and X'21' (hexadecimal 20 and 21), are used to exchange information between the SIOP and CPU. These are in addition to the SIOP-CPU interface lines. During the early phases of an SIO instruction, the CPU writes the address of the device/device controller and the address of the first command doubleword (stored in the command list), into memory location X'20'. The function code (SIO) and SIOP address are sent directly to the SIOP on the SIOP-CPU interface lines. The addressed SIOP then reads memory location X'20', stores the address of the command doubleword in its internal registers and places the device/device controller address on the interface lines. After the addressed device controller responds by sending status and condition code information to the SIOP, the SIOP loads the status and other information (if the SIO so specifies) into memory locations X'20' and X'21' to make it available to the CPU. The condition code is then sent directly to the CPU.

When the device controller starts making service requests at the conclusion of the SIO instruction, the SIOP accesses core memory for the 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 so that it will know what function to perform. The balance of the command doubleword is related to the I/O operation, and is retained by the SIOP. This information directs the operations of the SIOP until the operation is concluded. The exchange of I/O data takes place after the device has received the order. The SIOP keeps track of the number of bytes of data transmitted and their core memory locations, performs buffering operations, checks parity, records the SIOP status for future interrogation by the CPU, and performs the other operations related to the I/O operation.

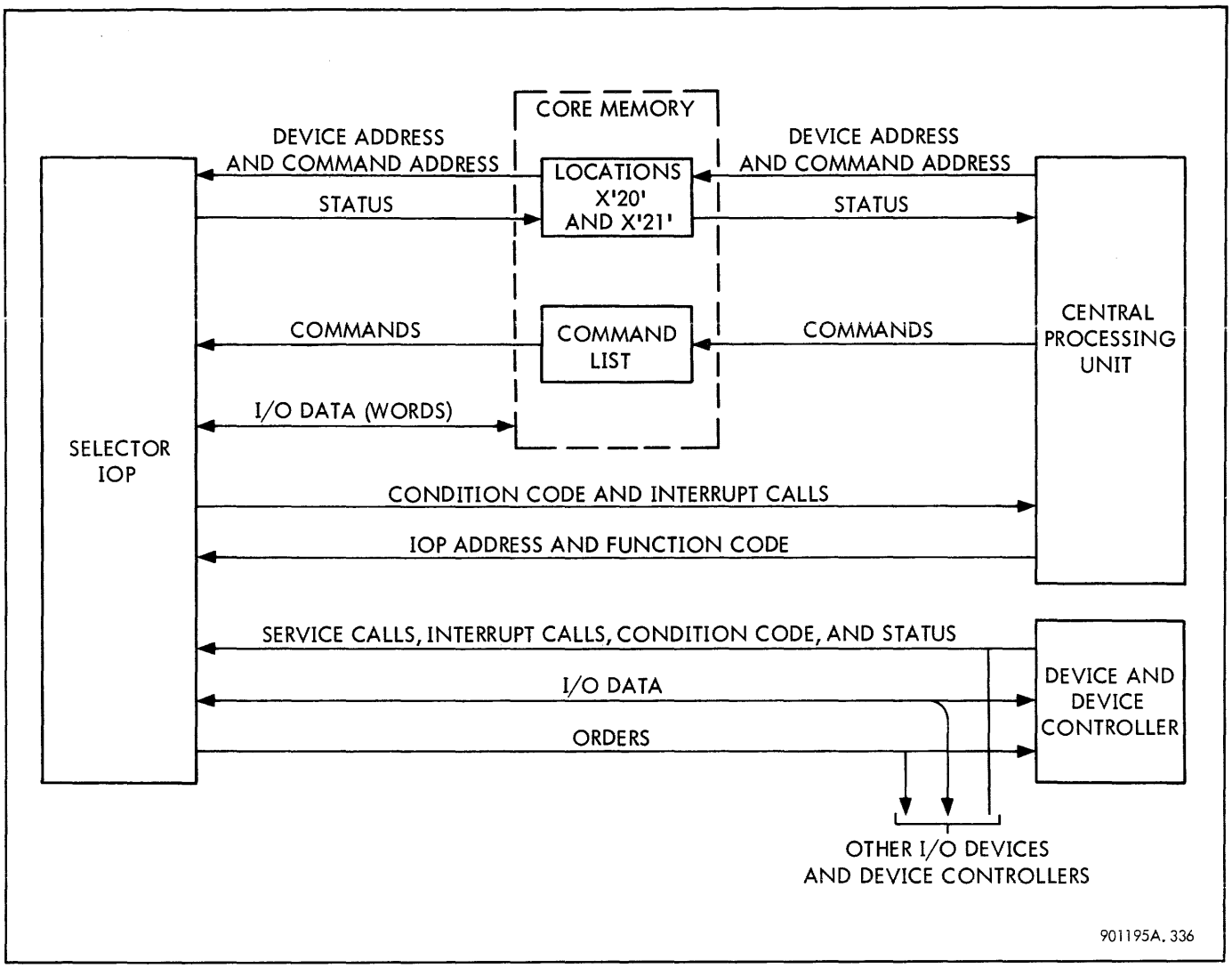


Figure 1-1. I/O System, Simplified Overall Block Diagram

The SIOP is capable of performing command chaining and data chaining operations. 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 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 of the CPU. Command chaining causes the SIOP to load a new command doubleword at the end of a record, send the new order to the device, and start processing the new record. The SIOP allows only one of the 32 possible device controllers attached to it to be in the busy state at any given time. The SIOP also allows a device controller to exchange a complete record with core memory as the result of a single service request from the device controller.

Interrupt calls made by a device controller are passed along to the CPU by the SIOP. 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 SIOP and device controller. The highest priority device controller with an interrupt pending sends its address (along with status and condition code information) to the SIOP. The SIOP 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 the appropriate action based on the status and condition code information it has just received.

1-4 SPECIFICATIONS AND LEADING PARTICULARS

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

Table 1-1. General Specifications

Power (supplied by PT16 power supply)	+8 Vdc (15A) -8 Vdc (3.3A) +4 Vdc (35A) Total watts: 286
Logic signal levels	One, +4 Vdc 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 max
Operating	10,000 feet max

SECTION II
OPERATION AND PROGRAMMING

2-1 GENERAL

The SIOP contains no controls or indicators other than the address switches and bus-share switches contained on the switch comparator module (LT26) in slot 8 of chassis F (see figure 2-1). The SIOP address switches (S1-1, S2-1, and S3-1) are set as indicated in table 2-1. Switch S4-1 must be set to 1 for the lowest priority IOP on the CPU interface; otherwise, S4-1 must be set to 0. Two switches (S1-2 and S2-2) must be set if the bus-sharing option is installed. The first switch (S2-2) must be set to 1 on both of the bus-sharing SIOPs. This designates that the option is installed. The second switch (S1-2) must be set to 1 for the highest priority (A) IOP, and set to 0 for the lowest priority (B) IOP. Switch S2-2 designates the relative priority of the two SIOPs in gaining use of the core memory bus when requests by each are made at the same time.

The following definitions apply to instructions, commands (command doublewords), and orders: the CPU executes instructions, the SIOP executes commands, and the devices

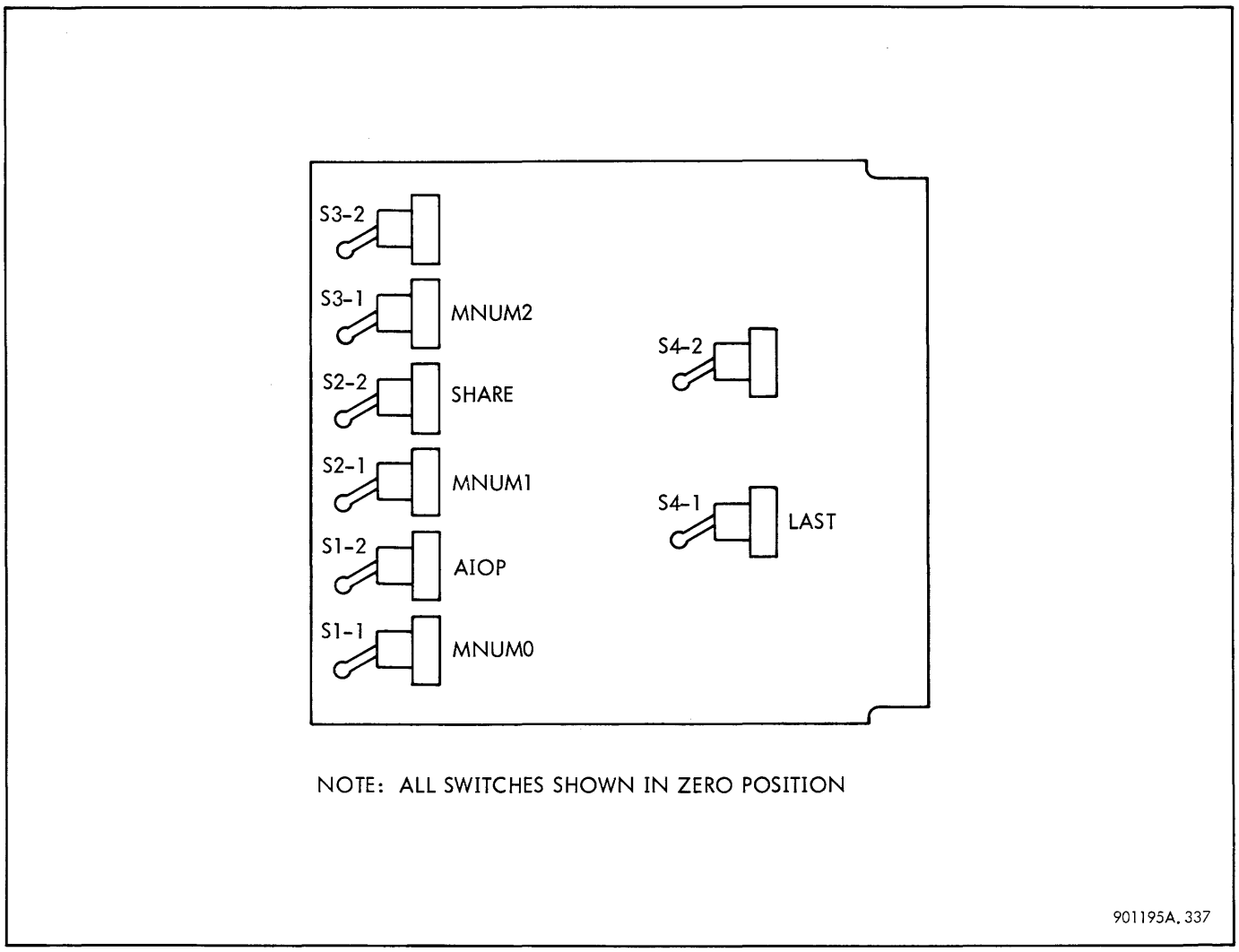


Figure 2-1. Selector IOP Switch Settings

execute orders. For example, the CPU may execute an SIO instruction to initiate an I/O operation. During the course of the operation, the SIOP fetches a command from core memory and stores it (except the order bits) in its internal registers. The command provides the SIOP with information needed to perform its function; commands are therefore executed by the SIOP. The order bits of the command are transmitted to the devices and define the operations they are to perform; the I/O devices therefore execute orders.

For a description of I/O instructions, commands, and orders, see the Sigma 7 computer reference manual 900950 and the Sigma 5 computer reference manual 900959.

Table 2-1. Selector IOP Address Settings

SIOP ADDRESS	SWITCH SETTING		
	S1-1 MNUM0	S2-1 MNUM1	S3-1 MNUM2
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

SECTION III
PRINCIPLES OF OPERATION

3-1 INTRODUCTION

This section describes the principles of operation of the SIOP on a general information level and the logic elements on a detailed level. The detailed principles include a description of interface signals, the operation at each interface, a description of a typical I/O operation, the function of each register and flip-flop, timing signal generation, memory bus sharing, and a description of each of the instructions and service cycles based on phase sequence. A glossary of terms is included at the end of section III.

The SIOP allows only one of the several possible devices attached to it to be in a transmission operation at a given time. This transmission operation may occur in the burst mode (that is, when complete transmission of a record occurs without the device relinquishing and regaining access to the interface. [Not all devices are capable of burst mode operation]). When operating in the burst mode, the SIOP does not raise the end data line to the device until the entire byte count has been reduced to zero, or until an HIO or AIO causes the service call to be interrupted. This operation differs from that of a multiplexing IOP, which always signals end data when it reaches the end of a word for transmission to memory; that is, a multiplexing IOP accepts data from a device or transmits data to a device until one word has been exhausted or the device signals end data.

3-2 GENERAL PRINCIPLES OF OPERATION

The following paragraphs explain briefly the operation of the SIOP during instruction execution and transfer of data. Also included is a description of timing signal generation. Figure 3-1 is a simplified block diagram of the SIOP.

3-3 OPERATION DURING INSTRUCTION EXECUTION

When the CPU issues an SIO instruction to the SIOP, the CPU places the SIOP address on the address lines and the function code on the function code lines and drives the control strobe line. The address (three bits) is decoded by the SIOP address logic to determine if this is the SIOP to which the SIO is addressed. If it is, a delay line in the timing section is started. The delay line generates timing signals that time the logic functions needed to perform I/O operations. All I/O operations start with the beginning phase. As the SIOP progresses through an operation, different phase latches are set; the particular latch depends upon the operation being performed.

The function code (three bits) is decoded by the function decode logic, which in turn drives the appropriate function code indicator line (in this case, SIO), and raises the function strobe line to the device controller. The function indicator lets the device controller know how to handle information received on the other lines.

As the SIOP progresses to the next phase of the operation, X'20' is forced into the memory address register and a memory read operation is performed. As a result, the contents of memory location X'20' are set into the SIOP memory assembly register. Previously, the CPU had placed into location X'20' the device/device controller address, an R field coding, and the address of the first command doubleword to be processed by the SIOP for the current I/O operation. The R field coding tells the SIOP how much information it shall return to the CPU via memory locations X'20' and X'21'. The device controller address is transferred directly from the memory assembly register to the address register and I/O register. Information in the I/O register is also present on the data lines, where it is available to the device controllers. The address of the first command doubleword is transferred from the memory assembly register to the command address counter during a later phase of the SIO instruction only if the SIO is successful. The command doubleword address is then retained by the command address counter until the SIO has been completed and the service cycles begin.

After the SIOP has stored the information it received from memory location X'20', it starts loading information into memory locations X'20' and X'21' (if the R field of the instruction so specifies). If both locations are to be written into, the SIOP performs two write operations. The first write operation stores into memory location X'20' the address of the command last executed by the SIOP. The second write operation stores into memory location X'21' the status information that the addressed device controller sent to the SIOP via the FR lines, the SIOP status, and the byte count currently stored in the byte count counter. The SIOP then drives the proceed line to the CPU. If the SIO is successful, the SIOP and device controller both enter the busy state and the device controller raises its service call (SC) line.

3-4 OPERATION DURING DATA TRANSFER

In response to the service call made by the device controller, the SIOP begins an operation which causes that device controller to be connected to the SIOP for service. There are four possible types of service cycles that may

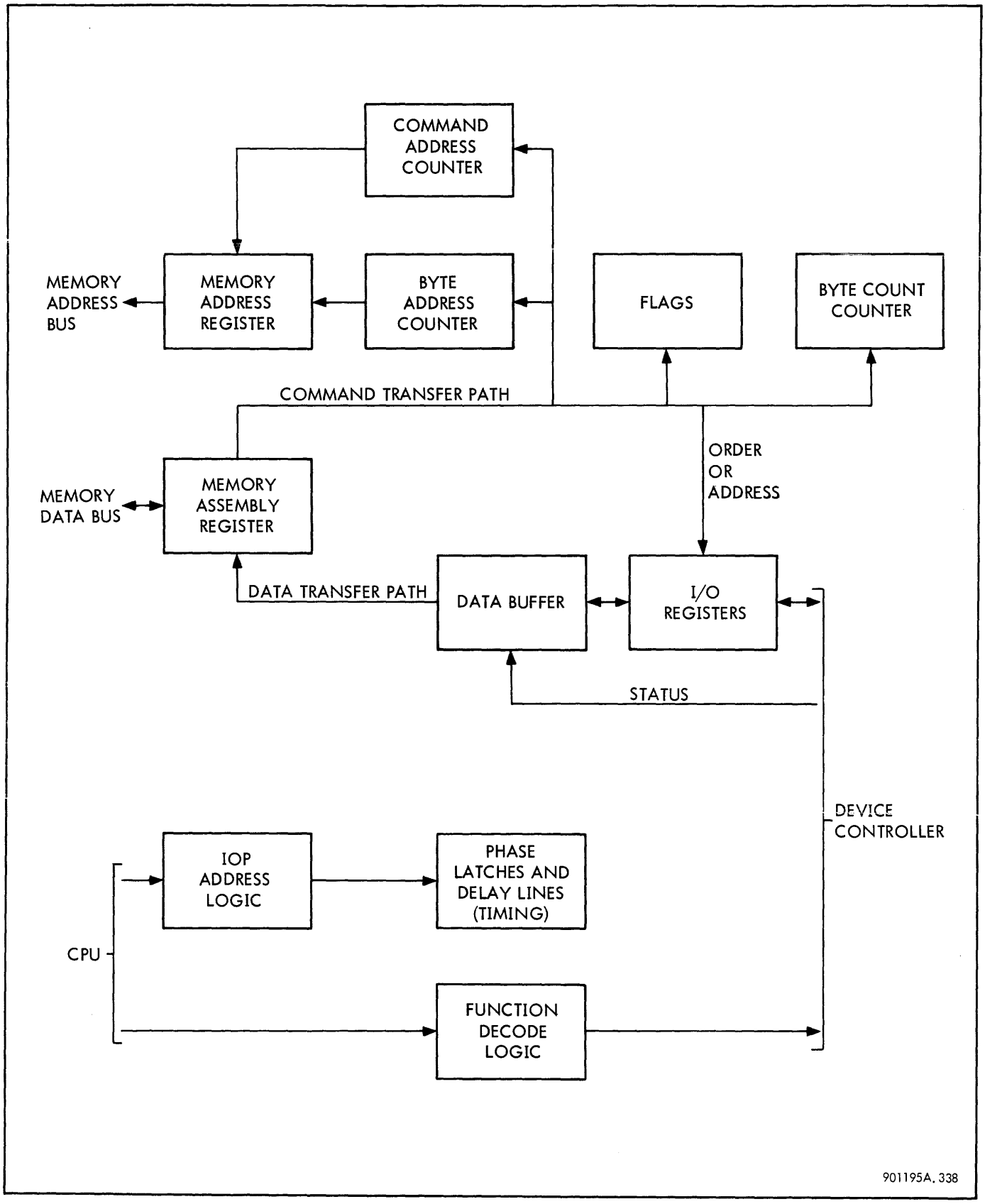


Figure 3-1. SIOP Simplified Block Diagram

follow an SIO instruction: order out, order in, data out, or data in. The device controller is mechanized so that it automatically defines the first service cycle following an SIO as an order out. The order out service cycle causes the SIOP to fetch the command doubleword stored in core memory. The address of the command doubleword is the one set into the command address counter during execution of the SIO instruction. Two memory accesses are required to fetch the two words of the command doubleword from core memory. As a result of the first memory access, the first word (even numbered) is set into the memory assembly register. The order encoded in the first word is transferred from the memory assembly register directly to the I/O register. Information contained in the I/O register is also present on the data lines and is therefore available to the device controller. The byte address encoded in the first word is transferred from the memory assembly register to the byte address counter. The SIOP then increases the command address by one and makes the second memory access. The second memory access sets the second word (odd numbered) of the command doubleword into the memory assembly register. The flags and byte count encoded in the second word are transferred from the memory assembly register to their respective registers, and the order out service cycle is concluded.

The device controller again raises its service call line and is again connected to the SIOP. This time, however, the device controller defines the service cycle as either a data in or data out, depending upon the order it has just received. The data in service cycle applies to a read or read backwards order. During a data in service cycle, data is accepted from the device, via its device controller, and is written into ascending core memory locations. A read backwards order causes the data to be written into descending core memory locations. If the order sent to the device controller is a write order, the device controller defines the service cycle as a data out. During a data out service cycle, the SIOP reads data from ascending memory locations and transfers it to the device via the device controller.

The SIOP may operate with device controllers having a one-byte, two-byte, or four-byte wide data path. After it has received these wide transmissions, the SIOP handles the information much the same as it would a collection of eight-bit bytes.

During a data in service cycle, bytes are received by the I/O register from the device controller. From the I/O register, they are transferred through the data buffer to the memory assembly register. The bytes enter the appropriate byte position of the memory assembly register so that they will be written into the memory word and byte location specified by the byte address counter. As each byte is transferred to the memory assembly register, the byte address is incremented by one (if the order is read) or decremented by one (if the order is read backwards) and the byte count is decremented by one.

During a data out service cycle, the operation is essentially the reverse of that during a data in cycle. Words are accessed from core memory into the byte positions of the memory assembly register specified by the byte address counter. The bytes selected by the byte address counter are transferred through the data buffer to the I/O register and are therefore available to the device controller. The byte count is decremented and the byte address is incremented for each byte transferred at the memory interface.

3-5 TIMING

Timing signals for all SIOP I/O operations are supplied by three delay lines and their associated phase latches. If the bus-sharing option is included, an additional delay line is used. Each delay line and its associated phases are generally associated with a particular part of an I/O operation.

3-6 DETAILED PRINCIPLES OF OPERATION

3-7 SIOP INTERFACE SIGNALS

A description of the operation at each interface and a brief description of each interface signal are provided under the three interface headings. The interface signals are also listed in tables, which include the signal name, designator, number of lines for signal groups, and direction of flow. See the Sigma Computer Systems Interface Design Manual, SDS publication 900973, for connection of SIOP into a Sigma 5 or 7 computer system. Figure 3-2 is an overall block diagram of the SIOP.

3-8 SIOP-Memory Interface

The SIOP 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. Each SIOP connects to a separate core memory bus unless the bus-sharing option is used. The bus-sharing option permits two SIOPs to share a core memory bus; the option must be present in the two SIOPs that share the memory bus.

The SIOP uses core memory locations X'20' and X'21' when responding to CPU requests for service. The memory (memories if interleaving is used) that possesses SIOP memory locations X'20' and X'21' must also possess CPU memory locations X'20' and X'21' if mapping is in effect. This is also a requirement for an SIOP which is to provide the CPU with its initial information during a bootstrap load operation. (Since the information transmitted through memory locations X'20' and X'21' applies to the CPU and the SIOP, communication via these two locations is presented under the SIOP-CPU interface heading, paragraph 3-11.)

3-9 SIOP-MEMORY INTERFACE OPERATION. When information is read from core memory, a full 32-bit word is always transferred to the SIOP. During a core memory write operation, however, information may be written into

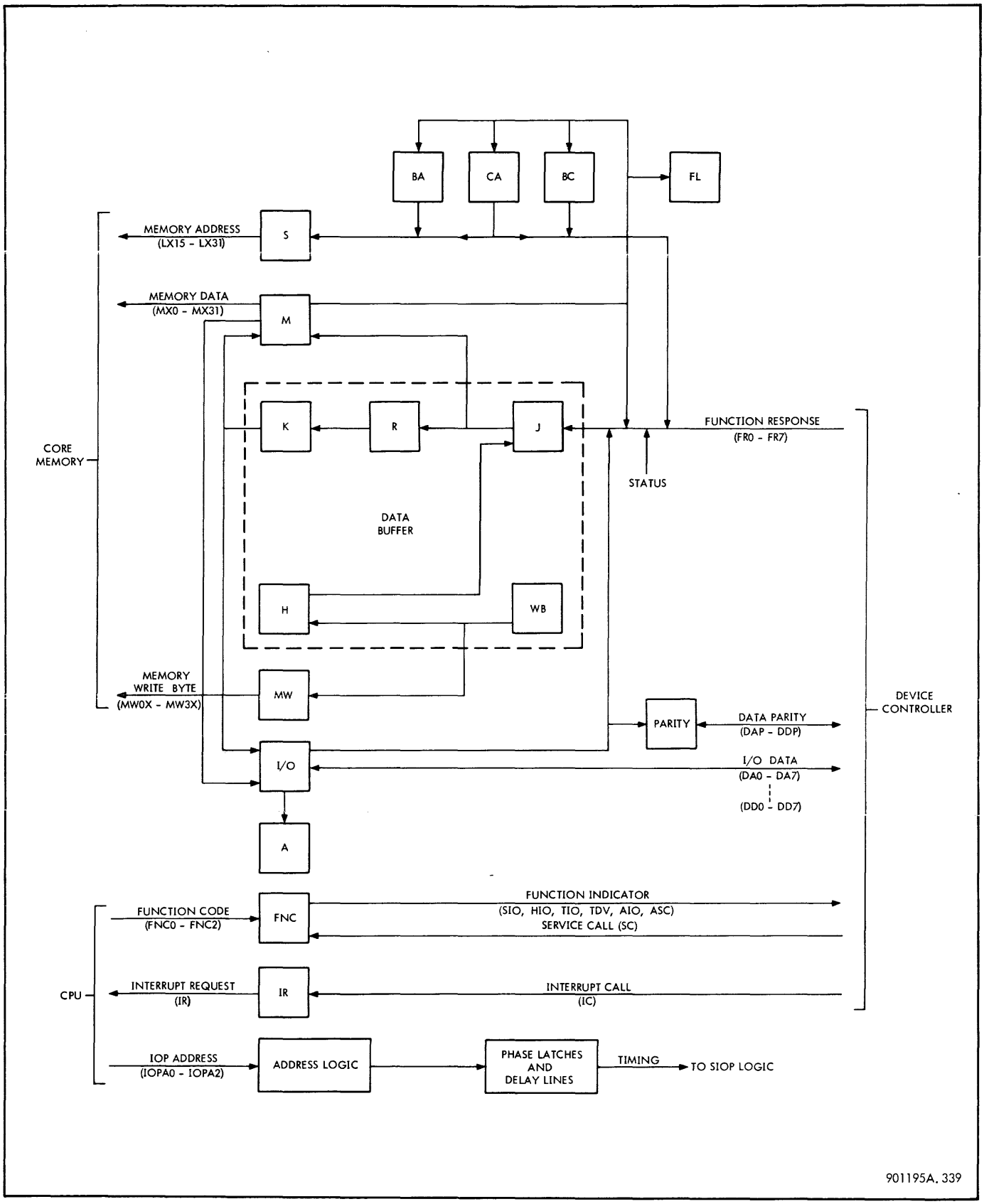


Figure 3-2. SIOP Overall Block Diagram

core memory on an eight-bit byte basis. One, two, three, or four selected bytes may be written; the number of bytes is controlled by the SIOP.

A memory access is initiated when the SIOP drives the memory request MQ (driven by signal MRQ) line (see figure 3-3). Before driving the memory request line, the SIOP places a 17-bit address on memory address lines L15 through L31, holds the write byte lines (MW0 through MW3) false to define a read operation, and waits for signal address release (AR) true, or address here (AH) false from core memory. During a core memory write operation, the SIOP drives the memory request line and then supplies the data to be written on memory data lines M0 through M31 simultaneously with the write byte signals on write byte lines MW0 through MW3. The memory request line is held true until the memory acknowledges the state by driving address release signal AR true or address here signal AH false. If no further requests are to be made, the request line and address lines are released. The SIOP does not make a memory request before receiving an address release signal or a false address here signal from the prior request.

3-10 SIOP-MEMORY INTERFACE SIGNALS. A brief description of the SIOP-memory interface signals follows. See table 3-1. All times are as measured at SIOP-cable interface.

a. Address here signal (AH). The memory drives the address here line to inform the SIOP that the address encoded on the address lines is an address implemented in the memory or memories to which the SIOP is attached. The address here line is strobed by the SIOP no sooner than 260 ns after the address lines have settled.

b. Address release signal (AR). The address release signal (not less than 40 ns duration) is generated by the memory to inform the SIOP that the address lines may be changed and that the memory request line may go false.

c. Address signals L15 through L31. The SIOP places the memory address on the 17 address lines to define the core memory word location to be read from or written into. The signals on the address lines are stable not less than 90 ns before the memory request signal goes true and remain stable until 20 ns after the memory request signal goes false.

d. Data gate signal (DG). The data gate signal is driven by core memory to strobe the data on the data lines into the SIOP memory assembly register during a core memory read operation. The data gate signal is a positive-going pulse of at least 40 ns duration.

e. Data signals M0 through M31.

1. Data input to memory. During a core memory write operation, signals on the data lines are present and stable within 120 ns after the memory request signal goes true, and remains stable for a minimum of 160 ns after the address release signal is received by the SIOP.

2. Data output from memory. During a core memory read operation, signals on the data lines are strobed by the data gate signal into the SIOP memory assembly register. The data signals do not change during the interval 80 ns before the data gate signal goes true until 65 ns after the data gate signal goes false.

f. Memory request signal (MQ). The memory request signal is driven true by the SIOP when it wishes to make a memory access. This signal is held true by the SIOP until it receives the address release signal or the address here signal is false when strobed.

g. Parity error signal (PE) and parity OK signal (POK). The parity error and the parity OK signals are positive-going pulses of not less than 40 ns duration. The memory drives one or the other to indicate the result of the parity check performed by memory during a memory read or partial write operation.

h. Write byte signals MW0 through MW3. The SIOP drives the appropriate write byte lines to designate the core memory byte or bytes into which data is to be written. These signals are present and stable within 120 ns after the memory request signal and are not dropped until the address release signal is received by the SIOP.

Table 3-1. SIOP-Memory Interface Signals

SIGNAL	DESIGNATOR	NO. OF LINES	DIRECTION OF SIGNAL FLOW	
			SIOP to Memory	Memory to SIOP
Address here	/AH/	1		X
Address release	/AR/	1		X
Address signals	/L15/-/L31/	17	X	
Data gate	/DG/	1		X
Data signals	/M0/-/M31/	32	X	X
Memory request	/MQ/	1	X	
Parity error	/PE/	1		X
Parity OK	/POK/	1		X
Write byte signals	/MW0/-/MW3/	4	X	

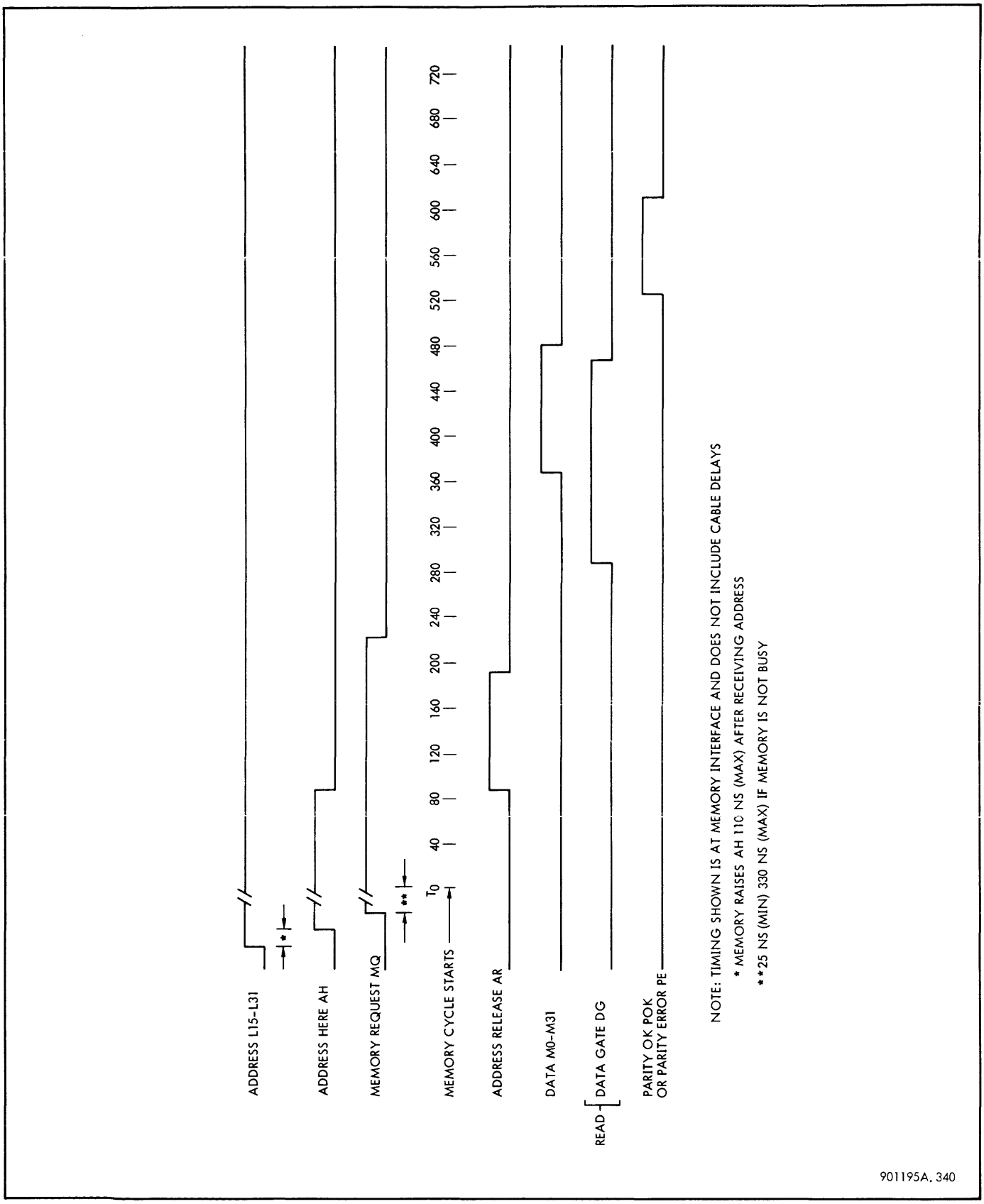


Figure 3-3. SIOP-Memory Interface Signal Timing Diagram (Port A or B)

3-11 SIOP-CPU Interface

Since much of the information transfer between the SIOP and CPU is via core memory, only one 14-wire cable is required at the SIOP-CPU interface. The SIOPs are arranged in a priority sequence such that during the execution of an AIO instruction, the SIOP nearest the CPU with an interrupt request pending responds to the CPU request for service.

3-12 SIOP-CPU INTERFACE OPERATION. The SIOP-CPU interface operations resulting from standard I/O instructions (if normally executed) occur in three distinct phases, as described below:

a. The CPU decodes the instruction and stores information related to the I/O operation in core memory location X'20'. The SIOP-CPU control and address lines are energized and service is requested of the SIOP.

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

c. When the proceed signal is received, the CPU performs the necessary operations with the data it has received, and terminates the operation as far as the CPU is concerned. The SIOP, device controller, and core memory then carry on the I/O operation to its completion without intervention by the CPU.

A description of SIOP-CPU communication via memory during the five I/O instructions is presented under paragraphs 3-13 through 3-16.

3-13 SIO. If the IOP busy status bit is true, or if a zero count, channel end, or unusual end interrupt is pending, the SIOP partially executes the SIO instruction. If the IOP busy status bit is false and a zero count, channel end, or unusual end interrupt is not pending, the SIOP executes the SIO instruction normally. (See the glossary of terms, table 3-36.) During normal execution, the SIOP makes a memory request to read memory location X'20'. If a memory parity error is detected, the SIOP terminates the operation immediately. If a memory parity error is not detected, the SIOP interprets the contents of memory location X'20' as shown in figure 3-4. Each group of bits of memory location X'20' is described under a through d below:

a. Bits 0 through 7 contain the device controller address. These bits are transmitted to the device controller on data lines DA0 through DA7. They are also stored in the SIOP if the SIOP goes to the busy state during the SIO instruction.

b. Bits 8 and 9 specify the SIOP response to the SIO instruction as follows:

Bit 8	Bit 9	SIOP Response
0	0	SIOP responds on NCOND1 and NCOND2 lines only
0	1	SIOP responds on NCOND1 and NCOND2 lines and writes in memory location X'21'
1	1	SIOP responds on NCOND1 and NCOND2 lines and writes in memory locations X'20' and X'21'

c. Bits 10 through 15 are ignored by the SIOP.

d. Bits 16 through 31 contain the command doubleword address that is subsequently stored in the SIOP if the SIOP goes to the busy state during the SIO instruction.

If the SIOP reads memory location X'20' without detecting an error, it communicates with the addressed device controller to receive the current device controller status. The SIOP then writes into memory locations X'20' and X'21' according to the state of bits 8 and 9 when memory location was read. The format of the information is shown in figure 3-5. Each group of bits is described under a through e below:

a. Memory location X'20' bits 0 through 15 contain the command doubleword address currently stored in the SIOP and decremented by one.

b. Memory location X'20' bits 16 through 31 contain the same data as was read from memory location X'20' bits 16 through 31.

c. Memory location X'21' bits 0 through 7 contain the device controller status information obtained from the function response lines at the device controller interface.

d. Memory location X'21' bits 8 through 15 contain SIOP status information stored in the SIOP. This status information is cleared prior to termination of the CPU service if the SIOP goes to the busy state. Each status bit is defined separately under 1 through 8 below:

1. Bit 8 – incorrect length. This bit is set true if incorrect length was signaled by the device controller via the order in operational status byte or if the SIOP detected data in the buffer at count done during a data in service cycle.

2. Bit 9 – transmission data error. This bit is set true if a transmission error was signaled by the device controller via the order in operational status byte or if the SIOP detected a byte parity error during a data in service cycle.

3. Bit 10 – transmission memory error. This bit is set true if a memory parity error was detected by the SIOP during a data out service cycle (memory read), or during a partial write operation of a data in service cycle.

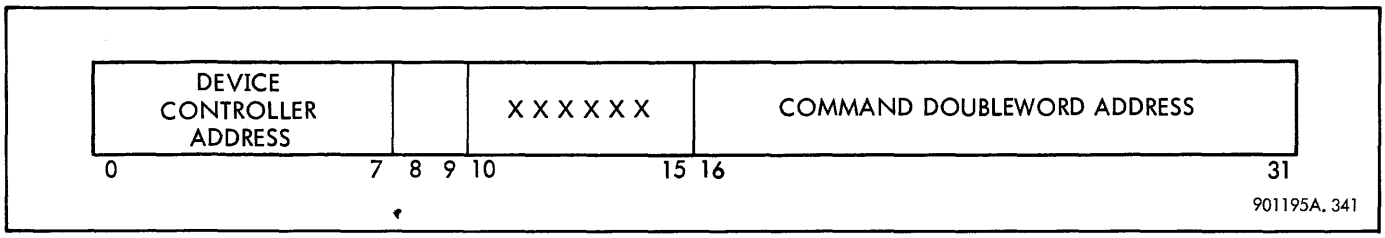


Figure 3-4. Format of Information Read from Core Memory Location X'20' During an SIO Instruction

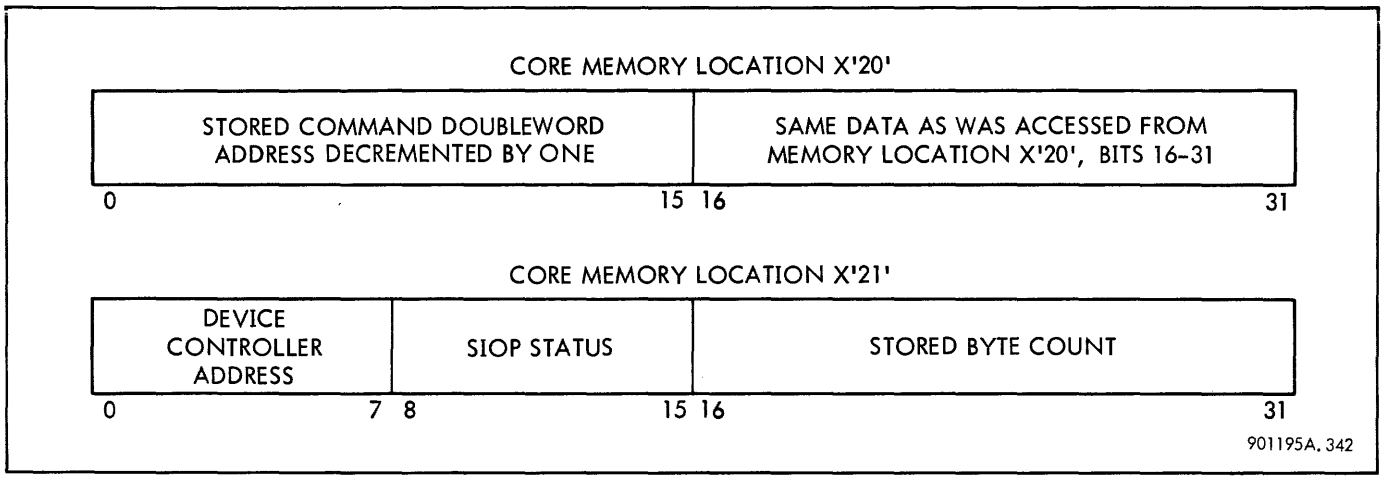


Figure 3-5. Format of Information Written Into Core Memory Locations X'20' and X'21' During an SIO Instruction

4. Bit 11 – memory address error. This bit is set true if a nonimplemented memory address was detected by the SIOP during a chaining operation or a data in service cycle.

5. Bit 12 – IOP memory error. This bit is set true if the SIOP detected a memory parity error while fetching a command doubleword.

6. Bit 13 – IOP control error. This bit is set true if the SIOP detected two successive transfer-in-channel commands.

7. Bit 14 – IOP halt. This bit is set true if the SIOP detected an IOP halt condition.

8. Bit 15 – IOP busy. This bit is set true during the current SIO instruction if the device controller responded by driving both the DOR and IOR lines true, indicating a condition code of CC1 = 0 and CC2 = 0. The SIOP does not report IOP busy during an SIO instruction. Bit 15 is cleared under the following conditions:

(a) The reset I/O line is true.

(b) During an HIO operation, the device controller address stored in the SIOP matched the device controller address transmitted at the device controller interface.

(c) During an order in service cycle, the device controller signaled an unusual end via the operational status byte.

(d) During an order in service cycle, the device controller signaled a channel end via the operational status byte and the command chain flag was false, or the IOP halt status byte was true.

e. Memory location X'21' bits 16 through 31 contain the byte count currently stored in the SIOP.

3-14 TIO and TDV. The TDV or TIO instruction is partially executed if all of the three following conditions prevail:

a. The SIOP is in the busy state.

b. The SIOP is currently responding to a device controller service request.

c. The SIOP has signaled neither end data nor request strobe acknowledge to the device controller before receiving the TDV or TIO function code from the CPU.

The TDV or TIO instruction is normally executed if any one of the following three conditions prevails:

a. The SIOP is not in the busy state.

b. The SIOP is in the busy state and is not currently responding to a device controller service request.

c. The SIOP is in the busy state, is currently responding to a device controller service request, and has signaled end data and request strobe acknowledge to the device controller before receiving the TDV or TIO function code from the CPU.

During normal execution of a TDV or TIO instruction, the SIOP makes a memory request to read memory location X'20'. If a parity error is detected when memory location X'20' is read, the SIOP terminates the operation immediately. If a memory parity error is not detected, the SIOP interprets the contents of memory location X'20' as shown in figure 3-6. Each group of bits of memory location X'20' is described under a through c below:

a. Bits 0 through 7 contain the device controller address. These bits are transmitted to the device controller on data lines DA0 through DA7.

b. Bits 8 and 9 of memory location X'20' have the same meaning as they do in an SIO instruction (see paragraph 3-13b).

c. Bits 10 through 31 are ignored by the SIOP.

If the SIOP reads memory location X'20' without detecting an error, it communicates with the addressed device controller to receive the current device controller status and condition code information. The SIOP then completes the response to the CPU service request by sending condition code information to the CPU according to the condition code specified by the device controller, and if requested, by writing into memory locations X'20' and X'21'. The format of the responses and definition of the individual bits are the same for the TDV and TIO instructions as for an SIO instruction (see paragraph 3-13, a through e).

3-15 HIO. The HIO instruction is always executed normally[1] by the SIOP. If the SIOP is currently responding to a device controller service request, the SIOP terminates the service to the device controller by driving the end data line true. The SIOP responds to the CPU service request as soon as the current service is terminated. Since the HIO instruction is executed normally, the SIOP reads memory location X'20'. If a memory parity error is detected while memory is read, the SIOP holds the NCOND1 and NCOND2 lines false and drives the proceed line true immediately, thus terminating the operation. If the SIOP reads memory location X'20' without detecting a memory parity error, it interprets the contents of memory location X'20' as shown in figure 3-6. Each group of bits of memory location X'20' is described under a through c below:

[1]Normal execution here and in the following descriptions means instruction execution in which data is accessed from core memory (except for AIO), communicating with a device controller if a parity error is not detected when memory was read, possible storing of data in memory, and controlling of the NCOND1 and NCOND2 lines. Partial execution involves only controlling the NCOND1 and NCOND2 lines. (Memory location X'20' is not accessed.)

a. Bits 0 through 7 contain the device controller address. These bits are transmitted to the device controller on data lines DA0 through DA7. They are also compared with the device controller address currently stored in the SIOP. If the addresses compare, the SIOP clears the four following status bits:

1. IOP busy
2. Zero count interrupt
3. Channel end interrupt
4. Unusual end interrupt

b. Bits 8 and 9 of memory location X'20' have the same meaning as they do in an SIO instruction (see paragraph 3-13b).

c. Bits 10 through 31 are ignored by the SIOP.

If the SIOP reads memory location X'20' without detecting an error, it performs the same operations as it does during a TDV or TIO instruction (see paragraph 3-14).

3-16 AIO. The SIOP always executes an AIO instruction normally if an interrupt is pending. If the SIOP is responding to a device controller service request, the SIOP terminates the service to the device controller by driving the end data line true, and then responds to the CPU request for service. The SIOP obtains the address and status from the interrupting device controller and assembles it in its memory assembly register. The SIOP also assembles its own address and status in the memory assembly register along with the address and status received from the device controller, and then writes the contents of this register into memory location X'20'. The format of the information written into memory location X'20' is shown in figure 3-7. Each group of bits of memory location X'20' is described under a through e below:

a. Bits 0 through 7 contain the device controller status information received on data lines DA0 through DA7.

b. Bits 8 through 15 contain SIOP status information. Each bit is defined under 1 through 8 below:

1. Bit 8 – incorrect length. (Same as for SIO instruction. See paragraph 3-13d1.)

2. Bit 9 – transmission data error. (Same as for SIO instruction. See paragraph 3-13d2.)

3. Bit 10 – zero byte count interrupt. This bit is set true if the interrupt on zero count flag is true and zero byte count is detected during a data in service cycle, or if zero byte count is detected during a data out service cycle and data chaining ensues or count done is reached.

4. Bit 11 – channel end interrupt. This bit is set true if the interrupt at channel end flag is true and channel

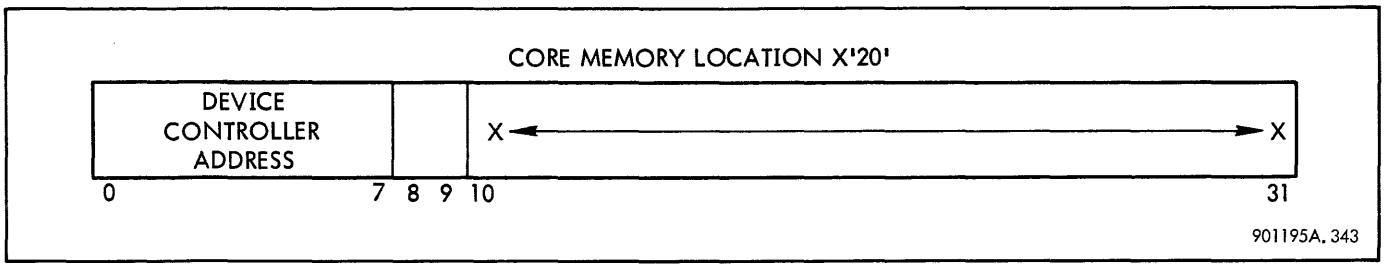


Figure 3-6. Format of Information Read from Core Memory X'20' During a TDV, TIO, or HIO Instruction

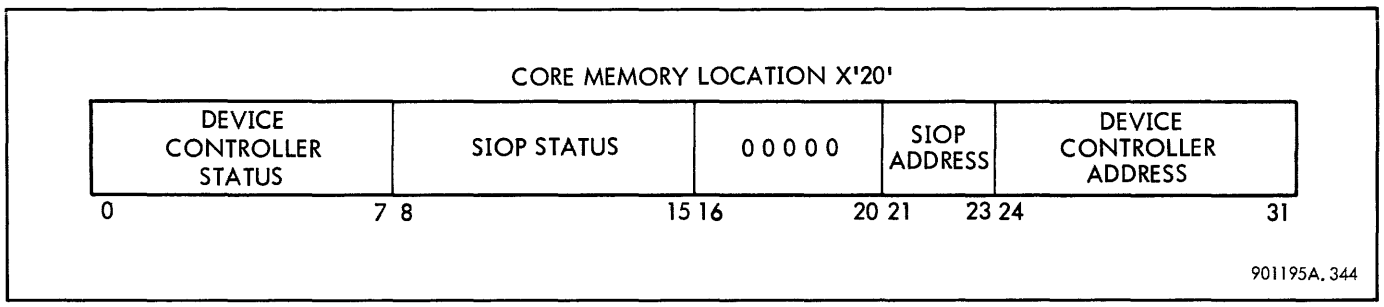


Figure 3-7. Format of Information Written Into Memory Location X'20' in Response to an AIO Instruction

end is signaled during an order in service cycle via the operational status byte.

5. Bit 12 – unusual end interrupt. This bit is set true if the interrupt at unusual end flag is true and channel end is signaled during an order in service cycle via the operational status byte, or if the unusual end flag is true and IOP halt is signaled to the device controller in the order in terminal order.

6. Bit 13 is false.

7. Bit 14 is false.

8. Bit 15 is false.

3-17 SIOP-CPU INTERFACE SIGNALS. A brief description of the SIOP-CPU interface signals follows (see table 3-2).

a. 1-MHz source clock signal (CL1S). The CPU sends a 1-MHz clock signal to the SIOP. The SIOP sends this signal to the device controllers on the device controller interface line clock line.

b. Control strobe (CNST). The CPU drives the control strobe line true during the execution of an I/O instruction to request service from an IOP. The control strobe line operates in a closed-loop manner with the proceed line; the CPU releases the control strobe line upon receiving the proceed signal. If the function code lines designate an SIO, HIO, TIO, or TDV operation and SIOP address recognition exists, or if the function code lines designate an AIO operation and the interrupt request line is true, the SIOP responds to the control strobe and does not pass it along to the next lower priority IOP. If an SIO, HIO, TIO, or TDV operation is designated and no IOP address recognition exists, or if an SIO operation is designated and the interrupt request line is false, the SIOP does not respond to the control strobe. Instead, it passes the control strobe along to the next lower priority IOP. If, however, the SIOP is designated as the last IOP on the tail-trunk cable by the state of a switch contact in the SIOP, the SIOP drives the proceed signal true and drives no other lines.

c. Function code lines FNC0 through FNC2. The CPU places a code representing the function code on the three function code lines when it makes a request to the SIOP for service. Signals on the function code lines are interpreted by the SIOP as follows:

Function Code Line	SIO	TIO	TDV	HIO	AIO
/FNC0/	0	0	0	0	1
/FNC1/	0	0	1	1	1
/FNC2/	0	1	0	1	0

d. Interrupt request signal (IR). All requests from device controllers are made to the CPU via the SIOP. When any device controller requests an interrupt, the SIOP 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.

Table 3-2. SIOP-CPU Interface Signals

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

e. IOP address lines IOPA0 through IOPA2. During execution of an SIO, TIO, TDV, or HIO instruction, the CPU places on the address lines the coded signals representing the address of the IOP from which service is requested. During execution of an AIO instruction, the address lines have no meaning.

f. Not condition code 1 and not condition code 2 lines. The SIOP may normally or partially execute the response to a CPU request for service. The SIO, TIO, or TDV instructions may either be normally or partially executed. The HIO and AIO instructions are always executed normally; this may involve prematurely terminating a device controller requested service by the SIOP. If during normal execution of an SIO, TIO, TDV, or HIO instruction the SIOP detects a memory parity error while reading memory location X'20', the SIOP does not communicate with any device controller. It also holds the NCOND1 and NCOND2 false. During normal execution of an SIO, TIO, TDV, or HIO instruction, if no memory parity error is detected while reading memory location X'20', the SIOP drives lines NCOND1 and NCOND2 with the same state that the device controller dictated on the data order request (DOR) and input/output request (IOR) lines, respectively. During execution of an AIO instruction, the SIOP drives the NCOND1 and NCOND2 lines as dictated by the device controller; however, if the responding device controller has the address stored by the SIOP and an unusual end interrupt is pending, the SIOP holds the NCOND2 line false. During partial execution of an instruction, the SIOP holds the NCOND1 line false and drives the NCOND2 line true. The SIOP does not send the proceed signal to the CPU until the condition code lines have settled, and does not release the condition code lines until the CPU has dropped the control strobe.

g. Proceed line (PR). In response to a CPU request for service on the control strobe line, the SIOP drives the proceed line to indicate termination of the service. The proceed line operates in a closed-loop manner with respect to the control strobe line; the SIOP raises the proceed line in response to the CPU raising the control strobe line, and drops the proceed line in response to the CPU dropping the control strobe line. The proceed line is driven at the times specified under 1 through 3 below for partial execution, normal execution, and no address recognition conditions:

1. Partial execution. During execution of an SIO, TIO, or TDV instruction, the SIOP responds to the control strobe by driving the NCOND2 and proceed lines true immediately.

2. Normal execution. If during an SIO, TIO, TDV, or HIO instruction the SIOP detects a memory parity error while accessing memory location X'20', the SIOP holds the NCOND1 and NCOND2 lines false and immediately drives the proceed line true. If during these instructions the SIOP does not write in memory location X'20' or X'21', the SIOP responds to the control strobe after receiving the device controller condition code response on the DOR and IOR lines. If the SIOP writes in memory locations X'20' and X'21', it drives the proceed line true after receiving the address release signal from core memory in response to the last memory request.

3. No address recognition. During the response to an SIO, TIO, TDV, or HIO instruction with no address recognition, or to an AIO instruction with the interrupt request line false, the SIOP does not respond to the control strobe; instead, it passes the control strobe along to the next lower priority IOP. However, if the SIOP is the last one on the cable interconnecting the IOPs to the CPU, the SIOP responds to the control strobe by holding both condition code lines false and driving the proceed line true immediately.

h. Reset I/O signal RIO. The CPU drives the reset I/O line true when the I/O system is to be initialized. In response to this signal, the SIOP is prepared for a service request from the CPU. In addition, the SIOP passes the reset I/O signal along to the device controllers on the I/O reset line.

3-18 SIOP-Device Controller Interface

All I/O devices controlled by the SIOP are connected to the SIOP by means of device controllers. All device controllers incorporate identical subcontroller sections; therefore, a common set of interface lines and signals serve all I/O devices. Except for the exchange of address, status, and condition code information during the five I/O instructions, most of the activity at the device controller interface occurs during the service cycles and immediately prior to the service cycles, when the device is requesting service. Generally, the type and number of service cycles depend upon the operation being performed, the number of bytes of data that are transferred, and the type of device with which communication is established.

3-19 SIOP-DEVICE CONTROLLER INTERFACE OPERATION. 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 SIOP, enters the busy condition, and drives its service call line true. The device is now started but not electrically connected to the SIOP; no data has been transferred between the device and core memory. In response to the service call from the device controller, the SIOP raises the acknowledge service call (ASC) line, delays, and then raises the function strobe (FS) line. A device controller indicates that it is connected for service by raising the function strobe acknowledge (FSL) line; otherwise, the lowest priority device controller drives the device controller available (AVO) line, and the service request is aborted.

A device controller is connected for service when the service connect flip-flop in the subcontroller section of the 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 is always an order out. The order out service cycle results in an order being transmitted from the SIOP to the device controller so that it will know what operation to perform. If the I/O operation involves transfer of data, the next service cycle will be a data out or data in, depending upon whether the device is to perform a write operation or a read operation.

After being started by an SIO instruction and connected to the SIOP for service during the ASC communication cycle, the device controller initially begins the service cycle by raising the request strobe (RS) line and coding the input/output request and data/order request (IOR and DOR) lines. The device controller specifies the required service cycle by appropriately coding these lines. Receipt of the RS signal permits the SIOP to strobe the IOR and DOR lines to determine the type of service cycle the device controller has specified. If an order in service cycle is specified, the SIOP strobes the operational status byte on the data lines (DA0 through DA7). If a data in service cycle is specified, the SIOP strobes the appropriate data lines (see paragraph 3-25 a, b, and c) for the input data. If an order out service cycle is specified, the SIOP accesses memory for the command doubleword and places the order from the first word on data lines DA0 through DA7. If a data out service cycle is specified, the SIOP accesses memory for the data and places the data on the appropriate data lines. The states of the end data and end service lines specify how the service terminates. The SIOP controls the end service line so that a terminal order terminates every device controller service request. Paragraphs 3-20 through 3-24 describe the action of the SIOP in response to the specific types of service cycles (order in, data in, order out, and data out) and a terminal order.

3-20 Order Out Service Cycle. An order out service cycle is specified by the device controller by raising the DOR and IOR lines. When the SIOP senses signal RS true, it performs the operations detailed under a through e below:

a. The SIOP examines the state of the chaining modifier bit. If it is true, the SIOP increments the current command doubleword address by one and clears the chaining modifier bit.

b. The SIOP reads the core memory location specified by the command doubleword address. If a memory address error is detected, the SIOP sets the memory address error and IOP halt status bits. If a memory parity error is detected, the SIOP sets the IOP memory error and IOP halt status bits. In either case, the SIOP responds immediately to the initial RS signal by raising signal RSA and reports IOP halt (possibly in addition to other information) in the ensuing terminal order.

c. If a memory address or memory parity error is not detected, the SIOP operates on the first (even) command word as follows (see figure 3-8).

If bits 4 through 7 are coded as shown in figure 3-8, the first word is interpreted by the SIOP as a transfer-in-channel command. Then the SIOP ignores bits 0 through 3 and 8 through 15 and stores the command doubleword address, bits 16 through 31, in the command address counter. If this is the second successive transfer-in-channel command detected by the SIOP, it sets the IOP control error and IOP halt status bits and immediately responds to the initial RS signal by raising RSA and by at least reporting IOP halt in the ensuing terminal order.

If the command accessed from memory is not a transfer-in-channel and a memory address error or memory parity error is not detected, the SIOP interprets the first word of the command doubleword (figure 3-9) as described in the following paragraphs and performs the following operations.

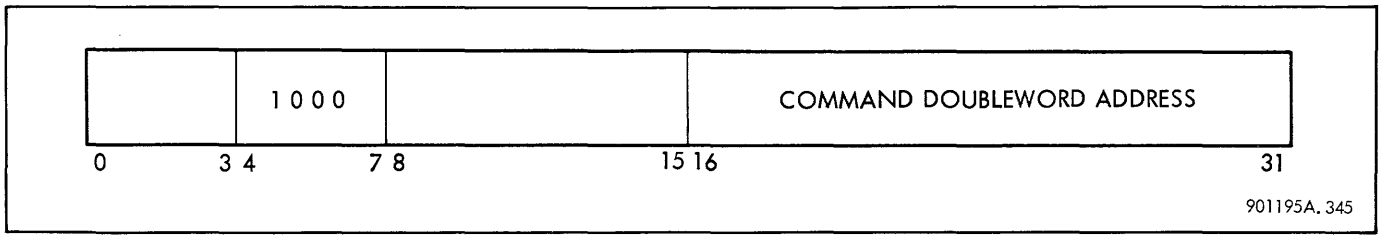


Figure 3-8. Transfer-in-Channel Command Format

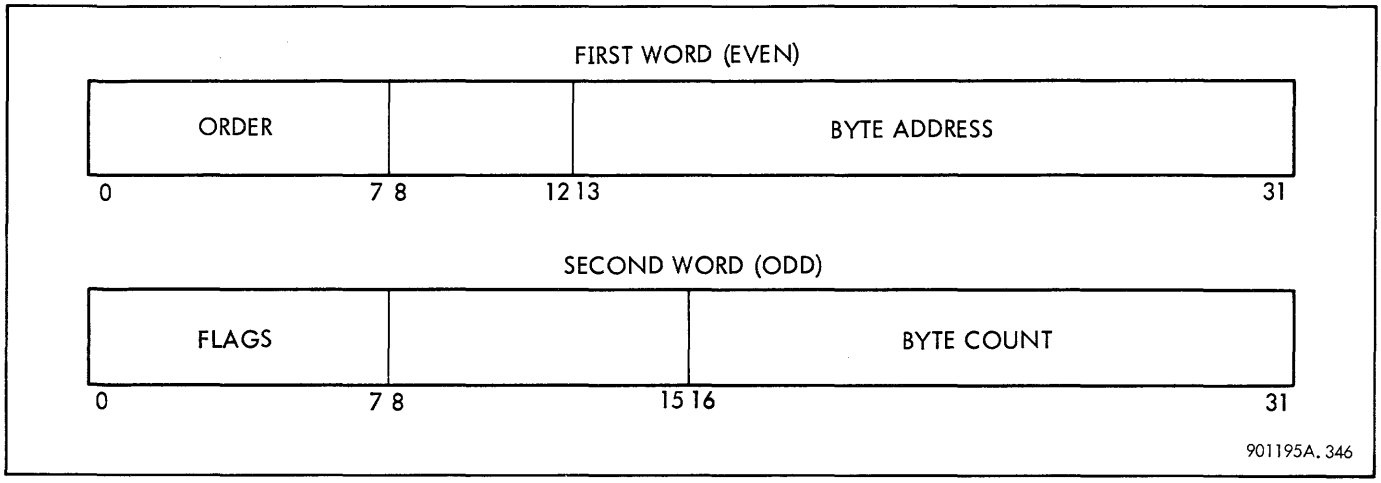


Figure 3-9. Command Doubleword Format

1. The SIOP transfers the order (bits 0 through 7) unchanged to the device controller via data lines DA0 through DA7, delays, and then raises RSA.

2. The SIOP examines the order to determine whether the read backward order (XXXX1100) has been coded. If it has, the SIOP stores this state as a control function for use during data in service cycles.

3. The SIOP stores the byte address, bits 13 through 31, in the BA counter.

d. The SIOP increments the even command word address by one, accesses core memory for the odd command word, and again increments the command address register (containing the command doubleword address) by one. The SIOP interprets the odd command word as shown in figure 3-9. Bits 0 through 7 contain flags, which are stored in the flag register, and bits 16 through 31 contain the byte count stored in the byte count register. Bits 8 through 15 are ignored by the SIOP. The SIOP response to the flags is described under 1 through 8 below:

1. Bit 0 - data chain flag (DC). During a data in or data out service cycle, the SIOP accesses the next command doubleword when the byte count goes to zero if the data chain flag is true. During the data chaining operation, the SIOP performs all operations prescribed by the order out service cycle except that it does not send the order to the device controller. During a data in service cycle, if the fast device controller line (FAST) is false and a zero byte count interrupt has been requested, the SIOP sets the end data flip-flop, empties the data buffer, and encodes at least the interrupt bit of the ensuing terminal order. During a data out service cycle, if the fast line is false and a zero byte count interrupt has been requested, the SIOP sets the end data flip-flop, and encodes at least the interrupt bit of the ensuing terminal order.

2. Bit 1 - interrupt on zero byte count flag (IZC). During a data in or data out service cycle, if the byte count has decreased to zero and the IZC flag is true, the SIOP sets the zero byte count interrupt status bit and drives the ZBCI line true at the device controller interface. If the data chain flag is true and the FAST line is false, the SIOP terminates the service cycle and reports at least interrupt in the ensuing terminal order. Otherwise, the SIOP terminates normally and reports interrupt in the terminal order.

3. Bit 2 - command chain flag (CC). During each terminal order in which the command chain flag is true, the SIOP reports command chain.

4. Bit 3 - interrupt at channel end flag (ICE). During an order in service cycle, if the device controller reports channel end and the ICE flag is true, the SIOP sets the channel end interrupt status bit and reports interrupt in the ensuing terminal order.

5. Bit 4 - halt on transmission error flag (HTE). During an order in service cycle, if the HTE flag is true, the device controller reports a transmission error or incorrect length, and the suppress incorrect length flag is false, the SIOP sets the IOP halt status bit and reports IOP halt in the ensuing terminal order. During a data in service cycle, if the HTE flag is true and the SIOP detects a transmission data error (a byte parity error at device controller interface), the SIOP sets the IOP halt status bit and reports IOP halt in the terminal order at the next termination. During data in and data out service cycles, if the HTE flag is true and the SIOP detects a transmission memory error (memory parity error during read or partial write for a data exchange), the SIOP sets the IOP halt status bit and reports IOP halt in the terminal order at the next termination.

6. Bit 5 - interrupt on unusual end flag (IUE). During an order in service cycle, if the device controller reports unusual end or if the IOP halt state exists and the IUE flag is true, the SIOP sets the unusual end interrupt status bit and reports at least interrupt in the ensuing terminal order.

7. Bit 6 - suppress incorrect length flag (SIL). During an order in service cycle, if the SIL flag is false, the HTE flag is true, and the incorrect length status bit has been set, the SIOP sets the IOP halt status bit and reports at least IOP halt in the ensuing terminal order. The incorrect length status bit may be set during an order in service cycle if the device controller reported incorrect length, or during a data in service cycle if the data chain flag is false, zero byte count is detected, and the SIOP data buffer is not empty.

8. Bit 7 - skip flag (S). During data in and data out service cycles, the SIOP performs all functions specified except core memory accesses if the skip flag is true.

e. After operating on the command doubleword as specified under paragraph 3-20 d above, the SIOP waits for the terminal order request strobe from the device controller. After receipt of the request strobe, the SIOP drives the end service line true, places the terminal order on data lines DA0 through DA7, delays, and then raises RSA to terminate the order out service cycle. (See paragraph 3-24 for a definition of information exchanged during a terminal order.)

3-21 Order In Service Cycle. An order in service cycle is specified by the device controller by holding the IOR line false and raising the DOR line with the initial request strobe. The functions performed by the SIOP during an order in service cycle are described under a and b below:

a. The SIOP strobes the end data line and data lines DA0 through DA7. Sensing that the device controller has raised the end data line, the SIOP also drives the end data line. SIOP response to each bit of the operational status byte on the data lines is presented under 1 through 5 below (data lines DA5 through DA7 convey no information):

1. Data line DA0 - transmission error. If this line is true, the SIOP sets the transmission data error status bit. If the HTE flag is true, the SIOP also sets the IOP halt status bit and reports IOP halt in the ensuing terminal order.

2. Data line DA1 - incorrect length. If this line is true, the SIOP sets the incorrect length status bit. If the SIL flag is false and the HTE flag is true, the SIOP also sets the IOP halt status bit and reports IOP halt in the ensuing terminal order.

3. Data line DA2 - chaining modifier. If this line is true, the SIOP sets the chaining modifier control flip-flop. During the next order out service cycle, the SIOP increments the command doubleword address in the command address register by one and clears the chaining modifier control flip-flop. The SIOP also clears the chaining modifier control flip-flop during a successful SIO operation.

4. Data line DA3 - channel end. If this line is true and the interrupt on channel end flag is true, the SIOP sets the channel end interrupt status bit and signals interrupt in the ensuing terminal order. If the command chain flag is false or if the IOP halt status bit is true, the SIOP also clears the IOP busy status bit.

5. Data line DA4 - unusual end. If this line is true and the interrupt at unusual end flag is true, the SIOP sets the unusual end interrupt status bit and signals interrupt in the ensuing terminal order. The SIOP also clears the IOP busy status bit.

b. The SIOP delays after raising the end data line and then raises the RSA line. After the device controller receives signal RSA, it drives the RS line for the terminal order. When the SIOP receives the terminal order request, it places the terminal order on data lines DA0 through DA7, raises the end service line, delays, and then raises RSA to terminate the order in service cycle.

3-22 Data Out Service Cycle. A data out service cycle is specified by the device controller by holding the DOR line false and raising the IOR line with the initial request strobe. The functions performed by the SIOP during a data out service cycle are described under a through c below:

a. The SIOP accesses the core memory address specified by the byte address counter, and loads the data into the data buffer. The SIOP increments the byte address by one, two, three, or four and decrements the byte count by one, two, three, or four as required. This cycle is repeated until either the byte count goes to zero or the end data flip-flop is set. If the byte count goes to zero and the data chaining flag is true, the SIOP performs the data chaining process. If the data chaining flag is not true when the byte count goes to zero, the SIOP terminates the data out service with a terminal order in which count done is reported.

b. At the device controller interface, the SIOP accesses the data buffer for the data, aligns the data as required by the interface data path width and the state of the byte address and byte count counters, places the data on the data lines, generates the parity bit (DAP), delays, and raises the RSA line.

c. The SIOP performs the operations specified in a and b above until the end data line has been raised. The SIOP then waits for the terminal order request strobe, places the terminal order on data lines DA0 through DA7, raises the end service line, delays, and raises the RSA line to terminate the data out service cycle.

3-23 Data In Service Cycle. A data in service cycle is specified by the device controller by holding both the IOR and DOR lines false with the initial request strobe. The functions performed by the SIOP during a data in service cycle are described under a through c below:

a. After each request strobe received from the device controller, the SIOP delays, strobes the data lines (except during the terminal order), checks parity if required, loads the data into the data buffer, and drives the RSA line true. The SIOP repeats these operations at the device controller interface until the end data line has been raised.

b. The SIOP performs the three following functions at the core memory interface:

1. It accesses the data buffer register for the data.

2. It aligns the data in the memory assembly register as required by the current state of the byte count and byte address.

3. It requests core memory full write or partial write operations.

During each core memory interface cycle the SIOP increments the byte address counter as required (or decrements it if a read backwards order was sent to the device controller), and decrements the byte count counter as required. The SIOP repeats these operations until a memory address error occurs, the byte count goes to zero, or the end data line is driven. (An AIO or HIO causes the SIOP to drive the end data line.)

If a memory address error occurs, the SIOP terminates the service cycle by driving the end data line to the device controller. If the byte count goes to zero and the data chaining flag is true, the SIOP performs the chaining operation and then continues to exchange data between the data buffer register and core memory. If the byte count goes to zero and the data chain flag is false, the SIOP drives the end data line to the device controller and terminates. If the device controller drives the end data line, the SIOP empties the data buffer register (except if a memory address error was detected or zero byte count occurred and the data chain flag is false), and terminates the data in service cycle. Incorrect length may be signalled if data is left in the buffer at count done.

c. The SIOP performs the operations specified in a and b above until end data and RSA have been signaled. The SIOP then waits for the terminal order request strobe, places the terminal order on data lines DA0 through DA7, raises the end data line, delays, and raises the RSA line.

3-24 Terminal Order. The SIOP terminates each device controller service sequence with a terminal order exchange. The terminal order is transmitted to the device controller on data lines DA0 through DA3. The conditions under which each of these lines is driven true are described under a through d below:

a. Data line DA0 - interrupt. The SIOP drives this line true to cause the device controller to drive its interrupt call line (IC) true at the conclusion of the terminal order. The SIOP drives the interrupt line (DA0) during the terminal order under any of the following four conditions:

1. If, during an order in service cycle, the interrupt at channel end flag was true and the device controller signaled channel end

2. If, during an order in service cycle, the interrupt at unusual end flag was true and the device controller signalled in unusual end or the IOP halt status bit was true

3. If, during a data out service cycle, the byte count goes to zero and the data chain and interrupt at zero byte count flags are true, or if count done is specified in the terminal order and the interrupt at zero byte count flag is true

4. If, during a data in service cycle, the byte count goes to zero and the interrupt at zero byte count flag is true

b. Data line DA1 - count done. The SIOP specifies count done via the terminal order when the byte count reaches zero under the following two conditions:

1. If, during a data out service cycle, the byte count has gone to zero, the data chain flag was false, and the SIOP had transmitted all data in the data buffer register to the device controller

2. If, during a data in service cycle, the byte count has gone to zero and the data chain flag was false

c. Data line DA2 - command chain. The SIOP drives this line true during every terminal order in which the command chain flag is true.

d. Data line DA3 - IOP halt. The SIOP drives this line true during every terminal order in which the IOP halt status bit is true.

3-25 SIOP-DEVICE CONTROLLER INTERFACE SIGNALS. A brief description of the SIOP-device controller interface signals follows (see table 3-3):

a. Data lines DA0 through DA7. Data lines are driven by either the SIOP or the device controller. Lines DA0 through DA7 are multipurpose lines. Their uses during the various communications with the device controller are described under 1 through 7 below:

1. During normal execution of an SIO, HIO, TIO, or TDV instruction, the SIOP places the device controller address on data lines DA0 through DA7, delays 185 ns, and drives the FS line. The SIOP does not change the state of DA0 through DA7 until either signal FSL or AVO has been received.

2. During execution of an AIO instruction, the device controller places the device controller status on lines DA0 through DA7. The SIOP delays 185 ns after receiving signal FSL and then strobes the lines. The device controller does not change the state of the data lines until the SIOP drops signal FS.

3. During an order out service cycle, in response to the initial RS, the SIOP places the order on data lines DA0 through DA7, delays 185 ns, and drives the RS line true. The SIOP does not change the state of the data lines until the next RS is received from the device controller.

4. During an order in service cycle, the device controller places the operational status byte on data lines DA0 through DA7 when it issues the initial RS. The SIOP delays 185 ns after receiving signal RS, then strobes the data lines. The SIOP interprets the data lines as follows:

Data Line	Meaning
DA0	Transmission error
DA1	Incorrect length
DA2	Chaining modifier
DA3	Channel end
DA4	Unusual end
DA5-DA7	None

The device controller does not change the state of the data lines until it receives signal RSA from the SIOP.

5. During a data out service cycle, if ES is false, the SIOP places the data on data lines DA0 through DA7, delays 185 ns, and drives the RSA line. The SIOP does not change the state of the data lines until the next RS signal is received.

6. During a data in service cycle, if ES is false, the device controller places the data on data lines DA0 through DA7 and drives line RS. The SIOP delays 185 ns after receiving RS, and then strobes the lines. The device controller does not change the state of the data lines until it receives signal RSA from the SIOP.

7. In response to the final RS of any service cycle, the SIOP raises ES, places the terminal order on data lines DA0 through DA7, delays 185 ns, and drives the RSA line. The device controller interprets the information on the data lines as follows:

Data Line	Meaning
DA0	Interrupt
DA1	Count done
DA2	Command chain
DA3	IOP halt
DA4-DA7	None

The SIOP does not change the state of the data lines until they are again needed during the next communication with the device controller.

b. Data parity signal (DAP). The data parity line is used to provide an odd parity bit for the data byte on data lines DA0 through DA7 during data in and data out service cycles. If the PC line is true and the DX2 and DX4 lines are false, the SIOP checks for odd parity during a data in service cycle. If a parity error is detected during a data in service cycle, the SIOP sets the transmission data error status bit. Timing for the DAP line is the same as the timing for data lines DA0 through DA7 during data in and data out service cycles.

c. Data signals DB0 through DB7, DC0 through DC7, and DD0 through DD7. The lines for these signals are used only during the exchange of data during data in and data out service cycles. Timing for these lines is the same as that for data lines DA0 through DA7.

d. Two- and four-byte interface control lines DX2 and DX4. These lines are controlled by the device controller during data in and data out service cycles to define the width of the data interface. Either one, two, or four bytes may be exchanged. If lines DX2 and DX4 are both false, the SIOP exchanges data with the device controller on data lines DA0 through DA7 only. If line DX2 is true, the SIOP exchanges data with the device controller on data lines DA0 through DA7 and DB0 through DB7; lines DA0 through DA7 carry the most significant byte. If line DX4 is true, the SIOP exchanges data with the device controller on data

Table 3-3. SIOP-Device Controller Interface Signals

SIGNAL	DESIGNATOR	NO. OF LINES	DIRECTION OF SIGNAL FLOW	
			SIOP to D/C*	D/C* to SIOP
Data input/output (1st byte)	/DA0/-/DA7/	8	X	X
Data input/output (2nd byte)	/DB0/-/DB7/	8	X	X
Data input/output (3rd and 4th bytes)	/DC0/-/DC7/, /DD0/-/DD7/	16	X	X
Data parity	/DAP/	1	X	X
2-byte interface control	/DX2/	1		X
4-byte interface control	/DX4/	1		X
End data	/ED/	1	X	X
End service	/ES/	1	X	
Parity check	/PC/	1		X
Data/order request	/DOR/	1		X
Input/output request	/IOR/	1		X
Function response	/FR0/-/FR7/	8		X
Request strobe	/RS/	1		X
Request strobe acknowledge	/RSA/	1	X	
Function strobe	/FS/	1	X	
Function strobe acknowledge	/FSL/	1		X
Device controller available	/AVO/	1		X
AVI from IOP	/TRUE-1/, /TRUE-2/	2	X	
Service call	/SC/	1		X
Interrupt call	/IC/	1		X
I/O reset	/RST/	1	X	
Reset I/O	/TEST RIO/	1		X
Clock, 1 MHz	/CL1/	1	X	
Zero byte count interrupt	/ZBCI/	1	X	
Fast device controller	/FAST/	1		X
Start I/O function indicator	/SIO/	1	X	
Halt I/O function indicator	/HIO/	1	X	
Test I/O function indicator	/TIO/	1	X	
Test device function indicator	/TDV/	1	X	
Acknowledge interrupt function indicator	/AIO/	1	X	
Acknowledge service call indicator	/ASC/	1	X	

*D/C = Device controller

lines DA0 through DA7, DB0 through DB7, DC0 through DC7, and DD0 through DD7; lines DA0 through DA7 carry the most significant byte, and lines DD0 through DD7 carry the least significant byte.

e. End data signal (ED). The ED line designates the action taken by the SIOP and the device controller during all four service cycles. The ED line is controlled by either the SIOP or the device controller. It is always driven true by the device controller with the initial RS of an order in or order out service cycle. If the ED line is driven true by either the SIOP or the device controller during a data in or data out service cycle, no more data is exchanged. The ED line is driven true by the SIOP under the following four conditions:

1. When the ED line has been driven true by the device controller

2. During a data in or data out service cycle if the CPU requests the SIOP to perform an HIO operation, or if the CPU requests that an AIO operation be performed and the SIOP has an interrupt pending

3. During a data out service cycle if one or more of the following conditions occur:

(a) The SIOP detects an IOP halt condition.

(b) The SIOP detects zero byte count, the data buffer is empty, and the data chain flag is false (count done).

(c) The SIOP detects zero byte count, the data chain flag is true, the interrupt on zero byte count flag is true, and the FAST line is false.

4. During a data in service cycle if one or more of the following conditions occur:

(a) The SIOP detects a memory address error.

(b) The SIOP detects zero byte count and the data chain flag is false (count done).

(c) The SIOP detects zero byte count, the data chain flag is true, the interrupt on zero byte count flag is true, and the FAST line is false.

During a data in service cycle, the SIOP always empties the data buffer register after ED is set but before the terminal order is issued, unless ED was set because an IOP halt condition was detected. When the SIOP drives the ED line, it delays at least 185 ns before driving the RSA line.

f. End service signal (ES). The ES line is driven by the SIOP only. The SIOP drives the ES line when a service cycle is to be terminated. When the device controller senses ES true, it does not issue another request strobe during the current service cycle. The SIOP controls the ES line so that a terminal order always concludes each of the four service cycles. The timing of the ES line with respect to the RSA line is the same as the timing of data lines DA0 through DA7 during the terminal order exchange.

g. Parity check signal (PC). The PC line is driven by device controllers in which its use has been specified. The SIOP, however, responds to the PC line only if lines DX2 and DX4 are false. If the PC line is true under this condition, the SIOP performs an odd parity check on data lines DA0 through DA7 and the data parity line. The PC line, if driven, is raised with the initial RS of a data in service cycle and is not dropped until RSA and ES have been signalled by the SIOP.

h. Data/order request (DOR) and input/output request (IOR) signals. These lines are multipurpose lines that are controlled by the device controller. During the normal execution of a CPU request for service, they carry condition code information as described under 1 below. During a service cycle, they define the type of service cycle as described under 2 below.

1. During normal execution of a CPU service request, if the SIOP drives the FS line true, the device controller places condition code information on the DOR and IOR lines. The SIOP waits until it senses the FSL line true, delays a minimum of 185 ns, and then strobes the DOR and IOR lines. The information obtained from these lines is stored by the SIOP for later transmission to the CPU on the NCOND1 and NCOND2 lines. The device controller does not change the state of the DOR and IOR lines until the SIOP drops the FS line.

2. During a device controller service request that is not aborted, the device controller connects for service at the time the SIOP drops the FS line. The device controller then drives the DOR and IOR lines with coded signals, drives the data lines if IOR is false, and issues the initial request strobe. The SIOP delays a minimum of 185 ns after receiving the initial request strobe of the service cycle and then strobes the IOR and DOR lines. The SIOP interprets the state of these lines as follows:

IOR	DOR	Type of Service Cycle Defined
1	1	Order out
1	0	Order in
0	1	Data out
0	0	Data in

The device controller does not change the state of these lines until it receives the initial request strobe acknowledge signal from the SIOP. After strobing the DOR and IOR lines to determine the type of service cycle, the SIOP performs the functions defined by that service cycle (see paragraphs 3-20 and 3-23).

i. Function response signals FR0 through FR7. The function response lines are driven only by the device controller. During the normal execution of an SIO, TIO, TDV, or HIO instruction, the device controller supplies device controller status on these lines. During an AIO instruction, the device controller places its address on these lines. The SIOP ignores the state of the function response lines during the service cycles and during the ASC communication cycle. The SIOP delays a minimum of 185 ns after the FSL line has been driven true and then strobes the function response lines for later transmission of the information on the lines to core memory. The device controller does not change the state of the function response lines until the SIOP has dropped the FS line.

j. Request strobe (RS) and request strobe acknowledge (RSA) signals. The RS line is driven only by the device controller, and the RSA line is driven only by the SIOP. These lines operate in a closed-loop manner (that is, RS is applied by the device controller until the acknowledge signal RSA is received). Then the acknowledge signal remains applied until RS is removed, at which time the acknowledge signal is removed. The device controller, after connecting for service, sends a succession of RS signals to the SIOP until the SIOP drives the end service line in conjunction with the RSA line. The SIOP, after receiving the initial RS of a service cycle, delays a minimum of 185 ns and then strobes the IOR and DOR lines to determine the type of service requested by the device controller. If the service is either an order out or data out service cycle, the SIOP immediately accesses core memory for the order or data, places it on the data lines, delays a minimum of 185 ns and raises the RSA lines. This sequence continues until either the SIOP or the device controller signals end data. The SIOP then waits for the final RS for the terminal order, places the terminal order on the data lines, drives the end service line true, delays at least 185 ns, and then drives the RSA line to terminate the service cycle.

k. Function strobe (FS), function strobe acknowledge (FSL), and device controller available signals (AVO). The function strobe line operates in a closed-loop manner with either the FSL or the AVO line. After the SIOP has driven the FS line true, the device controller drives either the FSL or the AVO line, and drops the FSL or AVO line in response to the SIOP dropping the FS line. During normal execution of an SIO, HIO, TIO, or TDV instruction (if no memory parity error is detected while reading memory location X'20'), the SIOP places the device controller address on data lines DA0 through DA7, drives the appropriate function indicator line, and drives the FS line.

If the device controller address is not recognized, or if the addressed device controller is off-line, the SIOP will receive signal AVO in response to the FS signal. Otherwise, the addressed device controller responds by placing the condition code information on the DOR and IOR lines, and the device controller status on the function response lines, and driving the FSL line. During execution of an AIO instruction, the SIOP drives the AIO function indicator line, delays, and then drives the FS line. If there is no device controller interrupt pending, the SIOP receives signal AVO in response to signal FS. Otherwise, the responding device controller places its address on the function response lines, condition code information on the IOR and DOR lines, and the device controller status on data lines DA0 through DA7, and then drives line FSL true.

In response to a device controller request for service on the SC line, the SIOP drives the ASC line, delays, and then drives the FS line. If the SC line has dropped prior to the time that the SIOP rasises the ASC line, the SIOP receives a response on the AVO line.

l. AVI from IOP signals (TRUE-1 and TRUE-2). These two lines are driven by the SIOP when the SIOP logic power supply is on. They provide the AVI signal to the highest priority device controller connected to the SIOP.

m. Service call signal (SC). The SC line is driven by the device controller to request service from the SIOP. The SIOP responds to the request by initiating one of the four service cycles or an aborted service request. The device controller drives the SC line at least until the SIOP has driven the ASC function indicator line; otherwise, the SIOP aborts the service request and waits for the next service request from either the CPU or a device controller. CPU requests have priority over device controller service requests.

n. Interrupt call signal (IC). This line is driven by device controllers to make an I/O interrupt request to the CPU via the SIOP. The SIOP stores the interrupt request whenever the IC line is true if the SIOP is not currently engaged in processing its part of an AIO instruction. The SIOP drives the interrupt request line (IR) at the CPU interface as long as the interrupt request is stored in the SIOP. The SIOP clears the interrupt request when the reset I/O line is driven, the CPU is executing an AIO instruction, or an HIO instruction is directed to the device controller whose address is stored in the SIOP.

o. I/O reset signal (RST). The I/O reset line is driven by the SIOP. It has the same state as the RIO line driven by the CPU or the TESTRIO line driven at the device controller interface by the JX58 tester when the tester is connected.

p. Reset I/O signal (TESTRIO). The TESTRIO line is driven at the device controller interface by the JX58 tester when the tester is connected to the SIOP. The TESTRIO signal causes the same SIOP response as does the RIO signal generated by the CPU (that is, the I/O system is initialized).

q. 1 MHz clock signal (CL1). Signal CL1S is received from the CPU and passed along to the device controllers as signal CL1; it is not used by the SIOP.

r. Zero byte count interrupt (ZBCI) and fast device controller (FAST) signals. The FAST line is driven by device controller in which its use has been specified during data in and data out service cycles. It is raised with the initial RS of a data in or data out service cycle and is not dropped until RSA and ES have been driven by the SIOP. If both the FAST line and the data chain flag are true, the SIOP does not terminate the service cycle to send a zero byte count interrupt to the device controller via a terminal order. Instead, the SIOP drives the ZBCI line. This line has the same state as the zero byte count interrupt status bit. If the device controller drives the FAST line, it also reacts to the ZBCI line when it goes true during a data in or data out service cycle. The action taken by the device controller when the ZBCI line is true is the same as when the SIOP specifies an interrupt (data line DA0 true) during a terminal order exchange.

s. Function indicator signals SIO, TIO, TDV, HIO, AIO, and ASC. During normal execution of a CPU request to perform an SIO, HIO, TIO, TDV, or AIO operation, the SIOP drives the appropriate function indicator line. Similarly, in response to a device controller request for service via the SC line, the SIOP drives the ASC line. The driven line is held stable for at least 185 ns before the function strobe line is driven, and is not dropped until at least 185 ns after the function strobe line is dropped.

3-26 TYPICAL I/O OPERATION

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 3-10 shows the sequence of events for such an operation and figure 3-11 is the related timing diagram. This operation is one in which data is transferred from core memory to the device and there are no unusual conditions.

If the SIOP is busy, the SIO instruction is partially executed – that is, core memory location X'20' is not accessed, but the SIOP sends condition code information to the CPU to inform the CPU of the busy condition. If the SIOP is not busy, the SIO instruction is executed normally. In this case, the SIOP accesses core memory location X'20' for the address of the device controller and the address of the first command doubleword (command) that the SIOP is to execute. The SIOP then drives the SIOP function indicator line, places the address of the device controller on the data lines, drives the FS line, and waits for a response from the addressed device controller. The addressed device controller responds by placing its status information on the function response lines (FR0 through FR7) and condition code information on the IOR and DOR lines, and by then driving the FSL line. The SIOP sends this information, in addition to other information stored in its own registers, to the CPU by way of core memory locations X'20' and X'21' (see paragraph 3-13).

Following this, the SIOP and the device controller enter the busy state, the SIO instruction is concluded, and the device controller drives its service call line true. The CPU is released from the I/O operation at the conclusion of the SIO instruction. The SIOP and the device controller function together to execute the program (the portion of the command list) associated with the current I/O operation. The operation may consist of one command (chaining not specified), or more than one command (chaining specified).

In response to the service call from the device controller, the SIOP drives the acknowledge service call line and function strobe line true, during which time the device is connected to the SIOP for service (the service connect flip-flop [FSC] in the device subcontroller is set). The device controller also places its address on the function response lines and driver the FSL line; however, the SIOP ignores the function response lines at this time, since the SIOP has only one set of registers (CA, BA, flags, and status). This procedure differs from that of the multiplexing IOP. The multiplexing IOP may be controlling several device controllers, all of which may be requesting service at the same time.

The first service cycle following an SIO is always defined by the device controller as an order out. An order out service cycle is defined when the device controller drives both the IOR and DOR lines true. The order out service cycle causes the SIOP to fetch a command doubleword from the command list in core memory. The address of this command was stored in the SIOP command address counter during execution of the SIO instruction. The SIOP stores the command doubleword (except the order bits) in the appropriate registers and counters. The order bits are sent to the device controller. The SIOP logic is such that a terminal order always concludes an order in or order out service cycle even though the SIOP may have nothing to report to the device controller. The device controller disconnects (the FSC flip-flop in the device controller is reset) from the SIOP at the conclusion of the terminal order.

After the device controller disconnects, it again drives its service call line true, and the SIOP again responds with an acknowledge service call signal. The service cycle following an order out is normally a data in or data out. For this example, the device controller specifies the service cycle as a data out by driving the IOR line true and holding the DOR line false. In response to the initial request strobe from the device controller, the SIOP starts accessing data from core memory starting with the address specified by the byte address. As data is read into the data buffer, the byte address is incremented and the byte count is decremented. With each request strobe from the device controller, the SIOP places one, two, or four bytes of data in parallel on the data lines for transfer to the device controller. The number of parallel bytes depends upon the

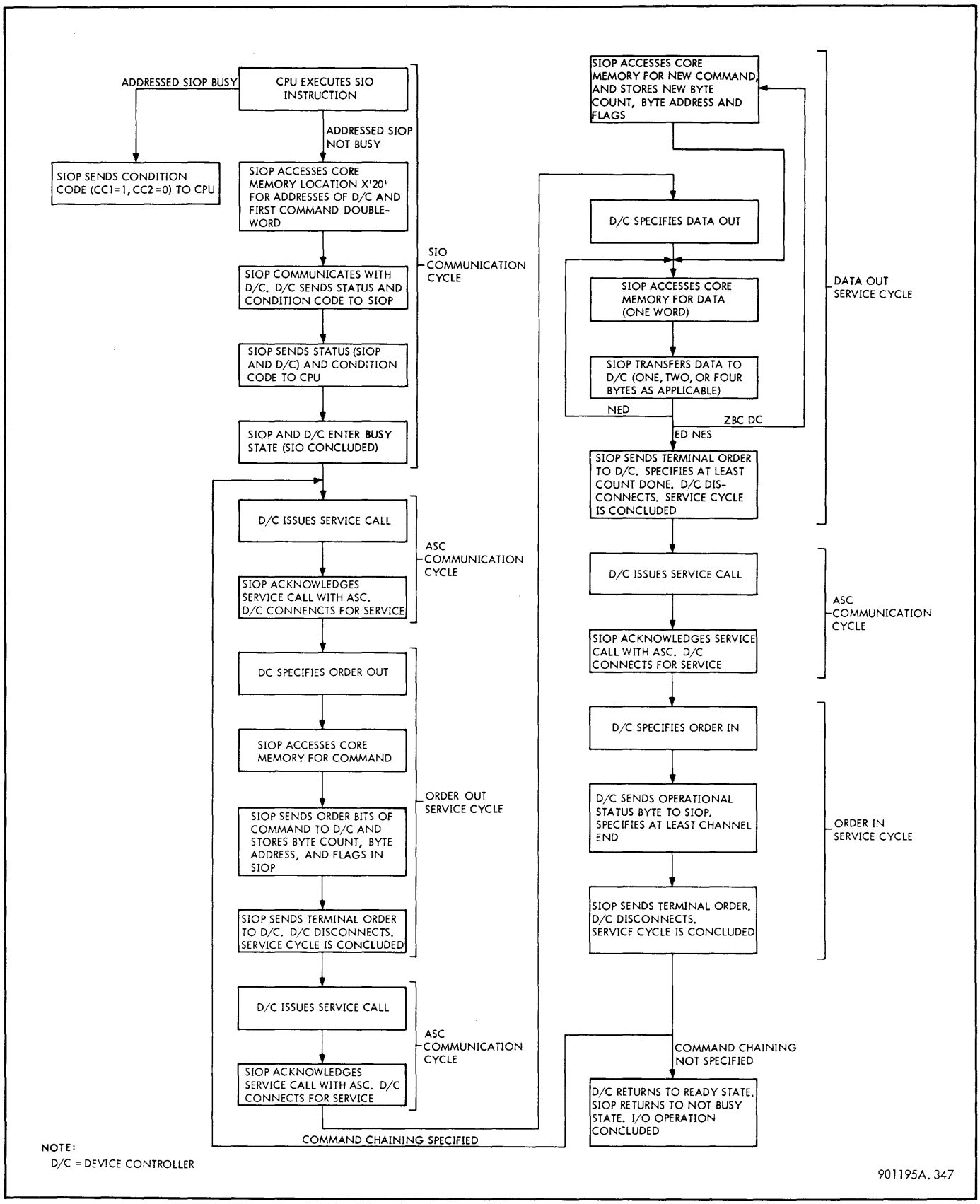


Figure 3-10. Typical I/O Operation, Sequence Diagram

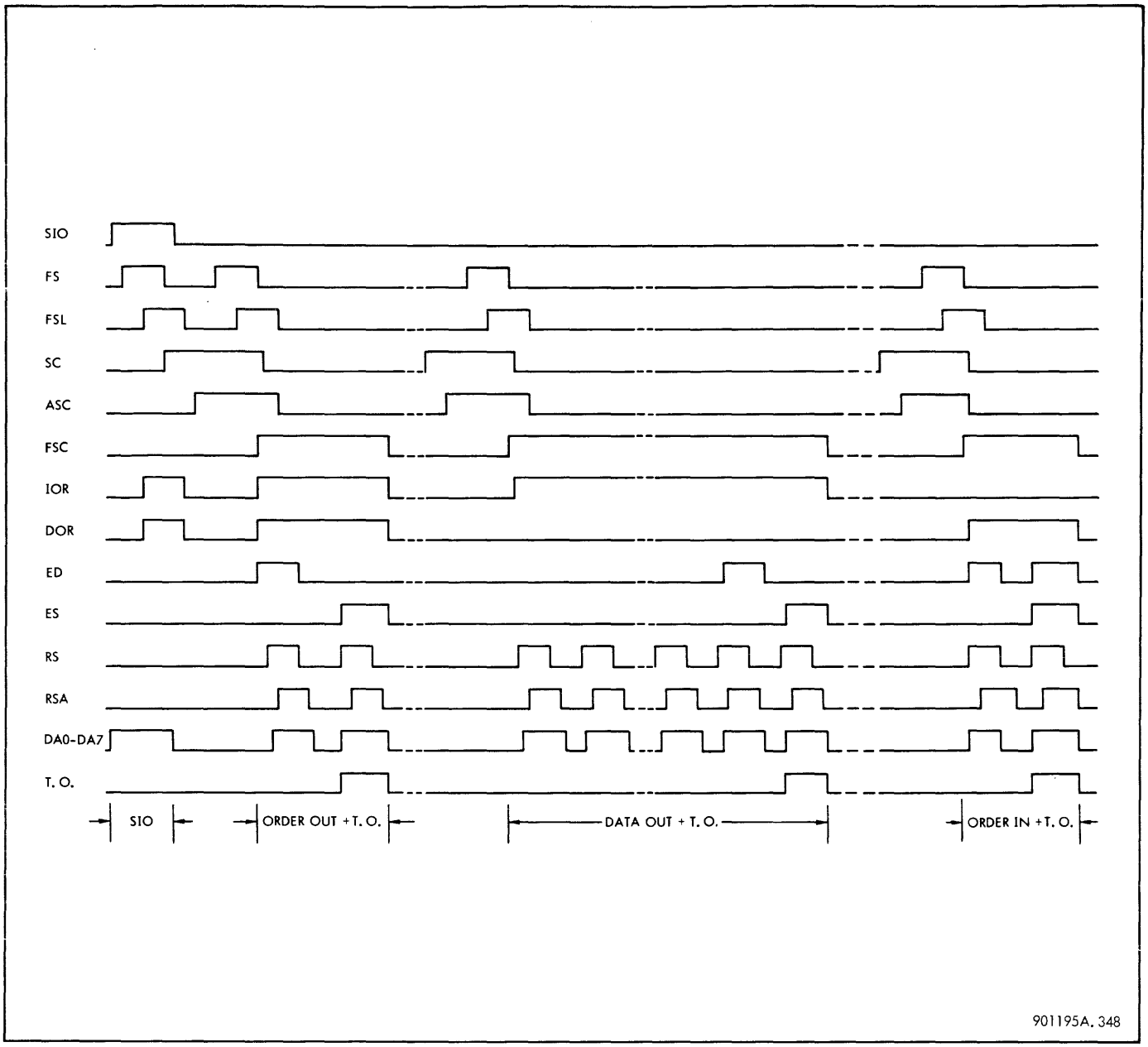


Figure 3-11. Typical I/O Operation, Timing Diagram

state of lines DX2 and DX4, which are controlled by the device controller.

The process of accessing data from core memory and transferring it to the device controller continues until the byte count goes to zero (ZBC) or end data (ED) is set. If the byte count goes to zero and the data chaining (DC) flag is set, the SIOP accesses core memory for the next command doubleword, and stores the new byte address, byte count, and flags into their appropriate registers. At the same time, the SIOP continues transmitting data from its data buffer, by way of the I/O register, to the device controller. After the last byte of data specified by the old command doubleword has been transferred to the device controller, the SIOP starts transferring data to the device controller that was specified by the new command doubleword. There is no interruption in the service to the device controller; it receives its byte (or parallel bytes) of data for each request strobe issued.

If the byte count goes to zero and the data chain flag is false, the SIOP terminates the service cycle with a terminal order in which count done is specified. When the SIOP wants to send a terminal order to the device controller, it drives the end data line true and holds the end service line false. The service connect flip-flop in the device controller does not reset and thus disconnect from the SIOP until it receives signal ES true.

The device controller responds to count done by requesting service, and when connected (via an ASC communication cycle), specifies an order in service cycle. At this time, the device controller reports channel end to the SIOP, and if there are any unusual conditions, these are also reported. (See paragraph 3-21.) If the device controller detects an error before all of the data has been exchanged, it will at that time drive the end data line true. This causes the SIOP to prematurely terminate the data service cycle so that the device controller can report the error condition via the ensuing order in service cycle.

The device controller inspects the contents of the terminal order following the order in service cycle to determine what action must be taken. (See paragraph 3-24 for a description of terminal orders.) If command chaining is specified in the terminal order following the order in service cycle, the device controller will again request service. After being connected, it will request another order from the SIOP via an order out cycle. The SIOP performs the same functions during this order out cycle as it did during the one following the SIO instruction. If command chaining is not specified via the terminal order, the device controller disconnects and then performs its final operations. These are operations that do not require the services of the SIOP. The conclusion of these activities is defined as device end. If the program is such that the CPU is to be notified when device end occurs, a request for an interrupt at device end may be encoded in the M bits of the original order the device controller received during the order out service cycle. If an interrupt is to occur for other reasons, it is specified during the terminal order following the order in service cycle.

3-27 TIMING SIGNAL GENERATION

Timing signals for all SIOP operations are derived from three delay lines and their associated phase latches. An additional delay line is used if the SIOP incorporates the bus-sharing option. This delay line is described under the heading of the memory bus-sharing option, paragraph 3-73. The triggering of a delay line starts a series of timing pulses. The combination of a particular timing signal generated by the delay line and the signal from a particular phase latch provides a unique timing signal. As the SIOP sequences through various phases, timing signals are generated for all SIOP logic operations. Since there are four delay lines, different types of operations may occur simultaneously. For example, the TI delay line may be providing timing signals for the transfer of data into the I/O register at the same time the TR delay line is providing signals for data buffer operation. During service sequences for the data operations, the TI, TM, and TR delay lines may run concurrently. Each delay line and its associated phase latches are described separately below.

3-28 FI Phases and TI Delay Line

Timing signals generated by the FI phases and TI delay line are used primarily to control operations at the device controller interface. The FI phases consist of FI01, FI11, FI12, and FI13. Two latches are used to define each phase. The latch with the letter A in its reference designator (first rank) is set first, and the latch with the letter B in its reference designator (second rank) is set next. A logic diagram of the TI delay line is shown in figure 3-12. Figure 3-13 includes, in addition to other associated signals, the timing signals generated by the TI delay line. The timing signals shown in figure 3-13 assume a zero ns delay through all circuits except the delay line.

A brief description of TI delay line operation follows. Assuming the delay line is empty (no pulse going down the line), signal TICYCLE, the output of a latch, is true. The latch was set either by the RESET signal or by signal TIS560, which was true the last time the delay line was triggered. Since signal TICYCLE is true, signal S/TI goes true when signal TIRQ goes true. When signal S/TI goes true, a voltage step is launched in the delay line. When the voltage wavefront reaches the 120 ns tap on the delay line, signal NTIS120 goes false and resets the latch. Signal TICYCLE, therefore, goes false and cuts off the input to the line. Cutting off the input 120 ns after it has started creates a 120 ns pulse that travels down the line. As the pulse passes each tap on the line, that tap is true for 120 ns.

3-29 FM Phases and TM Delay Line

Timing signals generated by the FM phases and TM delay line are used primarily to control operations at the memory interface. The FM phases consist of FM01, FM11, FM12, FM13, FM14, FM21, FM22, FM24, FM31 through FM34, and FM41. Similar to the FI phases, the FM phases are also defined by two latches each (the first and second rank latches). A logic diagram of the TM delay line is shown in figure 3-14. Timing signals associated with the TM delay line are shown in figure 3-15. The timing signals shown in figure 3-15 assume a zero ns delay through all circuits except the delay line.

Unlike the TI delay line, the TM line runs twice for each phase; once when the first rank phase latch is set, and again when the second rank phase latch is set. Signals FMA and FMB in figure 3-15 represent the first and second rank respectively of any of the phase latches. The TM line is triggered in a manner similar to the TI line. Signal TMCYCLE, the output of a latch, is true. The latch was set either by the RESET signal or by signal TMS280, which was true the last time the delay line was triggered. Signal S/TM goes true and triggers the delay line when either signal TMRQA or TMRQB goes true.

The logic for TMRQB is such that the delay line must have been triggered first during an FM phase by signal TMRQA. When the voltage wavefront reaches the 80 ns tap on the delay line, signal NTMS080 goes false and resets the latch. The pulse that travels down the delay line therefore has a duration of 80 ns. The delay line cannot be triggered the second time until the first pulse reaches the 280 ns delay line tap. It is therefore possible for two pulses with wave

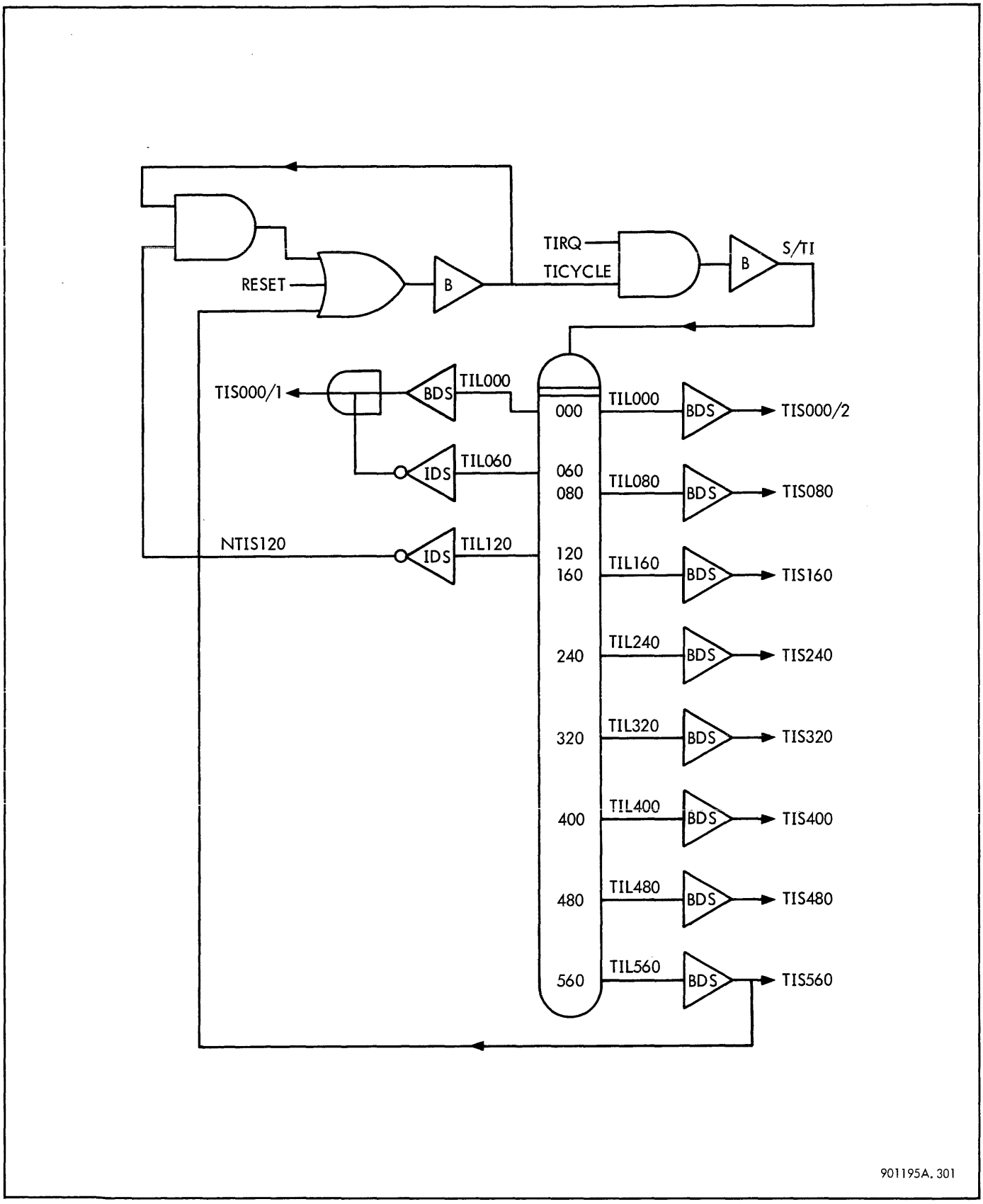


Figure 3-12. TI Delay Line Logic Diagram

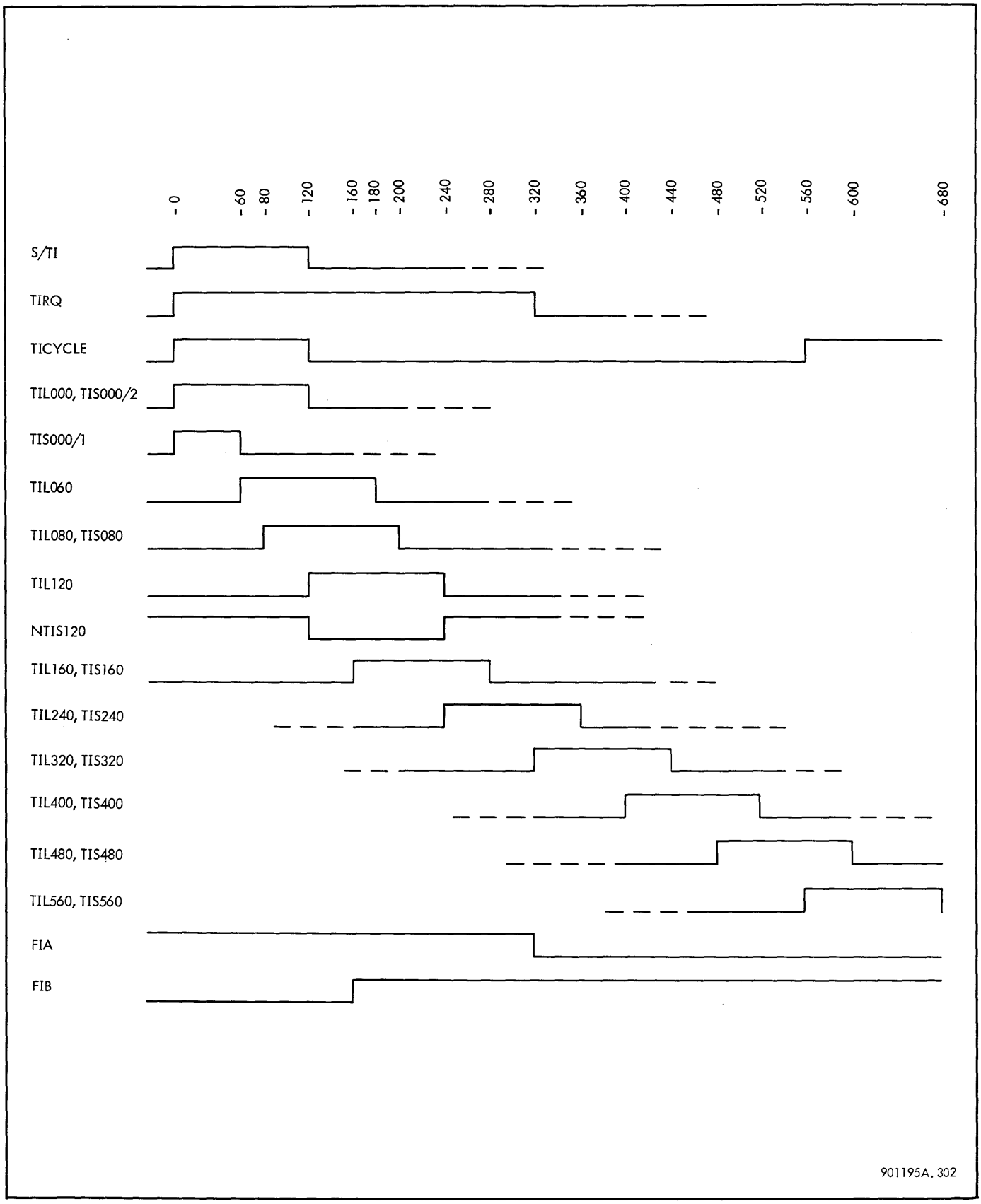


Figure 3-13. TI Delay Line Timing Diagram

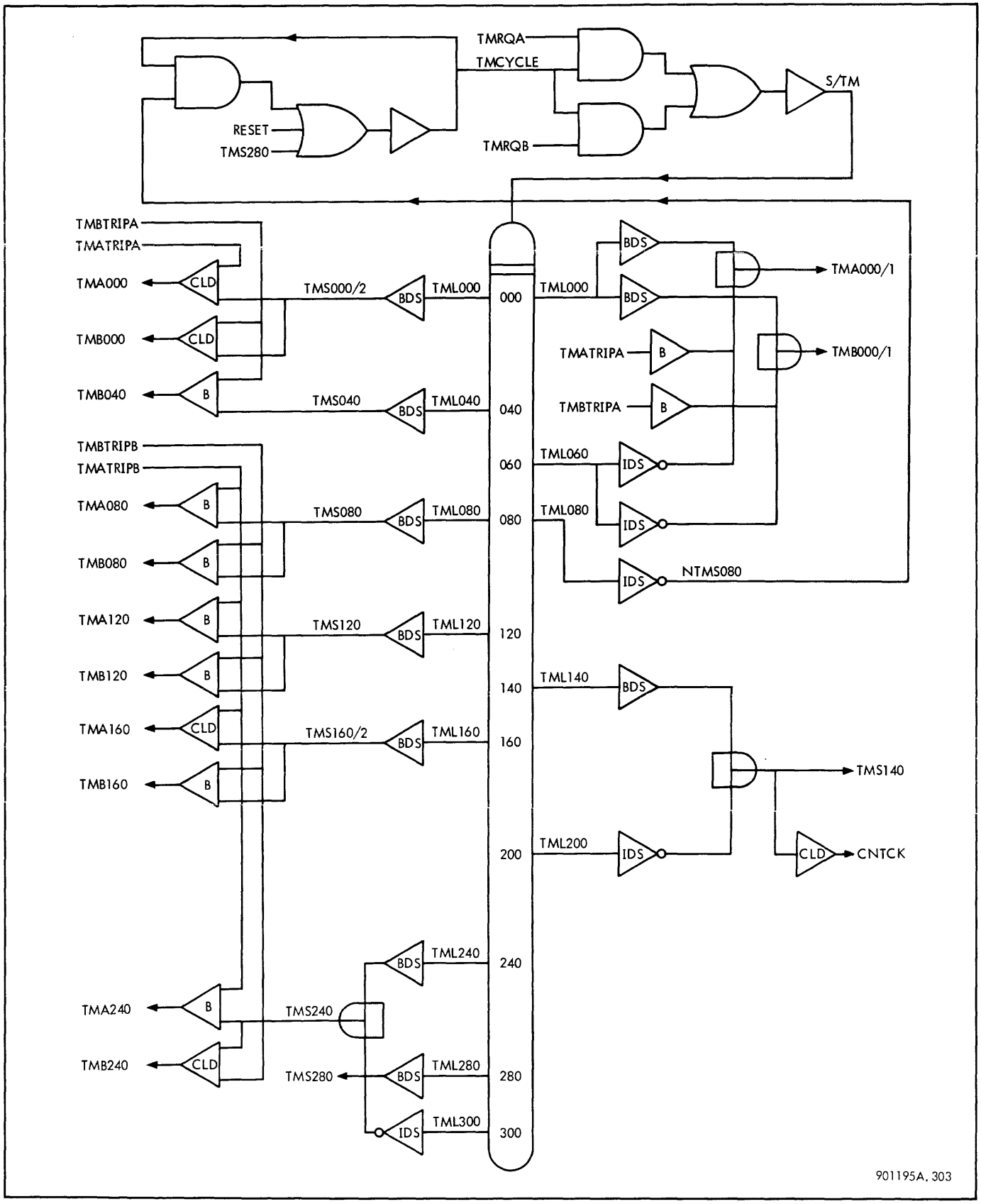


Figure 3-14. TM Delay Line Logic Diagram

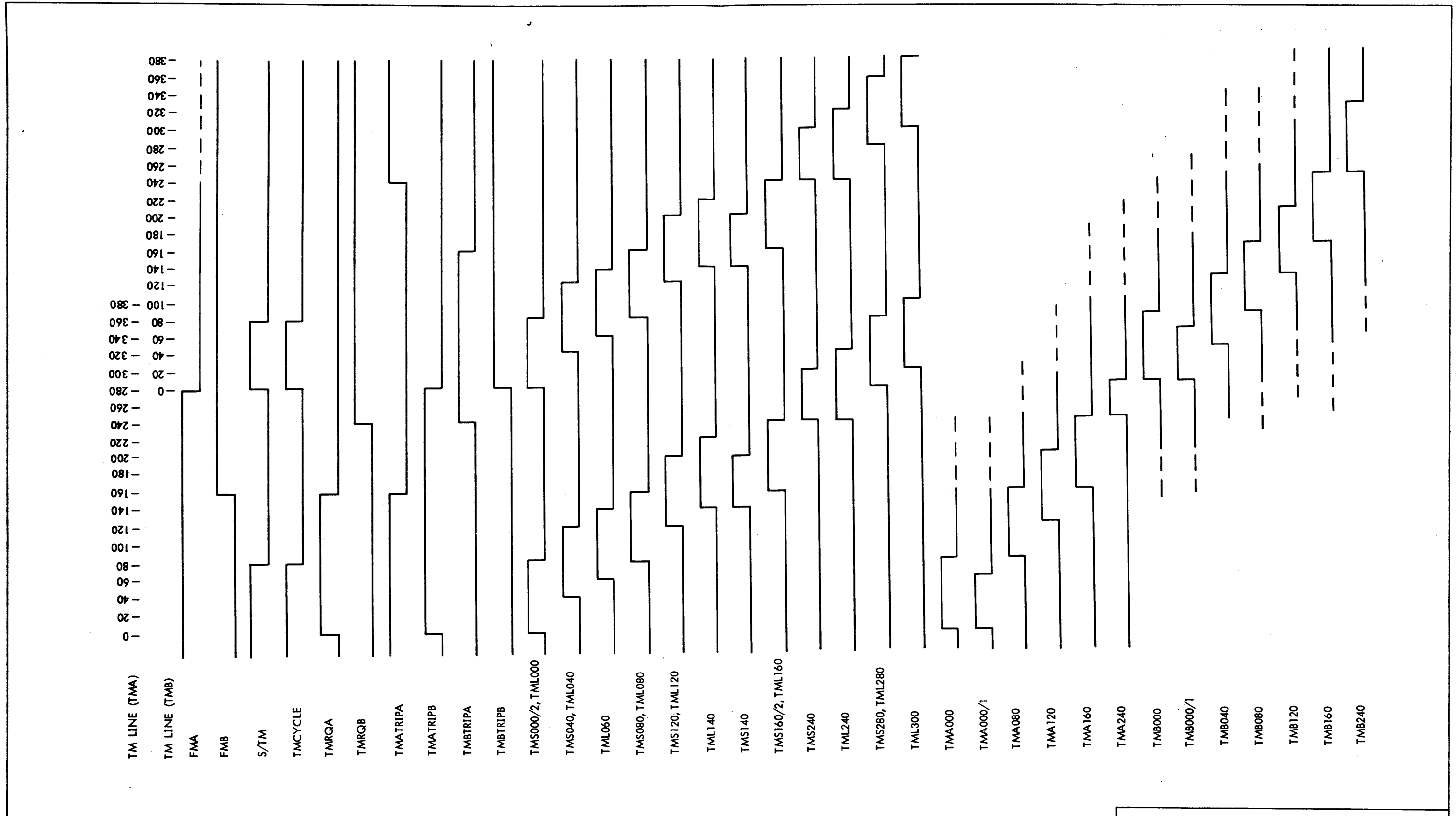


Figure 3-15. TM Delay Line Timing Signals

901195A.304

fronts 280 ns apart to travel down the delay line at the same time.

Each time the delay line is triggered, the voltage at the taps (signals TMLXXX) goes high for a period of 80 ns. The output of the delay line sensors (signals TMSXXX), which are controlled by these taps, are applied with other signals to buffers and clock drivers. The other signals (TMATRIPA, TMATRIPB, TMBTRIPA, and TMBTRIPB) effectively divide the clock drivers and buffers into two groups (TMAXXX and TMBXXX). The TMAXXX group of timing signals is associated with the operation of the line during the FMXXA phases, and the TMBXXX group is associated with the operation of the line during the FMXXB phases. This results in a group of unique timing pulses for each FM phase.

3-30 READ and WRITE Phases and TR Delay Line

The signals generated by the READ and WRITE phases and the TR delay line are used to control operation of the data buffer register (buffer). When data is to be written into the buffer, the WRITE latch is set, and when data is to be read from the buffer, the READ latch is set. A logic diagram of the TR delay line is shown in figure 3-16. Timing signals associated with the TR delay line is shown in figure 3-17. Zero ns delay is assumed for all circuits except the delay line.

The delay line starts when either of signals SREAD or SWRITE goes true, since signal CYCLE, the output of a latch, is true. The latch was set either by the RESET signal or by timing signals generated the last time the delay line was triggered. When signal SREAD or SWRITE goes true, signal S/TR goes true and triggers the delay line. When the voltage wave front reaches the 40 ns tap on the line, signal NTRS040 goes false and resets the latch. This causes a pulse having a duration of 40 ns to travel down the delay line.

Operation of the buffer is such that, during a data in operation, the SIOP tries to keep the buffer empty, and during a data out operation, it tries to keep it full. Therefore, priority for the read and write operation is not established until after the delay line has started. For example, if during a data in operation, signal SWRITE may be true and SREAD may come true immediately afterward. In this case, the read operation takes priority over the write operation by virtue of the logic for the READ and WRITE latches. If the operation is a data out, write takes priority over read to keep the buffer filled. If the operation is a data in, read takes priority over write to keep the buffer empty.

3-31 SIOP REGISTERS

The various data and control registers within the SIOP are shown in the SIOP overall block diagram (figure 3-2). A brief description of each register is presented below. Included with each description 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. The registers that comprise the data buffer are described following the heading Data Buffer Operation (paragraph 3-40).

3-32 Address (A) Register Inputs

The A-register (figure 3-18) consists of eight flip-flops, A0 through A7, and normally contains the device/device controller address. During execution of an SIO instruction, the A-register receives the address from the CPU via core memory location X'20', the M-register, and the I/O-register, if the SIO is successful.

During an HIO instruction, the address in the A-register is compared with the address in byte position A of the I/O-register. If they compare, signal MATCH goes true, which indicates that the HIO instruction (address bits in the I/O-register) is addressed to the device controller currently being serviced by the SIOP. The address in the A-register is also used during an AIO instruction to permit clearing of the interrupt status. An SIO instruction cannot be accepted by the SIOP until the interrupt status is cleared. The current address in the A-register is not cleared until the subsequent successful SIO, at which time it receives the new address.

3-33 Byte Count (BC) Counter

The BC counter consists of 16 flip-flops, BC00 through BC15 (figure 3-19), and contains the current byte count. The byte count is initially loaded by signal BCXM into the BC counter after the second word of the command doubleword is accessed from core memory and placed into the M-register. The BC counter is decremented by one for each byte transmitted at the core memory interface (signal BCNT).

During execution of an instruction (when the current byte count is to be sent to the CPU as part of the response to the instruction), the byte count is transferred to byte positions C and D of the M-register via the J-register. During data in and data out service, the byte count (certain decoded states) is applied (with the byte address, BA17 and BA18 decoded) to the byte presence input register (figure 3-40), which in turn controls the count generator to supply updating information for the BC and BA counters. Updating of the BC counter as a function of the count generation logic is shown in table 3-4.

3-34 Byte Address (BA) Counter

The BA counter consists of 19 flip-flops, BA00 through BA18 (figure 3-20), and contains the current byte address. The byte address is initially loaded into the BA counter by signal BAXM after the first word of the command doubleword is accessed from core memory and placed into the M-register. The byte address is updated as data is exchanged between core memory and the SIOP; it is incremented when the device is executing a write or read order, and is decremented when the device is executing a read backwards order.

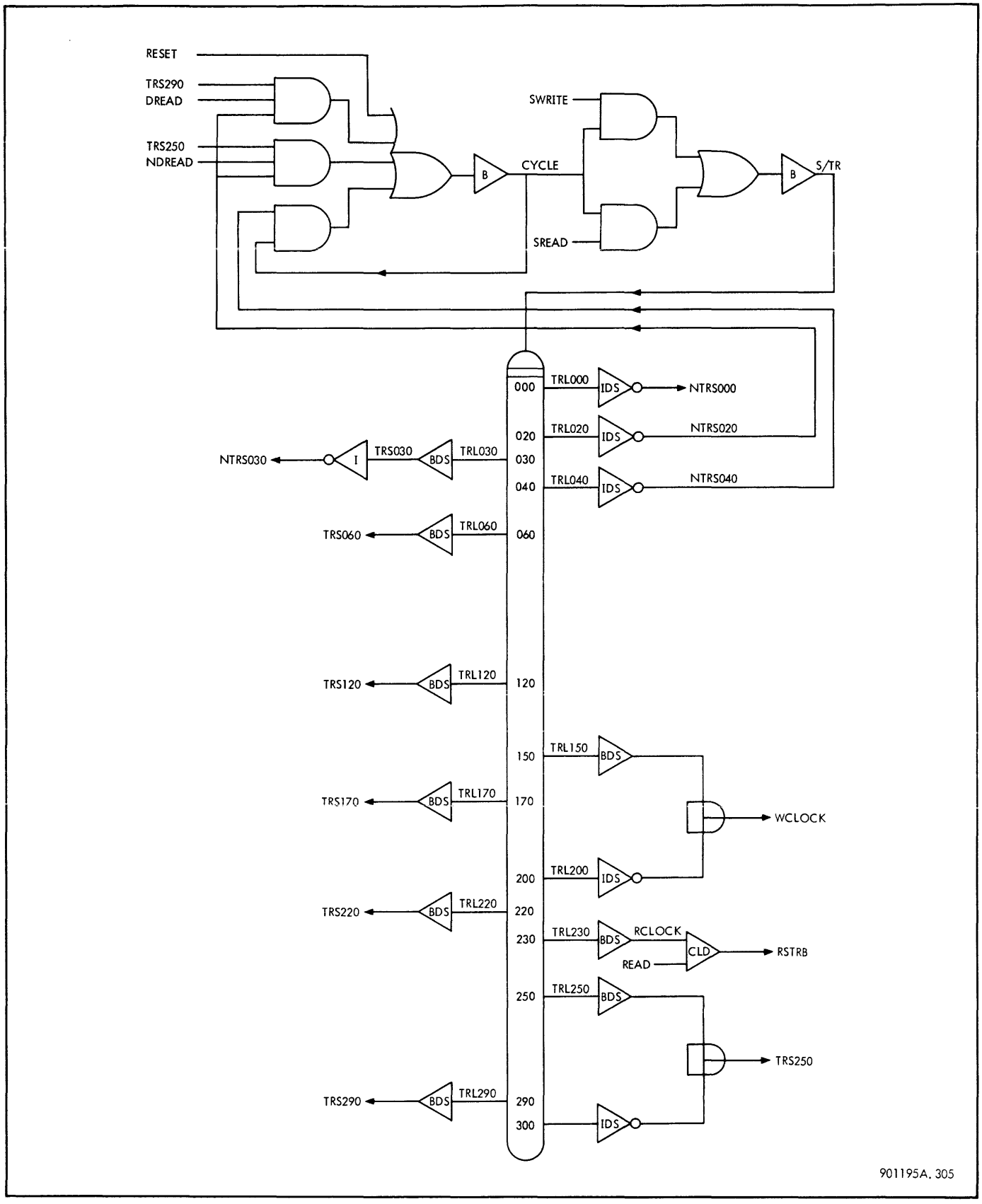


Figure 3-16. TR Delay Line Logic Diagram

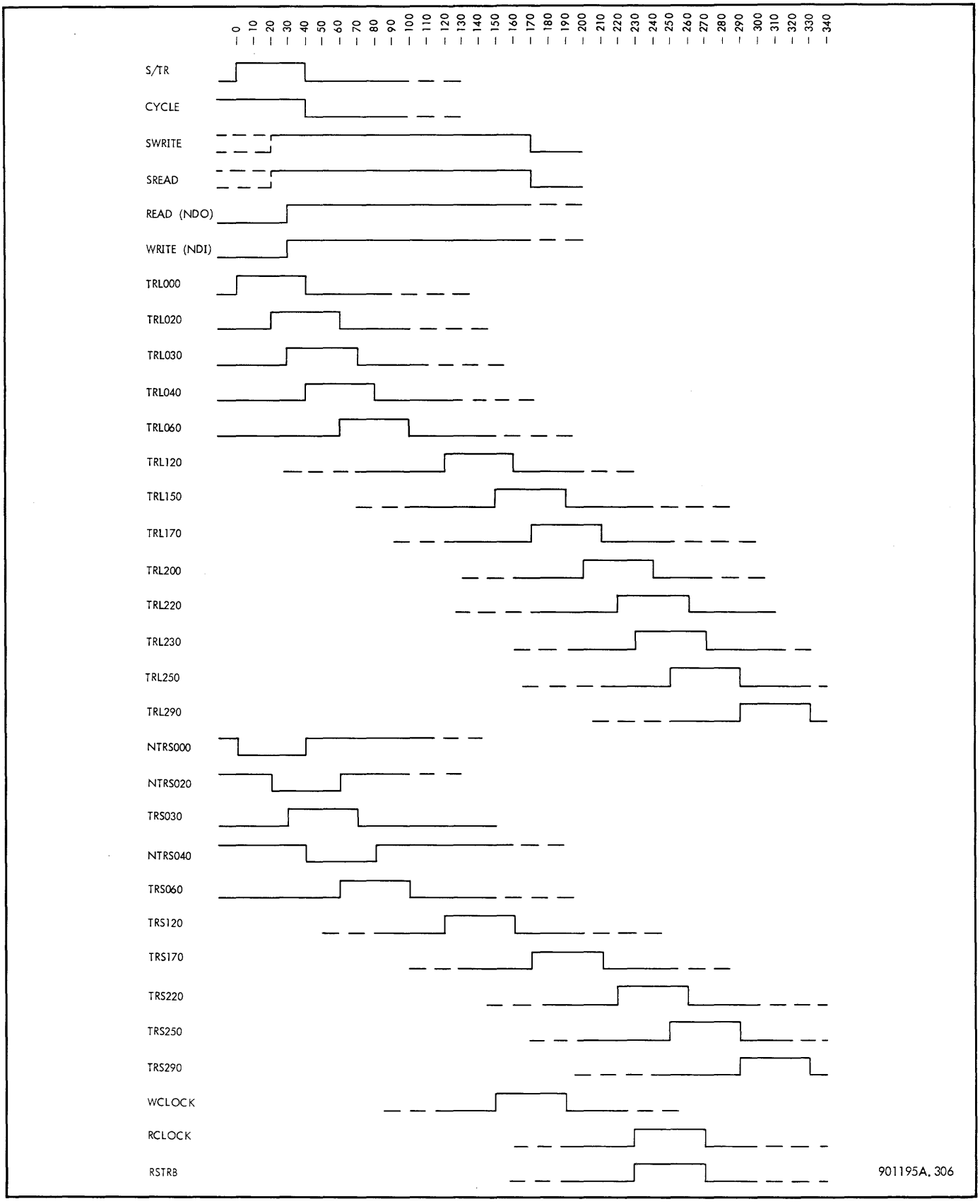


Figure 3-17. TR Delay Line Timing Signals

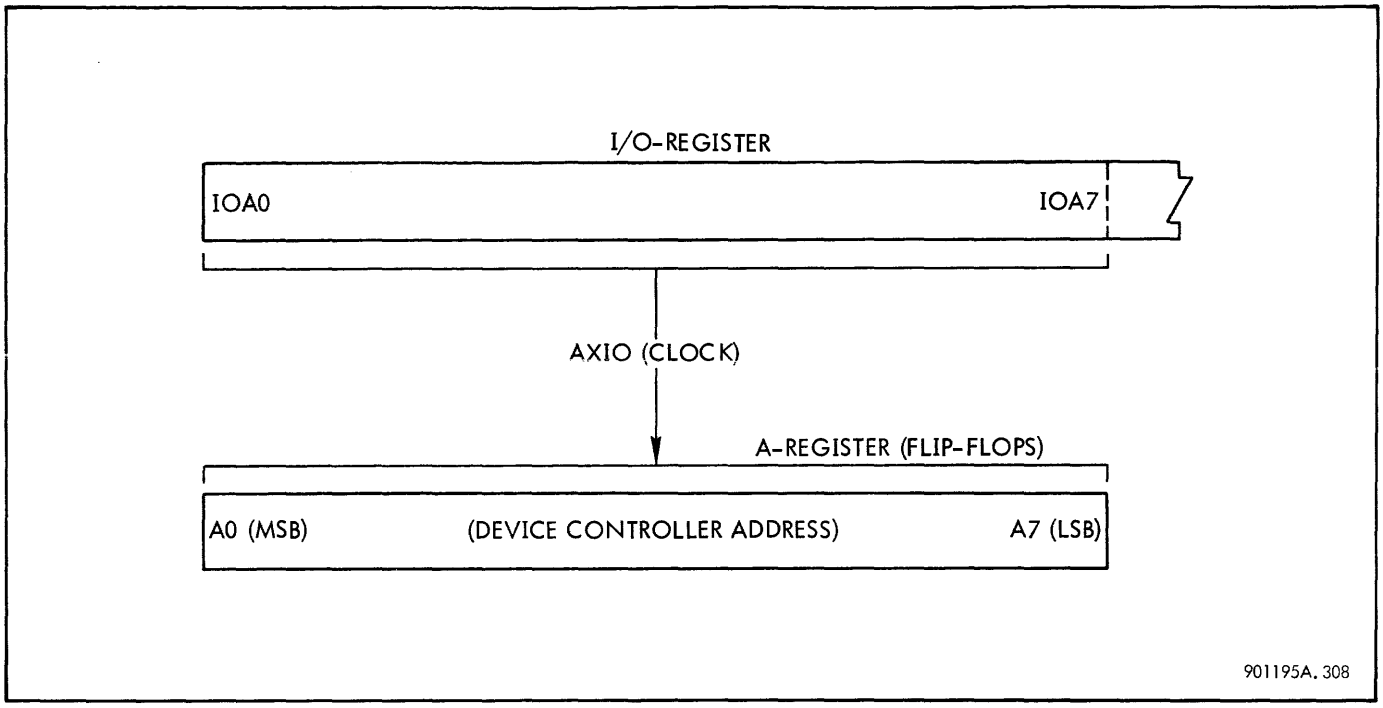


Figure 3-18. Address (A) Register Inputs

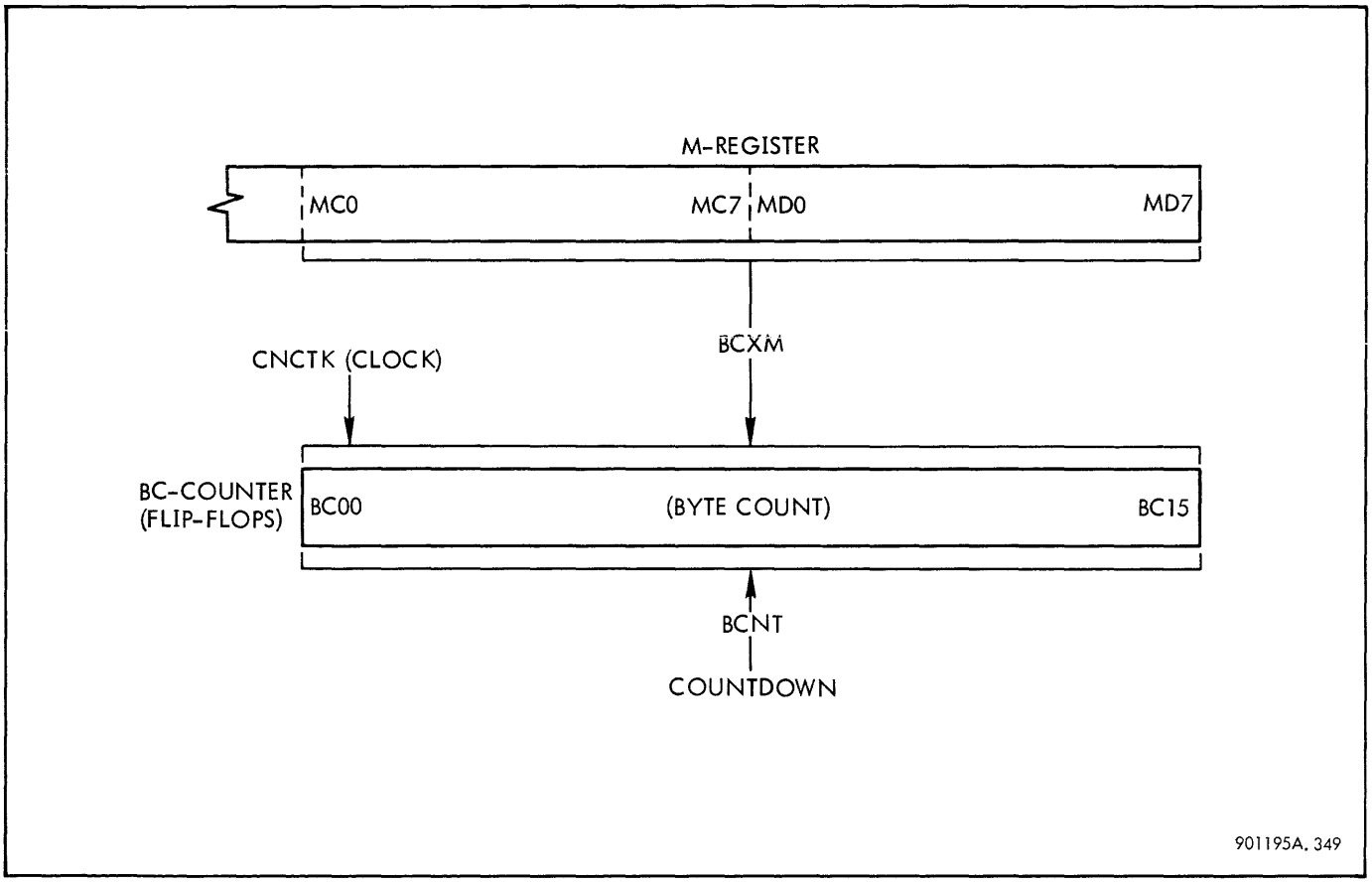


Figure 3-19. Byte Count (BC) Counter Inputs

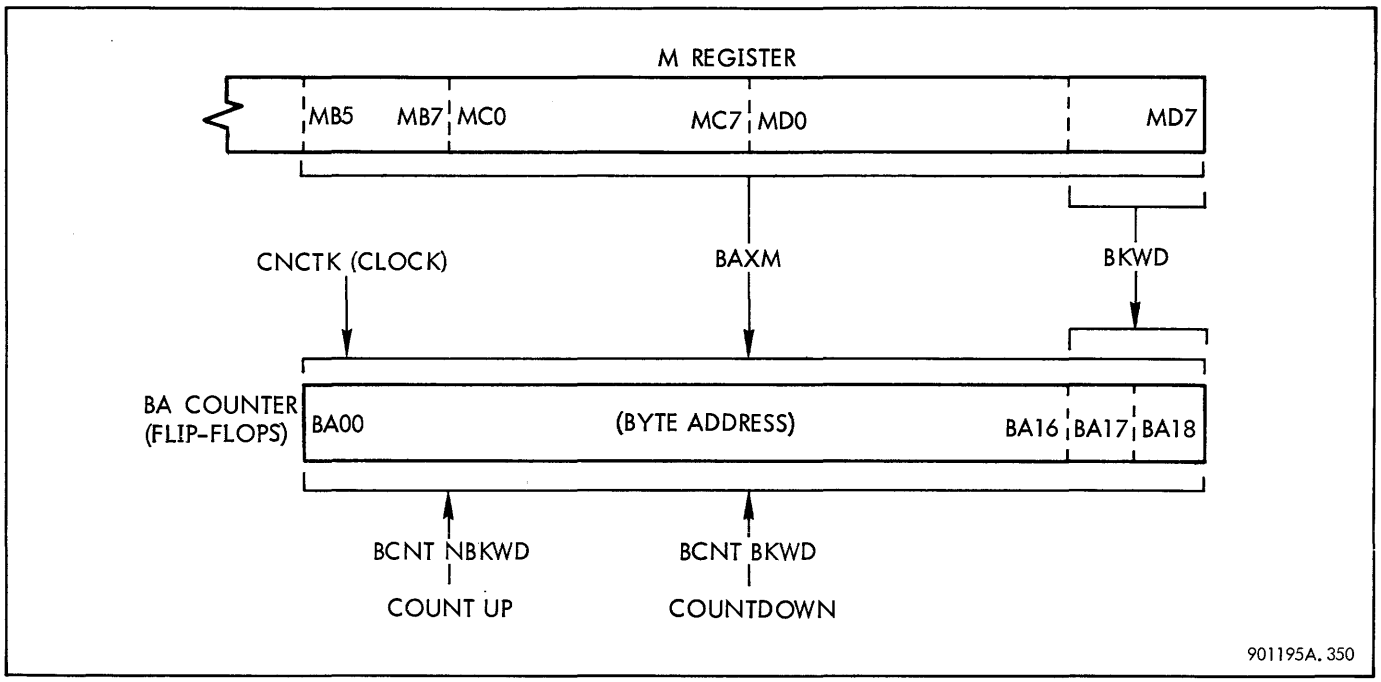


Figure 3-20. Byte Address (BA) Counter Inputs

Table 3-4. Byte Count Counter Updating

PRESENT STATE			NEXT STATE											
			Down 1 CBC1			Down 2 CBC2			Down 3 CBC3			Down 4 CBC4		
BC13	BC14	BC15	BC13	BC14	BC15	BC13	BC14	BC15	BC13	BC14	BC15	BC13	BC14	BC15
0	0	0	1	1	1	1	1	0	1	0	1	1	0	0
0	0	1	0	0	0	1	1	1	1	1	0	1	0	1
0	1	0	0	0	1	0	0	0	1	1	1	1	1	0
0	1	1	0	1	0	0	0	1	0	0	0	1	1	1
1	0	0	0	1	1	0	1	0	0	0	1	0	0	0
1	0	1	1	0	0	0	1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	0	0	0	1	1	0	1	0
1	1	1	1	1	0	1	0	1	1	0	0	0	1	1

During data in and data out service, the byte address (BA17 and BA18 decoded) is applied (with certain decoded states of the byte count) to the presence input register (figure 3-40), which in turn controls the count generator to supply updating information for the BC and BA counters. Updating of the BA counter when the device is executing a write, read (NBKWD), or read backwards order (BKWD) is shown in table 3-5. Initialization of the BA counter during chaining is shown in figure 3-21. The initial state of BA17 and BA18 is a function of the state of the logic signal BKWD. If signal NBKWD is true, the two LSBs of the byte address in bit positions MD6 and MD7 are transferred unchanged to bit positions BA17 and BA18 of the BA counter. If signal BKWD is true, bit positions BA17 and BA18 are set or reset as required so that they become the inverse of MD6 and MD7, respectively.

During data out service, signal BKWD is false; therefore, the BA counter is updated by incrementing BA00 through BA18. This is also true for data in service if signal BKWD is false. However, during data in service, if signal BKWD is true, the BA counter is updated by incrementing BA16 through BA18 and decrementing BA00 through BA16. (BA00 through BA16 and BA16 through BA18 can be considered as two distinct counters.) An example of the updating of the BA counter during data in service when the byte count is equal to seven is shown in table 3-6.

3-35 Command Address (CA) Counter

The CA counter (see figure 3-22) consists of 16 flip-flops, CA00 through CA15, and normally contains the current command doubleword address (command address) plus one. The 16-bit command address is set into the 16 MSBs of the S-register whenever a new command is to be accessed, that is, during an order out service cycle or a data service cycle with chaining. The LSB (S31) of the S-register is controlled separately (see paragraph 3-25).

Each time a new command is accessed from core memory, the command address is incremented by one by signal CACNTU. During execution of an instruction (when the current command address is to be sent to the CPU as part of the response to the instruction), the command address is decremented by one by signal CACNTD and is transmitted via the J-register to the M-register (byte positions A and B).

The command address is initially loaded into the CA counter during the later part of an SIO instruction (if it is successful) from byte positions C and D of the M-register. Clock signal CNTCK, for the CA counter flip-flops, goes true every time the TM delay line runs.

3-36 Flags (FL) Register

The FL-register consists of eight buffered latches shown in figure 3-23. During an order out service cycle, the second word of the command doubleword, which contains the flags,

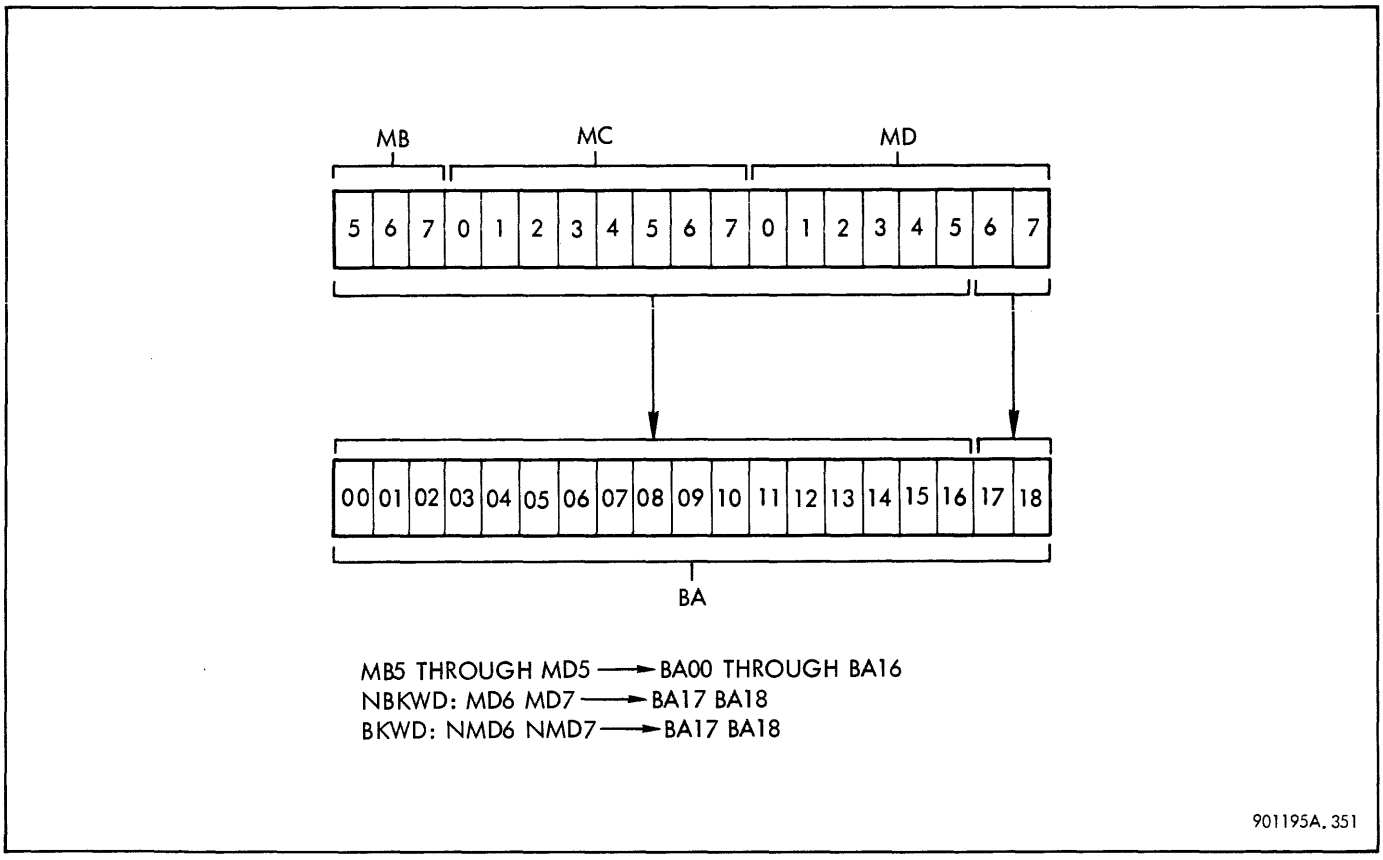


Figure 3-21. BA Counter Initial States

Table 3-5. BA Counter Updating for NBKWD or BKWD

STATE PRESENT			NEXT STATE											
			Up 1 CBC1			Up 2 CBC2			Up 3 CBC3			Up 4 CBC4		
BA16	BA17	BA18	BA16	BA17	BA18	BA16	BA17	BA18	BA16	BA17	BA18	BA16	BA17	BA18
0	0	0	0	0	1	0	1	0	0	1	1	1	0	0
0	0	1	0	1	0	0	1	1	1	0	0	1	0	1
0	1	0	0	1	1	1	0	0	1	0	1	1	1	0
0	1	1	1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	0	0	0
1	0	1	1	1	0	1	1	1	0	0	0	0	0	1
1	1	0	1	1	1	0	0	0	0	0	1	0	1	0
1	1	1	0	0	0	0	0	1	0	1	0	0	1	1

Table 3-6. Sample Updating of BA Counter During Data In Service

State	BKWD False	BKWD True
Initial state: MD5 MD6 MD7	0 1 0	0 1 0
Initial state: BA16 BA17 BA18 (see figure 3-21)	0 1 0	0 0 1
S/WB-register (see table 3-13)	See BAA2: set WBC, WBD	See BAA1: set WBB, WBC, WBD
Transfer WB to MW (see figure 3-41)	WBC to MWC WBD to MWD	WBB to MWC WBC to MWB WBD to MWA
Count generator (see table 3-14)	See CBC2	See CBC3
Next state: BA16 BA17 BA18 (see table 3-5)	1 0 0	1 0 0

Note: Byte count = 7

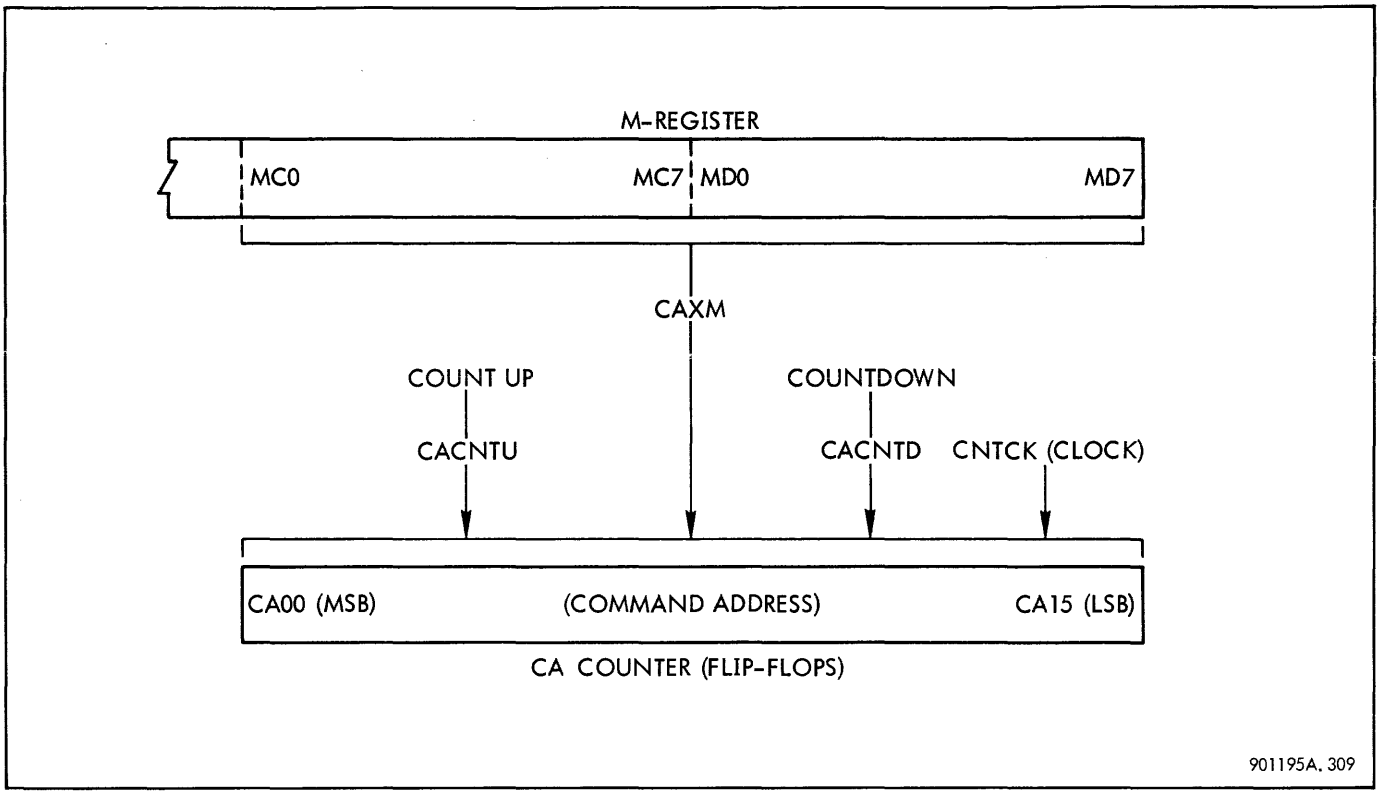


Figure 3-22. Command Address (CA) Counter Inputs

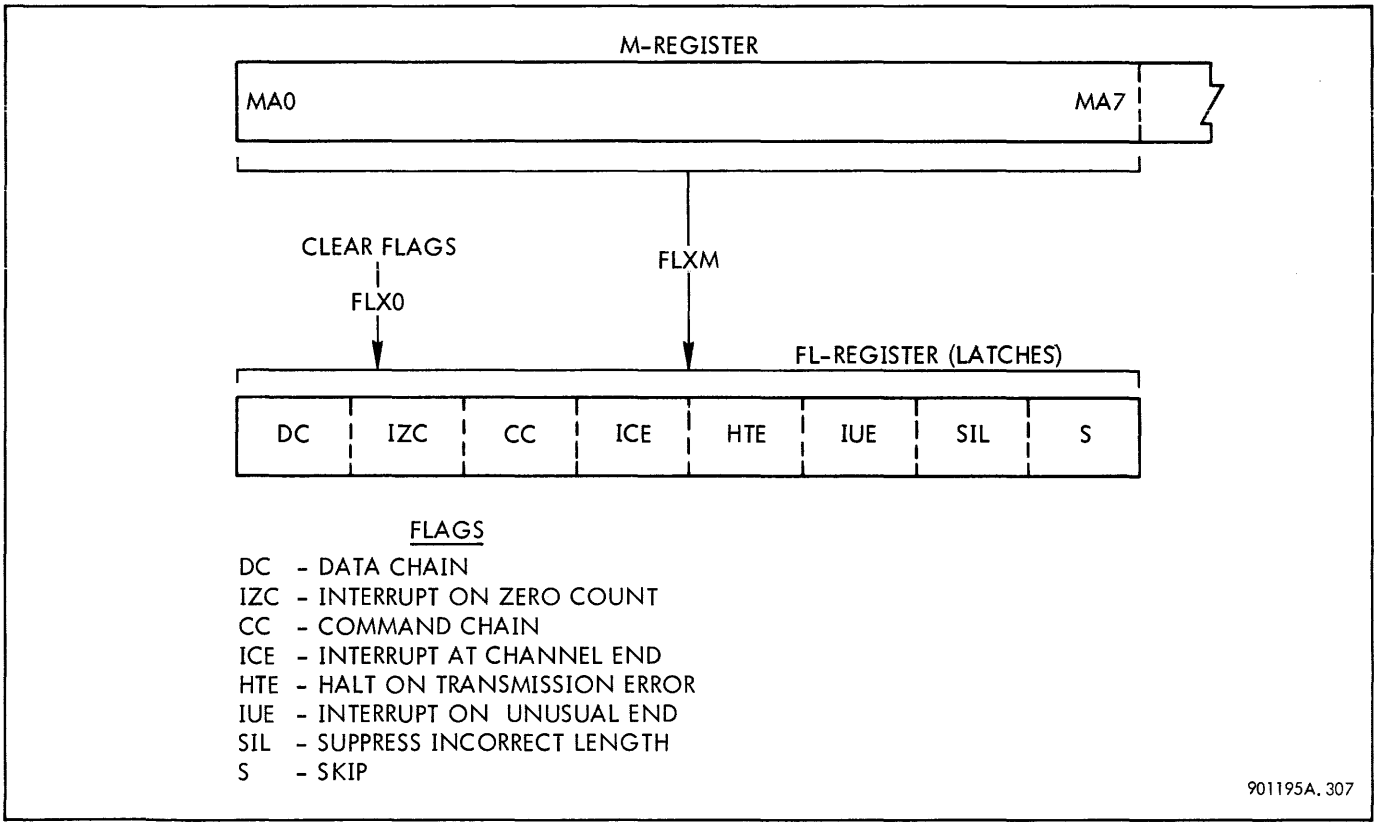


Figure 3-23. Flags (FL) Register Inputs

is set into the memory assembly (M) register after core memory is accessed. If no core memory errors are detected during the memory access, the flags are then transferred from the M-register to the FL-register, where they are used to control the operations of subsequent service cycles.

For a detailed description of the flags, see paragraph 3-20 of this manual, Sigma 5 Computer Reference Manual, SDS publication 900959, and Sigma 7 Computer Reference Manual, SDS publication 900950.

3-37 Input/Output (I/O) Register

The I/O-register (see figure 3-24) consists of 32 buffered latches, IOA0 through IOA7, IOB0 through IOB7, IOC0 through IOC7, and IOD0 through IOD7. During output operations, the I/O-register drives the data lines to the device controller, and during input operations, information placed on the data lines by the device controller is set into the I/O-register. During instruction execution, the address encoded in the instruction is transferred via the M-register and I/O-register to the A-register.

During the termination phase of a data in or data out operation, the terminal order is assembled in the most significant byte position (byte position A) of the I/O-register for transmission to the device controller. During data out service, the I/O-register receives data from the K-register (output of the data buffer), and during data in service, the I/O-register transmits data to the J-register (input of the data buffer).

3-38 Memory Address (S) Register

The S-register (figure 3-25) consists of 17 buffered latches, S15 through S31, the outputs of which drive the 17 address

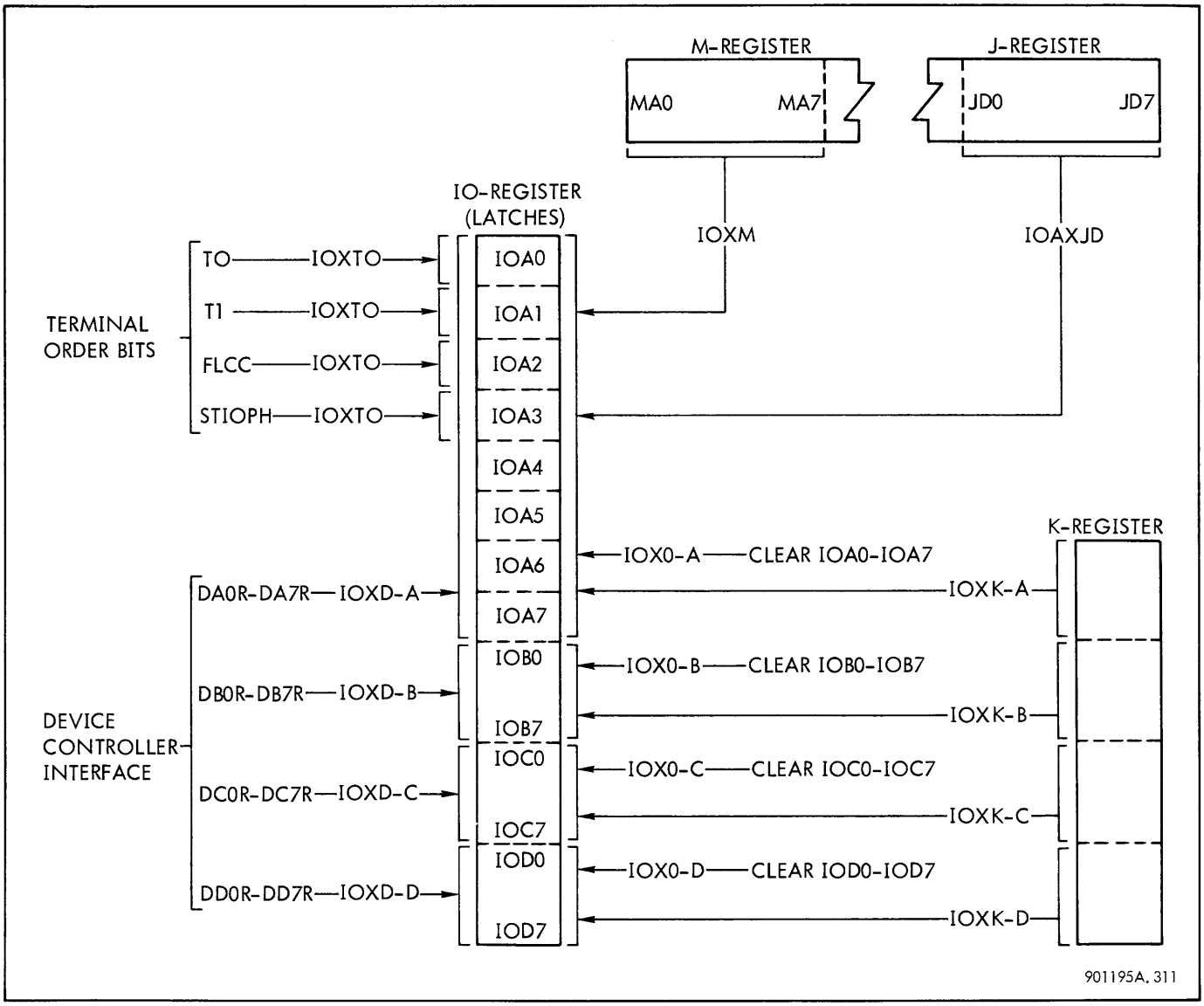


Figure 3-24. Input/Output (I/O) Register Inputs

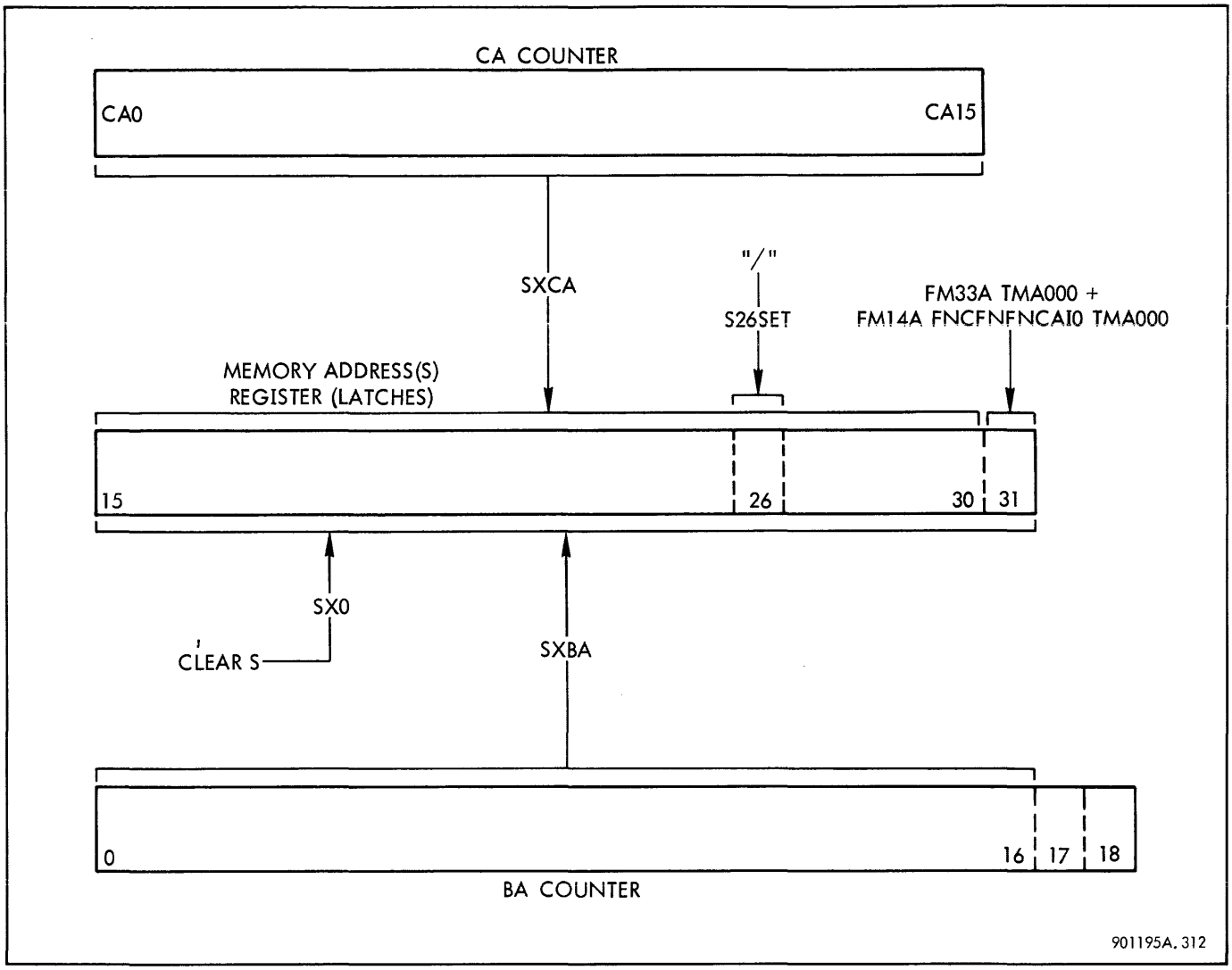


Figure 3-25. Memory Address (S) Register Inputs

lines, /LX15/ through /LX31/, to core memory. Bit S15 is the MSB and bit S31 is the LSB of the word address.

During the data service cycles, the S-register receives its inputs from the byte address (BA) register. Signal SXBA transfers the most significant 17 bits of the BA-register (BA0 through BA16) to the S-register to define the core memory word location. Bits 17 and 18 of the BA-register define the particular byte of the word specified by the S-register.

Whenever a command doubleword is accessed from core memory, the S-register receives its inputs from the command address (CA) register. This occurs during order out service following an SIO, or because of command chaining. Signal SXCA, which transfers bits 0 through 15 of the CA-register to bits 15 through 30 of the S-register, goes true during FM31A of an order out service cycle. Bit 31 is zero, since the S-register is cleared at the beginning of the phase. The first word (even numbered) of the command doubleword is therefore accessed from core memory after the memory request signal goes true.

The current command doubleword address remains in the S-register until FM33A (the next phase of an order out service cycle), at which time S31 is forced to a one by signal FM33A TMA000. The second (odd numbered) word of the command doubleword is therefore accessed after the memory request signal again goes true.

During the early part of the CPU-initiated instructions (SIO, HIO, TIO, and TDV), the SIOP must read the contents of core memory location X'20'. This is accomplished during FM01A by first clearing the S-register, and by then forcing a one into bit S26. During the later part of an SIO, HIO, TIO, or TDV instruction, the SIOP may have to write information back into core memory locations X'20' and X'21'. If core memory location X'20' is to be written into, the SIOP sequences through FM12, at which time the memory request

signal is sent to core memory. The information is written into location X'20', since that was the address previously set into the S register and has not been changed. When the SIOP sequences to FM14, if the instruction is not an AIO, a one is forced into bit S31 by signal FM14A FNCFNFMCAIO TMA000, thus addressing location X'21'. If the instruction is an AIO, the S-register addresses location X'20' during FM01 and remains at X'20' until FM14, at which time the response information is written into location X'20'.

3-39 Memory Assembly (M) Register

The M-register (figure 3-26) consists of 32 buffered latches, MA0 through MA7, MB0 through MB7, MC0 through MC7, and MD0 through MD7. The M-register receives data from core memory during core memory read operation, and also drives the memory data lines during memory write operations.

During data out service cycles, the M-register receives data from core memory and transmits it to the I/O-register via the data buffer. During data in service cycles, the M-register receives data from the I/O-register via the data buffer and transmits it to core memory. The enabling signals, MXM-A through MXM-D, are controlled by signals data gate DG from core memory and MPESHARE from the bus-sharing option, when core memory is accessed.

When the device is excuting a read backwards order, enabling signals MXKB-A through MXKB-D enable transfer of data from the least significant byte position of the K-register (part of data buffer), to the most significant byte position of the M-register, the second least significant byte position of the K-register to the second most significant byte position of the M-register, and so on. When the device is executing a read forward order, the data is transferred from the K-register to the M-register in a straight forward manner, that is, from least significant to least significant. When the SIOP is responding to a CPU instruction, the response information that is sent to core memory from the M-register is first assembled in the J-register. Enabling signals MUJX and MLJX transfer the information to the M-register.

3-40 DATA BUFFER OPERATION

The data buffer provides for the exchange of data between core memory and a device controller at the maximum possible rate. This rate is dictated by maximum core memory rate. When the IOP-memory interface is operating at near maximum memory data rate, any disturbance at this interface that tends to reduce the instantaneous data rate at the device controller interface to below the average rate required by the device controller also tends to cause a rate error in the device controller. The data buffer smooths the data rate at the device controller interface so as to prevent data rate errors related to memory port interference, data chaining, and IOP byte alignment timing delays.

Circuitry associated with the data buffer also enables realignment of the data boundaries with the IOP. For example, during a data out service cycle, data may start at a designated byte address at the core memory interface (defined by BA17 and BA18) and be aligned in the IOP to start on a word boundary at the device controller interface. Similarly, during a data in service cycle, data aligned on a word boundary at the device controller interface is aligned in the IOP to start at the designated byte address at the core memory interface. See figure 3-27 for a block diagram of the data buffer. The data buffer registers are described below.

3-41 High Speed Memory (R) Register

The R-register consists of four fast-access memory (FT39) modules, RA0 through RA8, RB0 through RB8, RC0 through RC8, and RD0 through RD8. The organization of the R-register is shown in figure 3-28. The input and output signals are shown in figure 3-29. Each module contains eight location. Each location has nine bits. All bits are used to store data except RA8, RB8, RC8, and RD8, which function as byte presence indicators (figure 3-30). The R-register is organized on a byte basis. If the device controller interface data path is four bytes wide, all four of the fast access memory modules are used. If the data path is one byte wide, only the modules containing bits RA0 through RA7 are used, and if the data path is two bytes wide, only the modules containing bits RA0 through RA8 and RB0 through RB8 are used.

During data out service, the data moves from the memory interface (M) register, through the data buffer to the device controller interface (I/O) register (see figure 3-31). Similarly, for data in service, the data moves from the I/O-register through the data buffer to the M-register (see figure 3-32). All data commutation occurs at the R-register input during data out service for device controller interface paths that are one byte wide (DX1) or two bytes wide (DX2). Data alignment also occurs at the R-register output for both data out and data in service. Decommutation of the data occurs at the data buffer output during data in service when DX1 and DX2 are true.

3-42 Address Selection (LR) Register

The LR-register consists of three buffered latches, LR1, LR2, and LR3. The LR-register is driven from the IA-register during buffer write cycles and from the OA-register during buffer read cycles. The purpose of the LR-register is to select one of eight locations in each of the four fast-access memory modules of the R-register during a buffer read or write operation. (See figure 3-33.) During a write operation, LR1 through LR3 are controlled by signals IA1 through IA3 respectively, and during a read operation, LR1 through LR3 are controlled by signals OA1 through OA3, respectively. (See figure 3-34.) Table 3-7 shows the location of the R-register as a function of the LR-register output signals.

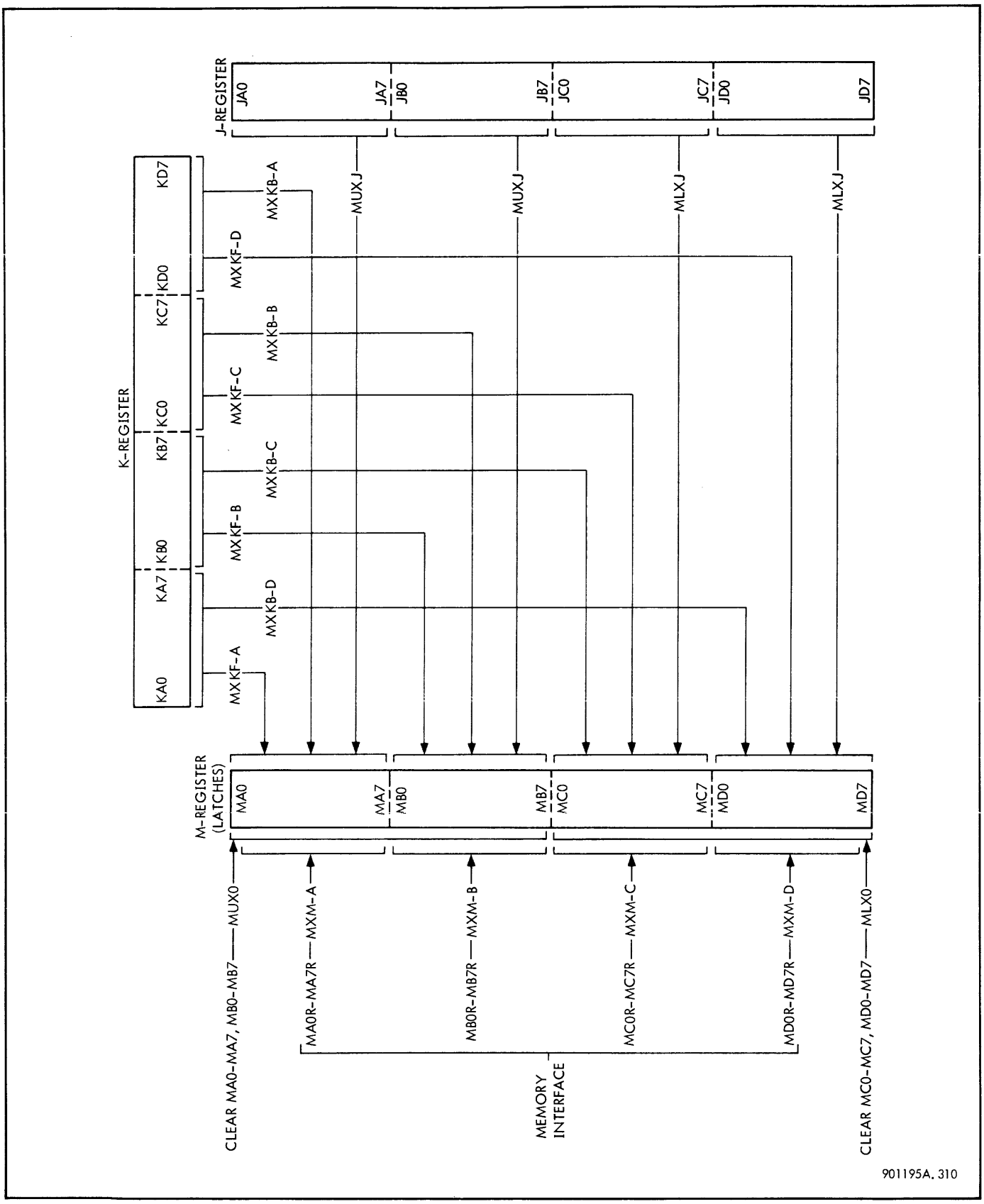


Figure 3-26. Memory Assembly (M) Register Inputs

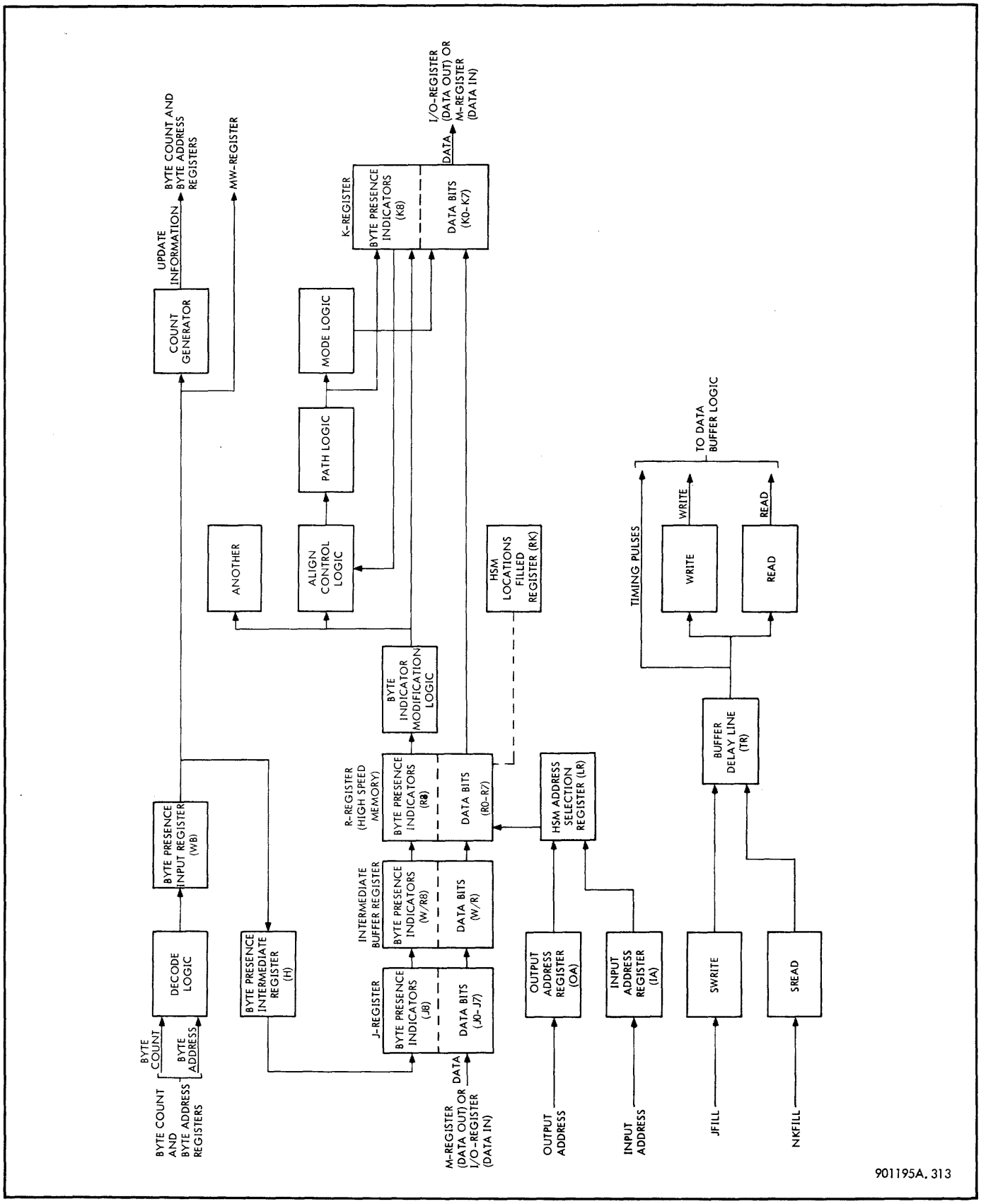
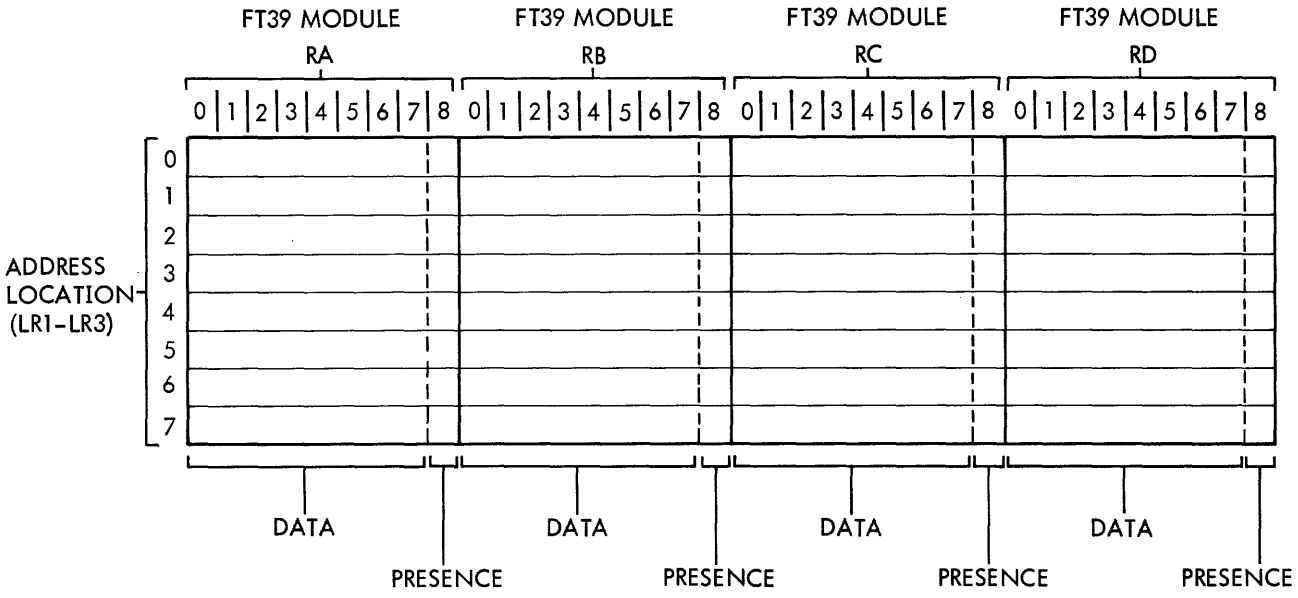


Figure 3-27. Data Buffer Block Diagram



NOTES :

1. EACH MODULE CONTAINS 8 LOCATIONS (0 THROUGH 7)
2. EACH LOCATION OF EACH MODULE CONTAINS 9 BITS (0 THROUGH 8)
3. BYTES ARE USED AS FOLLOWS FOR THE THREE INTERFACE WIDTHS :

INTERFACE WIDTH	MODULE RA	RB	RC	RD
DX1	X			
DX2	X	X		
DX4	X	X	X	X

901195A. 314

Figure 3-28. R-Register Physical-Functional Correlation

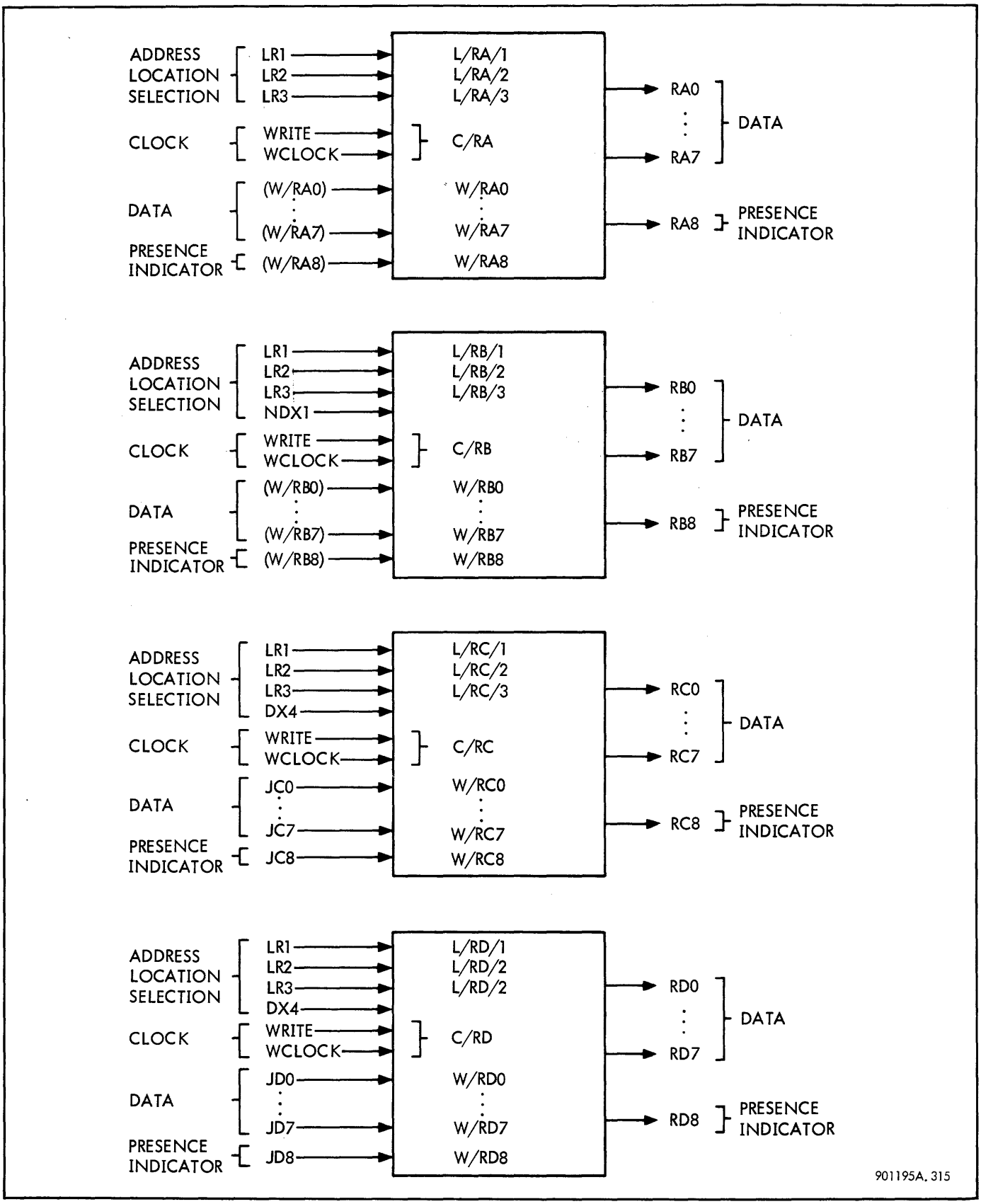


Figure 3-29. R-Register Input/Output Signals

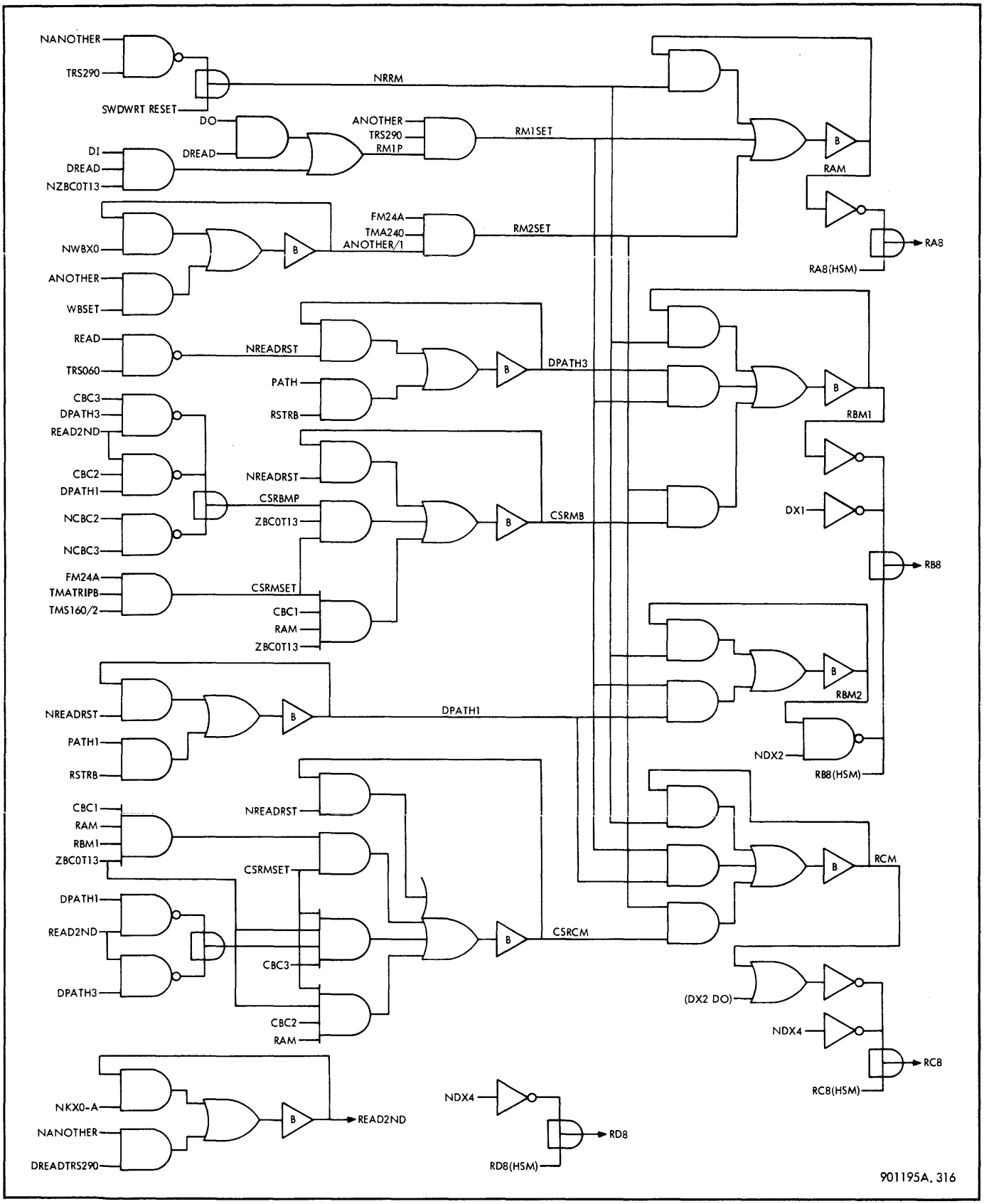


Figure 3-30. R-Register Presence Indicator Bits, Logic Diagram

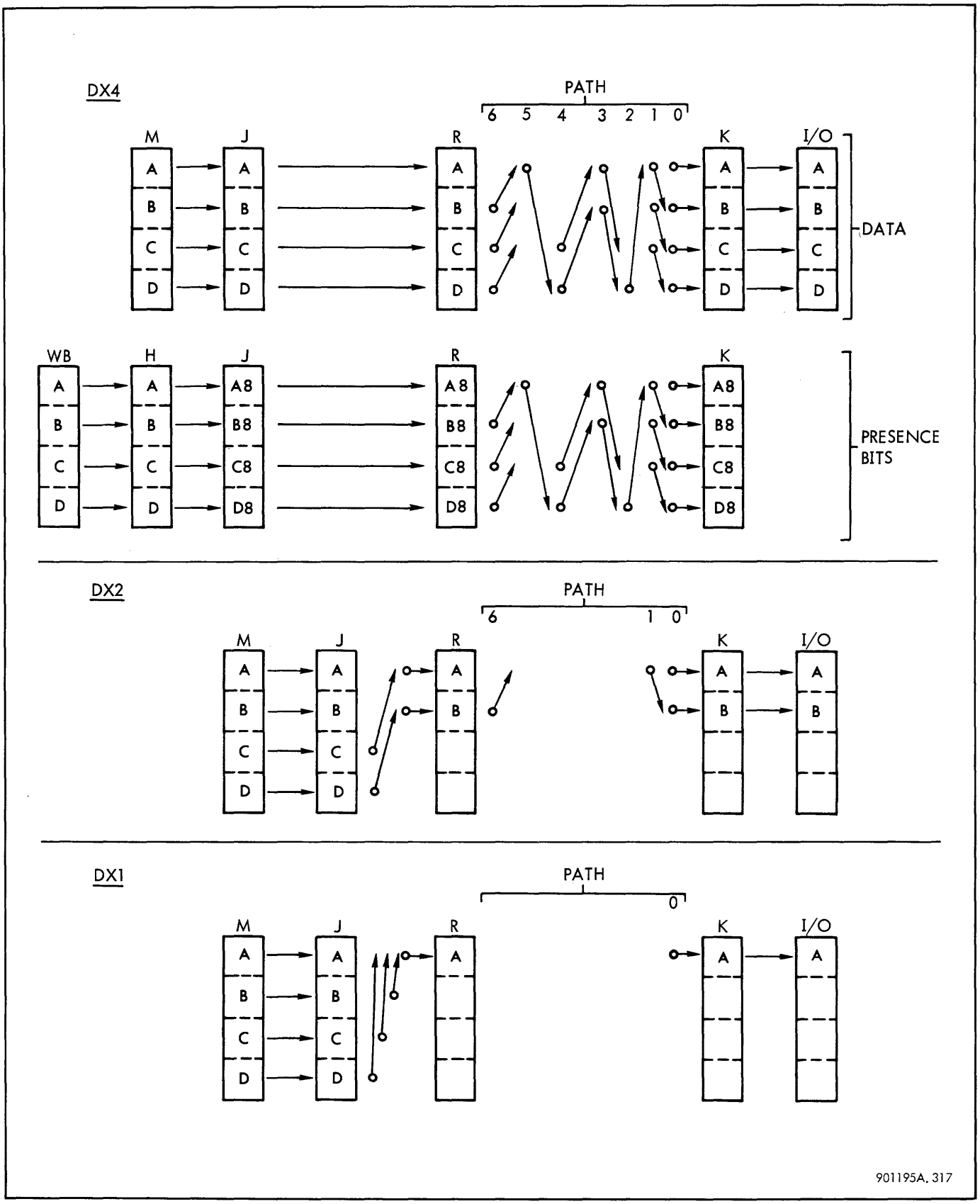


Figure 3-31. Buffer Data Path During Data Out Service

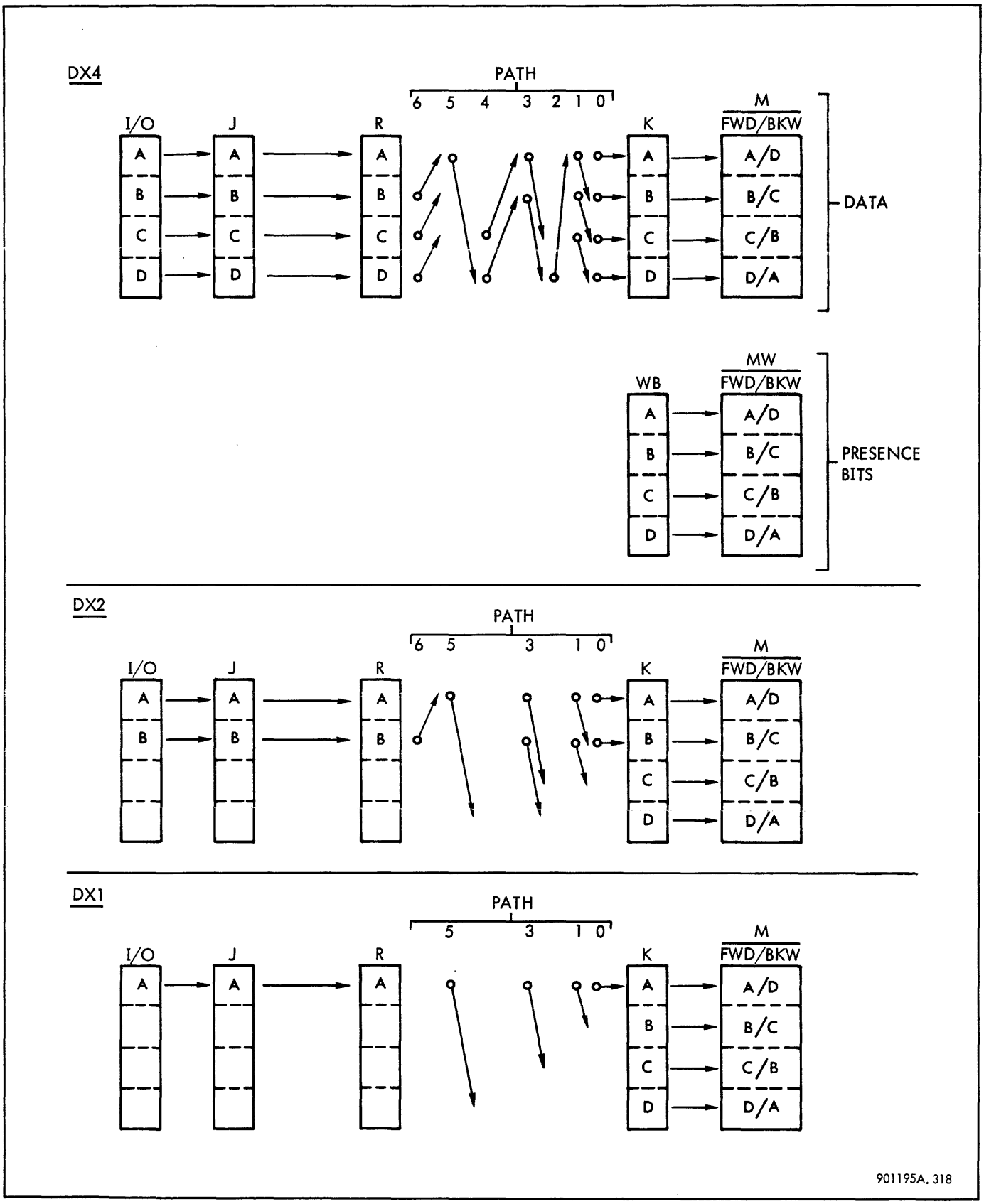


Figure 3-32. Buffer Data Path During Data In Service

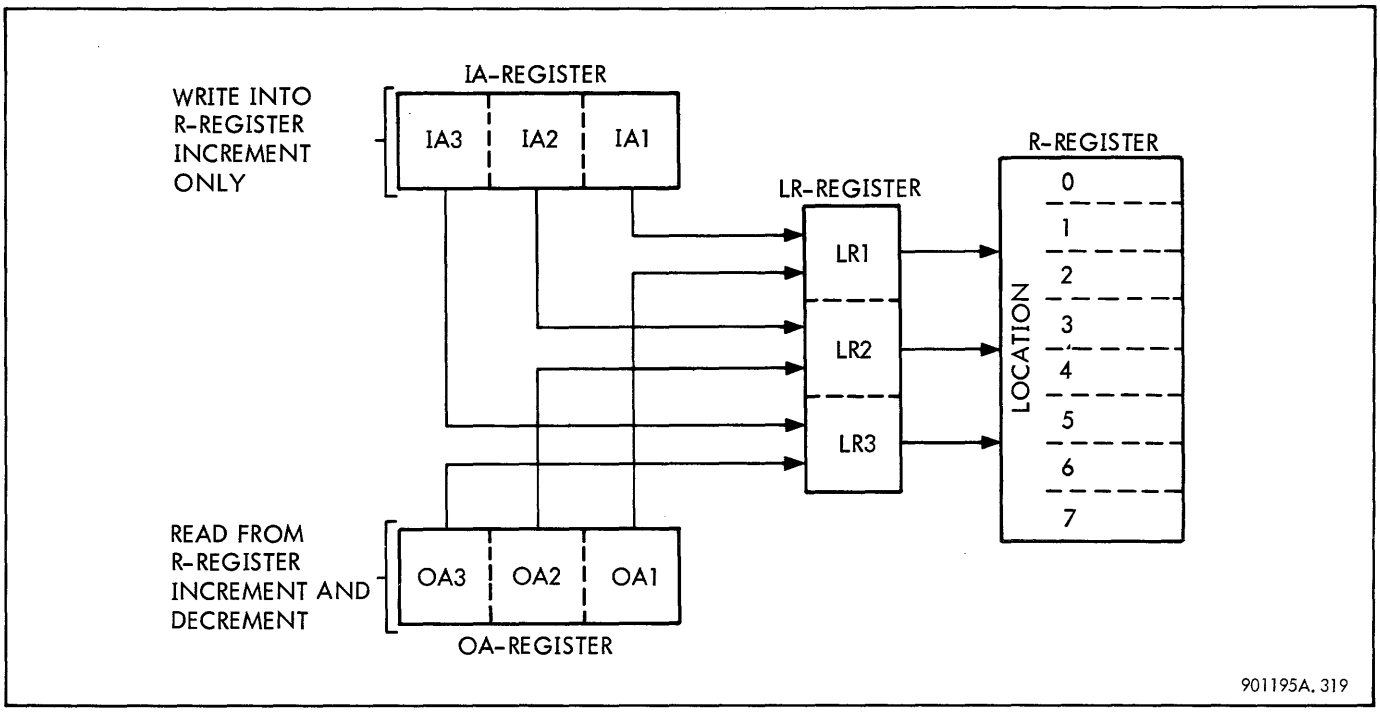


Figure 3-33. R-Register Location Selection (LR) Register

Table 3-7. Decoding of LR-Register

LR1	LR2	LR3	R-Register Location
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

3-43 High Speed Memory Input Address (IA) Register

The IA-register consists of three flip-flops, IA1 through IA3. (See figure 3-34.) The output of IA-register is applied to the LR-register. After updating (incrementing by one) during a buffer write cycle, the IA-register designates the high speed memory location to be written during the next write cycle. The initial state of the IA-register after command chaining is 000.

3-44 High Speed Memory Output Address (OA) Register

The OA-register consists of three flip-flops, OA1 through OA3. (See figure 3-34.) The output of the OA-register is applied to the LR-register. After updating during a buffer read cycle, the OA-register designates the high speed memory location to be read from during the next buffer read cycle. The initial state of the OA-register after command chaining is 000.

3-45 High Speed Memory Locations Filled (RK) Register

The RK-register consists of four flip-flops, RK0 through RK3. (See figure 3-35.) This register designates the number of high speed memory locations that contain data. The RK-register is decremented during buffer write cycles, and is incremented during buffer read cycles. Table 3-8 shows the relationship between the register state and the number of locations filled in the high speed memory. The initial state of the RK-register after command chaining is 1111.

During modification of the R-register presence indicator bits, the OA-register is decremented by one. Signals DREAD, TRS290, and ANOTHER provide the clock signal when the OA-register is decremented.

3-46 Buffer Input (J) Register

The J-register consists of 32 buffered latches, JA0 through JA8, JB0 through JB8, JC0 through JC8, and JD0 through

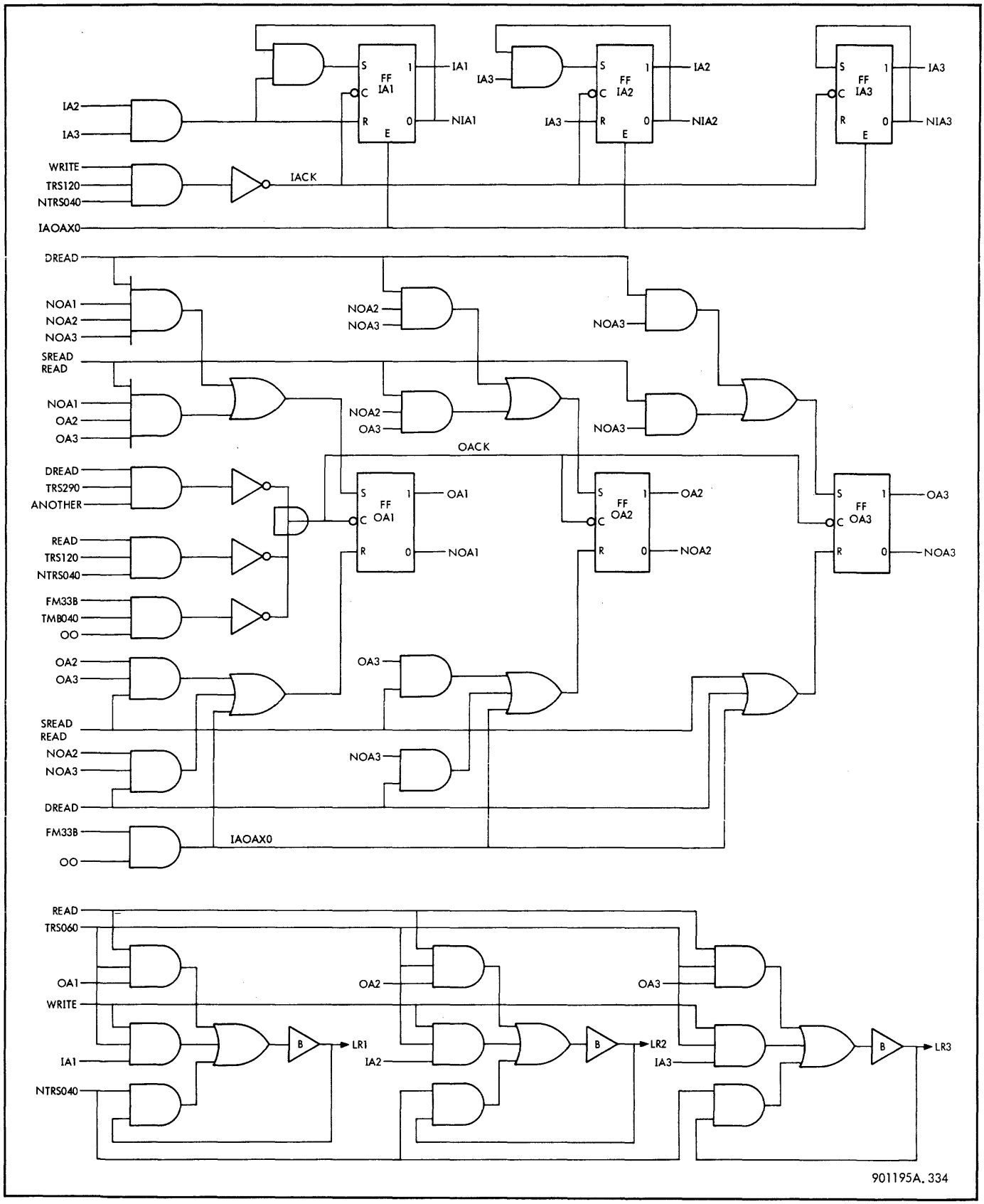


Figure 3-34. OA-, IA-, and LR-Registers Logic Diagram

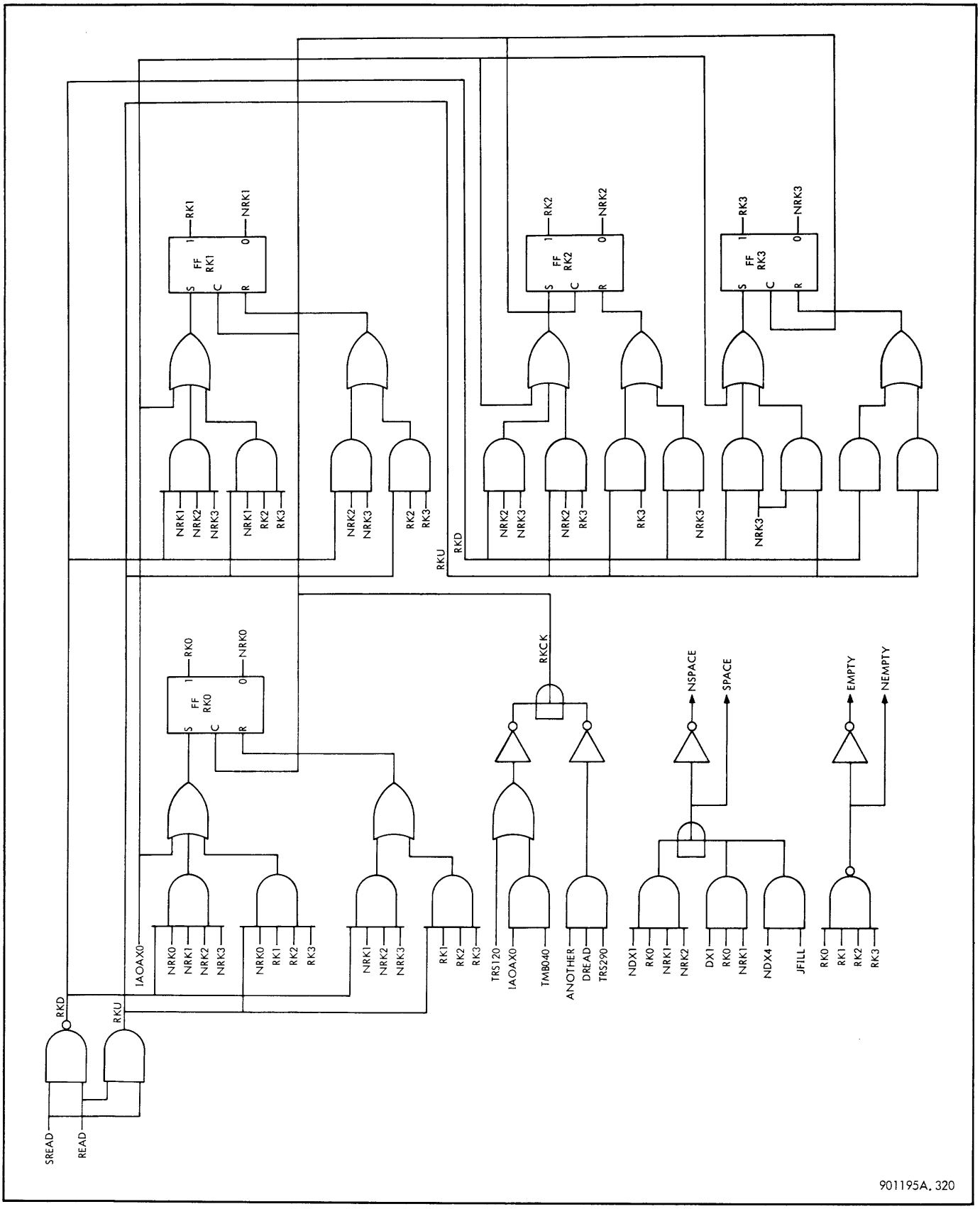


Figure 3-35. HSM Locations Filled (RK) Register, Logic Diagram

Table 3-8. Decoding of RK-Register

RK0	RK1	RK2	RK3	R-Register Locations Filled
1	1	1	1	0*
1	1	1	0	1
1	1	0	1	2
1	1	0	0	3
1	0	1	1	4†
1	0	1	0	5†
1	0	0	1	6†**
1	0	0	0	7†**
0	1	1	1	8††

*Empty

†NSPACE (DX1)

**NSPACE (NDX1)

††Full

JD8 (see figure 3-36). Bits 0 through 7 of all bytes contain data, and bit 8 of all bytes (JA8, JB8, JC8, and JD8) contains the byte presence indicator. During data out service, the J-register receives data from the M-register and the byte presence indicators from the byte presence indicator (H) register. During data in service, the J-register receives data from the I/O-register, and each byte presence bit is set true. Figure 3-37 is a logic diagram of the byte presence bits of the J-register.

3-47 Intermediate Buffer Input (WR) Register

The WR-register consists of 18 buffered latches, W/RA0 through W/RA8 and W/RB0 through W/RB8 (see figure 3-38). This register enables commutation of the data from the J-register into the R-register during data out service when signal DX4 is false, that is, when the device controller interface is one byte or two bytes wide (see figure 3-31).

The data commutation path from the J-register to the R-register is controlled by the byte presence bits (JA8, JB8, JC8, and JD8) of the J-register. The byte presence bits are in turn controlled by signals RJA8, RJB8, RJC8, and RJD8 (see figures 3-37 and 3-38). Listed below is the sequence of SIOP operations that leads to the transfer of the byte presence indicators from the H-register to the byte presence bits of the J-register during data out service with a two-byte interface (NDX4):

a. Signal NMRQ triggers the TM delay line to generate timing signals during FM21A.

b. FM21A.

1. TMA000/1: the WB-register is cleared.

2. TMA000: the WB-register bits (WBA, WBB, WBC, and WBD) are set as required.

3. TMS140: latches MRQ (memory request) and MRQ1 are set by signal BCNT.

4. CNTCK: the byte count and byte address registers are updated as required when signal CNTCK goes true.

c. Signal NJFILL triggers the TM delay line to generate timing signals during FM21B.

d. FM21B.

1. TMS140: the H-register is cleared.

2. TMS160: the contents of the WB-register are transferred to the H-register.

e. Signal NMRQ triggers the TM delay line to generate timing signals during FM22A.

f. Signal NJFILL triggers the TM delay line to generate timing signals during FM22B.

g. FM22B.

1. TMB000/1: the J-register is cleared.

2. TMB040: the byte presence indicators in the H-register are transferred to the byte presence bits of the J-register (JA8, JB8, JC8, and JD8).

3. TMB080: the JFILL latch is set.

The relationship between the state of the byte presence bits of the J-register and the commutation path, and the state of the byte presence bits after each buffer write cycle is shown in tables 3-9 through 3-12.

The data commutation paths are defined by the following logic equations:

RAXJA = NRAXJB NRAXJC1 NRAXJC2 NRAXJD

RAXJB = (DX1 DO) NJA8 JB8

RAXJC = (DX1 DO) NJA8 NJB8 JC8
+ (DX2 DO) NJA8 NJB8

RAXJD = (DX1 DO) NJA8 NJB8 NJC8

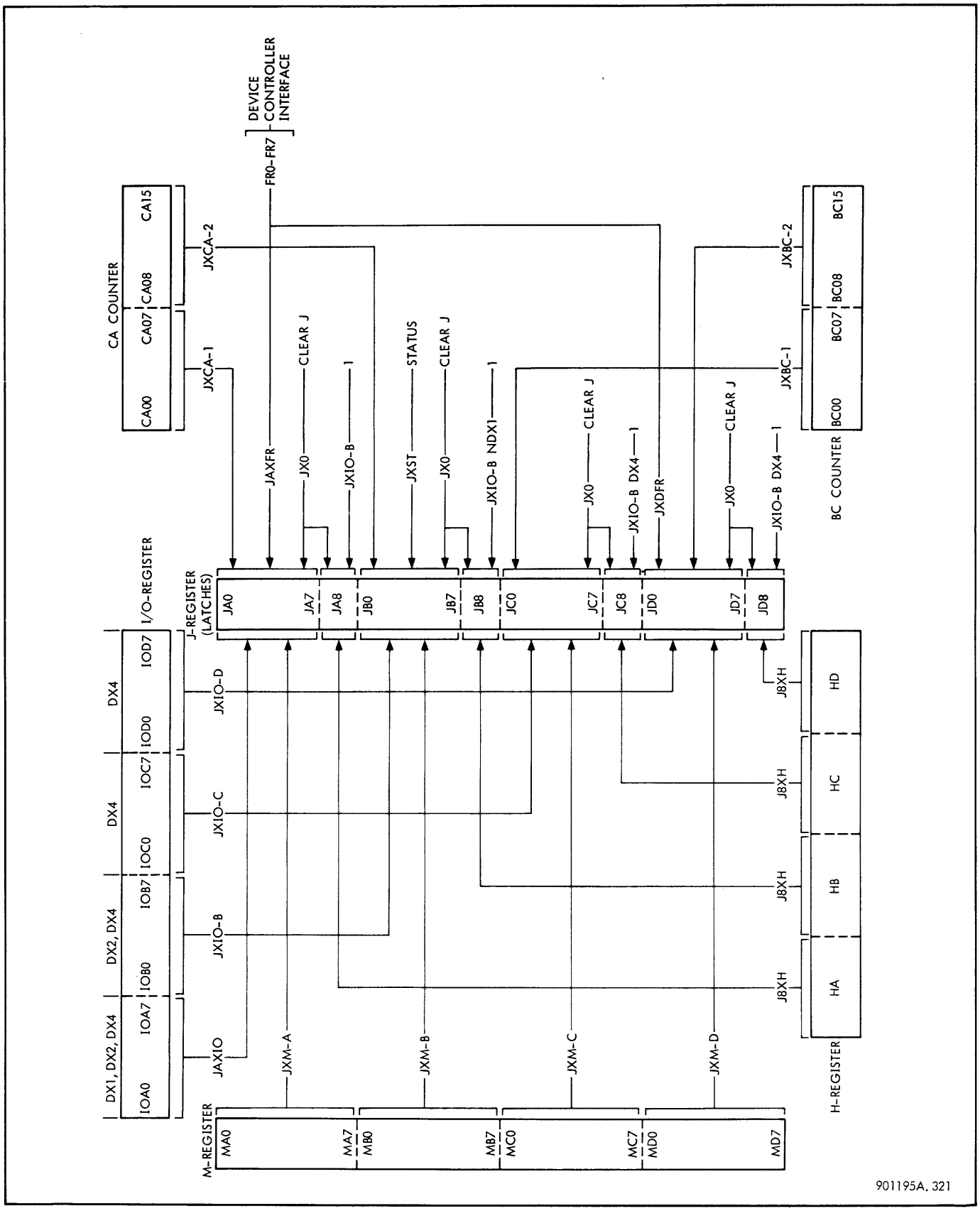


Figure 3-36. J-Register Inputs

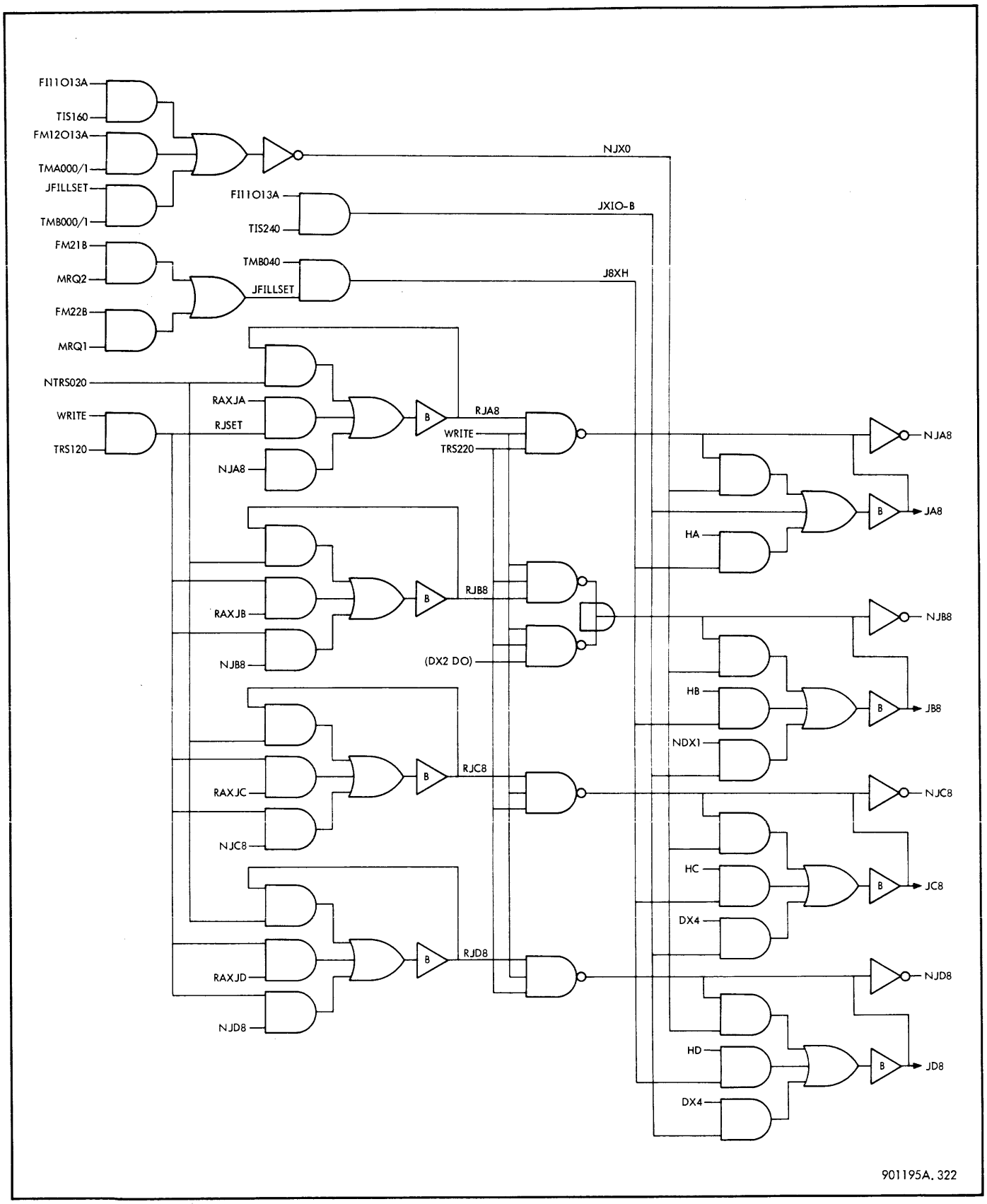


Figure 3-37. J-Register Byte Presence Bits, Logic Diagram

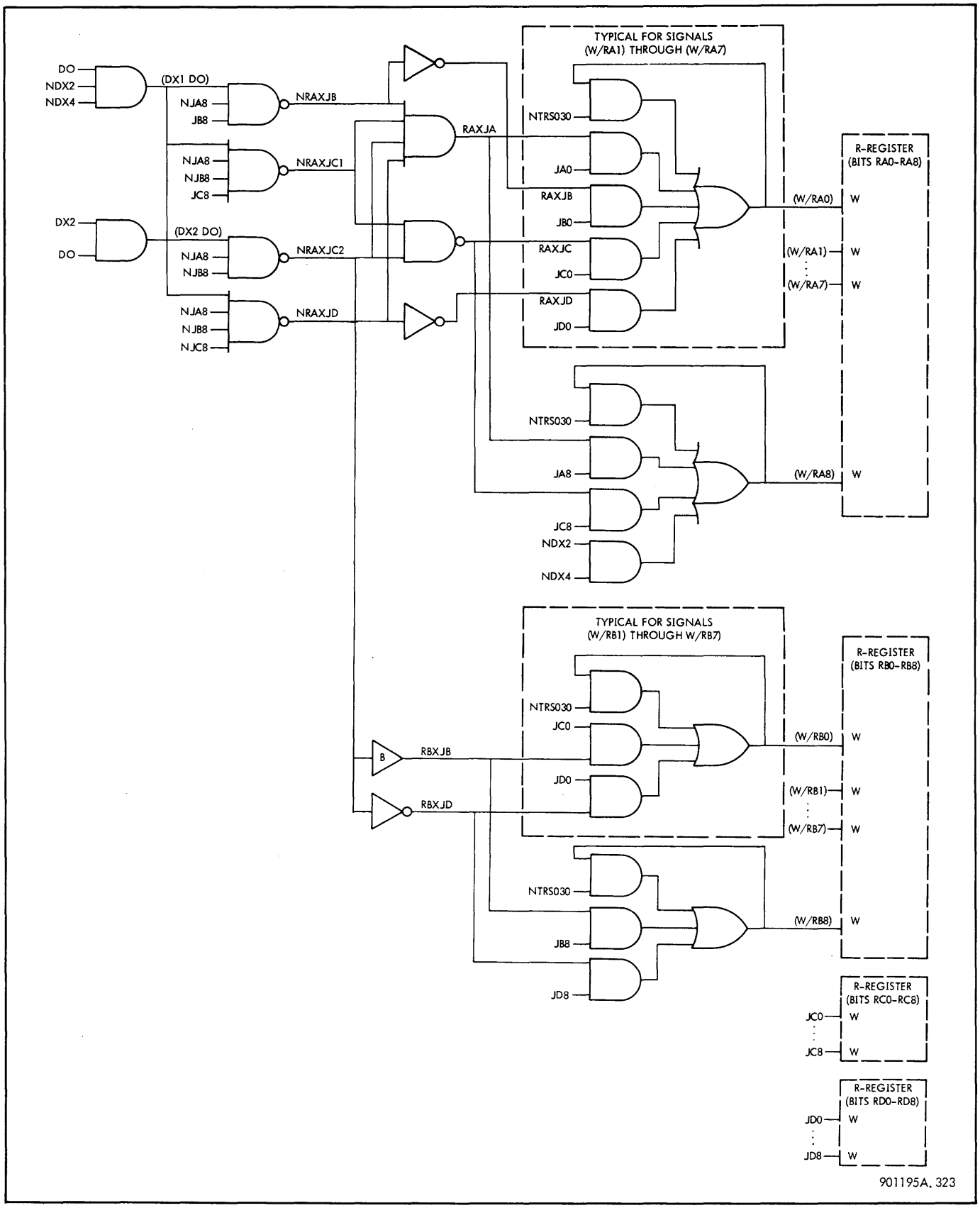


Figure 3-38. Intermediate Buffer (WR) Register Logic Diagram

Table 3-9. Modification of J-Register Byte Presence Bits After Buffer Write Operations (DX1)

INITIAL STATE				NEXT STATE AFTER			
JA8	JB8	JC8	JD8	First Write	Second Write	Third Write	Fourth Write
1	0	0	0	0 0 0 0			
1	1	0	0	0 1 0 0	0 0 0 0		
1	1	1	0	0 1 1 0	0 0 1 0	0 0 0 0	
1	1	1	1	0 1 1 1	0 0 1 1	0 0 0 1	0 0 0 0
0	1	0	0	0 0 0 0			
0	1	1	0	0 0 1 0	0 0 0 0		
0	1	1	1	0 0 1 1	0 0 0 1	0 0 0 0	
0	0	1	0	0 0 0 0			
0	0	1	1	0 0 0 1	0 0 0 0		
0	0	0	1	0 0 0 0			

Notes: 1. Data out service
2. One byte wide interface (DX1)
3. Initial states shown are after transfer of H-register to J-register

Table 3-10. Data Commutation Path from J-Register to R-Register

INITIAL STATE				COMMUTATION PATH FOR			
JA8	JB8	JC8	JD8	First Write	Second Write	Third Write	Fourth Write
1	0	0	0	RAXJA			
1	1	0	0	RAXJA	RAXJB		
1	1	1	0	RAXJA	RAXJB	RAXJC	
1	1	1	1	RAXJA	RAXJB	RAXJC	RAXJD
0	1	0	0	RAXJB			
0	1	1	0	RAXJB	RAXJC		
0	1	1	1	RAXJB	RAXJC	RAXJD	
0	0	1	0	RAXJC			
0	0	1	1	RAXJC	RAXJD		
0	0	0	1	RAXJD			

Notes: 1. Data out service
2. One byte wide interface (DX1)

Table 3-11. Modification of J-Register Byte Presence Bits After Buffer Write Operations (DX2)

INITIAL STATE				NEXT STATE AFTER	
JA8	JB8	JC8	JD8	First Write	Second Write
1	0	0	0	0 0 0 0	
1	1	0	0	0 0 0 0	
1	1	1	0	0 0 1 0	0 0 0 0
1	1	1	1	0 0 1 1	0 0 0 0
0	1	0	0	0 0 0 0	
0	1	1	0	0 0 1 0	0 0 0 0
0	1	1	1	0 0 1 1	0 0 0 0
0	0	1	0	0 0 0 0	
0	0	1	1	0 0 0 0	
0	0	0	1	0 0 0 0	

Notes: 1. Data out service

2. Two byte wide interface (DX2)

3. Initial states shown are after transfer of H-register to J-register

Table 3-12. Data Commutation Path from J-Register to R-Register

INITIAL STATE				COMMUTATION PATH FOR	
JA8	JB8	JC8	JD8	First Write	Second Write
1	0	0	0	RAXJA, RBXJB	
1	1	0	0	RAXJA, RBXJB	
1	1	1	0	RAXJA, RBXJB	RAXJC, RBXJD
1	1	1	1	RAXJA, RBXJB	RAXJC, RBXJD
0	1	0	0	RAXJA, RBXJB	
0	1	1	0	RAXJA, RBXJB	RAXJC, RBXJD

(Continued)

Table 3-12. Data Commutation Path from J-Register to R-Register (Cont.)

INITIAL STATE				COMMUTATION PATH FOR	
JA8	JB8	JC8	JB8	First Write	Second Write
0	1	1	1	RAXJA, RBXJB	RAXJC, RBXJD
0	0	1	0	RAXJC, RBXJD	
0	0	1	1	RAXJC, RBXJD	
0	0	0	1	RAXJC, RBXJD	

Notes: 1. Data out service

2. Two byte wide interface (DX2)

3-48 Byte Presence Intermediate (H) Register

The H-register consists of four buffered latches, HA, HB, HC, and HD (figure 3-39). The H-register provides intermediate storage of the byte presence indicators between the WB-register and the J-register during data out service. For example, during FM21A TMA000 (paragraph 3-47b2), the setting of the WB-register is controlled by the byte count and byte address in the BC- and BA-registers respectively. Following this, a core memory access is made and the BC- and BA-registers are updated. During FM21B TSM160, the byte indicators in the WB-register are transferred to the H-register (paragraph 3-47 d2), where they are saved until FM22B TMB040, when they are transferred to the byte presence bits of the J-register (paragraph 3-47g2). The WB-register at this time contains the updated byte presence indicators.

3-49 Byte Presence Input (WB) Register

The WB-register consists of four buffered latches, WBA, WBB, WBC, and WBD (figure 3-40). The WB-register generates byte presence indicators for the buffer during data out service, and for core memory during data in service (figure 3-41). The WB-register also provides the input to the count generator logic.

The generation of the byte indicators as a function of the state of the BA- and BC-registers is shown in table 3-13. Signals BA17 and BA18 are the two LSBs of the byte address. The byte indicator generation logic for data out service follows:[1]

S/WBA = [(FM21A + FM22A) TMA000 NDI] BAA0

S/WBB = [(FM21A + FM22A) TMA000 NDI] NBA17 N(BAA0 BCC1)

[1]The part of these equations not included in brackets is shown in table 3-13.

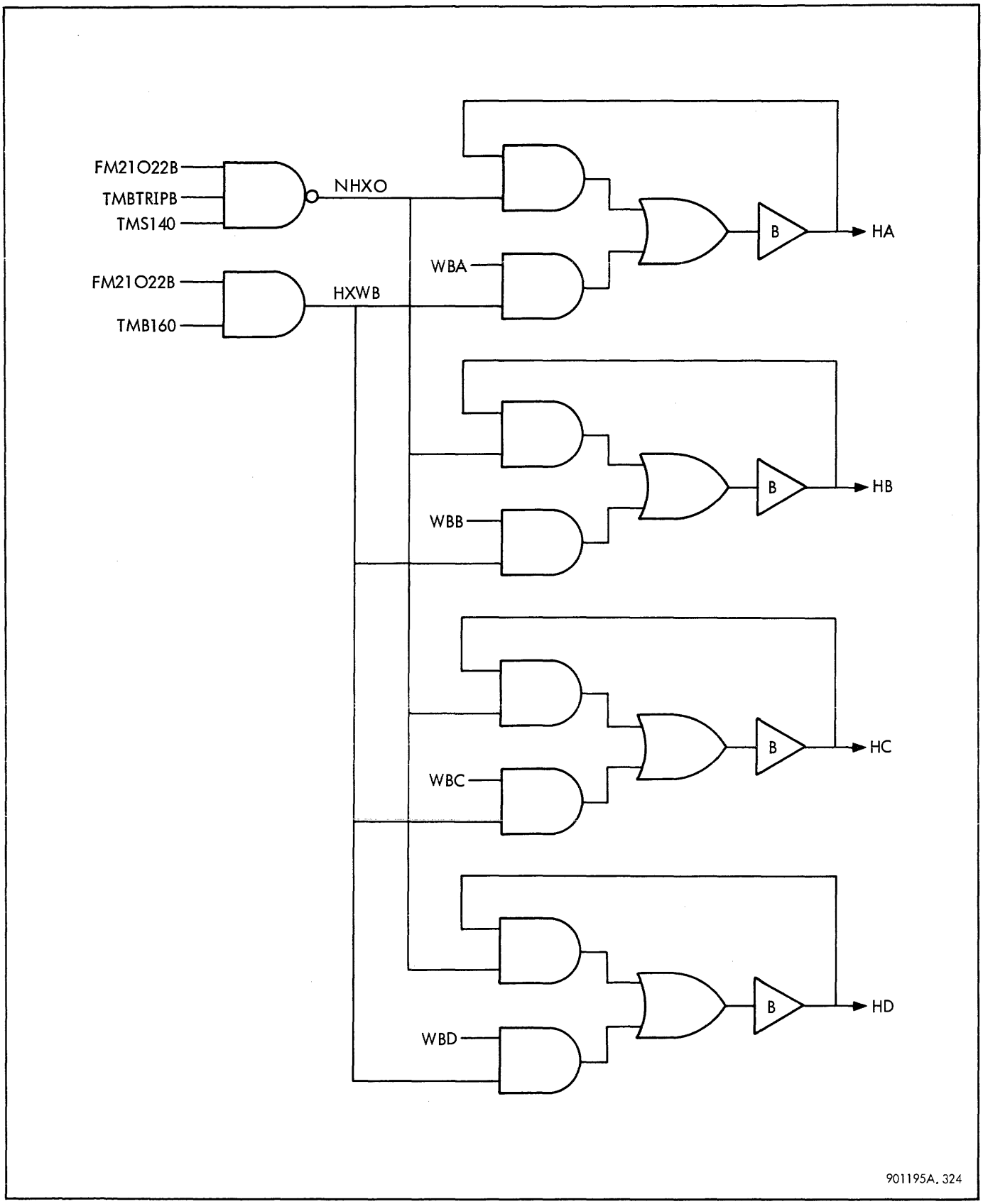


Figure 3-39. Byte Presence Intermediate (H) Register Logic Diagram

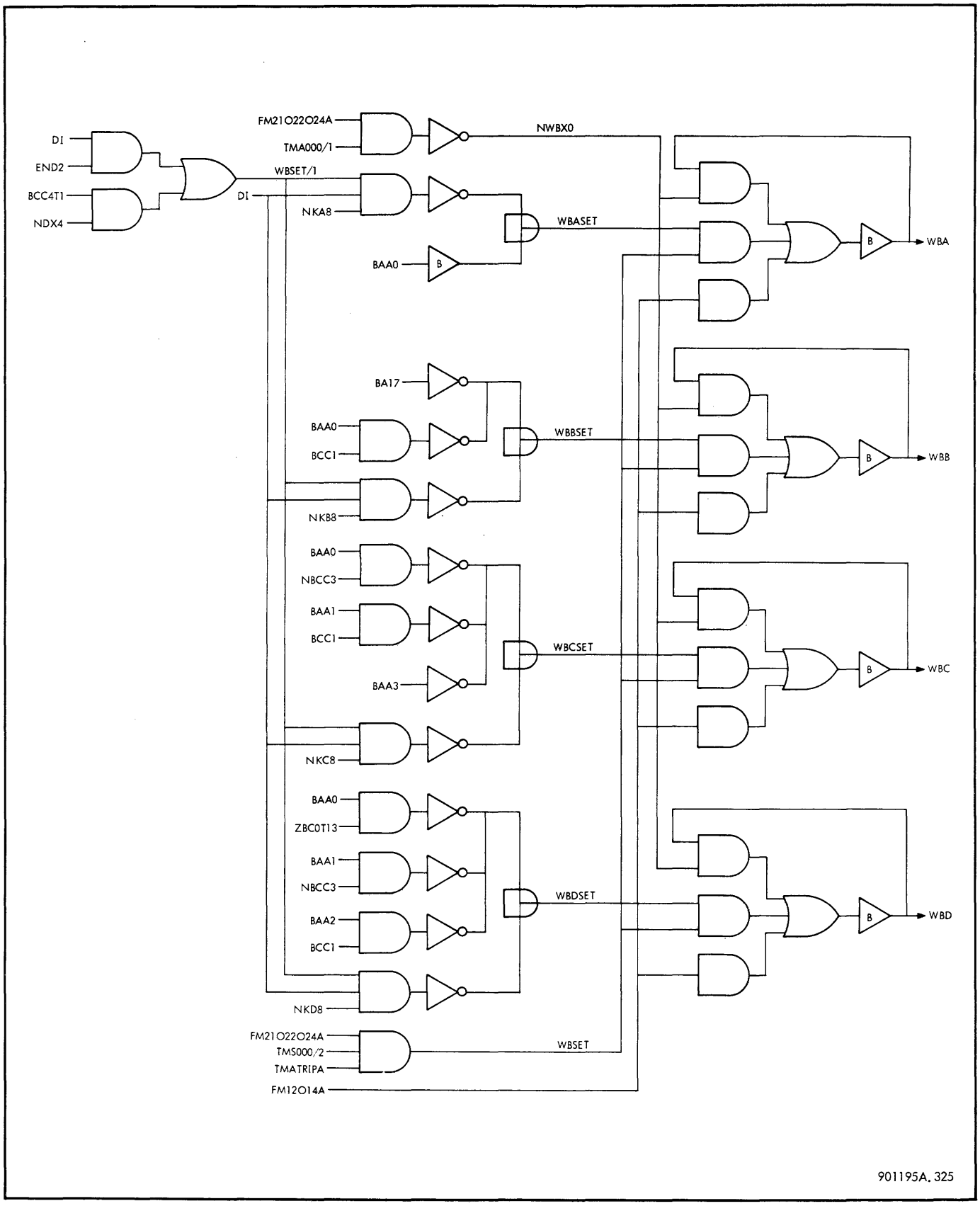


Figure 3-40. Byte Presence Input (WB) Register Logic Diagram

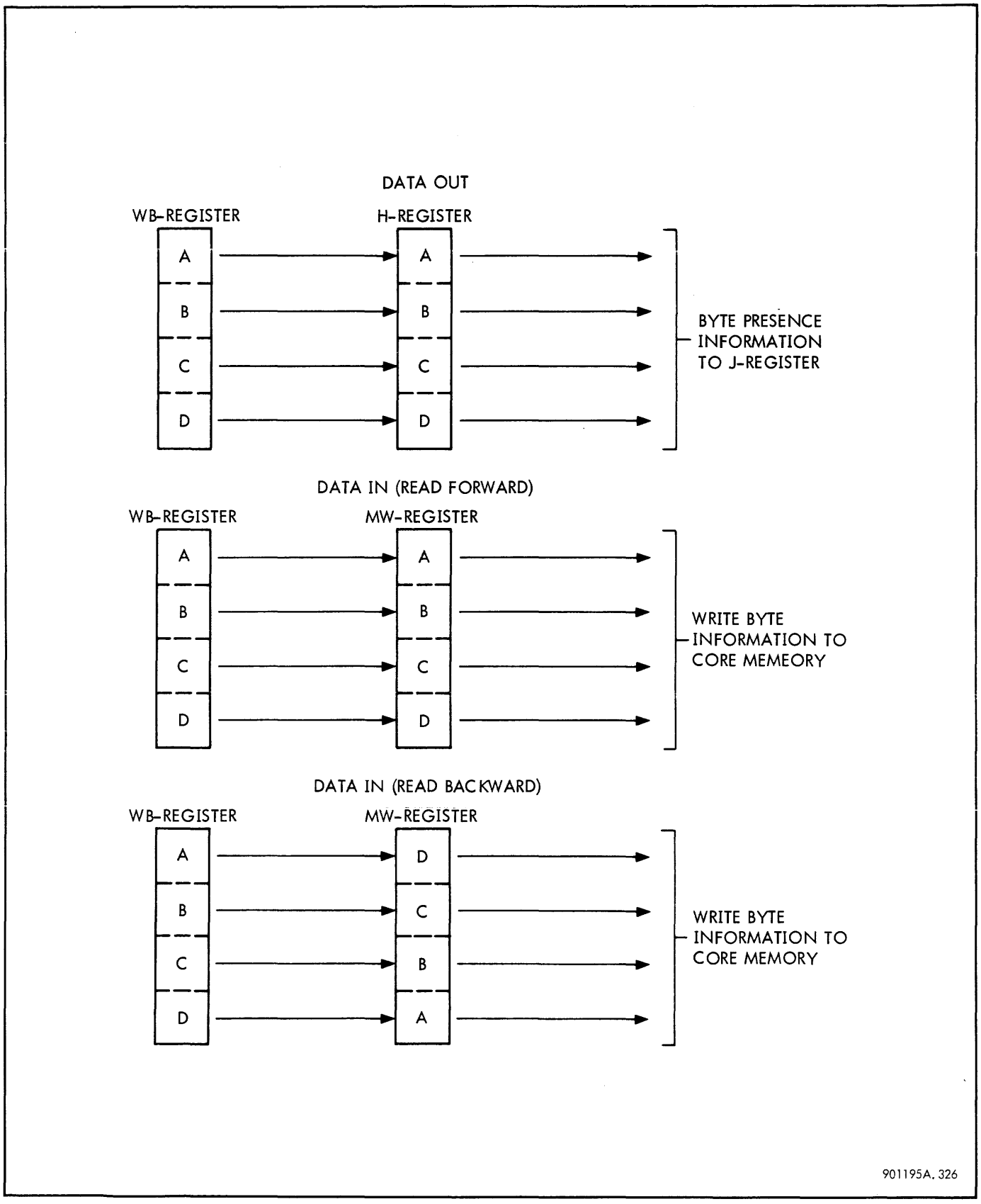


Figure 3-41. Transfer Paths from WB-Register

S/WBC = [(FM21A + FM22A) TMA000 NDI] NBAA3 N(BAA0 NBCC3) N(NAA2 BCC1)

S/WBD = [(FM21A + FM22A) TMA000 NDI] N(BAA0 ZBC0T13) N(BAA1 NBCC3) N(BAA2 NBCC1)

The byte indicator generation logic for data in service follows:

S/WBA = (FM24A TMA000) N [NKA8 DI (NDX4 BCC4T1 + END2)] BAA0

S/WBB = (FM24A TMA000) N [NKB8 DI (NDX4 BCC4T1 + END2)] NBA17 N(BAA0 BCC1)

S/WBC = (FM24A TMA000) N [NKC8 DI (NDX4 BCC4T1 + END2)] NBAA3 N(BAA0 NBCC3) N(BAA1 NBCC1)

S/WBD = (FM24A TMA000) N [NKD8 DI (NDX4 BCC4T1 + END2)] N(BAA0 ZBC0T13) N(BAA1 NBCC3) N(BAA2 BCC1)

Note

Signal END2 is true when end data is true and the buffer is empty.

Signal NDX4 is true when the interface is not four bytes wide.

Signal BCC4T1 is true when the byte count is less than five.

3-50 Count Generation Logic

The count generation logic consists of gating that generates count signals CBC1, CBC2, CBC3, and CBC4 (figure 3-42). These signals are applied to the BA- and BC-registers and define the number of counts required for their updating

Table 3-13. Byte Presence Indicator Generation

BYTE COUNT	NBA17 NBA18 (BAA0) WB				NBA17 BA18 (BAA1) WB				BA17 NBA18 (BAA2) WB				BA17 BA18 (BAA3) WB			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
0*t																
1*t	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
2*t	1	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1
3*	1	1	1	0	0	1	1	1	0	0	1	1	0	0	0	1
4	1	1	1	1	0	1	1	1	0	0	1	1	0	0	0	1
5	1	1	1	1	0	1	1	1	0	0	1	1	0	0	0	1
6	1	1	1	1	0	1	1	1	0	0	1	1	0	0	0	1
7	1	1	1	1	0	1	1	1	0	0	1	1	0	0	0	1

*BCC1: ZBC0T13

tNBCC3

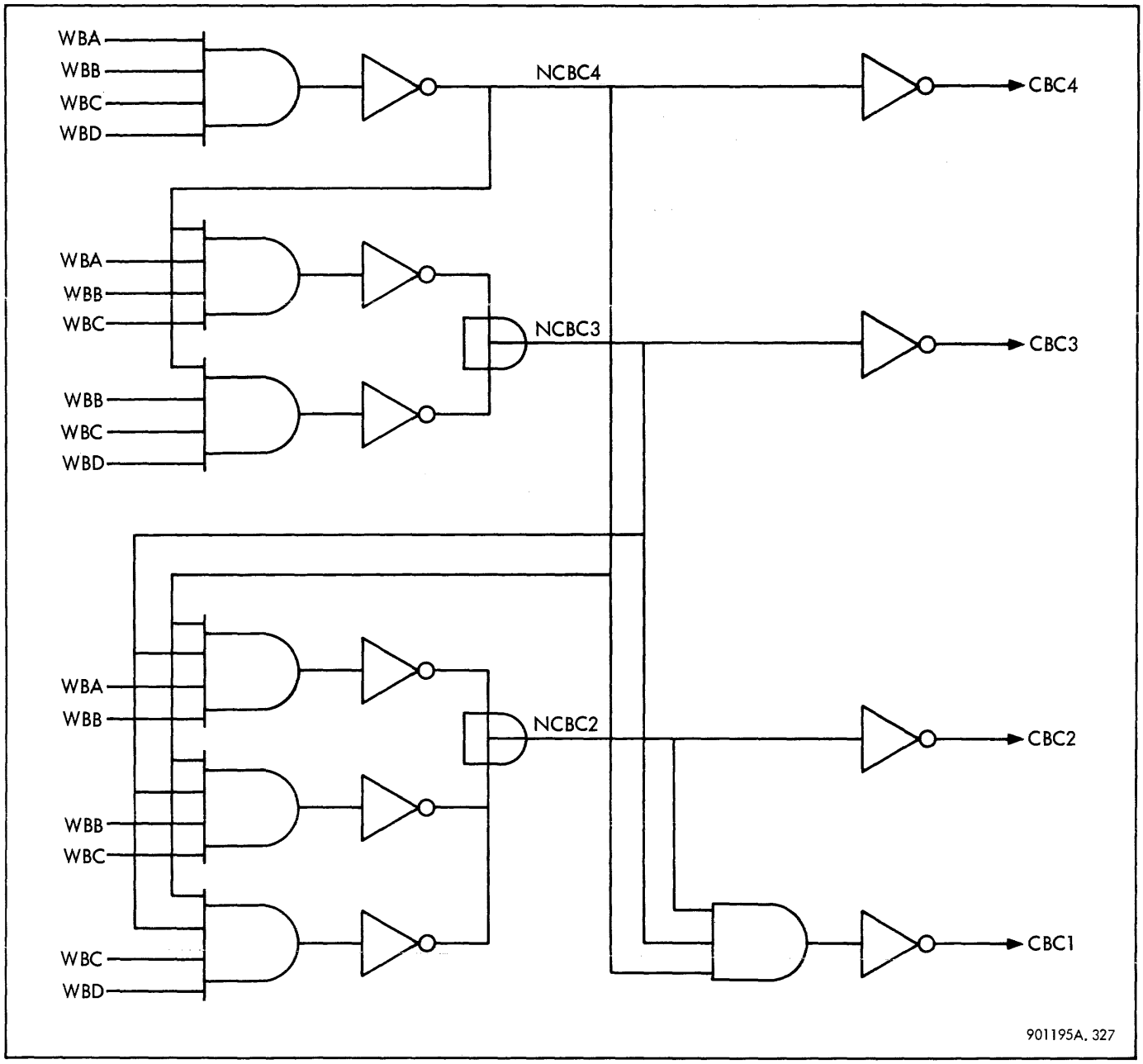


Figure 3-42. Count Generator Logic Diagram

during data in and data out service. The count signals (CBC1 through CBC3) are generated as a function of the WB-register, as shown in table 3-14. Simplified logic equations for the count generator logic follow:

CBC1 = NCBC2 NCBC3 NCBC4

CBC2 = NCBC4 NCBC3 (WBA WBB + WBB WBC + WBC WBD)

CBC3 = NCBC4 (WBA WBB WBC + WBB WBC WBD)

CBC4 = WBA WBB WBC WBD

3-51 Buffer Output (K) Register

The K-register consists of 36 buffered latches, KA0 through KA8, KB0 through KB8, KC0 through KC8, and KD0 through KD8. Bits 0 through 7 of each byte store data (figure 3-43), and bit 8 of each byte contains the byte presence indicator (figure 3-44). The byte presence indicators designate those bytes that contain valid data as a result of a buffer read cycle.

During data out service the K-register supplies data to the I/O-register, and during data in service, the K-register supplies data to the M-register. The state of the byte

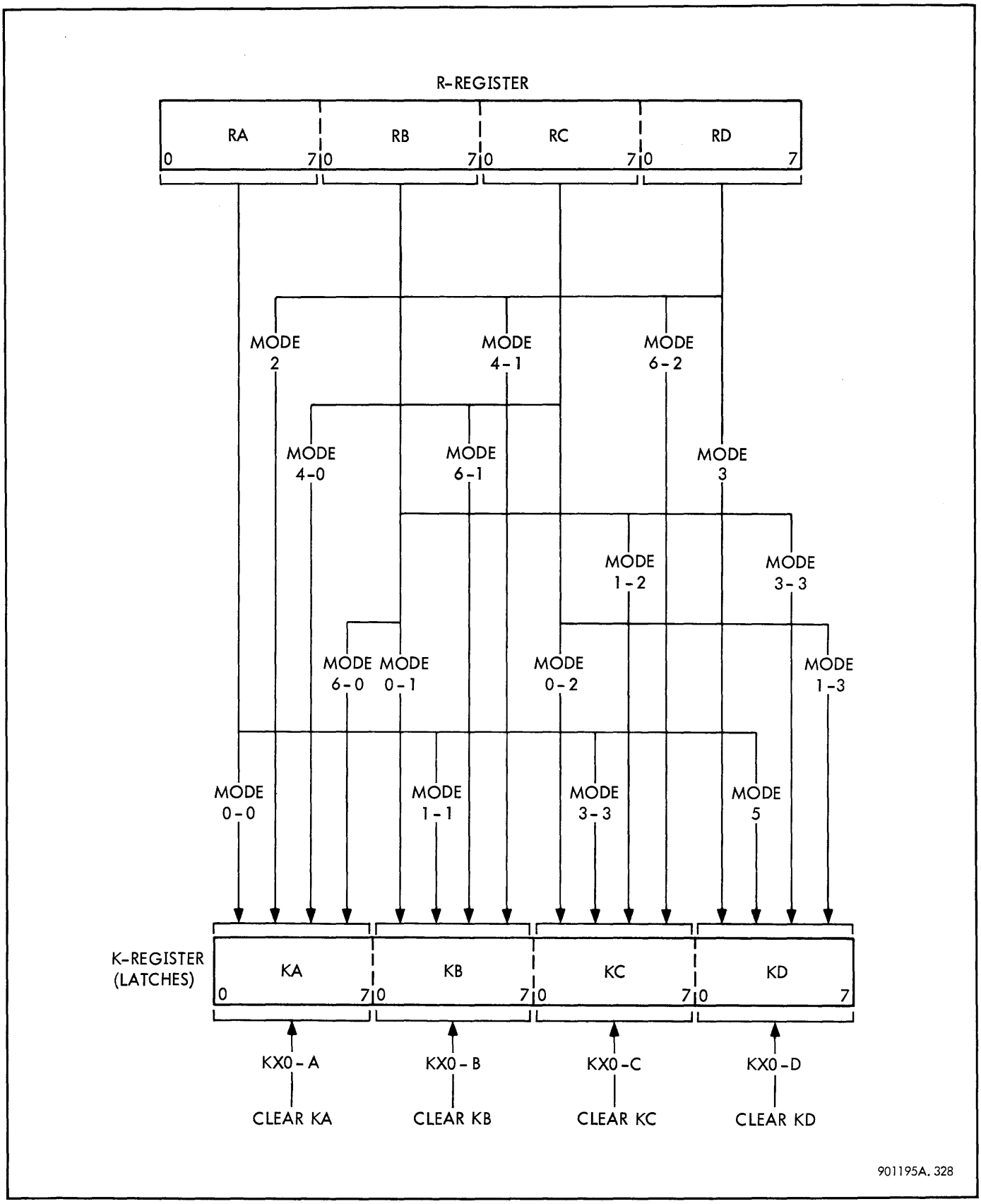


Figure 3-43. K-Register Inputs, Data Bits

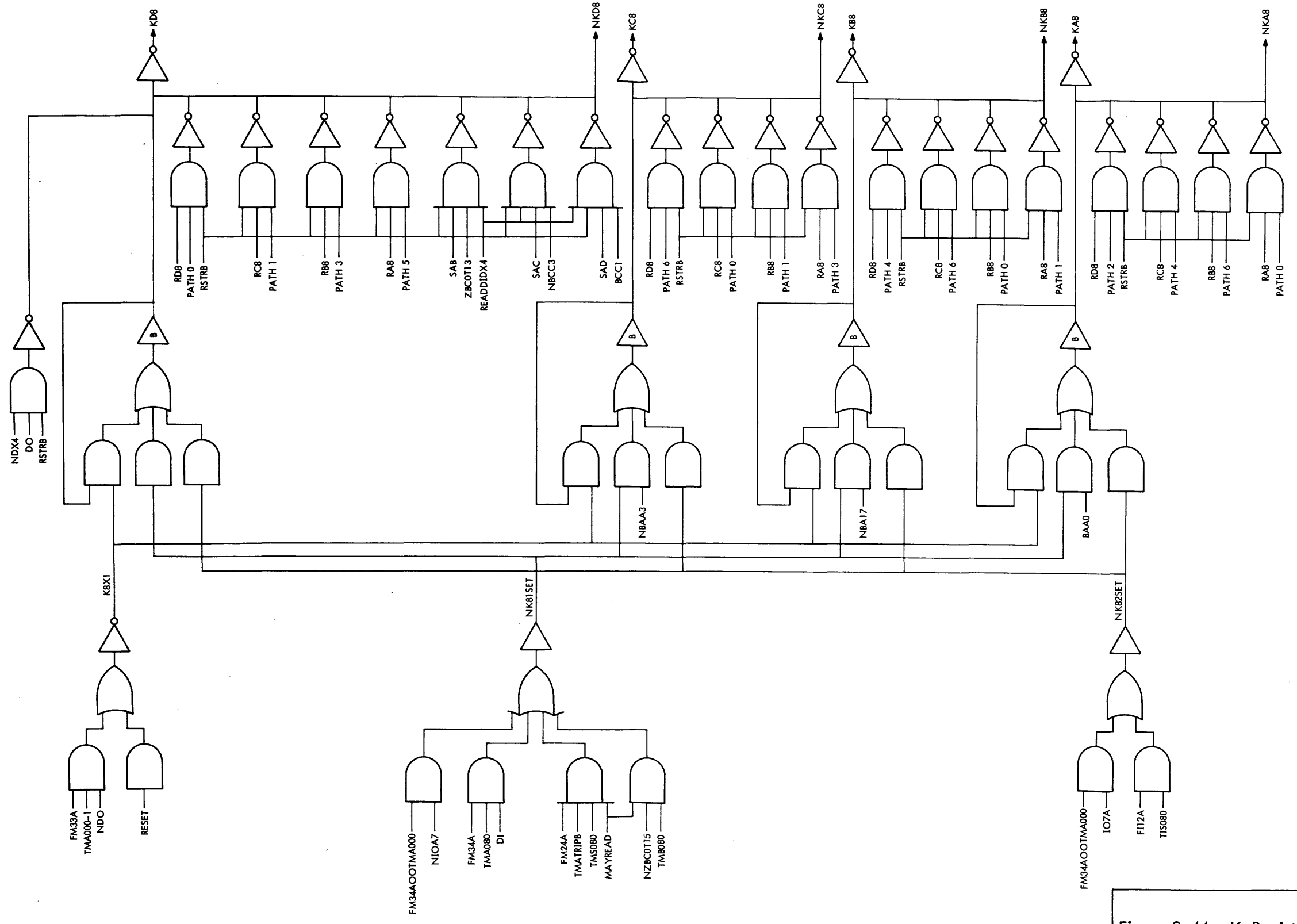


Figure 3-44. K-Register Byte Presence Indicator Bits Logic Diagram

901195A. 329

Table 3-14. Count Generator Signal Definition

BYTE PRESENCE SIGNALS				COUNT SIGNALS
WBA	WBB	WBC	WBD	
1	0	0	0	CBC1 (up 1 or down 1)
0	1	0	0	
0	0	1	0	
0	0	0	1	
1	1	0	0	CBC2 (up 2 or down 2)
0	1	1	0	
0	0	1	1	
1	1	1	0	CBC3 (up 3 or down 3)
0	1	1	1	
1	1	1	1	CBC4 (up 4 or down 4)

presence indicator bits of the K-register provides control information for a number of purposes, including (1) record byte presence indication for subsequent control of the WB-register, (2) designation that an output cycle may start at the device controller interface or an input cycle may start at the core memory interface, and (3) control of the path logic during subsequent read cycles from the same R-register location.

Control of the byte presence bits of the K-register during data out service is described below.

Bits KA8, KB8, KC8, and KD8 are initially reset by signal NK82SET. They are updated during subsequent read cycles as a function of the path logic and the byte presence indicator bits of the R-register, as shown in table 3-15.

Table 3-15. Byte Presence Bit Updating of K-Register

R-REGISTER PRESENCE BIT	PATH						
	5	3	1	0	6	4	2
RA8	KD8	KC8	KB8	KA8			
RB8		KD8	KC8	KB8	KA8		
RC8			KD8	KC8	KB8	KA8	
RD8				KD8	KC8	KB8	KA8

Signals KFILL1 or KFILL2 provide entry into FI12A. Latch KFILL1 is set by the following signals when the device controller interface is one or four bytes wide:

KFILL1 = KD8 DREADTRS290 N(DX2 DO) + ...

Latch KFILL1 is set by the following signals when the device controller interface is two bytes wide:

KFILL1 = KB8 DREADTRS290 DX2 DO + ...

Signal KFILL2 provides entry into FI12A for those cases in which either the byte count has gone to zero and KB8 was not reached (the device controller interface is two bytes wide DX2), or KD8 was not reached (the device controller interface is four bytes wide DX4).

Control of the byte presence bits of the K-register during data in service is described below.

Bits KA8, KB8, KC8, and KD8 are initialized by signal NK81SET and the state of signals BAA0 through BAA3 (the two LSBs of the byte address), as shown below:

Signal	BAA0	BAA1	BAA2	BAA3
KA8	0	1	1	1
KB8	0	0	1	1
KC8	0	0	0	1
KD8	0	0	0	0

During subsequent read cycles the byte presence bits of the K-register are updated according to table 3-15. Byte presence bit KD8 must be true to enter FM24 whether or not KD0 through KD7 contain valid data. When the device controller interface is four bytes wide (DX4), KD8 is also set for certain byte count conditions where the byte count will go to zero in subsequent phase FM24A before KD8 is otherwise reached according to the following:

KD8 = SAB ZBC0T13 (path 6)
+ SAC NBCC3 (path 4, 6)
+ SAD BCC1 (path 2, 4, 6)

Signal KFILL1 or END2 provides entry into FM24A. Latch KFILL1 is set as follows:

If DX4 or (NDX4 NBCC4T1):

KFILL1 = DREAD TRS290 KD8

If NDX4 BCC4T1:

KFILL1 = DREAD TRS290

Signal END2 signifies that end data was raised at the device controller interface and that the data buffer is empty.

3-52 Align Control Logic

The align control logic consists of gating that generates signals SAA, SAB, SAC, and SAD and buffered latches that generate signals KAA, KAB, KAC, and KAD (see figure 3-45). The gating that generates signals SAA through SAD decodes the state of the presence bits of the R-register in such a way that only one of these signals is true at any one time, as shown below:

Signal	RA8	RB8	RC8	RD8
SAA	1	X	X	X
SAB	0	1	X	X
SAC	0	0	1	X
SAD	0	0	0	1

Signals SAA through SAD designate the least significant byte of the R-register that contains data. For example, if signal SAC is true, the RC byte contains data and the RA and RB bytes do not.

The latches that generate signals KAA through KAD are controlled by the presence indicator bits of the K-register, as shown below, and designate the least significant K-register byte with space. For example, if KAC is true, the two least significant bytes (KA and KB) contain data, and the third least significant byte (KC) is empty. Only one output may be true at a time. The output signals from the align control logic are applied to the path logic.

Signal	KA8	KB8	KC8	KD8
KAA	0	X	X	X
KAB	1	0	X	X
KAC	1	1	0	X
KAD	1	1	1	X

3-53 PATH and MODE Logic

The PATH logic consists of gating that generates signals PATH0 through PATH6. Input signals to the PATH logic consists of the output signals from the align control logic. The PATH signals are applied to the input of the MODE logic and also to the presence bits of the K-register. (See figure 3-46.) The PATH logic controls the path the data takes from the R-register to the K-register during buffer read cycles. Generation of the PATH signals as a function of the align control signals is shown below, following the logic equations.

PATH0 = SAA KAA + SAB KAB + SAC KAC + SAD KAD

PATH1 = SAA KAB + SAB KAC + SAC KAD

PATH2 = SAD KAA

PATH3 = SAA KAC + SAB KAD

PATH4 = SAC KAA + SAD KAB

PATH5 = SAA KAD

PATH6 = SAB KAAA + SAC KAB + SAD KAC

Signal	SAA	SAB	SAC	SAD
KAA	0	6	4	2
KAB	1	0	6	4
KAC	3	1	0	6
KAD	5	3	1	0

The MODE signals are derived from the PATH signals, interface width signals DX4 and NDX1, and the buffer read strobe RSTRB, as shown by the following logic equations:

MODE0-0 = PATH0 RSTRB

MODE0-1 = PATH0 RSTRB NDX1

MODE0-2 = PATH0 RSTRB DX4

MODE0-3 = PATH0 RSTRB DX4

MODE1-1 = PATH1 RSTRB

MODE1-2 = PATH1 RSTRB NDX1

MODE1-3 = PATH1 RSTRB DX4

MODE2 = PATH2 RSTRB

MODE3-2 = PATH3 RSTRB

MODE3-3 = PATH3 RSTRB NDX1

MODE4-0 = PATH4 RSTRB

MODE4-1 = PATH4 RSTRB

MODE5 = PATH5 RSTRB

MODE6-0 = PATH6 RSTRB NDX1

MODE6-1 = PATH6 RSTRB DX4

MODE6-2 = PATH6 RSTRB DX4

The MODE signals control the transfer of data from the R-register to the K-register, as shown in figure 3-47. The data is aligned from a particular byte position of the R-register to a particular byte position of the K-register

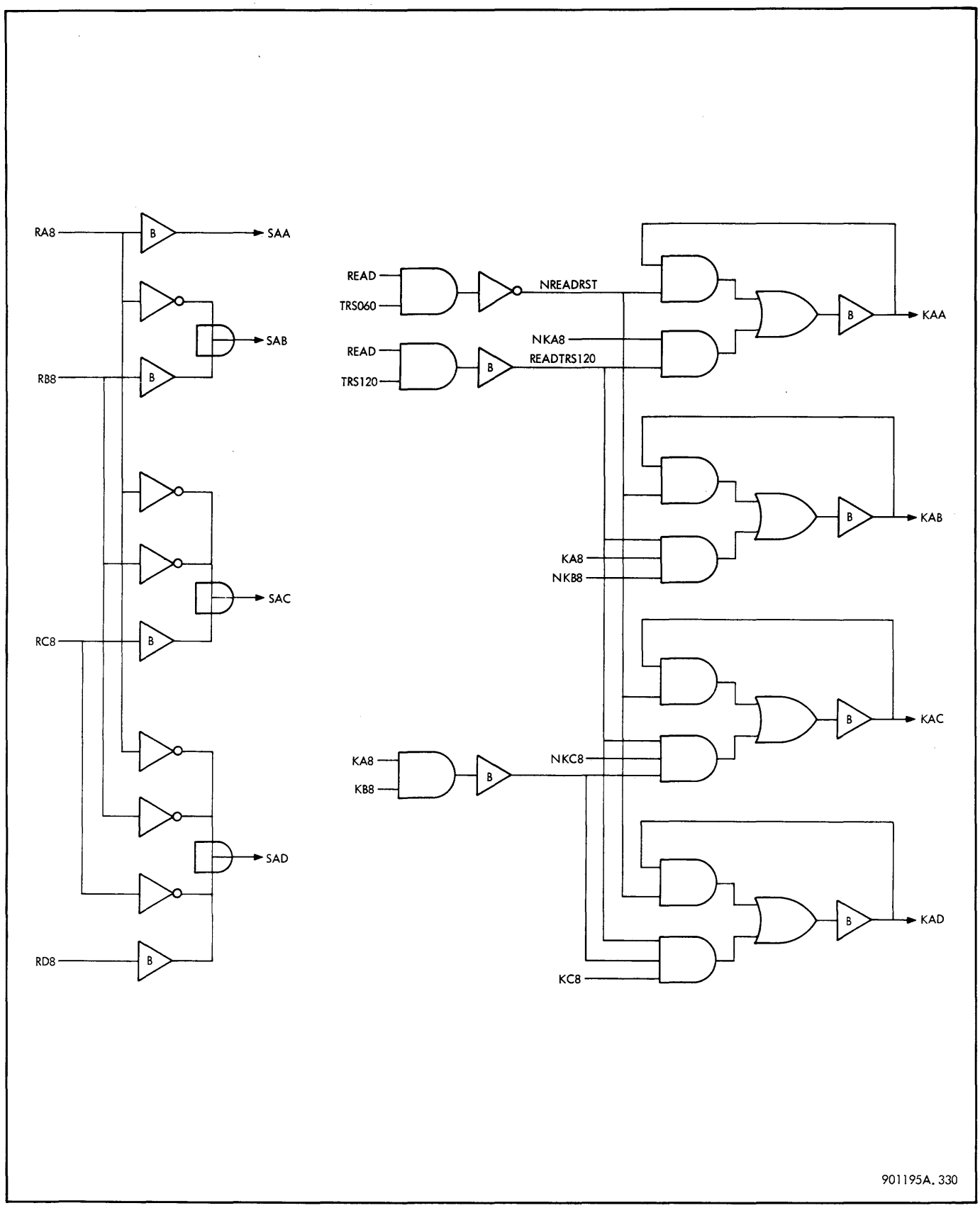


Figure 3-45. Align Control Logic Diagram

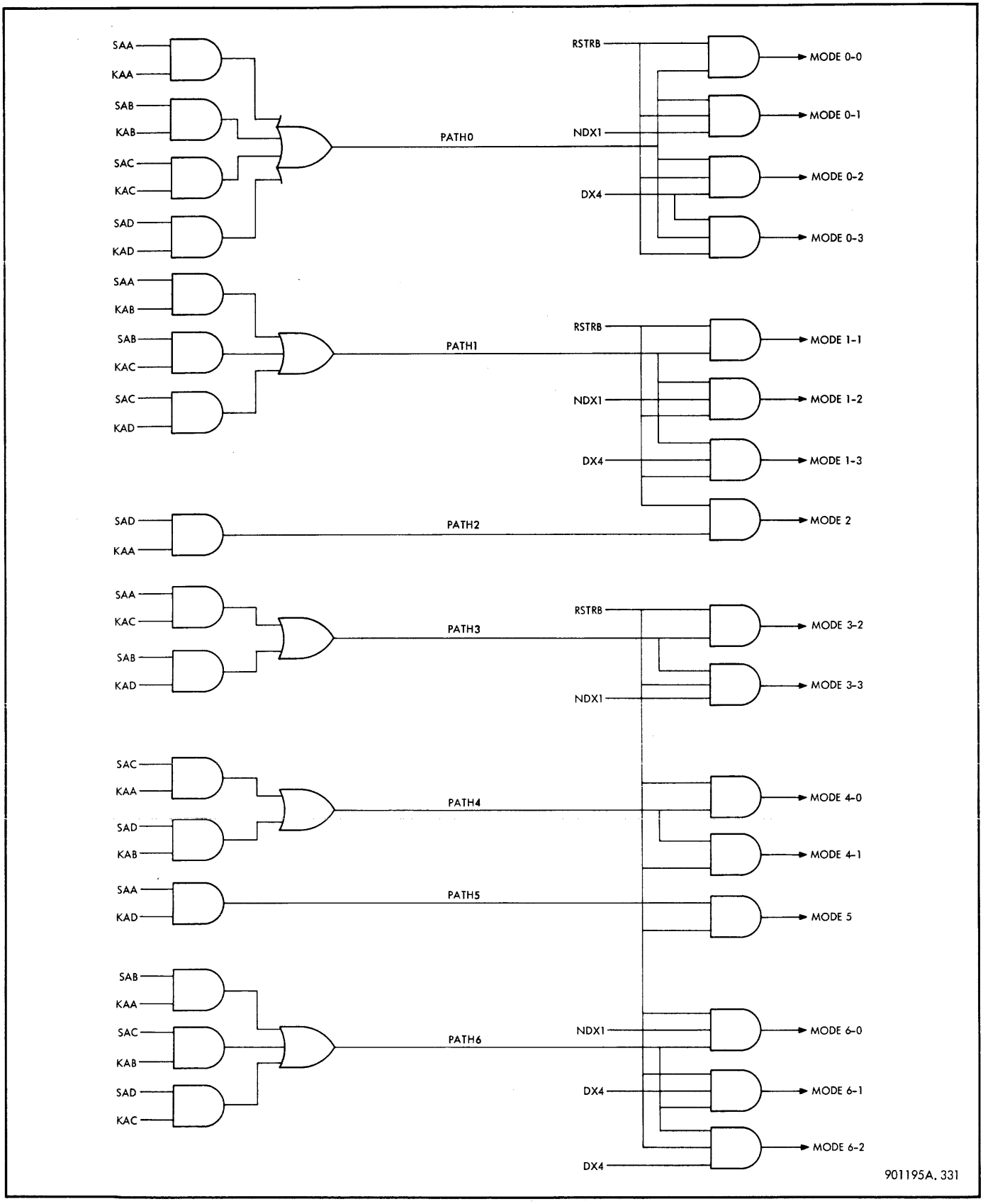


Figure 3-46. PATH and MODE Logic Diagram

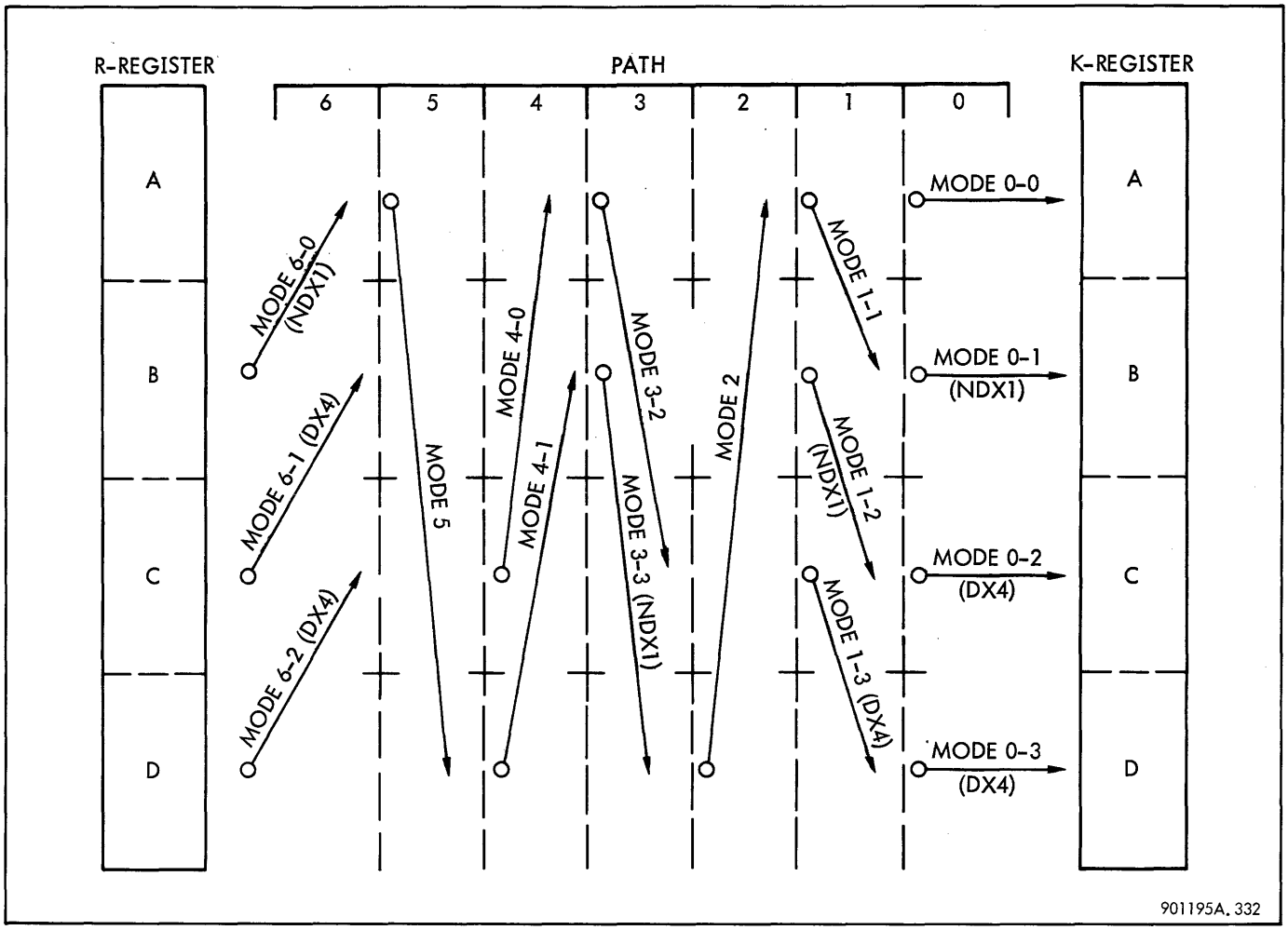


Figure 3-47. R-Register to K-Register Transfer Paths

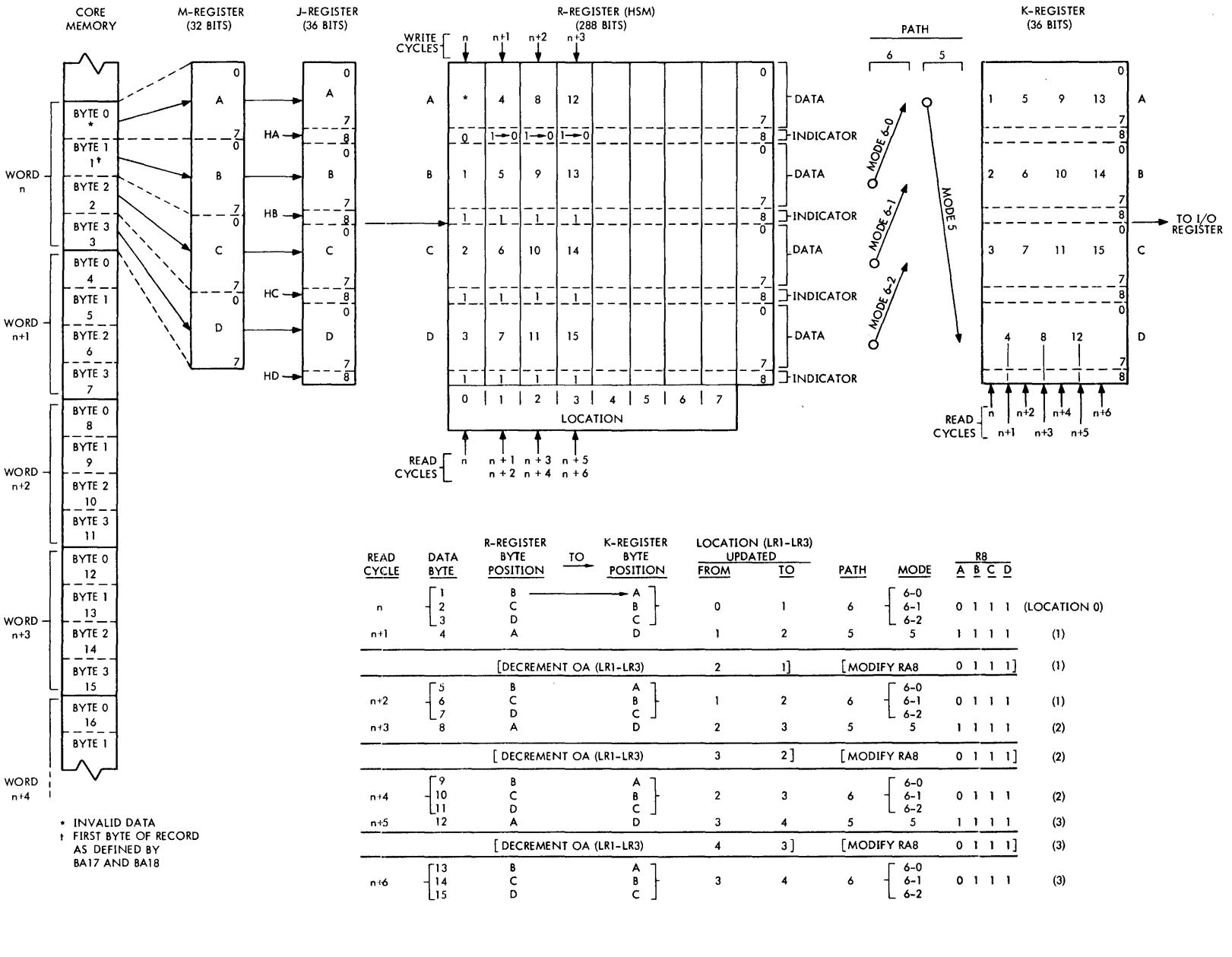
according to the MODE signals. For example, if signal PATH0 is true and the interface is four bytes wide (DX4), signals MODE0-0 through MODE0-3 will be true. Under these conditions, the data in byte positions A through D of the R-register is transferred to the corresponding byte position of the K-register.

Another example: during data out service, if PATH0 is true and the interface is one byte wide (DX1), only signal MODE0-0 will be true, which results in the transfer of the data in byte position A of the R-register to byte position A of the K-register. In this case, byte position A of the R-register is the only position of the R-register that contains valid data (see figure 3-31).

In the above two examples, no data alignment takes place. During data out service, however, commutation of the data takes place during transfer from the J-register (all four bytes) to byte position A of the R-register when the interface is one byte wide (DX1). In this case, each byte of data from the J-register is written into successive locations of byte position A of the R-register, since the OA-register, and thus the LR-register, are incremented by one with each buffer write cycle.

When any other PATH signal except PAHT0 is true, data alignment takes place. For example, if signal PATH6 is true and the interface is four bytes wide, signals MODE6-0 through MODE6-2 will be true. These signals cause the data in byte positions B through D of the R-register to transfer to byte positions A through C, respectively, of the K-register. If byte position A of the R-register contains valid data, the next read strobe will cause signals PATH5 and, therefore, MODE5 to go true. Signal MODE5 causes the data in byte position A of the R-register to transfer to byte position D of the K-register. The data is thus aligned within the data buffer to start on a word boundary at the device controller interface even though signals BA17 and BA18 specified a byte in core memory that did not start at a word boundary.

Figure 3-48 shows an example of the above case. In the example, the operation is data out, the first byte of a record is assumed to start at core memory word n, byte



READ CYCLE	DATA BYTE	R-REGISTER BYTE POSITION	TO	K-REGISTER BYTE POSITION	LOCATION (LR1-LR3) UPDATED FROM	TO	PATH	MODE	R8 A B C D	
n	1	B	→	A	0	1	6	6-0	0 1 1 1	(LOCATION 0)
	2	C		B				6-1		
	3	D		C				6-2		
n+1	4	A		D	1	2	5	5	1 1 1 1	(1)
		[DECREMENT OA (LR1-LR3)			2	1]	[MODIFY RA8		0 1 1 1]	(1)
n+2	5	B		A	1	2	6	6-0	0 1 1 1	(1)
	6	C		B				6-1		
	7	D		C				6-2		
n+3	8	A		D	2	3	5	5	1 1 1 1	(2)
		[DECREMENT OA (LR1-LR3)			3	2]	[MODIFY RA8		0 1 1 1]	(2)
n+4	9	B		A	2	3	6	6-0	0 1 1 1	(2)
	10	C		B				6-1		
	11	D		C				6-2		
n+5	12	A		D	3	4	5	5	1 1 1 1	(3)
		[DECREMENT OA (LR1-LR3)			4	3]	[MODIFY RA8		0 1 1 1]	(3)
n+6	13	B		A	3	4	6	6-0	0 1 1 1	(3)
	14	C		B				6-1		
	15	D		C				6-2		

Figure 3-48. Sample of Data Transfer Through Data Buffer

position 1, and the device controller interface is four bytes wide (DX4). In addition, the OA- and IA-registers are cleared so that the location specified by the LR-register is location 0. After the first core memory access is made, the first three bytes of the record, as specified by signals BA17 and BA18, are set into byte positions B, C, and D, respectively, of the M-register. Byte 0 of word n is set into byte position A; however, since it is not part of the record, it is invalid data. The data in all four byte positions is parallel transferred to the corresponding byte positions of the J-register, and the presence indicator bits of the J-register are set according to the H-register. In this example, HA, and thus JA8, will be false, and the other indicators will be true.

During the first buffer write cycle (n), the contents of the J-register are written into location 0 of the R-register. Byte position A contains invalid data, and byte positions B, C, and D contain the first three bytes of the record. The presence indicator bits of the J-register are also transferred to the corresponding bit positions of the R-register. Since a zero in any indicator bit of a particular location (byte position A, B, C, or D) inhibits transfer of the data in the associated byte position, invalid data, or data that has already been read, will not be transferred to the K-register during a particular read cycle. Read cycle n, therefore, transfers the data in location 0, bytes B, C, and D, to byte positions A, B, and C respectively of the K-register, since signal PATH6 is true. The next read cycle, n + 1, transfers the fourth byte of the record, in location 1, byte position A of the R-register to byte position D of the K-register, since signal PATH5 is true.

3-54 R-Register Byte Modification Logic

R-register byte presence indicators RA8, RB8, and RC8 may be modified (set to 0) after a read cycle of a data in or data out operation to indicate that data in a particular byte position (or positions) of the addressed location has been used.

This modification is necessary, since a second read cycle may be necessary to empty all byte positions of the addressed location. A zero indicator bit of a particular byte position prevents the data in that position from being read. Signal ANOTHER is true when the same location is read twice in succession.

In addition to modification of the indicator bits of the R-register, the OA-register, which serves as an input to the location register (LR1 through LR3), must also be updated (decremented). This is necessary because it is incremented (to specify the next location) after every read cycle. During buffer read cycles, the LR-register is controlled by the OA-register. The OA-register is decremented when clock signal OACK, controlled by signals ANOTHER, DREAD, and TRS290, goes true.

For example, assume that during data out service when signal DX4 is true, that all four byte positions of location 1 of the R-register contain valid data (bytes 4, 5, 6, and 7 in figure 3-48). The table in the figure describes the sequence of events for the above stated conditions. During read cycle n + 1, the data in byte position A of the R-register is transferred via PATH5 to byte position D of the K-register, and the presence indicator bit of byte position D (KD8) is set. The OA-register is incremented from location 1 to location 2.

Since bytes 5, 6, and 7 must also be read from location 1, the OA-register is decremented to again specify location 1, and the indicator bit of byte position A of R-register location 1 must be modified (from true to false). This modification will cause signal PATH6, and thus signals MODE6-0, MODE6-1, and MODE6-2, to go true when the next read cycle (n + 2) occurs. During read cycle n + 2, bytes 5, 6, and 7, in byte positions B, C, and D respectively of location 1 are transferred to byte positions A, B, and C of the K-register. After the data in the four byte positions of the K-register has been used, the K-register is cleared in preparation for receipt of the next four bytes of data from the R-register.

The conditions for updating the presence indicator bits of the R-register during data in and data out service are given in paragraphs 3-55 through 3-58 below.

3-55 DATA OUT SERVICE (FOUR-BYTE INTERFACE). The presence indicator bits of the R-register (RA8 through RD8) are cleared according to the portion of table 3-16 related to four-byte interface if another read cycle is indicated. See latches RAM, RBM1, RBM2, and RCM, in figure 3-30. During data out service, these latches are controlled by signals DO, DREAD, TRS290, and ANOTHER.

3-56 DATA OUT SERVICE (TWO-BYTE INTERFACE). The presence indicator bits of the R-register are cleared according to the portion of table 3-16 relating to two-byte interface if another read cycle is indicated. When the interface is two bytes wide, only byte positions RA and RB of the R-register are used.

3-57 DATA IN SERVICE (FOUR-BYTE INTERFACE). The following four provisions apply during data in service when the interface is four bytes wide:

a. All four byte indicators (RA8, RB8, RC8, and RD8) are written.

b. Indicator RA8 is always false after the initial read operation from any location of byte position RA.

c. Indicator RD8 is always true if another byte is to be read from the same location of byte position RD.

d. Indicators RA8, RB8, and RC8 are modified during the current read cycle (RM1SET) if the byte count is greater than three (table 3-17), or they are modified during subsequent phase FM24A (RM2SET) if the byte count is less than four (table 3-18).

Table 3-16. Byte Indicator Modification During Data Out Service

SIGNAL	STATE OF Ri8, FOUR-BYTE INTERFACE (DX4)															
	I*	A†			I*	A†		I*	A†		I*	A†	I*	A†	I*	A†
		5	3	1		5	3		3	1		5		3		1
RA8	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
RB8	1	1	0	0	1	1	0	1	0	0	1	1	1	0	0	0
RC8	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	0
RD8	1	1	1	1	0	0	0	1	1	1	0	0	1	0	1	1

STATE OF Ri8, TWO-BYTE INTERFACE (DX2)			
SIGNAL	I*	A†	
		1	
RA8	1	0	
RB8	1	1	

*Initial state of Ri8

†State after read cycle involving path shown

Table 3-17. Byte Modification During Data In Service (DX4) (Byte Count Greater Than Three)

SIGNAL	STATE OF Ri8			
	I	A		
		5	3	1
RA8	1	0	0	0
RA8	1	1	0	0
RC8	1	1	1	0
RD8	1	1	1	1

Table 3-18. Byte Modification During Data In Service (DX4) (Byte Count Less Than Four)

COUNT CONDITIONS				RESULT OF MODIFICATION
Signal	CBC3	CBC2	CBC1	
RA8 ↓	X	X	X	RA8, RB8, RC8 RA8, RB8 RA8

Table 3-18. Byte Modification During Data In Service (DX4) (Byte Count Less Than Four) (Cont.)

COUNT CONDITIONS				RESULT OF MODIFICATION
Signal	CBC3	CBC2	CBC1	
NRA8 RB8 ↓	X	X	X	- RB8, RC8 RB8
NRB8 RC8 ↓	X	X	X	- - RC8

3-58 DATA IN SERVICE (TWO-BYTE INTERFACE). During data in service, if the interface is not four bytes wide (NDX4) and the byte count is four or less, the SIOP partially writes at core memory (one byte per memory cycle if the interface is one byte wide [DX1] or a maximum of two bytes per memory cycle if the interface is two bytes wide [DX2]) so that only the correct amount of data is accessed from the data buffer.

The following four provisions apply during data in service when the interface is not four bytes wide:

a. Byte presence indicators RC8 and RD8 are always false.

b. Indicator RA8 is always false after initial read from any location.

c. Indicator RB8 is always true if another read is to occur from the same location.

d. Indicator RB8 is not cleared if signal PATH5 is true and the byte count is greater than three, or if signals RA8 and BCC1 are true and the byte count is less than four.

3-59 ANOTHER Logic

Logic signal ANOTHER designates that the buffer location accessed during the current buffer read cycle will also be accessed during the next buffer read cycle. ANOTHER logic consists of a buffered latch with the inputs shown in figure 3-49. The conditions for another read cycle from the same location are given in table 3-19.

3-60 Data Buffer Timing Sequence

Most of the events that occur in the data buffer are controlled by the TR delay line during either a write or a read cycle. The initialization of the buffer, however, occurs under the control of the TM delay line during the chaining sequence. A summary of the buffer initialize sequence is given below:

a. Signal NRESET clears latches READ and WRITE.

b. Signal RESET sets latch CYCLE, clears latch JFILL, and causes K8X1 to go true.

c. The following occurs during an order out service cycle:

1. Signal OO clears latch DREAD.

2. Signals OO, FM32A, and TMA240 cause KX0 to go true.

Table 3-19. Conditions Required to Set ANOTHER Latch

Latch ANOTHER at READ TRS220	Comments
Data out or data in	
KAB (RA8 RB8 RC8 RD8)	PATH1, DX4 only: data left in RD
KAC (RA8 RB8 RC8	PATH3, DX4 only: data left in RC and possibly RD
+ RB8 RC8 RD8)	PATH1, DX4 only: data left in RD
KAD (RA8 RB8	PATH5, DX4 DO + NDX1 DI): data left in RB and possibly RC, RD
+ RB8 RC8	PATH3, DX4: data left in RC and possible RD
+ RC8 RD8)	PATH1, DX4: data left in RD
Data out, DX2	
DAB (RA8 RB8)	PATH1, DX2: data left in RB
Data in, DX4	
SAA ZBC0T13	Byte count < 4: data left in RD and possibly RC, RB
SAB NBCC3	Byte count > 3: data left in RD and possibly RC
SAC BCC1	Byte count = 1: data left in RD
Data in	
BCC1 (RA8 RB8)	Byte count = 1: If DX2, data left in RB

Note: If DX1 is true, RB8, RC8, and RD8 are always false. If DX2 is true, RC8 and RD8 are always false

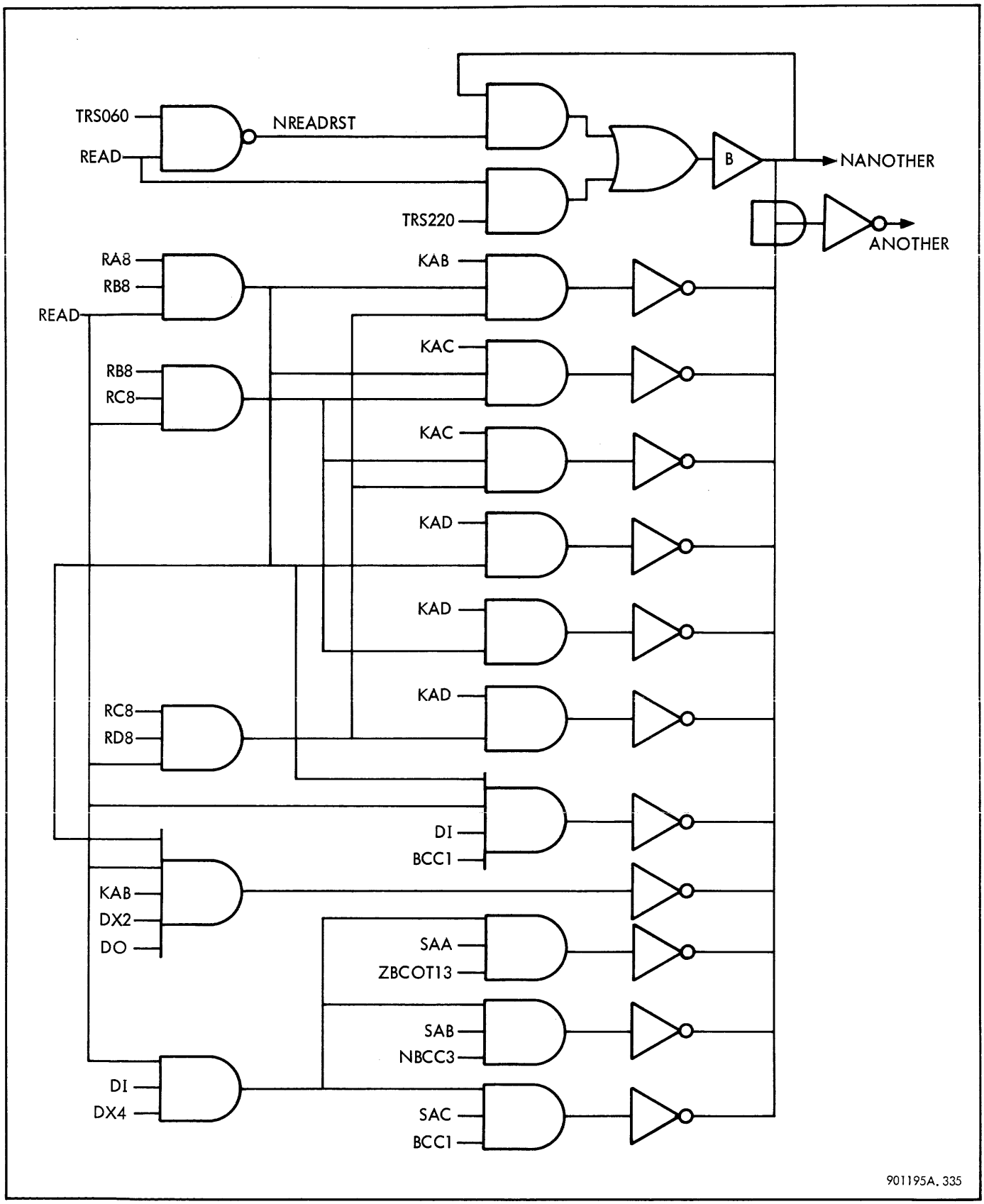


Figure 3-49. ANOTHER Logic Diagram

3. Signals OO FM33A and TMA000 clear latch JFILL.

4. Signals NDO, FM33A, and TMA000 cause K8X1 to go true.

5. Signals O0, FM33A, and TMA160 clear latches SWRITE, SREAD, HWRITE, RAM, RBM1, RBM2, and RCM.

6. Signals O0 and FM33B cause signal IAOAX0 to go true. The IA-register is initialized by signal IAOAX0, and the OA- and RK-registers are initialized by signals IAOAX0 and TMB040.

7. Signals O0, FM34A, TMA000, and NIOA7 cause signal NK81SET to go true during a data in service cycle. Signal NK81SET clears the presence indicator bits (KA8, KB8, KC8, and KD8) of the K-register and clears latch KFILL1.

8. Signals O0, FM34A, and IOA7 cause signal NK82SET to go true during a data out service cycle. Signal NK82SET clears the presence indicator bits of the K-register, and clears latch KFILL1. Latch KFILL2 is cleared by signals FM34A and TMA240.

d. The following occurs during a data chaining sequence:

1. Signals NDO, FM33A, and TMA000 cause signal K8X1 to go true. Signal K8X1 sets the presence indicator bits of the K-register.

2. Signals DI, FM34A, and TMA080 cause signal NK81SET to go true.

3-61 PHASE SEQUENCES

The phase sequences for the five CPU I/O instructions and the four service cycles are described separately in the following paragraphs. The sequence of phases is shown in figure 3-50. The two primary groups of phase sequences are those resulting from a CPU request for service, and those resulting from a device controller request for service.

Both groups start with two preliminary phases, FI01 and FM01. The CPU sequences are terminated in FM14, and the device controller sequences are terminated in FM41. Both groups advance from the termination phase to the first of the preliminary phases, FI01, where the SIOP waits for the next request for service. Since many events are unique to the CPU service requests, and other events are unique to device controller service requests, the SIOP sequences are described under separate headings.

3-62 SIOP Responses to CPU Service Requests

Before the delay line starts for FI01, the SIOP decodes the function code lines and the address lines at the CPU interface. If logic term ME is true when the control strobe (CNSTR) is received from the CPU, the TI delay line is started for FI01. Otherwise, the SIOP passes the control strobe along to the next lower priority IOP as signal CNSTD, or sends the proceed (PR) signal to the CPU if the SIOP is the lowest priority IOP. An exception to the above is made in the case of partial execution of an SIO, TIO, or TDV service request.

3-63 SIOP RESPONSES TO AN SIO FUNCTION CODE. The TI delay line starts when signal TIRQ goes true. Signal TIRQ goes true when signal ME goes true, which indicates that that the SIO was directed to this IOP (MYSIO), that the SIOP is not busy (NSTIOPB1), and that all interrupt status bits are false (NSTI).

To terminate the CPU request for service, the SIOP responds to the control strobe by driving the proceed line and the not condition code lines (NCOND1D and NCOND2D). Signal NCOND1D indicates that for a normally executed SIO, the addressed IOP and device controller recognized the address originally contained in the SIO instruction. Signal NCOND2D indicates that for a normally executed SIO, the SIO will be successful. However, for a partially executed SIO, signal NCOND2D is always true.

Since the SIOP can exchange data with only a single busy device controller, the SIOP exhibits a busy state if the addressed device controller is busy. An SIO is partially executed if directed to a busy SIOP. The SIOP leaves the busy state during an HIO operation (if directed to the device controller whose address is in the A-register) or during an order in service cycle (if the device controller goes to the ready state following the order in sequence). See table 3-20 for a detailed description of the operations that occur during an SIO sequence.

3-64 SIOP RESPONSES TO TIO AND TDV FUNCTION CODES. The events that occur in the SIOP during a TIO or TDV operation are similar to those that occur during an SIO operation, and the sequence of phases is identical (figure 3-50). Signal ME goes true if the function is a TIO or TDV addressed to this SIOP (MYTEST), and signal NSERVICE1 is true (enables a TIO or TDV to be executed normally). For partially executed TIO or TDV

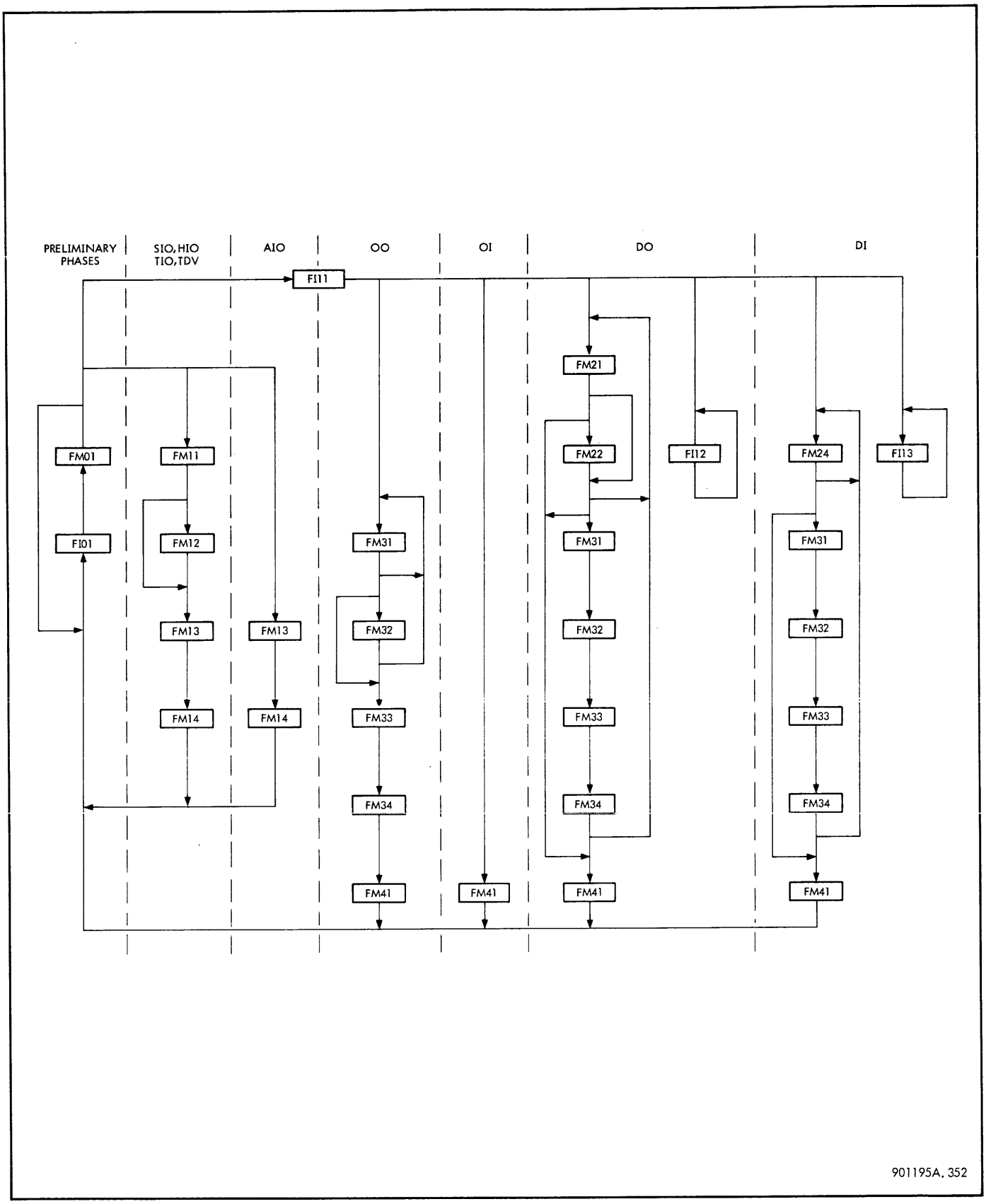


Figure 3-50. SIOP Phase Diagram

Table 3-20. SIO Instruction Phase Sequence

Phase/Clock	Logical Operation	Comments
FI01A	NFSL (SC + CNST ME PR2) ⇏ S/TI	The TI delay line starts when signal S/TI goes true
TIS000/2	CNST ME PR2 ⟹ S/FNCF	SIOP responds to CPU request
TIS160	S/FI01B	Second rank phase latch set
TIS240	FNCF ⟹ S/FNCG, R/PR1	Signal FNCG gates CPU function indicator signals at device controller (D/C) interface. Signal PR1 initializes proceed logic
	FNCF FGSIO ⟹ S/FNCSIO	SIO function indicator line raised at D/C interface
TIS320	R/FI01A	First rank phase latch cleared
	S/SX0/1	Enables clearing of S-register
TIS560	S/FM01A	Next phase of preliminary phases
	FM01A FNCF ⇏ MEMRQ	Request for the memory bus made if bus-sharing option is installed
FM01A	TMRQASHARE NMRQ (FNCF NFNCAIO + ...) ⇏ S/TM	TM delay line starts when signal S/TM goes true
TMA000/1	SX0/1 ⟹ SX0	S-register is cleared
TMA000	FNCF NFNCAIO ⟹ S/S26	Core memory address X'20' forced into S-register by setting bit S26
	FNCF ⟹ R/PR2	Signal NPR2 primes proceed logic
	R/MRQ1, R/MRQ2	Latches initialized for current operation
TMS080	R/SX0/1	
TMS140	MUX0, MLX0	M-register latches cleared
	MWX0	MW-register latches cleared (defines a core memory read operation)
	FNCF NFNCAIO ⟹ S/MRQ	Core memory request to read location X'20'
	FNCF NFNCAIO ⟹ S/MRQ1	Designates that MRQ was raised

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-20. SIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM01A (Cont.)		
TMA160	S/FM01B	Second rank phase latch set
	R/MPE	
CNTCK	TMATRIPB FNCF NFNCAIO ⟹ CACNTD	Enables countdown of command address counter
FM01B	TMRQB ⇏ S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM01A	
	MRQ1 ⟹ R/MPR	Signal MPR signifies core memory parity release
TMB160	R/ORD, R/OUT, R/TRA1, R/CHAIN	Latches initialized
TMB240	FNCF NFNCAIO ⟹ S/FM11A	FM11 is next phase of SIO sequence
FM11A	TMRQASHARE NMRQ MPR NRS ⇏ S/TM AR ⇏ R/MRQ, MPESHARE DG ⇏ MXM, MPESHARE PE ⇏ S/MPE, POK + MPE ⇏ S/MPR	TM delay line starts when signal S/TM goes true. Latch MRQ is cleared by signal address release (AR) from core memory. Signal MXM enables memory data to be set into M-register when signal data gate (DG) is received from core memory. Latch memory parity release (MPR) is set when signal parity OK is received from core memory. Signal MPESHARE controls data transfer from core memory when bus-sharing option is installed
TMA000/1	IOX0-A	Byte position A of I/O-register is cleared
TMA000	IOAXMA	D/C address in byte position A of M-register is transferred to byte position A of I/O-register
	R/MRQ1	
TMA080	MB1 ⟹ S/ORD	Latch ORD is set if signal MB1 is true. (Designates that core memory location X'21' is to be written into.) Signal MB1 reflects R-field coding of SIO instruction
TMA160	S/FM11B	Second rank phase latch set
FM11B	TMRQB ⇏ S/TM	TM delay line is started for the second rank phase latch

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-20. SIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM11B (Cont.)		
TMB000	NMPE ⟹ S/FS	Request D/C response
	R/FM11A	First rank phase latch cleared
TMB240	NMPE MB0 ⟹ S/FM12A	SIOP advances to FM12 if core memory location X'20' is to be written into (MB0), and no memory parity errors occurred (NMPE)
	MPE + NMB0 ⟹ S/FM13A	SIOP advances to FM13 if memory parity error was detected (MPE), or core memory location X'20' is not to be written into
FM12A	TMRQASHARE NMRQ ⇏ S/TM	TM delay line is started for FM12A. Latch MRQ is cleared when address release (AR) is received from core memory for previous memory request
TMA000/1	JX0	J-register is cleared
TMA000	JXCA	Command address previously stored in CA-register is transferred to byte positions A and B of J-register. Command address was decremented by one during FM01A
TMS140	S/MRQ	Core memory request to write into location X'20'
	S/MRQ1	Designates that MRQ was raised
	MUX0	Upper half of M-register cleared
TMA160	MUXJ	Command address is transferred from byte positions A and B of J-register to byte positions A and B of M-register
	MWX1	Defines core memory write operation (all four bytes)
	S/FM12B	Second rank phase latch set
FM12B	TMRQB ⇏ S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM12A	First rank phase latch cleared
	S/FM13A	Next phase of SIO sequence

Notes:

1. Skips of phases are possible

2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-20. SIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM13A	TMRQASHARE NMRQ (FSL + NFS) =/=>S/TM AR=/=>R/MRQ AVO=/=>R/FS	TM delay line is started when signal S/TM goes true. Latch MRQ is cleared when signal AR is received from core memory for previous memory request. Latch FS is cleared if signal AVO instead of FSL is received at D/C interface, that is, if the operation is aborted
TMA000/1	JX0	J-register is cleared
TMA000	S/CC1, S/CC2	Initialize condition code latches
	R/MRQ1	
TMA160	DOR FS ⟹ R/CC1	D/C raises signal DOR if it recognizes its address on the data lines
	IOR FS ⟹ R/CC2	D/C raises signal IOR if SIO is accepted
	NFNCAIO ⟹ JAXFR	D/C status information on FR lines is set into byte position A of J-register
	S/FM13B	Second rank phase latch is set
CNTCK	TMATRIPB FNCF NFNCAIO⇒CACNTU	Enables count up of CA-register
FM13B	TMRQB=/=>S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM13A	Clear first rank phase latch
	R/FS	
	FNCSIO NCC1 NCC2 ⟹ AXIO, R/STIOPB2	Signal AXIO transfers D/C address currently in byte position A of I/O-register to A-register. Previous information is cleared from latch STIOPB2
	NFNCAIO (NORD + MPE) ⟹ S/PR1	Signal PR1 enables proceed logic
	PR1 NPR2=/=>PRD	Proceed (PR) signal is sent to the CPU
CNTCK	TMBTRIPB FNCSIO NCC1 NCC2 ⟹ CAXM	Signal CAXM transfers new command address in byte positions C and D of M-register to CA-register if D/C recognized its address (NCC1) and the SIO was accepted (NCC2)
TMB240	S/FM14A	Next phase of SIO sequence

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol =/=> indicates a function not controlled by a clock pulse

(Continued)

Table 3-20. SIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM14A	TMRQASHARE NMRQ ⟹S/TM	TM delay line is started for FM14A
TMA000	S/S26, NFNCAIO ⟹S/S31	Address X'21' is forced into S-register when bits S26 and S31 are true
	R/FNCG	
	FNCF NFNCAIO ⟹JXBC, JXST	Signal JXBC transfers previously stored byte count from the BC-register to byte positions C and D of J-register. Signal JXST transfers status stored in SIOP to byte position B of J-register. (D/C status was set into byte position A of J-register during FM13A)
TMS140	MUX0, MLX0	M-register latches are cleared
	NFNCAIO NMPE ORD ⟹S/MRQ, S/MRQ1	Request core memory write address X'21' (MRQ). Signal MRQ1 designates that MRQ was raised
TMA160	FNCSIO NCC1 NCC2=STX0, FLX0, CMX0, S/STIOPB1	If SIO is successful (NCC1 NCC2), the following occurs: SIOP status is cleared (STX0), flags are cleared (FLX0), chaining modifier (CM) is cleared (CMX0), and SIOP busy status bit STIOPB1 is set
	MUXJ, MLXJ	Contents of the J-register (SIOP status, D/C status, and byte count) are transferred to M-register. Memory request has already been made to write this information into core memory location X'21'
	MWX1	Defines core memory write operation (all four bytes)
	S/FM14B	Second rank phase latch set
TMA240	R/MEMCYC	Core memory bus is not in use. (Applies when bus-sharing option is installed)
FM14B	TMRQB ⇏S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM14A	Clear first rank phase latch
	R/FNCF	

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-20. SIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM14B (Cont.)		
TMB240	R/FNCSIO	
	FM14B MPESHARE AR =#⇒ R/MEMCYC/1	Latch MEMCYC/1 is cleared when address release is received from core memory. Signal NMEMCYC/1 designates that SIOP has completed its memory bus operations. (Applies when bus-sharing option is installed)
	S/FI01A	SIOP returns to FI01 and waits for service call from D/C

Notes:

1. Skips of phases are possible
2. The symbol ⇒ indicates a function controlled by a clock pulse. The symbol =#⇒ indicates a function not controlled by a clock pulse

(MYTEST SERVICE1) operations, the SIOP always drives the NCOND2D line to the CPU true. See table 3-21 for a detailed description of the operations that occur during a TIO or TDV sequence.

3-65 SIOP RESPONSES TO AN HIO FUNCTION CODE. The events that occur in the SIOP during the HIO function are similar to those that occur during an SIO function, and the sequences of phases are identical (figure 3-50). Signal ME goes true if the function code is an HIO addressed to this SIOP (FGHIO MYNUM). The HIO is always executed normally and may prematurely terminate a D/C service request. See table 3-22 for a detailed description of the operations that occur during an HIO sequence.

3-66 SIOP RESPONSES TO AN AIO FUNCTION CODE. The sequence of phases for an AIO function code is shown in figure 3-50. The SIOP always executes an AIO instruction normally if an interrupt is pending. The delay line starts for FI01 if an AIO function has been decoded and the interrupt latch (IR) has been set (signal MYAIO is true). Signal MYAIO causes signal ME to go true. See table 3-23 for a detailed description of the operations that occur during an AIO sequence.

3-67 SIOP Responses to Device Controller Service Requests.

The SIOP responds to D/C service requests by sequencing through the two preliminary phases and one of the groups of phases (designated as OO [order out], OI [order in], DO [data out], or DI [data in]), as shown in figure 3-50.

The SIOP, if waiting in the initial phase (FI01), responds to the service call from the D/C by starting the TI delay line and sequencing through the two preliminary phases. During the preliminary phases the D/C is connected to the D/C interface for service – that is, the service connect flip-flop in the D/C is set. If, however, the D/C does not connect for service (signal AVO is high at the SIOP) the SIOP aborts the service request by returning to the initial phase during timing pulse TMB240 of FM01B. If the service call is not aborted, the SIOP advances to FI11A during timing pulse TMB240 of FM01B and waits for the initial request strobe from the D/C.

During FI11, the SIOP strobes the DOR and IOR lines (controlled by the D/C) to determine which of the four service cycles has been specified. The service cycle type is retained by latches ORD (set by signal DOR) and OUT (set by signal IOR) and is decoded as follows:

ORD	OUT	Service Cycle
1	1	Order out
1	0	Order in
0	1	Data out
0	0	Data in

The SIOP status bits, with the exception of the busy status bit, are set during the service sequences. The busy status

Table 3-21. TIO and TDV Instructions, Phase Sequence

Phase/Clock	Logical Operation	Comments
FI01A	NFSL NAVO (SC + CNST ME PR2)=/=>S/TI	The TI delay line starts when signal S/TI goes true
TIS000/2	CNST ME PR2⟹S/FNCF	SIOP responds to CPU request
TIS160	S/FI01B	Second rank phase latch set
TIS240	FNCF⟹S/FNCG, R/PR1	Signal FNCG gates CPU function indicator signals at D/C interface. Signal PR1 initializes proceed logic
TIS320	R/FI01A	First rank phase latch is cleared
	S/SX0/1	Enables clearing of S-register
TIS560	S/FM01A	Next phase of preliminary phases
	FM01A FNCF=/=>MEMRQ	Request for memory bus made if bus-sharing option is installed
FM01A	TMRQASHARE NMRQ (FNCF NFNCAIO +...)=/=>S/TM	TM delay line starts when signal S/TM goes true
TMA000/1	SX0/1⟹SX0	S-register is cleared
TMA000	FNCF⟹R/PR2	Signal NPR2 primes proceed logic
	FNCF NFNCAIO⟹S/S26	Core memory address X'20' forced into S-register by setting bit S26
	R/MRQ1, R/MRQ2	Latches initialized for current operation
TMS080	R/SX0/1	
TMS140	MUX0, MLX0	M-register latches cleared
	MWX0	MW-register latches cleared (defines a core memory read operation)
	FNCF NFNCAIO⟹S/MRQ	Core memory request to read location X'20'
	FNCF NFNCAIO⟹S/MRQ/1	Designates that MRQ was raised
TMA160	S/FM01B	Second rank phase latch set
	R/MPE	

Notes:

1. Skips of phases are possible

2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol =/=> indicates a function not controlled by a clock pulse

(Continued)

Table 3-21. TIO and TDV Instructions, Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
CNTCK	TMATRIPB FNCF NFNCAIO ⟹ CACNTD	Enables countdown of command address counter
FM01B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM01A	First rank phase latch cleared
	MRQ1 ⟹ R/MPR	Signal MPR signifies core memory parity release
TMB160	R/ORD, R/OUT, R/TRA1, R/CHAIN	Latches initialized
TMB240	FNCF NFNCAIO ⟹ S/FM11A	FM11 is next phase of a TIO or TDV sequence
FM11A	TMRQASHARE NMRQ MPR NRS ⇏ S/TM AR ⇏ R/MRQ, MPESHARE DG ⇏ MXM, MPESHARE PE ⇏ S/MPE, POK + MPE ⇏ S/MPR	TM delay line starts when signal S/TM goes true. Latch MRQ is cleared by signal address release (AR) from core memory. Signal MXM enables memory data to be set into M-register when signal data gate (DG) is received from core memory. Latch memory parity release (MPR) is set when signal parity OK (POK) is received from core memory. Signal MPESHARE controls data transfer from core memory when bus-sharing option is installed
TMA000/1	IOX0-A	Byte position A of I/O-register is cleared
TMA000	IOAXMA	D/C address in byte position A of M-register is transferred to byte position A of I/O-register
	R/MRQ1	
TMA080	MB1 ⟹ S/ORD	Latch ORD is set if signal MB1 is true. (Designates that core memory location X'21' is to be written into.) Signal MB1 reflects R-field coding of TIO or TDV instruction
TMA160	S/FM11B	Second rank phase latch set
FM11B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	NMPE ⟹ S/FS	Requests D/C response
	R/FM11A	First rank phase latch cleared

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-21. TIO and TDV Instructions, Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
TM11B (Cont.)		
TMB240	NMPE MB0 ⟹ S/FM12A	SIOP advances to FM12 if core memory location X'20' is to be written into (MB0), and no memory parity errors occurred (NMPE)
	MPE + NMB0 ⟹ S/FM13A	SIOP advances to FM13 if a memory parity error was detected (MPE), or if core memory location X'20' is not to be written into
FM12A	TMRQASHARE NMRQ ⇏ S/TM	TM delay line is started for FM12A. Latch MRQ is cleared when address release (AR) is received from core memory for previous memory request
TMA000/1	JX0	J-register is cleared
TMA000	JXCA	Command address previously stored in CA-register is transferred to byte positions A and B of J-register. Command address was decremented by one during FM01A
TMS140	S/MRQ	Core memory request to write into location X'20'
	S/MRQ1	Designates that MRQ was raised
	MUX0	Upper half of M-register cleared
TMA160	MUXJ	Command address is transferred from byte positions A and B of J-register to byte positions A and B of M-register
	MWX1	Defines core memory write operation (all four bytes)
	S/FM12B	Second rank phase latch set
FM12B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM12A	First rank phase latch cleared
	S/FM13A	Next phase of TIO or TDV sequence

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-21. TIO and TDV Instructions, Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM13A	TMRQASHARE NMRQ (FSL + NFS) ⇏ S/TM AR ⇏ R/MRQ AVO ⇏ R/FS	TM delay line is started when signal S/TM goes true. Latch MRQ is cleared when signal AR is received from core memory for previous memory request. Latch FS is cleared if signal AVO instead of FSL is received at D/C interface – that is, if the operation is aborted
TMA000/1	JX0	J-register is cleared
TMA000	S/CC1, S/CC2	Condition code latches initialized
	R/MRQ1	
TMA160	DOR FS ⟹ R/CC1	D/C raises signal DOR if it recognizes its address on the data lines and it can accept an SIO
	IOR FS ⟹ R/CC2	D/C raises signal IOR if it cannot accept an SIO
	NFNCAIO ⟹ JAXFR	D/C status information on FR lines is set into byte position A of J-register
	S/FM13B	Second rank phase latch is set
CNTCK	TMATRIPB FNCF NFNCAIO ⟹ CACNTU	Enables count up of CA register
FM13B	TMRQB ⇏ S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM13A	Clears first rank phase latch
	R/FS	
	PR1 NPR2 ⇏ PRD	Proceed (PR) signal is sent to CPU
	NFNCAIO (NORD + MPE) ⟹ S/PR1	
TMB240	S/FM14A	Next phase of TIO and TDV instructions
FM14A	TMRQASHARE NMRQ ⇏ S/TM	TM delay line is started for FM14A when signal S/TM goes true
TMA000	S/S26, NFNCAIO ⟹ S/S31	Address X'21' is forced into S-register when bits S26 and S31 are true
	R/FNCG	

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-21. TIO and TDV Instructions, Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM14A		
TMA000 (Cont.)	FNCF NFNCAIO ⟹ JXBC, JXST	Signal JXBC transfers previously stored byte count from the BC-counter to byte positions C and D of the J-register. Signal JXST transfers status stored in SIOP to byte position B of J-register. (D/C status was set into byte position A of J-register during FM13A)
TMS140	MUX0, MLX0	M-register latches are cleared
	NFNCAIO NMPE ORD ⟹ S/MRQ, S/MRQ1	Request core memory write address X'21' (MRQ). Signal MRQ1 designates that MRQ was raised
TMA160	MUXJ, MLXJ	Contents of J-register (SIOP status, D/C status, and byte count) are transferred to M-register. Memory request has already been made to write this information into core memory location X'21'
	MWX1	Defines core memory write operation (all four bytes)
	S/FM14B	Second rank phase latch set
TMB240	R/MEMCYC	Core memory bus is not in use. (Applies when bus-share option is installed)
FM14B	TMRQB ⇏ S/TM	TM delay line starts for second rank phase latch
TMB000	R/RNCF	
	R/FM14A	Clears first rank phase latch
TMB240	FM14B MPESHARE AR ⟹ R/MEMCYC/1	Latch MEMCYC/1 is cleared when address release is received from core memory. Signal MEMCYC/1 designates that SIOP has completed its memory bus operations. (Applies when bus-sharing option is installed)
	S/FI01A	SIOP returns to FI01 and waits for service call from CPU

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

Table 3-22. HIO Instruction Phase Sequence

Phase/Clock	Logical Operation	Comments
F101A	NFSL (SC + CNST ME PR2)⇏S/TI	TI delay line starts when signal S/TI goes true
TIS000/2	CNST ME PR2⟹S/FNCF	SIOP responds to CPU request
TIS160	S/FI01B	Second rank phase latch set
TIS240	FNCF⟹S/FNCG, R/PR1	Signal FNCG gates function indicator signals at D/C interface. Signal PR1 initializes proceed logic
TIS320	R/FI01A	First rank phase latch cleared
	S/SX0/1	Enables clearing of S-register
TIS560	S/FM01A	Next phase of preliminary phases
	FM01A FNCF⟹MEMRQ	Request for memory bus made if bus-sharing option is installed
FM01A	TMRQASHARE NMRQ (FNCF NFNCAIO + ...)⇏S/TM	TM delay line starts when signal S/TI goes true
TMA000/1	SX0/1⟹SX0	S-register is cleared
TMA000	FNCF⟹R/PR2	Signal NPR2 primes proceed logic
	FNCF NFNCAIO⟹S/S26	Core memory address X'20' is forced into S-register by setting bit S26
	R/MRQ1, R/MRQ2	Latches initialized for current operation
TMS080	R/SX0/1	
TMS140	MUX0, MLX0	M-register latches cleared
	MWX0	MW-register latches cleared (defines core memory read operation)
	FNCF NFNCAIO⟹S/MRQ	Core memory request to read location X'20'
TMA160	R/MPE	
	S/FM01B	Second rank phase latch set
CNTCK	TMATRIPB FNCF NFNCAIO⟹CACNTD	Enables countdown of command address counter
FM01B	TMRQB⇏S/TM	TM delay line is started for second rank phase latch

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-22. HIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM01B (Cont.)		
TMB000	R/FM01A	First rank phase latch cleared
	MRQ1 ⟹ R/MPR	Signal MPR signifies core memory parity release
TMB160	R/ORD, R/OUT, R/TRA1, R/CHAIN	Latches initialized
TMB240	FNCF NFNCAIO ⟹ S/FM11A	FM11 is next phase of HIO sequence
FM11A	TMRQASHARE NMRQ MPR NRS ⇏ S/TM AR ⇏ R/MRQ, MPESHARE DG ⇏ MXM, MPESHARE PE ⇏ S/MPE, POK ⇏ + MPE ⇏ S/MPR	TM delay line starts when signal S/TM goes true. Latch MRQ is cleared by signal address release (AR) from core memory. Signal MXM enables memory data to be set into M-register when signal data gate (DG) is received from core memory. Latch memory parity release (MPR) is set when signal parity OK (POK) is received from core memory. Signal MPESHARE controls data transfer from core memory when bus-sharing option is installed
TMA000/1	IOX0-A	Byte position A of I/O-register is cleared
TMA000	IOAXMA	D/C address in byte position A of the M-register is transferred to byte position A of I/O-register
	R/MRQ1	
TMA080	MB1 ⟹ S/ORD	Latch ORD is set if signal MB1 is true. (Designates that core memory location X'21' is to be written into.) Signal MB1 reflects R-field coding of HIO instruction
TMA160	S/FM11B	Second rank phase latch set
FM11B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	NMPE ⟹ S/FS	Request D/C response
	R/FM11A	First rank phase latch cleared
TMB240	NMPE MB0 ⟹ S/FM12A	SIOP advances to FM12 if core memory location X'20' is to be written into (MB0), and no memory parity error occurred (NMPE)

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-22. HIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM11B		
TMB240 (Cont.)	MPE + NMB0 ⟹ S/FM13A	SIOP advances to FM13 if a memory parity error was detected (MPE), or if core memory location X'20' is not to be written into
FM12A	TMRQASHARE NMRQ ⇏ S/TM	TM delay line is started for FM12A. Latch MRQ is cleared when address release (AR) is received from core memory for previous memory request
TMA000/1	JX0	J-register is cleared
TMA000	JXCA	Command address previously stored in CA-register is transferred to byte positions A and B of J-register. Command address was decremented by one during FM01A
TMS140	S/MRQ	Core memory request to write into location X'20'
	S/MRQ1	Designates that MRQ was raised
	MUX0	Upper half of M-register cleared
TMA160	MUXJ	Command address is transferred from byte positions A and B of J-register to byte positions A and B of M-register
	MWX1	Defines core memory write operation (all 4 bytes)
	S/FM12B	Second rank phase latch set
FM12B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM12A	First rank phase latch cleared
	S/FM13A	Next phase of HIO sequence
FM13A	TMRQASHARE NMRQ (FSL + NFS) ⇏ S/TM AR ⇏ R/MRQ AVO ⇏ R/FS	TM delay line is started when signal S/TM goes true. Latch MRQ is cleared when signal AR is received from core memory for previous

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-22. HIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM13A (Cont.)		memory request. Latch FS is cleared if signal AVO instead of FSL is received at D/C interface, that, if the operation is aborted
TMA000/1	JX0	J-register is cleared
TMA000	S/CC1, S/CC2	Condition code latches initialized
	R/MRQ1	
TMA160	DOR FS ⟹ R/CC1	D/C raises signal DOR if it recognizes its address on the data lines
	IOR FS ⟹ R/CC2	D/C raises signal IOR only if it is not busy
	NFNCAIO ⟹ JAXFR	D/C status information on FR lines is set into byte position A of J-register
S/FM13B		Second rank phase latch is set
CNTCK	TMATRIPB FNCF NFNCAIO ⟹ CACNTU	Enables count up of CA-register
FM13B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM13A	Clear first rank phase latch
	R/FS	
	FGHIO MATCH ⇏ STAX0	Interrupt status is cleared by signal STAX0. Signal MATCH is true if D/C address currently in A-register matches D/C address encoded in HIO instruction
	NFNCAIO (NORD + MPE) ⟹ S/PR1	If a memory parity error was detected, latch PR1 will be set, enabling the proceed signal to be sent to CPU
	PR1 NPR2 ⇏ PRD	Proceed signal is sent to CPU
TMB240	S/FM14A	Next phase of HIO instruction
FM14A	TMRQASHARE NMRQ ⇏ S/TM	TM delay line is started for FM14A

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-22. HIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM14A (Cont.)		
TMA000	S/S26, NFNCAIO ⟹ S/S31	Address X'21' is forced into S-register when bits S26 and S31 are true
	R/FNCG	
	FNCF NFNCAIO ⟹ JXBC, JXST	Signal JXBC transfers previously stored byte count from BC counter to byte positions C and D of J-register. Signal JXST transfers status stored in SIOP to byte position B of J-register. (The D/C status was set into byte position A of J-register during FM13A)
TMS140	MUX0, MLX0	M-register latches are cleared
	NFNCAIO NMPE ORD ⟹ S/MRQ, S/MRQ1	Request core memory write address X21' (MRQ). Signal MRQ1 designates that MRQ was raised
TMA160	MUXJ, MLXJ	Contents of J-register (SIOP status, D/C status, and byte count) are transferred to M-register. Memory request has already been made to write this information into core memory location X'21'
	MWX1	Defines core memory write operation (all 4 bytes)
	FGHIO MATCH ⟹ R/STIOPB1	HIO instruction takes SIOP out of busy state when latch STIOPB1 is reset
	S/FM14B	Second rank phase latch set
TMB240	R/MEMCYC	Designates that core memory bus is not in use. (Applies when bus-sharing option is installed)
	NMRQ ⟹ R/MRMCYC/1	Permits new request for the memory bus. (Applies when bus-sharing option is installed)
FM14B	TMRQB ⟹ S/TM	TM delay line starts for the second rank phase latch
TMB000	R/RNCF	

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-22. HIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM14B		
TMB000 (Cont.)	R/FM14A	Clear first rank phase latch
TMB240	FM14B MPESHARE AR ⇏ R/MEMCYC/1	Latch MEMCYC/1 is cleared when address release is received from core memory. It was cleared during FM14A if core memory location X'21' was not written into. Signal MEMCYC/1 designates that SIOP has completed its memory bus operations. (Applies when bus-sharing option is installed)
	S/FI01A	SIOP returns to FI01 and waits for next service call from CPU

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

Table 3-23. AIO Instruction Phase Sequence

Phase/Clock	Logical Operation	Comments
FI01A	NFSL (SC + CNST ME PR2) ⇏ S/TI	TI delay line starts when signal S/TI goes true
TIS000/2	CNST ME PR2 ⟹ S/FNCF	SIOP responds to CPU request
TIS160	S/FI01B	Second rank phase latch set
TIS240	FNCF ⟹ S/FNCG, R/PR1	Signal FNCG gates function indicator signals at D/C interface. Signal PR1 initializes proceed logic
	FNCF FGAIO ⟹ S/FNCAIO	AIO function indicator line raised at D/C interface
TIS320	R/FI01A	First rank phase latch cleared
	S/SX01/1	Enables clearing of S-register
TIS560	S/FM01A	Next phase of preliminary phases
	FM01A FNCF ⇏ MEMRQ	Request for memory bus made if bus-sharing option is installed

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-23. AIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM01A	TMRQASHARE NMRQ (FNCAIO NRS + ...) ⇏ S/TM	TM delay line starts when signal S/TM goes true
TMA000/1	SX0/1 ⟹ SX0	S-register is cleared
TMA000	FNCF ⟹ R/PR2	Signal NPR2 primes proceed logic
	R/MRQ1, R/MRQ2	Latches initialized for current operation
	FNCAIO ⟹ S/FS	Function strobe FS raised at D/C interface. Requests D/C response
TMS080	R/SX0/1	
TMS140	MUX0, MLX0	M-register latches cleared
	MWX0	MR-register latches cleared (defines a core memory read operation)
TMA160	R/MPE	
	S/FM01B	Second rank phase latch set
FM01B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM01A	First rank phase latch cleared
	MRQ1 ⟹ R/MPR	Signal MPR signifies core memory parity release
TMB160	R/ORD, R/OUT, R/TRA1, R/CHAIN	Latches initialized
	S/FM13A	FM13 is next phase of AIO sequence
FM13A	TMRQASHARE NMRQ (FSL + NFS) ⇏ S/TM AVO ⇏ R/FS	TM delay line is started when signal S/TM goes true. Latch FS is cleared if signal AVO is received at D/C interface instead of signal FSL
TMA000/1	JX0	J-register is cleared
	FNCAIO ⟹ IOX0	I/O-register is cleared
TMA000	S/CC1, S/CC2	Initialize condition code latches
	R/MRQ1	

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-23. AIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM13A (Cont.)		
TMA160	DOR FS ⟹ R/CC1	D/C raises DOR to indicate it has an interrupt pending
	IOR FS ⟹ R/CC2	D/C raises IOR to indicate a normal interrupt condition
	S/FM13B	Second rank phase latch is set
	FNCAIO ⟹ JDXFR	Transfers SIOP address to byte position C, and D/C address to byte position D of J-register
	FNCAIO ⟹ IOXD	Transfers D/C status to byte position A of I/O-register
TMA240	FNCAIO ⟹ JAXIO	Transfers D/C status from byte position A of I/O-register to byte position A of J-register
FM13B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM13A	Clear first rank phase latch
	R/FS	
	FNCAIO ⟹ IOAX0	Byte position A of I/O-register is cleared
TMB120	FNCAIO ⟹ IOAXJD	Transfers D/C address from byte position D of J-register to byte position A of I/O-register
TMB240	S/FM14A	Next phase of AIO instruction
FM14A	TMRQASHARE NMRQ ⇏ S/TM	TM delay line is started for FM14A when signal S/TM goes true
TMA000	S/S26	Address X'20' is forced into S-register
	R/FNCG	
	FNCAIO ⟹ R/IR	Interrupt latch is reset to clear pending interrupt
	FNCAIO NCC1 MATCH ⟹ JXST	If contents of A-regsiter equal contents of byte position A of I/O-register (MATCH), SIOP status is transferred to byte position B of J-register

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-23. AIO Instruction Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM14A (Cont.) TMS140	FNCAIO ⟹ S/MRQ, S/MRQ1	Request core memory write address X'20' (MRQ). Signal MRQ1 designates that MRQ was raised
	MUX0, MLX0	M-register latches are cleared
TMA160	MUXJ, MLXJ	Contents of J-register (D/C status, SIOP status, SIOP address, and D/C address) are transferred to M-register
	MWX1	Defines core memory write operation (all four bytes)
	S/FM14B	Second rank phase latch set
	FNCAIO NCC1 MATCH STUEI ⟹ S/CC2	CPU is informed of an unusual interrupt condition (STUEI) when condition code line NCOND2 is false (CC2)
TMA240	R/MEMCYC	Core memory bus is not is use. (Applies when bus-sharing option is installed)
	NMRQ ⇏ R/MEMCYC/1	Permits new request for memory bus. (Applies when bus-sharing option is installed)
FM14B	TMRQB ⇏ S/TM	TM delay line starts for the second rank phase latch
TMB000	R/FNCF	
	PR1 NPR2 ⟹ PRD NFNCF NMRQ ⟹ S/PR1	Proceed line to CPU is raised
	R/FM14A	Clear first rank phase latch
TMB160	FNCAIO NCC1 MATCH ⟹ STAX0	SIOP interrupt status is cleared
TMB240	R/FNCAIO	
	FM14B MPESHARE AR ⇏ R/MEMCYC/1	Latch MEMCYC/1 is cleared when address release is received from core memory. Signal MEMCYC/1 designates that SIOP has completed its memory bus operations. (Applies when bus-sharing option is installed)
	S/FI01A	SIOP returns to FI01 and waits for next service call from CPU

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

bit (STIOPB1) is set during an SIO instruction. Table 3-24 shows the service sequence and logical operation involved in setting each status bit.

All D/C service requests (if not aborted) are terminated with a terminal order. The conditions for sending the various terminal order bits from the SIOP to the D/C are shown in table 3-25.

Since the events that occur during the two preliminary phases are the same for all four service cycles, the sequence table for the preliminary phases (table 3-26) appears before the sequence tables for the service cycles. At the conclusion of the preliminary phases, the SIOP advances to FI11, at which time the D/C specifies the service cycle type. This phase is, therefore, included at the beginning of each of the four service cycle tables.

The WRITE and READ phases (paragraph 3-72) apply to both data in and data out service cycles. The WRITE and READ phase tables (3-33 and 3-34) should be used in conjunction with the data out phase sequences (table 3-31) and the data in phase sequences (table 3-32).

Table 3-24. Conditions for Setting SIOP Status Bits

SERVICE SEQUENCE					CONDITIONS	STATUS BIT SET
SIO	OO Chain	OI	DO	DI		
		●			FI11B TIS320 IOA1	
				●	+ FM41A TMA160 NFLDC ZBC1 NEMPTY	STIL
		●			FI11B TIS320 IOA0	DATAERR and STTDE
				●	+ (FI11B + FI13B) TIS400 DX1 PC EVEN	
				●	FM24A TMA080 MPE	DATAERR and STTME
			●	●	+ FM31A TMA080 MPE	
			●	●	+ FM41A TMA000 MPE	
	●				FM31B TMB080 MRQ NAH	
			●		+ FM21B TMB080 MRQ NAH	
			●		+ FM22B TMB080 MRQ NAH	
				●	+ FM24B TMB080 MRQ NAH	STMAE
	●				FM32A TMA000 MPE	
	●				+ FM34A TMA080 MPE	STIOPME
	●				FM32A TMA160 TRA TRA1	STIOPCE
		●			FI11B TIS560 STIL NFLSIL FLHTE	
	●		●	●	+ STMAE TMBTRIPB	
	●				+ STIOPCE TMBTRIPB	
	●				+ STIOPME TMBTRIPB	
		●	●	●	+ FM41A TMA080 FLHTE DATAERR	STIOPH
●					FM14A TMA160 FNCSIO NCC1 NCC2	STIOPB1

(Continued)

Table 3-24. Conditions for Setting SIOP Status Bits (Cont.)

SERVICE SEQUENCE					CONDITIONS	STATUS BIT SET
SIO	OO Chain	OI	DO	DI		
				●	FM24B TMB240 FLIZC ZBC0T15	
			●		FM34A TMA160 FLIZC	
			●		FM41A TMA160 FLIZC NK8 KFILL2	STZBCI
		●			FM41A TMA160 FLICE JA3	STCEI
		●			FM41A TMA160 FLIUE (JA4 + STIOPH)	STUEI

Table 3-25. Conditions for Setting Terminal Order Bits

SERVICE SEQUENCE				CONDITIONS	BIT SET	COMMENTS
OO	OI	DO	DI			
	●			FM41A TMA160 STCEISET	IOA0 (Interrupt)	Channel end interrupt
	●			FM41A TMA160 STUEISET		Unusual end interrupt
		●		FM41A TMA160 FLIZC (count done)		Zero count interrupt
		●	●	FM41A TMA160 STZBCI		Zero count interrupt
		●		FM41A TMA160 NK8 KFILL2	IOA1 (count done)	
			●	FM41A TMA160 NFLDC ZBC1		
●	●	●	●	FM41A TMA160 FLCC	IOA2 (command chain)	D/C examines bits only during OI
●	●	●	●	FM41A TMA160 STIOPH	IOA3 (IOP halt)	(See table 3-24 – STIOPH)

Table 3-26. Service Call Sequence of Preliminary Phases

Phase/Clock	Logical Operation	Comments
FI01A	NFSL (SC + ...) ⇏ S/TI	TI delay line starts when signal S/TI goes true

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-26. Service Call Sequence of Preliminary Phases (Cont.)

Phase/Clock	Logical Operation	Comments
FI01A (Cont.)		
TIS160	S/FI01B	Second rank phase latch set
TIS240	NFNCF SC ⟹ S/FNCASC	ASC function indicator line raised at D/C interface
TIS360	R/FI01A	First rank phase latch cleared
	S/SX0/1	Enables clearing of S-register
TIS560	S/FM01A	Next phase of preliminary phases
	FNCASC ⟹ S/FS	Function strobe raised at D/C interface. Requests D/C response
	FM01A NFNCF ⇏ TMRQASHARE	Signal TMRQASHARE must be true before TM delay line can be started when the bus-sharing option is installed
FM01A	TMRQASHARE NMRQ [NFNCF NRS (FSL + NFNCASC)] ⇏ S/TM	TM delay line starts when signal S/TM goes true
	AVO ⇏ R/FS, R/FNCASC	
	SX0/1 ⟹ SX0	S-register is cleared
	R/MRQ1, R/MRQ2	Latches initialized for current operation
TMS080	R/SX0/1	
TMS140	MUX0, MLX0	M-register latches cleared
	MWX0	MW-register latches cleared (defines a core memory read operation)
TMA160	S/FM01B	Second rank phase latch set
	R/MPE	
	FNCASC ⟹ R/FS	
FM01B	TMRQB ⇏ S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM01A	First rank phase latch cleared
TMB160	R/ORD, R/OUT, R/TRA1, R/CHAIN, FNCASC ⟹ R/ED, R/ES	Latches initialized for current operation

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-26. Service Call Sequence of Preliminary Phases (Cont.)

Phase/Clock	Logical Operations	Comments
FM01B (Cont.) TMB240	NFNCF NFNCASC ⟹ S/FI01A	Service call aborted (returns to initial phase, FI01)
	FNCASC ⟹ S/FI11A	D/C connects. SIOP advances to FI11 and waits for initial request strobe from D/C

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

3-68 SIOP RESPONSES TO AN ORDER OUT REQUEST. During FI11, if the D/C specifies an order out cycle by driving the DOR and IOR lines, the SIOP sets latches ORD and OUT and sequences through the group of phases specified OO (figure 3-50). This group of phases is also called the chaining sequence, since many of the events relating to data chaining and command chaining occur during these phases. (Command chaining includes all of the order out operations.)

During the termination phase (FM41) the SIOP transfers the new flags to the FL-register. SIOP responses to the flags are shown in table 3-27. See table 3-28 for a detailed description of the operations that occur during an order out service cycle.

3-69 SIOP RESPONSES TO AN ORDER IN REQUEST. During FI11, if the D/C specifies an order in cycle by driving the DOR line true and holding the IOR line false, the SIOP sets latch ORD (latch OUT remains false) and sequences through the group of phases specified OI (figure 3-50).

During order in service, the D/C sends the operational status byte to the SIOP on the data lines with the initial RS. The SIOP operates on the D/C input by setting certain status bits, as shown in table 3-29. The service cycle is terminated with a terminal order (see table 3-25). See table 3-30 for a detailed description of the operations that occur during an order in service cycle.

3-70 SIOP RESPONSES TO A DATA OUT REQUEST. During FI11, if the D/C specifies a data out cycle by driving the IOR line true and holding the DOR line false, the SIOP sets latch OUT (latch ORD remains false) and sequences through the group of phases specified DO (figure 3-50).

Figure 3-51 shows typical phase sequences during a data out service cycle with D/C interface data path widths of one (DX1), two (DX2), or four (DX4) bytes. During FI11 the SIOP enters FM21 where the initial core memory request (MRQ) is made. Data is accessed from core and transferred to the data buffer (R-register) during buffer write cycles (W). As soon as a read cycle (R) is permitted, data is transferred from the data buffer to the K-register. (The numbers below the W and R indicate the locations filled in the buffer after the read or write cycle.)

At the D/C interface, FI12 is started and data is transmitted to the I/O-register for subsequent transmission to the D/C. From this point the D/C interface and the core memory interface operate relatively independently until end data is raised either by the SIOP or by the D/C. The data chaining operation is included in figure 3-51 to show the following:

a. The D/C interface continuing to exchange data until the buffer has been depleted

b. The effect of data chaining with a near empty buffer (DX4)

c. The SIOP returning to FM21 after the exit from the data chaining sequence

After end data has been raised, the SIOP terminates after each data out service cycle with a terminal order in FM41 (see table 3-25). See table 3-31 for a detailed description of the operations that occur during a data out service cycle.

3-71 SIOP RESPONSES TO A DATA IN REQUEST. During FI11, if the D/C specifies a data in service cycle by holding both the DOR and IOR lines false, the SIOP responds by not setting latches ORD and OUT, and sequences through the group of phases specified DI (figure 3-50).

Table 3-27. SIOP Response to Flags (Continued)

Flag	OO	OI	DO	DI	SIOP Response	Comments
FLDC			●		FI12A TIS000/1 FLDC FLIZC NFAST CHAIN EMPTY/1⟹S/ED	Request ZBC interrupt
				●	FM24B TMB240 FLDC ZBC0T15 NSTIOPHSET1	
			●		+ FM21B TMB240 FLDC ZBC2 NSTIOPHSET1	
			●		+ FM22B TMB240 FLDC ZBC NSTIOPHSET1⟹S/FM31A	Data chain
			●		FM21B TMB240 ZBC2 NFLDC	
			●		+ FM22B TMB240 ZBC1 NFLDC	
				●	+ FM24B TMB240 ZBC0T15 NFLDC⟹S/FM41A	Count done termination
			●		NDREAD TRS250 DO NFLDC ZBC0T15 FM41A⟹S/ENDDIO	Last data in buffer
				●	FM41A TMA160 NFLDC DI ZBC1 NEMPTY⟹S/STIL	Data left in buffer at count done
				●	FM41A TMA160 NFLDC DI ZBCI⟹S/IOA1	Count done
FLIZC			●		FI12A TIS000/1 FLDC FLIZC NFAST CHAIN EMPTY/1⟹S/ED	Request ABC interrupt
			●		FM34A TMA160 NFAST FLIZC DO⟹S/NFLCLEAR	Terminate
			●		FM34A TMA160 FLIZC DO⟹S/STZBCI	Zero count interrupt
			●		FM41A TMA160 FLIZC DO NK8 KFILL2⟹S/STZBCI, S/IOA0	Zero count interrupt
				●	FM24B TMB240 FLIZC ZBC0T15⟹S/STZBCI	Zero count interrupt
FLCC	●	●	●	●	FM41A TMA160 FLCC⟹S/IOA2	Command chain
				●	FM41A TMA160 (NFLCC + STIOPH) JA3 OI⟹S/STIOPB2	Clear IOP busy

Note:

The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹/ indicates a function not controlled by a clock pulse

Table 3-27. SIOP Response to Flags (Cont.)

Flag	OO	OI	DO	DI	SIOP Response	Comments
FLICE				●	FM41A TMA160 FLICE JA3 OI ⟹ S/STCEI, S/IOA0	Channel end interrupt
FLHTE				●	FI11B TIS560 OI FLHTE NFLSIL STIL ⟹ S/STIOPH	IOP halt
		●	●	●	FM41A TMA080FLHTE DATAERR ⟹ S/STIOPH	IOP halt
FLIUE				●	FM41A TMA160 FLIUE OI (JA4 + STIOPH) ⟹ S/STUEI, S/IOA0	Unusual end interrupt
FLSIL		●			FI11B TIS560 OI NFLSIL FLHTE STIL ⟹ S/STIOPH	Incorrect length
FLSKIP			●	●	(FM21A + FM22A + FM24A) TMS140 BCNT NFLSKIP ⟹ S/MRQ	Perform all operations except memory request
CM	●				FM31A TMS140 NCM ⟹ S/MRQ	Update CA and do not access memory if CM true

Note:
The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

Table 3-28. Order Out Service Cycle Phase Sequence

Phase/Clock	Logical Operation	Comments
FI11A	RS $\not\Longrightarrow$ S/TI	TI delay line starts when request strobe RS is received from D/C
TIS000/1	IOX0	I/O-register latches are cleared
	CNST (IR FGAIO + MYNUM FGHIO) $\Longrightarrow$ S/ED	If this SIOP has an interrupt pending and the CPU sends an AIO, or if an HIO is addressed to this SIOP, the SIOP will terminate the D/C service request by setting end data (ED)
TIS000/2	S/MPR	Memory parity release
	R/END3, R/ENDFI, R/END1, R/END2	Initialize latches
	R/SERVICE2	Inhibits normal execution of TIO or TDV instruction. (NSERVICE1 SERVICE2 enable normal execution of a TIO or TDV instruction)
TIS160	S/FI11B	Second rank phase latch set
	S/SERVICE1	Initializes NSERVICE1 latch. (NSERVICE1 SERVICE2 enables execution of a TIO or TDV instruction)
	R/FNCASC	ASC function indicator line at D/C interface dropped
	JX0	J-register latches are cleared
	DOR $\Longrightarrow$ S/ORD, IOR $\Longrightarrow$ S/OUT	D/C drives DOR and IOR lines to specify an order out service cycle. Latches ORD and OUT retain this information
	EDI $\Longrightarrow$ S/ED	Signal EDI from D/C (sent on ED line) sets latch end data (ED)
TIS320	R/FI11A	First rank phase latch cleared
	OO $\Longrightarrow$ S/FM31A	Next phase of an order out service cycle
	FM31A $\not\Longrightarrow$ MEMRQ	Request made for memory bus. (Applies when bus-sharing option is installed)
	NOI $\Longrightarrow$ S/SX0/1	Enables clearing of the S-register
TIS480	ED RSA $\not\Longrightarrow$ S/SERVICE2	Enables normal execution of a TIO or TDV directed to this SIOP. D/C caused latch ED to be set when first RS was received

Notes:

1. Skips of phases are possible
2. The symbol $\Longrightarrow$ indicates a function controlled by a clock pulse. The symbol $\not\Longrightarrow$ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
TIS480 (Cont.)	SERVICE2 NMYTEST⇏R/SERVICE1	A TIO or TDV will be normally executed if signal NSERVICE1 is true. Signal MYTEST is true if a TIO or TDV is directed to this SIOP
FM31A	TMQRASHARE NMRQ MPR⇏S/TM	TM delay line is started for FM31A
TMA000/1	SX0/1⟹SX0	S-register latches are cleared
TMA000	SXCA	Command address (even) is set into S-register
	R/MRQ1, R/MRQ2	Latches initialized
TMS080	R/SX0/1	Latch cleared
TMA080	MPE⟹S/STTME, S/DATAERR	If a memory parity error was detected during a prior memory request when data chaining was was in progress, the transmission memory error status bit (STTME) and the transmission data error or transmission memory error latch (DATAERR) will be set
TMS140	MUX0, MLX0	M-register latches are cleared
	MWX0	MW-register latches are cleared. (Specifies a core memory read operation)
	NCM⟹S/MRQ	If chaining modifier bit is false (NCM), SIOP reads even command doubleword address from core memory
	NCM⟹S/MRQ1, S/MRQ2	Designates that MRQ was raised
TMA160	S/FM31B	Second rank phase latch set
	S/CHAIN	Controls transfer of new flags and clearing of old flags during termination phase
	R/MPE	Initialize memory parity error latch
CNTCK	TMATRIPB⟹CACNTU	Increments command address by one to specify next command doubleword address
FM31B	TMRQB⇏S/TM	TM delay line is started for the second rank phase latch

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM31B (Cont.)		
TMB000	T/FM31A	Clear first rank phase latch
	R/CM	Clear chaining modifier if true
	MRQ1 ⟹ R/MPR	Initialize memory parity release latch
TMB080	MRQ NAH ⟹ S/STMAE	Memory address error status bit set if a memory request was made (MRQ) and signal address here not received (NAH). (Address requested does not exist)
	STAME TMBTRIPB ⇏ S/STIOPH	IOP halt status bit (STIOPH) set if a core memory address error (STMAE) was detected
TMB240	NSTMAE NMRQ2 ⟹ S/FM31A	Reenter FM31A if chaining modifier (CM) was true and no memory address error detected (SNTMAE)
	NSTMAE NMRQ2 ⟹ S/SX0/1	Enables clearing of S-register latches
	NSTMAE MRQ2 ⟹ S/FM32A	Advance to FM32A if a memory access was made (MRQ2), and no memory address error was detected (NSTMAE)
	STMAE ⟹ S/FM33A	Advance to FM33A if memory address not present
	STMAE ⟹ S/MPR, R/MRQ	Signal MPR signifies core memory parity release. Core memory request latch MRQ is cleared. If the memory address requested does not exist (STMAE true – see FM31B TMB080), none of the strobes (AR, POK, etc) are returned from memory
FM32A	TMRQASHARE NMRQ MPR ⇏ S/TM AR ⇏ R/MRQ POK + MPE ⇏ S/MPR MPESHARE PE ⇏ S/MPE MPESHARE DG ⇏ MXM	TM delay line is started for FM32A. The memory request (MRQ) signal goes false when address release (AR) is received from core memory. Memory parity release (MPR), if not set during FM31B because of a memory address error, is set by signal parity OK (POK) from core memory. If a memory parity error (PE is detected, latch MPE will be set. Signal data gate (DG) from memory causes MXM to go true. Signal MXM gates data on memory data lines into M-register

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM32A (Cont.)		
TMA000/1	OO ⟹ IOAX0	I/O-register latches are cleared
TMA000	OO ⟹ IOAXMA, R/BKWD	Signal IOAXMA transfers order bits from byte position A of M-register to byte position A of I/O-register. Backward latch (BKWD) is reset
	MPE ⟹ S/STIOPME	Memory error status bit STIOPME is set if a memory parity error is detected
	R/MRQ1	Latch initialized
TMA160	S/FM32B	Second rank phase latch set
	TRA TRA1 ⟹ S/STIOPCE	Signals TRA and TRA1 both true indicate a control error and will cause control error status bit (STIOPCE) to be set
	OO IOA4 IOA5 NIOA6 NIOA7 ⟹ S/BKWD	Backward latch is set if order in I/O-register is an order to D/C to read backwards
CNTCK	TMATRIPB NMPE TRA NTRA1 ⟹ CAXM	If order bits specified a transfer-in-channel command (TRA), new command address (specified by TRA command) is loaded into CA counter (CAXM)
TMS240	TMATRIPB OO ⟹ KX0	K-register is cleared (buffer initialized)
FM32B	TMRQB ⇏ S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM32A	Clear first rank phase latch
TMB080	TRA ⇏ S/TRA1	If transfer-in-channel command (TRA), set latch TRA1. If this is second time through FM32 and TRA is true, an IOP control error exists
CNTCK	TMBTRIPB NMPE NTRA ⟹ BAXM	New byte address is transferred to BA counter
	BAXM BKWD NMD6 ⇏ S/BA17	Initializes least two significant bits (BA17 and BA18) of byte address as a function of signal BKWD
	BAXM BKWD NMD7 ⇏ S/BA18	

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM32B (Cont.)		
TMB240	NSTIOPH TRA ⟹ S/FM31A, S/SX0/1	If order was transfer-in-channel (TRA) and no IOP halt conditions exist, reenter FM31A to fetch command specified by TRA. Signal S/SX0/1 enables clearing of S-register
	STIOPH + NTRA ⟹ S/FM33A	Next phase of order out service cycle if last command accessed was not a transfer-in-channel command
	(STIOPME + STIOPCE) TMBTRIPB ⇏ S/STIOPH	IOP halt status bit is set if a memory error or control error was detected
FM33A	TMRQASHARE NMRQ MPR ⇏ S/TM	TM delay line is started for FM33A
TMA000	S/S31	Prepare to access odd word of command doubleword
	R/MRQ2	Latch initialized
	OO ⟹ R/JFILL	Initialize buffer
	NDO ⟹ K8X1	Byte presence indicator bits of K-register are initialized
TMS140	MUX0 MLX0	M-register latches are cleared
	MSTIOPH ⟹ S/MRQ	Access odd word of command doubleword
	NSTIOPH ⟹ S/MRQ1	MRQ was raised
TMA160	S/FM33B	Second rank phase latch set
	OO ⟹ R/SREAD, R/SWRITE, R/HFM21O22A, R/HWRITE, R/RAM, R/RBM1, R/RBM2, R/RCM	Initializes latches and buffer
FM33B	TMRQB ⇏ S/TM	TM delay line is started for FM33B
TMB000	R/FM33A	First rank phase latch cleared
	MRQ1 ⟹ R/MPR	Memory parity release latch initialized
	OO ⟹ S/RSA	Request strobe acknowledge (RSA) sent to D/C to permit D/C to accept order on data lines

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM33B		
TMB000 (Cont.)	RSA ED $\not\Rightarrow$ S/SERVICE2	Enables normal execution of a TIO or TDV directed to this IOP. D/C caused latch ED to be set when first RS was sent to IOP
	SERVICE2 NMYTEST $\not\Rightarrow$ R/SERVICE1	A TIO or TDV will be executed normally if signal NSERVICE1 is true. Signal MYTEST is true if a TIO of TDV is directed to this IOP
TMB040	OO $\Longrightarrow$ RKCK, OACK	Clock signals RKCK and OACK provide initialization of the RK- and OA-registers respectively (buffer initilize)
TMB160	R/TRA1	
TMB240	OO $\Longrightarrow$ S/MAYWRITE	Enables setting of BCNT, which is used to update the byte count and byte address counters
	FM33B OO $\not\Rightarrow$ R/IA, R/OA, R/RK	Registers initialized (buffer initialize)
	S/FM34A	Next phase of order out service cycle
FM34A	TMRQASHARE NMRQ MPR $\not\Rightarrow$ S/TM AR $\not\Rightarrow$ R/MRQ POK + MPE $\not\Rightarrow$ S/MPR MPESHARE PE $\not\Rightarrow$ S/MPE MPESHARE DG $\not\Rightarrow$ MXM	TM delay line is started for FM34A. (See FM32A for details)
TMA000	R/NFLCLEAR, R/ENDDIO, R/MRQ1	Latches initialized
	OO NIOA7 $\Longrightarrow$ NK81SET OO IOA7 $\Longrightarrow$ NK82SET	Latches NKA8, NKB8, NKC8, NKD8, and KFILL1 initialized (buffer initialize)
TMA080	MPE $\Longrightarrow$ S/STIOPME	IOP memory error status bit set if a memory parity error was detected
TMA160	S/FM34B	Second rank phase latch set
	R/MPE	Latch MPE cleared before leaving chaining sequence

Notes:

1. Skips of phases are possible
2. The symbol $\Longrightarrow$ indicates a function controlled by a clock pulse. The symbol $\not\Rightarrow$ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM34A		
TMA160 (Cont.)	OO ⟹ S/MAYREAD	Enables buffer read cycles (initializes buffer for ensuing data transfers)
	ORD + STIOPME + STIOPH + END3 ⟹ S/NFLCLEAR	Inhibits transfer of flags during this phase
CNTCK	TMATRIPB NSTIOPH NMPE ⟹ BCXM	New byte count transferred to BC counter if no errors are detected
TMA240	R/KFILL2, R/MEMCYC	Clears latches
FM34B	TMRQB ⇏ S/TM	TM delay line is started for FM34A
TMB000	R/FM34A	Clear first rank phase latch
	R/ZBC1, R/ZBC2	Initialize latches
	OO ⟹ R/DATAERR	Latch used for temporary storage of error conditions is cleared
TMB240	NFLCLEAR ⟹ S/FM41A	Next phase of order out service cycle
	FM41A ⇏ TMRQASHARE	Enables TM delay line to start. (Applies when bus-sharing option is installed)
	STIOPME TMBTRIPB ⇏ S/STIOPH	IOP halt status bit set if the memory error status bit is true
FM41A	TMRQASHARE NMRQ RS NRSA (ORD + ...) ⇏ S/TM	TM delay line starts when RS is received from D/C for terminal order (when signal S/TM goes true)
TMA000/1	IOX0	I/O-register latches cleared
TMA000	S/ES	Signal ES primes service connect flip-flop in D/C so that it will reset when it receives signal RSA
TMA160	FLCC ⟹ S/IOA2	Terminal order bits are set (see table 3-25)
	STIOPH ⟹ S/IOA3	
	S/FM41B	Second rank phase latch set
FM41B	TMRQB ⇏ S/TM	TM delay line is started for FM41B

Notes:

1. Skips of phases are possible

2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-28. Order Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM41B (Cont.)		
TMB000	R/FM41A	Second rank phase latch cleared
TMB040	CHAIN NSTIOPH ⟹ FLX0	Old flags cleared
TMB120	S/RSA	D/C reads terminal order and disconnects
TMB160	CHAIN NSTIOPH ⟹ FLXM	New flags are transferred to FL-register
TMB240	CHAIN ⟹ R/MEMCYC/1	Memory bus is not in use (applies when bus-sharing option is installed)
	S/FI01A	SIOP returns to initial phase

Notes:

1. Skips of phses are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

Table 3-29. SIOP Responses to Order In Operational Status Byte

Order In Status Bit	Logical Operation	Status Bit Set
TE (transmission error)	FI11B TIS320 IOA0	STTDE
IL (incorrect length)	FI11B TIS320 IOA1	STIL
CM (chaining modifier)	FI11B TIS400 IOA2	CM
CE (channel end)	FM41A TMA160 FLICE JA3	STCEI
	FM41A TMA160 JA3 (NFLCC + STIOPH) NMYSIO	R/STIOPB1
UE (unusual end)	FM41A TMA160 FLIUE (JA4 + STIOPH)	STUEI
	FM41A TMA160 JA4 NMYSIO	R/STIOPB1

Table 3-30. Order In Service Cycle Phase Sequence

Phase/Clock	Logical Operation	Comments
FI11A	RS ⇏ S/TI	TI delay line starts when RS is received from D/C
TIS000/1	IOX0	I/O-register latches are cleared
	CNST (IR FGAIO + MYNUM FGHIO) ⇏ S/ED	If this SIOP has an interrupt pending and the CPU sends an AIO or if an HIO is addressed to this SIOP, the SIOP will terminate the D/C service request by setting ED
TIS000/2	S/MPR	Memory parity release
	R/END3, R/ENDFI, R/END1, R/END2	Latches initialized
	R/SERVICE2	Inhibits normal execution of a TIO or TDV instruction
TIS160	S/FI11B	Second rank phase latch set
	S/SERVICE1	Initializes NSERVICE1 latch
	R/FNCASC	ASC function indicator line at D/C interface dropped
	IOXD	Operational status byte on data lines (from D/C) set into I/O-register
	JX0	J-register latches are cleared
	EDI ⟹ S/ED	Signal EDI from D/C sets latch end data ED
	DOR ⟹ S/ORD	D/C drives DOR line and holds IOR line false to specify an order in service cycle. Latch ORD is set and latch OUT remains reset
TIS240	JAXI0, JXI0	Operational status byte is transferred from I/O-register to J-register
TIS320	R/FI11A	First rank phase latch cleared
	OI IOA0 ⟹ S/STTDE, S/DATAERR	If D/C drives data line DA0 (IOA0) (indicating a transmission error), SIOP sets status bit STTDE and latch DATAERR, which temporarily stores the transmission data error

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-30. Order in Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FI11A (Cont.)		
TIS320 (Cont.)	OI IOA1 ⟹ S/STIL	D/C drives data line DA1 (IOA1) (indicating an incorrect length), SIOP sets status bit STIL
TIS400	OI IOA2 ⟹ S/CM	If D/C drives data line DA2 (IOA2), SIOP will set the chaining modifier bit (CM)
TIS480	NOUT ED ⟹ S/RSA	Request strobe acknowledge sent to D/C
	ED RSA ⇏ S/SERVICE2	Enables normal execution of a TIO or TDV directed to this IOP. D/C caused latch ED to be set when the first RS was received
	SERVICE2 NMYTEST ⇏ R/SERVICE1	A TIO or TDV will be executed normally if signal NSERVICE1 is true. Signal MYTEST is true if a TIO or TDV is directed to this IOP
TIS560	OI STIL NFLSIL FLHTE ⟹ S/STIOPH	SIOP sets IOP halt status bit if D/C specified incorrect length via operational status byte, suppress incorrect length flag (FLSIL) is false, and halt on transmission error flag (FLHTE) is true
	OI ⟹ S/FM41A	Next phase of order in service cycle
	FM41A ⇏ TMRQASHARE	Enables TM delay line to start. (Applies when bus-sharing option is installed)
FM41A	TMRQASHARE NMRQ RS NRSA (ORD + ...) ⇏ S/TM	TM delay line starts when RS is received from D/C for terminal order (when signal S/TM goes true)
TMA000/1	IOX0	I/O-register latches cleared
TMA000	S/ES	Signal ES primes service connect flip-flop in D/C so that it will reset when it receives signal RSA
TMA080	DATAERR FLHTE ⟹ S/STIOPH	IOP halt status bit set
TMA160	OI FLICE JA3 ⟹ S/IOA0, S/STCEI OI FLIUE (JA4 + STIOPH) ⟹ S/IOA0, S/STUEI FLCC ⟹ S/IOA2 STIOPH ⟹ S/IOA3 OI JA3 (NFLCC + STIOPH + JA4) ⟹ S/STIOPB2	Terminal order and status bits are set (see tables 3-24 and 3-25)
	STIOPB2 NMYSIO ⇏ R/STIOPB1	SIOP leaves busy state

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-30. Order In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM41B	TMRQB ⟹ S/TM	TM delay line is started for FM41B
TMB000	R/FM41A	Second rank phase latch cleared
TMB120	S/RSA	D/C reads terminal order and disconnects
TMB240	S/FI01A	SIOP returns to initial phase

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹/ indicates a function not controlled by a clock pulse

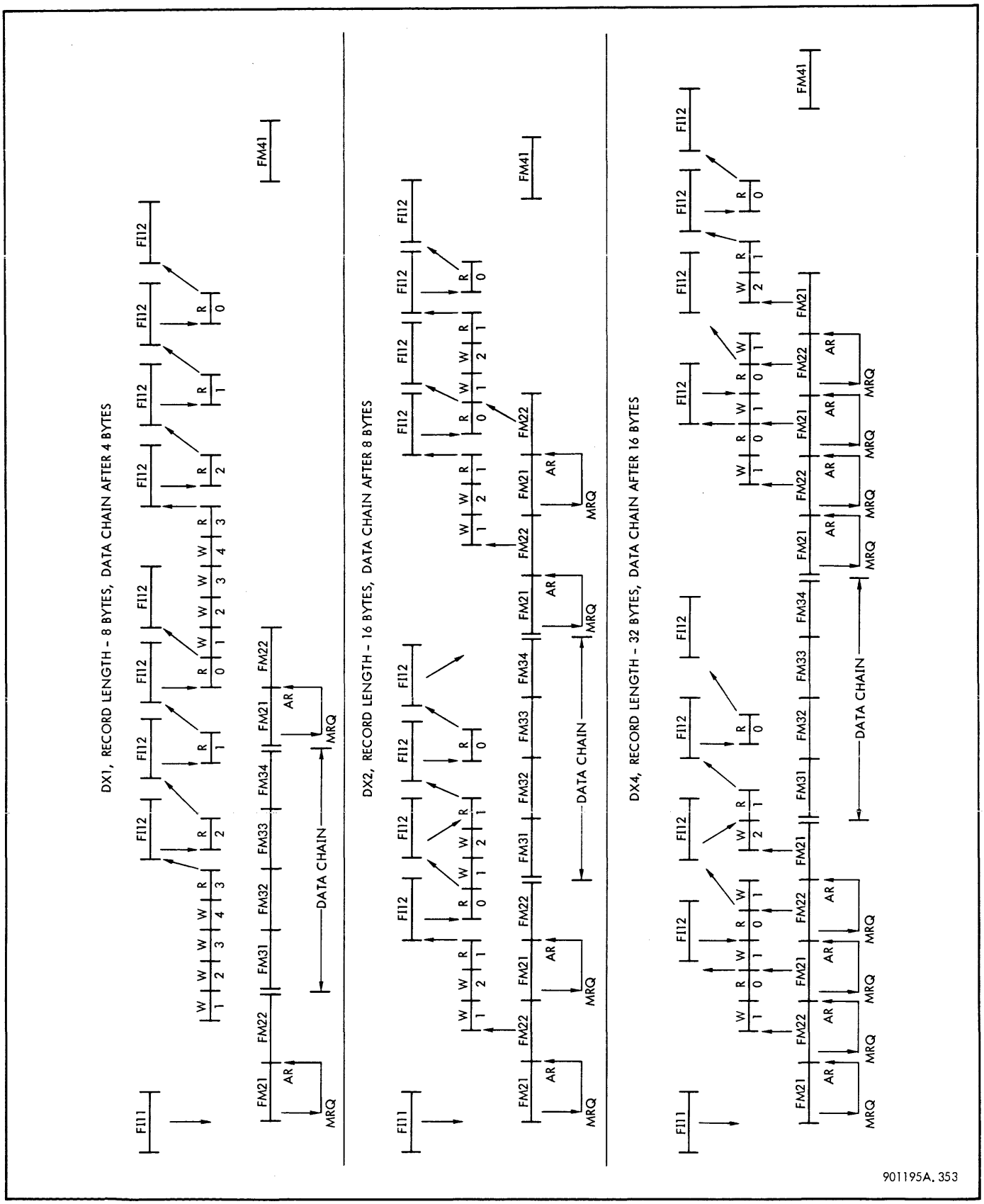


Figure 3-51. Typical Data-Out Sequences

Table 3-31. Data Out Service Cycle Phase Sequence

Phase/Clock	Logical Operation	Comments
FI11A	RS $\not\Longrightarrow$ S/TI	TI delay line started when RS is received from D/C
TIS000/1	IOX0	I/O-register latches are cleared
	CNST (IR FGAIO + MYNUM FGHIO) $\not\Longrightarrow$ S/ED	If this SIOP has an interrupt pending and CPU sends an AIO or if an HIO is addressed to this SIOP, the SIOP will terminate the D/C service request by setting ED
TIS000/2	S/MPR	Memory parity release
	R/END1, R/END2, R/END3, R/ENDFI	Latches initialized
	R/SERVICE2	Inhibits normal execution of a TIO or TDV instruction
TIS160	S/FI11B	Second rank phase latch set
	S/SERVICE1	Initializes NSERVICE1 latch
	R/FNCASC	ASC function indicator line at D/C interface dropped
	JX0	J-register latches are cleared
	EDI $\Longrightarrow$ S/ED	Signal EDI from D/C sets latch ED
	IOR $\Longrightarrow$ S/OUT	D/C drives IOR line true and holds IOR line false to specify a data out service cycle. Latch OUT is set and latch ORD remains false
TIS320	R/FI11A	First rank phase latch cleared
	DO $\Longrightarrow$ S/FM21A	Set initial phase for memory cycles
	FM21A $\Longrightarrow$ MEMRQ	Request memory bus (bus-sharing option)
	NOI $\Longrightarrow$ S/SX0/1	Enables clearing of S-register
TIS560	DO $\Longrightarrow$ S/FI12A	Data is transferred to the D/C during FI12 – that is, after data has been written into the buffer during buffer write cycles and read out of the buffer during buffer read cycles
FM21A	TMRQASHARE [(AR + NMRQ) NHFM21O22A + NMRQ END2] $\not\Longrightarrow$ S/TM	TM delay line starts when signal S/TM goes true

Notes:

1. Skips of phases are possible
2. The symbol $\Longrightarrow$ indicates a function controlled by a clock pulse. The symbol $\not\Longrightarrow$ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM21A (Cont.)		
TMA000/1	WBX0	WB-register latches are cleared
	SX0/1 ⟹ SX0	S-register latches are cleared
	NEND2 MAYWRITE NZBC2 ⟹ S/BCNT	Controls BC and BA counters
TMA000	SXBA	Byte address is transferred to S-register
	WBSET	Enables WB latches
	R/MRQ1	Latch initialized
TMS080	R/SX0/1	
TMA120	END2 ⟹ S/END3	Enables termination if ED had been set during FI12
TMS140	MWX0	Specifies core memory read operation
	BCNT NFLSKIP ⟹ S/MRQ	Core memory is accessed for data word if skip flag is false and BCNT is true. Signal BCNT implies that BA and BC counters will be updated next time CNTCK goes true. Signal BCNT also implies that latch ED had not been set during FI12 and that core memory accesses had not been inhibited because of lack of space in buffer
	BCNT ⟹ S/MRQ1	BA and BC counters were updated during FM21A
TMA160	S/FM21B	Second rank phase latch set
CNTCK	BCNT ⟹ Update BA and BC	Update BA and BC counters. Signal CNTCK is applied to clock input
TMA240	DX4 NSHARE NBCNT ⟹ S/HFM21O22A	Inhibit core memory accesses if data buffer is full
	HFM21O22A END1 ⇏ S/END2	Designates that ED is true and data buffer is full

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM21B	NJFILL ⇏ S/TM	TM delay line is started by signal NJFILL.
	MPESHARE DG ⇏ MXM	Signal DG from core memory causes signal MXM to go true, which enables memory data to enter M-register
TMB000/1	JFILLSET ⟹ JX0	J-register is cleared
TMB000	R/FM21A	First rank phase latch cleared
	R/BCNT	Latch initialized
	NREAD NSPACE ⟹ R/MAYWRITE	Inhibits setting of BCNT for updating of BA and CA counters
TMB040	JFILLSET Hi ⟹ JXMi	Data is transferred from M-register to J-register according to byte presence bit indications of H-register
	(i = A through D)	
	J8XH Hi ⟹ S/Ji8	Byte presence indicator bits of J-register are set according to H-register
	(i = A through D)	
TMB080	MRQ NAH ⟹ S/STMA	Memory address status bit set if address here is not yet received from core memory
	STMAE TMBTRIPB ⇏ S/STIOPH	IOP halt status bit set if memory address status bit set
	NREAD1 NANOTHER SPACE ⟹ S/MAYWRITE	BA and BC counters may be updated if buffer can accept data
	MRQ2 ⟹ S/JFILL	J-register is filled and buffer write cycles may start (see WRITE phase sequences, table 3-33). Also implies that BCNT was true in FM22A.
TMS140	MUX0, MLX0, HX0	M- and H-register latches are cleared
TMB160	HXWB	Contents of WB-register are transferred to H-register
	R/CHAIN	
	ZBC0T15 ⟹ S/ZBC1	Byte count has been decremented to zero

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
TMB080 (Cont.)		
TMB240	NSTMAE NZBC2 NEND3 ⟹ S/FM22A	If termination is not required, SIOP advances to FM22
	NSTMAE NZBC2 NEND3 ⟹ S/SX0/1	Enables clearing of S-register
	NSTMAE ZBC2 FLDC ⟹ S/FM31A	If data chaining is indicated and zero byte count was detected, SIOP will advance to chaining sequence of phases
	NSTMAE ZBC2 FLDC ⟹ S/SX0/1	Enables clearing of S-register
	STMAE + ZBC2 NFLDC + NZBC2 END3 ⟹ S/FM41A	SIOP advances to termination phase if (1) a memory address error had been detected, a zero byte count condition had been detected, and data chaining was not indicated, or (2) end data was set
	FM41A ⟹ TMRQASHARE	Enables TM delay line to start for termination phase
	STMAE ⟹ R/MRQ, R/MEMCYC, R/MEMCYC/1	Latches initialized
FM22A	TMRQASHARE [(AR + NMRQ) + NHFM21O22A + NMRQ END2] ⟹ S/TM	TM delay line starts when SIOP enters FM22A (when signal S/TM goes true)
TMA000/1	WBX0	WB-register latches are cleared. H-register is currently holding byte presence bits cleared from WB-register
	SX0/1 ⟹ SX0	S-register latches are cleared
	DX4 NSHARE NEND2 MAYWRITE NZBC1 ⟹ S/BCNT	Controls updating of BC and BA counters
TMA000	SXBA	Byte address transferred to S-register
	WBSET	Enables WB latches
	R/MRQ2	Latch initialized
TMS080	R/SX0/1	
TMA120	END2 ⟹ S/END3	Enables termination if ED had been set during FI12

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM22A (Cont.)		
TMS140	MWX0	Specifies core memory read operation
	BCNT NFLSKIP ⟹ S/MRQ	Core memory is accessed for data if skip flag is false and BCNT is true. Signal BCNT implies that latch ED had not been set during FI12 and that core memory access had not been inhibited because of lack of space in buffer
	BCNT ⟹ S/MRQ2	Designates that BA and BC counters were updated during FM22A
TMA160	S/FM22B	Second rank phase latch set
CNTCK	BCNT ⟹ Update BA and BC	Update BA and BC counters. Signal CNTCK is applied to clock input
TMA240	NBCNT ⟹ S/HFM21O22A	Inhibit core memory accesses. Data buffer can accept no more data
	HFM21O22A END1 ⇏ S/END2	ED is true and data buffer can accept no more data
FM22B	NJFILL ⇏ S/TM	TM delay line is started by signal NJFILL. Signal DG from core memory causes signal MXM to go true, which enables memory data to enter M-register
	MPESHARE DG ⇏ MXM	
TMB000/1	JFILLSET ⟹ JX0	J-register is cleared
TMB000	R/FM22A	First rank phase latch is cleared
	R/BCNT	Latch initialized
	NREAD NSPACE ⟹ R/MAYWRITE	Inhibits setting of BCNT for update of BA and BC counters
TMB040	JFILLSET Hi ⟹ JXMi	Data is transferred from M-register to J-register according to byte presence bit indications of H-register
	(i = A through D)	
	J8XH Hi ⟹ S/Ji8	Byte presence indicator bits of J-register are set according to H-register
	(i = A through D)	

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM22B (Cont.)		
TMB080	MRQ NAH ⟹ S/STMAE	Memory address status bit set if address here signal has not yet been received from core memory
	STMAE TMBTRIPB ⇏ S/STIOPH	IOP halt status bit set if memory address status bit set
	NREAD1 NANOTHER SPACE ⟹ S/MAYWRITE	BA and BC counters may be updated if buffer can accept data
	MRQ1 ⟹ S/JFILL	J-register is filled and buffer write cycles may start. (See WRITE phase sequences, table 3-33.)
TMS140	MUX0, MLX0, HX0	M- and H-register latches are cleared
TMB160	HXWB	Contents of WB-register are transferred to H-register
	ZBC0T15 ⟹ S/ZBC2	Byte count was decremented to zero during FM22
TMB240	NSTMAE NZBC1 NEND3 ⟹ S/FM21A	SIOP returns to FM21 to access more data from core memory if termination or chaining is not required
	NSTMAE NZBC1 NEND3 ⟹ S/SX0/1	Enables clearing of S-register
	FM21A ⇏ MEMRQ	Requests core memory bus (bus-sharing option)
	NSTMAE ZBC1 FLDC ⟹ S/FM31A, S/SX0/1	If data chaining is indicated and zero byte count was detected, SIOP will advance to chaining sequence of phases and enable clearing of S-register
	FM31A ⇏ MEMRQ	Requests core memory bus (bus-sharing option)
	STMAE + ZBC1 NFLDC + NZBC1 END3 ⟹ S/FM41A	SIOP advances to termination phase if (1) a memory address error had been detected, (2) zero byte count had been detected and data chaining flag is false, or (3) end data is true and zero byte count had not been detected
	FM41A ⇏ TMRQASHARE	Enables TM delay line to start for termination phase

Notes:

1. Skips of phases are possible

2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM22B TMB080 (Cont.)	STMAE ⟹ R/MRQ	Memory request latch initialized if a core memory address error was detected
	FLSKIP ⟹ R/MEMCYC/1	Enables requests for memory bus if skip flag is true
	FM22B NFLSKIP (POK + PE) ⇏ R/MEMCYC/1	Enables requests for memory bus if skip flag is false (after memory returns either parity OK or parity error signal)
FI12	RS NRSA (KFILL1 + KFILL2 + STIOPH) ⇏ S/TI	TI delay line starts when signal RS is received from D/C for data and K-register is filled (KFILL1 or KFILL2), or when signal RS is received for a terminal order (STIOPH)
TIS000/1	IOX0	I/O-register latches are cleared
	CNST (IR FGAIO + MYNUM FGHIO)	Latch end data is set if:
	+ KFILL2 + STIOPH	a. This SIOP has no interrupt pending and the CPU sends an AIO. (If an HIO is addressed to this SIOP, the SIOP will terminate the D/C service request by setting ED)
	+ FLDC FLIZC NFAST CHAIN EMPTY/1 ⟹ S/ED	b. Zero byte count was reached without data chaining and all data written into buffer has been emptied
		c. An error condition had been detected that will cause SIOP to halt
		d. Data chaining and interrupt at zero byte count flags are true, D/C is not driving FAST line, a chaining sequence is not in progress, and buffer is empty
TIS000/2	IOXK	Contents of the K-register (loaded from buffer during a read cycle) are transferred to I/O-register and therefore to D/C
TIS080	NK82SET; NK82SET ⇏ R/KFILL1, S/NKi8 (i = A through D)	Byte presence bits of K-register and KFILL1 are initialized
	NK8 ⇏ S/SREAD	A request to read data from buffer to K-register is made when K-register byte presence bits indicate K-register is empty

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FI12 (Cont.)		
TIS160	S/FI12B	Second rank phase latch set
	KX0	Data bits of all byte positions of K-register are cleared
	EDI ⟹ S/ED	Latch end data is set if D/C drives ED line (EDI)
TIS240	ED ⟹ S/END1	Designates that ED was set during FI12.
TIS320	R/FI12A	First rank phase latch set
	NED NDX1 ⟹ S/RSA	Request strobe acknowledge sent to D/C to permit D/C to read data on data lines. (D/C checks parity only on data lines DA0 through DA7)
TIS400	NED DX1 ⟹ S/RSA	RSA must be delayed if D/C interface is only one byte wide. (SIOP requires extra time to generate parity)
TIS480	ED ⟹ S/RSA	Signal RSA is delayed until TIS480 if ED is true
	ED RSA ⇏ S/SERVICE2	Enables normal execution of a TIO or TDV directed to this SIOP
	SERVICE2 MYTEST ⇏ R/SERVICE1	A TIO or TDV will be normally executed if signal NSERVICE1 is true. Signal MYTEST is true if a TIO or TDV is directed to this SIOP
TIS560	NED ⟹ S/FI12A	SIOP will sequence through FI12 again if more data is to be sent to D/C
	ED ⟹ S/ENDFI	No more data will be transferred to D/C
FM31A	TMRQASHARE NMRQ MPR ⇏ S/TM AR ⇏ R/MRQ MPESHARE PE ⇏ S/MPE POK + MPE ⇏ S/MPR	Delay line is started for FM31A. Latch MRQ is cleared when address release AR is received from core memory. Latch MPE is set if a core memory parity error was detected. Latch memory parity release MPR is set when core memory signals either parity OK (POK) or parity error (MPE)

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM31A (Cont.) TMA000/1	SX0/1 ⟹ SX0	S-register latches are cleared
TMA000	SXCA	Command address (even) is set into S-register
	R/MRQ1, R/MRQ2	Latches initialized
TMS080	R/SX0/1	Latch initialized
TMA080	MPE ⟹ S/STTME, S/DATAERR	If a memory parity error had been detected during a prior memory request when data chaining, the transmission memory error status bit (STTME) and the transmission data error or transmission memory error latch (DATAERR) will be set
TMS140	MUX0, MLX0	M-register latches are cleared
	MWX0	MW-register latches are cleared. (Specifies a core memory read operation)
	NCM ⟹ S/MRQ	During data chaining, chaining modifier bit is false (NCM); therefore, SIOP reads even command doubeword address from core memory
	NCM ⟹ S/MRQ1, S/MRQ2	Designates that MRQ was raised
TMA160	S/FM31B	Second rank phase latch set
	S/CHAIN	Controls transfer of new flags and clearing of old flags during termination phase
	R/MPE	Memory parity error latch initialized
CNTCK	TMATRIPB CACNTU	Increments command address by one to specify next command doubleword address
FM31B	TMRQB ⟹/ S/TM	TM delay line is started for the second rank phase latch
TMB000	R/FM31A	Clear first rank phase latch
	MRQ1 ⟹ R/MPR	Memory parity release latch initialized
TMB080	MRQ NAH ⟹/ S/STMAE	Memory address error status bit set if a memory request was made (MRQ), and signal address here not yet received (NAH)

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹/ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM31B		
TMB080 (Cont.)	STMAE TMBTRIPB $\not\Rightarrow$ S/STIOPH	IOP halt status bit (STIOPH) set if a core memory address error (STMAE) was detected
TMB240	NSTMAE NMRQ2 $\Longrightarrow$ S/SX0/1	Enables clearing of S-register latches
	NSTMAE MRQ2 $\Longrightarrow$ S/FM32A	Advance to FM32A if a memory access was made (MRQ2), and no memory address error detected (NSTMAE)
	STMAE $\Longrightarrow$ S/FM33A	Advance to FM33A if memory address not present
	STMAE $\not\Rightarrow$ S/MPR, R/MRQ	Signal MPR signifies core memory parity release. Core memory request latch MRQ is cleared
FM32A	TMRQASHARE NMRQ MPR $\not\Rightarrow$ S/TM AR $\not\Rightarrow$ R/MRQ POK + MPE $\not\Rightarrow$ S/MPR MPESHARE PE $\not\Rightarrow$ S/MPE MPESHARE DG $\not\Rightarrow$ MXM	TM delay line is started for FM32A. Memory request (MRQ) signal goes false when address release (AR) is received from core memory. Memory parity release (MPR), if not set during FM31B because of a memory address error, is set by signal parity OK (POK) from core memory. If a memory parity error (PE) is detected, latch MPE will be set. Signal data gate (DG) from memory causes MXM to go true. Signal MXM gates data on memory data lines into M-register
TMA000	MPE $\Longrightarrow$ S/STIOPME	Memory error status bit STIOPME is set if a memory parity error is detected
	R/MRQ1	Latch initialized
TMA160	S/FM32B	Second rank phase latch set
	TRA TRA1 $\Longrightarrow$ S/STIOPCE	Signals TRA and TRA1 both true indicate a control error and will cause control error status bit (STIOPCE) to be set
CNTCK	TMATRIPB NMPE TRA NTRA1 $\Longrightarrow$ CAXM	If order bits specified a transfer-in-channel command (TRA), new command address (specified by TRA command) is loaded into CA counter (CAXM)
FM32B	TMRQB $\not\Rightarrow$ S/TM	TM delay line is started for second rank phase latch

Notes:

1. Skips of phases are possible
2. The symbol $\Longrightarrow$ indicates a function controlled by a clock pulse. The symbol $\not\Rightarrow$ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM32B (Cont.)		
TMB000	R/FM32A	Clear first rank phase latch
TMB080	TRA ⟹ S/TRA1	If transfer-in-channel command (TRA), latch TRA1 is set
CNTCK	TMBTRIPB NMPE NTRA ⟹ BAXM	New byte address is transferred to BA counter
	BAXM BKWD NMD6 ⇏ S/BA17	Initializes least 2 significant bits (BA17 and BA18) of byte address as a function of signal BKWD
	BAXM BKWD NMD7 ⇏ S/BA18	
TMB240	NSTIOPH TRA ⟹ S/FM31A, S/SX0/1	If order was transfer-in-channel (TRA) and no IOP halt conditions exist, enter FM31A to fetch command specified by TRA. Signal S/SX0/1 enables clearing of S-register
	STIOPH + NTRA ⟹ S/FM33A	Next phase of chaining sequence if last command accessed from core memory was not a transfer-in-channel command or if IOP halt status bit is true
	(STIOPME + STIOPCE) TMBTRIPB ⇏ S/STIOPH	IOP halt status bit is set if a memory error or control error was detected
FM33A	TMRQASHARE NMRQ MPR ⇏ S/TM	TM delay line is started for FM33A
TMA000	S/S31	Prepare to access odd word of command doubleword
	R/MRQ2	Latches initialized
TMS140	MUX0 MLX0	M-register latches are cleared
	MSTIOPH ⟹ S/MRQ	Access odd word of command doubleword
	NSTIOPH ⟹ S/MRQ1	Designates that MRQ was raised
TMA160	S/FM33B	Second rank phase latch set
FM33B	TMRQB ⇏ S/TM	TM delay line is started for FM33B
TMB000	R/FM33A	First rank phase latch cleared
	MRQ1 ⟹ R/MPR	Memory parity release latch initialized
TMB160	R/TRA1	Latch initialized

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM33B (Cont.)		
TMB240	S/FM34A	Next phase of data out service cycle
FM34A	TMRQASHARE NMRQ MPR ⇏ S/TM	TM delay line is started for FM34A (see FM32A for details)
	AR ⇏ R/MRQ	
	POK + MPE ⇏ S/MPR	
	MPESHARE PE ⇏ S/MPE	
	MPESHARE DG ⇏ MXM	
TMA000	R/NFLCLEAR, R/ENDDIO, R/MRQ1	Initialize latches
TMA080	MPE ⟹ S/STIOPME	IOP memory error status bit set if a memory parity error was detected
TMA160	S/FM34B	Second rank phase latch set
	R/MPE	Latch cleared
	(DO FLIZC NFAST) + STIOPME + STIOPH + END3 ⟹ S/NFLCLEAR	Inhibits transfer of new flags from M-register to FL-register
CNTCK	TMATRIPB NSTIOPH NMPE ⟹ BCXM	New byte count transferred to BC counter if no errors are detected
TMA240	R/KFILL2, R/MEMCYC	Clears latches
FM34B	TMRQB ⇏ S/TM	TM delay line is started for FM34A
TMB000	R/FM34A	Clear first rank phase latch
	R/ZBC1, R/ZBC2	Latches initialized
TMB040	FLCLEAR ⟹ FLX0	Old flags are cleared from FL-register
TMB160	FLCLEAR ⟹ FLXM	New flags are transferred from M-register to FL-register
TMB240	NFLCLEAR ⟹ S/FM41A	The conditions that prevented the flags from being cleared (NFLCLEAR) also cause the SIOP to advance to the termination phase (FM41)
	FM41A ⇏ TMRQASHARE	Enables the TM delay line to start. (Applies when bus-sharing option is installed)

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM34B TMB240 (Cont.)	FLCLEAR DO ⟹ S/FM21A	If termination is not required at completion of chaining sequence, SIOP returns to FM21, during which time it will access more data. Core memory location and number of bytes will be determined by new byte count and byte address, and operation of SIOP will be directed by new flags
	FLCLEAR DO ⟹ S/SX0/1	Enables clearing of S-register
	FM21A ⇏ MEMRQ	Request made for memory bus. (Applies when bus-sharing option is installed)
	FLCLEAR ⟹ R/MEMCYC/1	Memory bus is not in use. (Applies when bus-sharing option is installed)
	STIOPME TMBTRIPB ⇏ S/STIOPH	IOP halt status bit set if memory error status bit is true
FM41A	TMRQASHARE NMRQ RS NRSA (NORD MPR ENDFI + ...) ⇏ S/TM AR ⇏ R/MRQ MPESHARE PE ⇏ S/MPE POK + MPE ⇏ S/MPR	TM delay line starts when RS is received from D/C if conditions a thru d exist: a. D/C has dropped RS from previous request, causing NRSA to be true b. Address release AR has been received from core memory (NMRQ) c. Parity OK or parity error has been received from core memory (MPR) d. End data has been received from D/C during FI12 (ENDFI)
TMA000/1	IOX0	I/O-register latches cleared
	MPE ⟹ S/STTME, S/DATAERR	Transmission memory error status bit STTME is set, and DATAERR (temporarily stores transmission data error or transmission memory error status) is set
TMA000	S/ES	Designates end of service. Signal ES is applied to reset input of service connect flip-flop in D/C so that it will reset when D/C receives next RSA from SIOP
	R/MRQ1	Latch initialized
TMA080	DATAERR FLHTE ⟹ S/STIOPH	IOP halt status bit is set if a transmission error had been detected and halt on transmission error flag FLHTE is true

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-31. Data Out Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM41A (Cont.)		
TMA160	S/FM41B	Second rank phase latch set
	DO KFILL2 NK8 FLIZC ⟹ S/IOA0, STZBCI DO KFILL2 ZBC1 ⟹ S/IOA1 FLCC ⟹ S/IOA2 STIOPH ⟹ S/IOA3	Terminal order bits are set (see table 3-25) and appropriate interrupt status bits are set. Bit IOA0 drives data line DA0 to request an interrupt from D/C. Zero byte count interrupt status bit STZBCI is set if an interrupt request is made Bit IOA1 drives data line DA1 to transmit count done to D/C Bit IOA2 drives data line DA2 to indicate command chaining to D/C Bit IOA3 drives data line DA3 to transmit an IOP halt to D/C
FM41B	TMRQB ⟹/ S/TM	TM delay line is started for FM41B
FMB000	R/FM41A	Second rank phase latch cleared
TMB040	CHAIN NSTIOPH ⟹ FLX0	Old flags are cleared from FL-register
TMB120	S/RSA	D/C reads terminal order and disconnects. Signal RSA permits service connect flip-flop in D/C to reset, thus disconnecting D/C from interface lines
TMB160	CHAIN NSTIOPH ⟹ FLXM	New flags are transferred from M-register to FL-register
TMB240	CHAIN ⟹ R/MEMCYC/1	Designates that memory bus is not in use. (Applies when bus-sharing option is installed)
	S/FIO1A	First of preliminary phase latches initialized

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹/ indicates a function not controlled by a clock pulse

Figure 3-52 shows typical phase sequences during a data in service cycle with D/C interface path widths of one (DX1), two (DX2), and four (DX4) bytes. During FI11, the SIOP strobes the data receivers at the D/C interface and transfers the data to the I/O-register. During a later part of FI11, the data is transferred from the I/O-register to the J-register and latch JFILL is set. Signal JFILL then initiates a buffer write cycle that transfers the data from the J-register to the buffer (R-register). After the valid data has been emptied from the J-register, signal NJFILL enables FI13, during which time more data is received from the D/C. This sequence is repeated until end data is signaled by either the SIOP or the D/C.

The operation of the SIOP at the core memory interface is somewhat independent of the operation at the D/C interface. When the buffer has data, buffer read cycles are initiated that transfer the data from the buffer to the

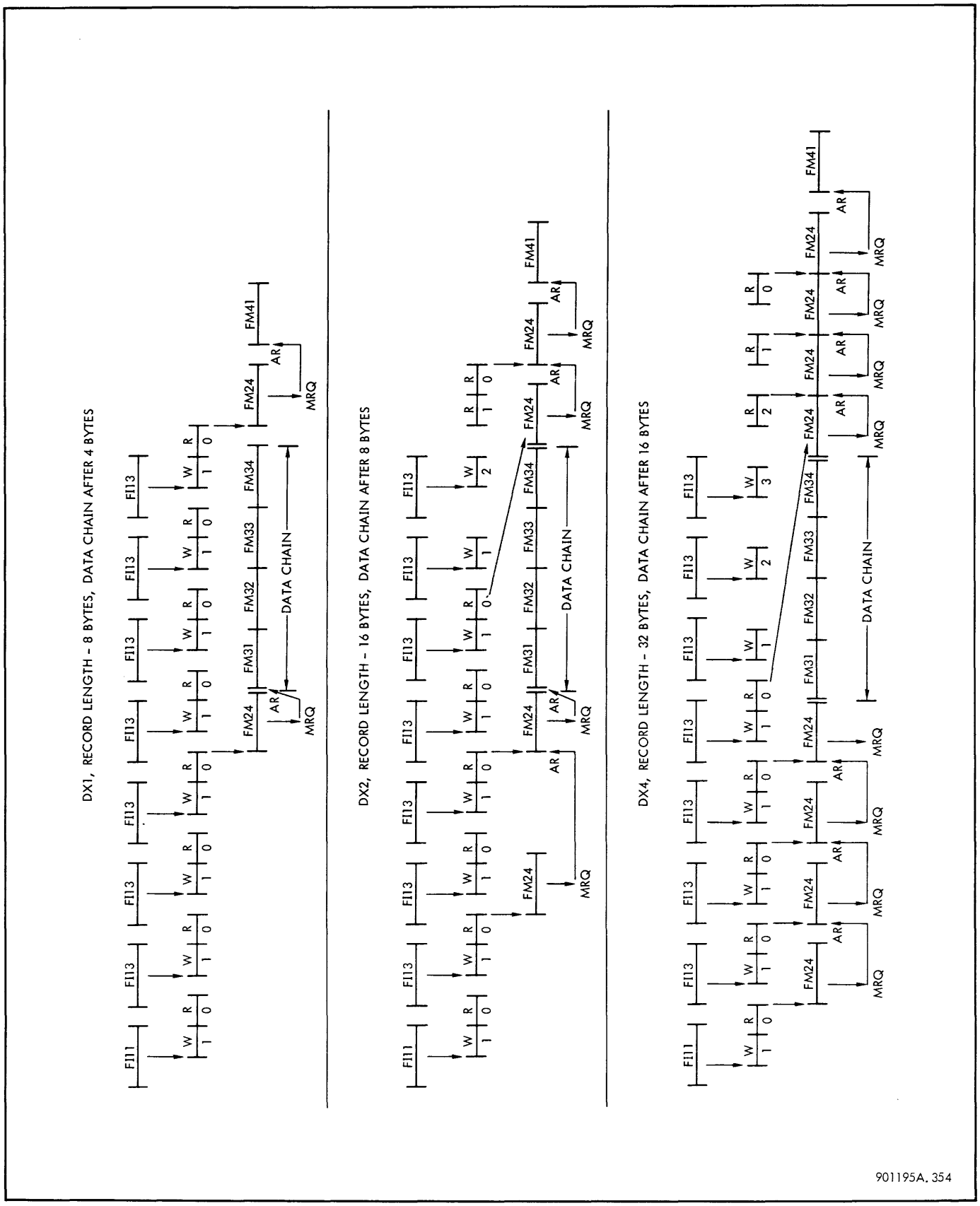


Figure 3-52. Typical Data-In Sequences

K-register. When the K-register has the specified data, signal KFILL enables FM24, during which time memory requests (MRQ) are made to transfer the data to core memory (via the M-register). Signal NK81SET clears the K-register, which in turn enables additional buffer read cycles. This sequence proceeds until the byte count is decremented to zero (ZBC0T15) or END3 is signaled.

If zero byte count is reached with the data chain flag true, the SIOP enters the chaining sequences (FM31 through FM34) and subsequently returns to exchange data in FM24. If zero byte count is reached with the data chain flag false, or if END3 is true, the SIOP terminates the service with a terminal order in phase FM41 (see table 3-25). See table 3-32 for a detailed description of the operations that occur during a data in service cycle.

3-72 WRITE AND READ PHASE SEQUENCES. A request to write data into the buffer is begun when the J-register (buffer input register) is full. This condition is defined by signal SWRITE, which is controlled by signal JFILL. During data in service, latch JFILL is set when the initial data is received from the D/C (FI11), and when subsequent data is received (FI13).

A request to read the data from the buffer into the K-register is initiated when the K-register is not full. This condition is defined by signal SREAD, which is controlled by signal NKD8 (K-register not full). The same applies during data out service, except that during data out service, latch JFILL is set during FM21 and FM22, when the data is accessed from core memory. See tables 3-33 and 3-34 for a detailed description of the write and the read phase respectively.

3-73 MEMORY BUS-SHARING OPTION

When the memory bus-sharing option is installed in two SIOPs, they may time share a core memory bus. Even though the two SIOPs may request service from s Sigma 5 or 7 core memory on the same bus, only one SIOP may be granted use of the bus at any one time. To resolve the conflict when both SIOPs request the bus at the same time, one of the two is given higher priority. (See section II for switch settings relating to bus-sharing priority.) Interconnection of two bus-sharing SIOPs is shown in figure 3-53.

Table 3-32. Data In Service Cycle Phase Sequence

Phase/Clock	Logical Operation	Comments
FI11	RS ⇏ S/TI	TI delay line starts when request strobe RS is received from D/C
TIS000/1	IOX0	I/O-register latches are cleared
	CNST (IR FGAIO + MYNUM FGHIO) ⟹ S/ED	If this SIOP has an interrupt pending and the CPU sends an AIO, or if an HIO is addressed to this SIOP, the SIOP will terminate the D/C service request by setting end data ED
TIS000/2	S/MPR	Memory parity release
	R/END3, R/ENDFI, R/END1, R/END2	Latches initialized
	R/SERVICE2	Inhibits normal execution of a TIO or TDV instruction
TIS160	S/FI11B	Second rank phase latch set

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FI11		
TIS160 (Cont.)	S/SERVICE1	Initializes NSERVICE1 latch
	R/FNCASC	ASC function indicator line at D/C is dropped
	IOXD	Data on data lines is set into I/O-register
	JX0	J-register latches are cleared
	DOR ⟹ S/ORD, IOR ⟹ S/OUT	D/C holds DOR and IOR lines false to specify a data out service cycle. Latches ORD and OUT remain false to retain this information
	EDI ⟹ S/ED	Signal EDI from D/C (sent on ED line) sets latch ED
TIS240	JAXI0, JXI0	Data received from D/C is transferred from I/O-register to J-register
TIS320	R/FI11A	First rank phase latch cleared
	DI ⟹ S/FM24A	Next phase of a data in service cycle
	FM24A ⇏ MEMRQ	Request made for memory bus. (Applies when bus-sharing option is installed)
	NOI ⟹ S/SX0/1	Enables clearing of S-register
	DI ⟹ S/JFILL	J-register is filled and buffer write cycles may start (see table 3-33)
	DI NED ⟹ S/RSA	Request strobe acknowledge sent to D/C
TIS400	DI PC DX1 IOAEVEN ⟹ S/STTDE, S/DATAERR	SIOP checks parity only on byte A of data lines (DX1) if parity check PC is true. If a parity error is detected (IOEVEN), status bits STTDE and DATAERR will be set
TIS480	NOUT ED ⟹ S/RSA	If end data is true, SIOP delays until TIS480 to send RSA. Otherwise, RSA is sent during TIS320. Delay is required to give ED line time to settle before being strobed by D/C
	ED RSA ⇏ S/SERVICE2	Enables normal execution of a TIO or TDV instruction directed to this SIOP

Notes:

1. Skips of phases are possible

2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FI11		
TIS480 (Cont.)	SERVICE2 NMYTEST ⇏ R/SERVICE1	A TIO or TDV will be normally executed if signal NSERVICE1 is true. Signal MYTEST is true if a TIO or TDV is directed to this SIOP
TIS560	DI ⇒ IOX0	I/O-register latches are cleared
	DI ED ⇒ S/ENDFI	This will be last TI delay line phase if ED is true
	DI NED ⇒ S/FI13A	If ED is not true, SIOP will accept more data from D/C during FI13
FM24A	TMRQASHARE MPR (AR + NMRQ) (KFILL1 + END2 + KD8 DREAD TRS290) ⇏ S/TM AR ⇏ R/MRQ MPESHARE PE ⇏ S/MPE POK + MPE ⇏ S/MPR	TM delay line starts for FM24A if conditions a, b, and c exist: a. A memory request had not yet been made (NMRQ) or if one was made and address release (AR) was received b. Core memory signaled either parity OK (POK) or parity error (MPE), signified by signal memory parity release (MPR) c. One or more of the following conditions: 1. Signal ED is true and last data has been written into and read from buffer (END2) 2. Buffer is filled and data may be written into core memory during FM24 (KFILL1) 3. During timing period TRS290 if KD8 and DREAD are true (a condition that controls KFILL1). (This condition provides earlier triggering of delay line than signal KFILL1 because circuit delays occur in setting KFILL1)
TMA000/1	WBX0	WB-register latches initialized
	S/BCNT	Updates BA and BC counters
	SX0/1 ⇒ SX0	S-register latches are cleared

Notes:

1. Skips of phases are possible
2. The symbol ⇒ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM24A (Cont.)		
TMA000	SXBA	Byte address is transferred to S-register
	WBSET	Enables setting of byte presence bits
	R/MRQ1	Latch initialized
	N(BAA0 NEND2) + BCC4T1 ⟹ R/MAYREAD	Inhibits buffer read cycles
TMS080	R/SX0/1	Latch initialized
	TMATRIPB MAYREAD ⟹ NK81SET NK81SET ⇏ R/KFILL1	Presence bits of K-register and KFILL1 initialized
TMA080	MPE ⟹ S/STTME, S/DATAERR	Status bits set if a memory error is detected
TMA120	END2 ⟹ END3	Enables termination if ED was set
TMA140	MUX0, MLX0, MWX0	Clears M-register latches and memory write byte latches
	NFLSKIP ⟹ S/MRQ, S/MRQ1	Memory request made to write data into core memory if skip flag is false. Signal MRQ1 designates that MRQ was raised
TMA160	S/FM24B	Second rank phase latch set
	R/MPE	Latch initialized
	NBKWD ⟹ MXKF, MWXWBF	If D/C is executing a read forward order (NBKWD), data is transferred from K-register byte positions A-D to M-register byte positions A-D respectively. Data is subsequently written into ascenting core memory locations
	BKWD ⟹ MXKB, MWXWBB	If D/C is executing a read backward order (BKWD), data is transferred from K-register byte positions A-D to M-register byte positions D-A respectively, thus providing for subsequent writing of data into descending core memory locations
CNTCK	BCNT ⟹ Update BA and BC	BA and BC counters are updated. Signal CNTCK is applied to clock input
TMS240	TMATRIPB ⟹ KX0	K-register latches are cleared
TMA240	R/MEMCYC	Enables bus-sharing priority logic PRQA and PRQB

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM24B	TMRQB ⟹ S/TM	TM delay line is started for FM24B
TMB000	R/FM24A	First rank phase latch cleared
	R/BCNT	Latch initialized
TMB040	MRQ1 WP ⟹ R/MPR	Memory parity release latch cleared if a partial write core memory request was made
	MRQ NAH ⟹ S/STMAE; STMAE TMBTRIPB ⇏ S/STIOPH	Memory error and IOP halt status bits set if address here (AH) is not received from core memory
	MAYREAD NZBC0T15 ⟹ NK81SET	Byte presence bits of K-register and KFILL1 initialized if zero byte count has not been reached
TMB160	R/CHAIN	Latch initialized
	ZBC0T15 ⟹ S/ZBC1	Zero byte count was reached during this phase
TMB240	NSTMAE NZBC0T15 NEND3 ⟹ S/FM24A, S/SX0/1	If termination is not required – that is, no memory address error, not zero byte count, and not end data – SIOP will again cycle through FM24 to transfer additional data to core memory. Signal SX0/1 enables clearing of S-register
	FM24A ⇏ MEMRQ	Request made for core memory bus
	NSTMAE ZBC0T15 FLDC ⟹ S/FM31A, S/SX0/1	If zero byte count was detected with no memory address errors and data chaining flag is true, SIOP will advance to chaining sequence of phases. Signal SX0/1 enables clearing of S-register
	FM31A ⇏ MEMRQ	Request made for core memory bus
	STMAE + ZBC0T15 NFLDC + NZBC0T15 END3 ⟹ S/FM41A	SIOP advances to termination phase if (1) a memory address error has been detected, (2) zero byte count has been detected and data chaining flag is false, or (3) zero byte count has not yet occurred and end data is true
	FM41A ⇏ TMRQASHARE	Enables TM delay line to start for the termination phase

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM24B		
TMB240 (Cont.)	NZBC0T15 ⟹ S/MAYREAD	Enables buffer read cycles
	ZBC0T15 FLIZC ⟹ S/STZBCI	Zero byte count interrupt status bit is set if zero byte count is reached and interrupt at zero byte count flag is true
	STMAE ⟹ R/MRQ, R/MEMCYC/1	Initializes core memory latches if a memory address error was detected
	FLSKIP ⟹ R/MEMCYC/1	Enables requests for core memory bus if skip flag is true
	FM24B NFLSKIP [NWP AR + WP (POK + PE)] ⟹ R/MEMCYC/1	Enables requests for memory bus if (1) skip flag is false and AR is received from memory during full write (NWP), or (2) if the skip flag is false during partial write (WP) and either parity OK (POK) or parity error (PE) has been received from memory
FI13	RS NRSA (NJFILL + FM41A) ⟹ S/TI	TI delay line starts when RS is received from D/C if J-register can accept data (NJFILL) or if a terminal order is specified (FM41A)
TIS000/1	IOX0	I/O-register latches are cleared
	CNST (IR FGAIO + MYNUM FGHIO) + FM41A + ENDDIO ⟹ S/ED	Latch ED is set if condition a, b, or c exists: a. This SIOP has an interrupt pending and the CPU sends an AIO, or an HIO is addressed to this SIOP, (the SIOP will terminate the D/C service request by setting ED) b. Termination phase latch has already been set c. Zero byte count is to be specified via a terminal order
TIS160	S/FI13B	Second rank phase latch is set
	IOXD	Data on data lines is set into I/O-register
	JX0	J-register latches are cleared
	EDI ⟹ S/ED	Signal EDI from D/C (received on ED line) sets latch ED

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FI13 (Cont.)		
TIS240	JAXIO, JXIO	Data received from D/C is transferred from I/O-register to J-register
	S/JFILL	J-register is filled and buffer write cycles may start (see table 3-33)
TIS320	R/FI13A	First rank phase latch cleared
	NED ⟹ S/RSA	Request strobe acknowledge sent to D/C
TIS400	DI PC DX1 IOAEVEN ⟹ S/STTDE, S/DATAERR	SIOP checks parity only on byte A of data lines (DX1) if parity check PC is true. If a parity error is detected (IOEVEN), status bits STTDE and DATAERR will be set
TIS480	ED ⟹ S/RSA	If ED is true, SIOP delays until TIS480 to send RSA. Otherwise, RSA is sent during TIS320. Delay is required for ED line to settle before being strobbed by D/C
	ED RSA ⇏ S/SERVICE2	Enables normal execution of a TIO or TDV instruction directed to this SIOP
	SERVICE2 NMYTEST ⇏ R/SERVICE1	A TIO or TDV will be executed normally if signal NSERVICE1 is true. Signal MYTEST is true if a TIO or TDV is directed to this SIOP
TIS560	IOX0	I/O-register latches are cleared
	NED ⟹ S/FI13A	If end data has not been specified by either S/C or SIOP, SIOP returns to FI13 to accept more data from D/C
	ED ⟹ S/ENDFI	This is last TI delay line phase if no more data is to be accepted
FM31A	TMRQASHARE NMRQ MPR ⇏ S/TM AR ⇏ R/MRQ MPESHARE PE ⇏ S/MPE POK + MPE ⇏ E/MPR	Delay line is started for FM31A. Latch MRQ is cleared when address release AR is received from core memory. Latch MPE is set if a core memory parity error was detected. Latch memory parity release MPR is set when core memory signals either parity OK (POK) or parity error (MPE)

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM31A (Cont.)		
TMA000/1	SX0/1 ⟹ SX0	S-register latches are cleared
TMA000	SXCA	Command address (even) is set into S-register
	R/MRQ1, R/MRQ2	Latches initialized
TMS080	R/SX0/1	Latch initialized
TMA080	MPE ⟹ S/STTME, S/DATAERR	If a memory parity error is detected when memory is read, transmission memory error status bit (STTME) and the transmission data error or transmission memory error latch (DATA ERR) will be set
TMS140	MUX0, MLX0	M-register latches are cleared
	MWX0	MW-register latches are cleared (specifies a core memory read operation)
	NCM ⟹ S/MRQ	Since chaining modifier bit is false (NCM), SIOP reads even command doubleword address from core memory
	NCM ⟹ S/MRQ1, S/MRQ2	Designates that MRQ was raised
TMA160	S/FM31B	Second rank phase latch set
	S/CHAIN	Controls transfer of new flags and clearing of old flags during termination phase
	R/MPE	Memory parity error latch initialized
CNTCK	TMATRIPB ⟹ CACNTU	Increments command address by one to specify next command doubleword address
FM31B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM31A	Clears first rank phase latch
	MRQ1 ⟹ R/MPR	Memory parity release latch initialized
TMB080	MRQ NAH ⇏ S/STMAE	Memory address error status bit set if a memory request was made (MRQ), and signal address here not yet received (NAH)
	STMAE TMBTRIPB ⇏ S/STIOPH	IOP halt status bit (STIOPH) set if a core memory address error (STMAE) was detected

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM31B (Cont.)		
TMB240	NSTMAE NMRQ2 ⟹ S/SX0/1	Enables clearing of S-register latches
	NSTMAE MRQ2 ⟹ S/FM32A	Advance to FM32A if a memory access was made (MRQ2), and no memory address error was detected (NSTMAE)
	STMAE ⟹ S/FM33A	Advance to FM33A if memory address not present
	STMAE ⟹ S/MPR, R/MRQ	Signal MPR signifies core memory parity release. Core memory request latch MRQ is cleared
FM32A	TMRQASHARE NMRQ MPR ⇏ S/TM AR ⇏ R/MRQ POK + MPE ⇏ S/MPR MPESHARE PE ⇏ S/MPE MPESHARE DG ⇏ MXM	TM delay line is started for FM32A. Memory request (MRQ) signal goes false when address release (AR) is received from core memory. Memory parity release (MPR), if not set during FM31B because of a memory address error, is set by signal parity OK (POK) from core memory. If a memory parity error (PE) is detected, latch MPE will be set. Signal data gate (DG) from memory causes MXM to go true. Signal MXM gates data on memory data lines into M-register
TMA000	MPE ⟹ S/STIOPME	Memory error status bit STIOPME is set if a memory parity error is detected
	R/MRQ1	Latch initialized
TMA160	S/FM32B	Second rank phase latch set
	TRA TRA1 ⟹ S/STIOPCE	Signals TRA and TRA1 both true indicate a control error, and will cause control error status bit (STIOPCE) to be set
CNTCK	TMATRIPB NMPE TRA NTRA1 ⟹ CAXM	If order bits specified a transfer-in-channel command (TRA), new command address (specified by TRA command) is loaded into CA counter (CAXM)
FM32B	TMRQB ⇏ S/TM	TM delay line is started for second rank phase latch
TMB000	R/FM32A	Clear first rank phase latch
TMB080	TRA ⇏ S/TRA1	If transfer-in-channel command (TRA), set latch TRA1

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM32B (Cont.)		
CNTCK	TMBTRIPB NMPE NTRA⟹BAXM	New byte address is transferred to BA counter
	BAXM BKWD NMD6⇏S/BA17 BAXM BKWD NMD7⇏S/BA18	Initializes least two significant bits (BA17 and BA18) of byte address as a function of signal BKWD
TMB240	NSTIOPH TRA⇏S/FM31A, S/SX0/1	If order was transfer-in-channel (TRA) and no IOP halt conditions exist, reenter FM31A to fetch command specified by TRA. Signal S/SX0/1 enables clearing of S-register
	STIOPH + NTRA⇏S/FM33A	Next phase of chaining sequence if last command accessed was not a transfer-in-channel command, or if IOP halt status bit is true
	(STIOPME + STIOPCE) TMBTRIPB⇏S/STIOPH	IOP halt status bit is set if a memory error or control error was detected
FM33A	TMRQASHARE NMRQ MPR⇏S/TM	TM delay line is started for FM33A.
TMA000	S/S31	Prepare to access odd word of command doubleword
	R/MRQ2	Initialize latches
	NDO⟹K8X1	Byte presence indicator bits of K-register are initialized
TMS140	MUX0 MLX0	M-register latches are cleared
	NSTIOPH⟹S/MRQ	Access odd word of command doubleword
	NSTIOPH⟹S/MRQ1	Designates that MRQ was raised
TMA160	S/FM33B	Second rank phase latch set
FM33B	TMRQB⇏S/TM	TM delay line is started for FM33B
TMB000	R/FM33A	First rank phase latch cleared
	MRQ1⟹R/MPR	Memory parity release latch initialized
TMB160	R/TRA1	Latch initialized
TMB240	S/FM34A	Next phase of chaining sequence

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM34A	TMRQASHARE NMRQ MPR ⇏ S/TM AR ⇏ R/MRQ POK + MPE ⇏ S/MPR MPESHARE PE ⇏ S/MPE	TM delay line starts for FM34A. (See FM32A for details)
TMA000	R/NFLCLEAR, R/ENDDIO, R/MRQ1	Latches initialized
TMA080	DI ⟹ S/MAYREAD	Enables buffer read cycles
	DI ⟹ NK81SET	Presence bits of K-register and latch KFILL1 initialized
	MPE ⟹ S/STIOPME	IOP memory error status bit set if a memory parity error was detected
TMA160	S/FM34B	Second rank phase latch set
	R/MPE	Latch cleared
	STIOPME + STIOPH + END3 + DI STZBCI EMPTY ED NJFILL NFI13A ⟹ S/NFLCLEAR	Inhibits transfer of new flags to FL-register
CNTCK	TMATRIPB NSTIOPH NMPE ⟹ BCXM	New byte count transferred to BC counter if no errors are detected
TMA240	R/KFILL2, R/MEMCYC	Clears latches
FM34B	TMRQB ⇏ S/TM	TM delay line is started for FM34A
TMB000	R/FM34A	Clears first rank phase latch
	R/ZBC1, R/ZBC2	Latches initialized
TMB040	FLCLEAR ⟹ FLX0	Old flags are cleared from FL-register
TMB160	FLCLEAR ⟹ FLXM	New flags are transferred from M-register to FL-register
TMB240	NFLCLEAR ⟹ S/FM41A	Conditions that prevented flags from being cleared (NFLCLEAR) also cause SIOP to advance to termination phase (FM41)
	FM41A ⇏ TMRQASHARE	Enables TM delay line to start. (Applies when bus-sharing option is installed)

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM34B		
TMB240 (Cont.)	FLCLEAR DI ⟹ S/FM24A, S/SX0/1	If termination is not required at completion of chaining sequence, SIOP returns to FM24, during which time more data will be written into core memory. Signal SX0/1 enables clearing of S-register
	FM24A ⇏ MEMRQ	Request made for memory bus. (Applies when bus-sharing option is installed)
	DI STBCI NFAST ⟹ S/ENDDIO	SIOP will send a zero byte count interrupt to D/C via a terminal order during FM41
	FLCLEAR ⟹ R/MEMCYC/1	Designates that memory bus is not in use. (Applies when bus-sharing option is installed)
	STIOPME TMBTRIPB ⇏ S/STIOPH	IOP halt status bit set if memory error status bit is true
FM41A	TMRQASHARE NMRQ RS NRSA (NORD MPR ENDFI + ...) ⇏ S/TM	TM delay line starts when RS is received from D/C if conditions a thru d all exist:
	AR ⇏ R/MRQ	a. D/C has dropped RS from previous data request, causing NRSA to be true
	MPESHARE PE ⇏ S/MPE	b. Address release AR has been received from core memory (NMRQ)
	POK + MPE ⇏ S/MPR	c. Parity OK or parity error has been received from core memory (MPR)
		d. No more data is to be accepted from D/C during FM24 (ENDFI)
TMA000/1	IOX0	I/O-register latches are cleared
	MPE ⟹ S/STTME, S/DATAERR	Transmission memory error status bit STTME is set, and DATAERR (temporary storage for transmission or memory error) is set
TMA000	S/ES	End of service. Signal ES primes reset input of service connect flip-flop in D/C
	R/MRQ1	Latch initialized
TMA080	DATAERR FLHTE ⟹ S/STIOPH	IOP halt status bit is set if a transmission error has been detected and halt on transmission error flag is true

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-32. Data In Service Cycle Phase Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
FM41A (Cont.)		
TMA160	S/FM41B	Second rank phase latch set
	STZBCI ⟹ S/IOA0	Terminal order bits are set (see table 3-25):
	DI NFLDC ZBCI ⟹ S/IOA1	a. Bit IOA0 drives data line DA0 to request an interrupt from D/C
	FLCC ⟹ S/IOA2	b. Bit IOA1 drives data line DA1 to transmit count done to D/C
	STIOPH ⟹ S/IOA3	c. Bit IOA2 drives data line DA2 to indicate command chaining to D/C
		d. Bit IOA3 drives data line DA3 to transmit an IOP halt to D/C
	DI NFLDC ZBC1 NEMPTY ⟹ S/STIL	Incorrect length status bit is set if buffer is empty, zero byte count has been reached, and data chaining flag is false
FM41B	TMRQB ⟹/ S/TM	TM delay line is started for FM41B
TMB000	R/RM41A	Second rank phase latch is cleared
TMB040	CHAIN NSTIOPH ⟹ FLX0	Old flags are cleared from FL-register
TMB120	S/RSA	D/C reads terminal order and disconnects. Signal RSA permits service connect flip-flop in D/C to reset, thus electrically disconnecting D/C from interface lines
TMB160	CHAIN NSTRIOPH ⟹ FLXM	New flags are transferred from M-register to FL-register if data chaining has occurred
TMB240	CHAIN ⟹ R/MEMCYC/1	Designates that memory bus is not in use. (Applies when bus-sharing option is installed)
	S/FI01A	First of preliminary phase latches initialized

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹/ indicates a function not controlled by a clock pulse

Table 3-33. WRITE Phase Sequence

Phase/Clock	Logical Operation	Comments
WRITE	(SWRITE + SREAD) CYCLE ⇏ S/TR	Either a request to write (SWRITE) into buffer or a request to read (SREAD) from buffer starts TR delay line
NTRS020	R/RJA8, R/RJB8, R/RJC8, R/RJD8	Latches initialized
TRS030	R/DREAD	Latch initialized
	SWRITE N(DI SREAD) ⇒ S/WRITE	Buffer write cycle is initiated
NTRS040	R/CYCLE	Cuts off input to delay line so that pulse traveling down line will to 40 ns in duration
	LRX0	LR-register latches cleared
TRS060	LRXIA	Enables address selection (LR) register to assume state of input address (IA) register
	DI ⇒ R/JFILL	Initializes latch JFILL
TRS120	C/IA, C/RK	Clock signal provided for input address register and locations filled (RK) register
	S/RJA8, S/RJB8, S/RJC8, or S/RJD8	Appropriate byte presence bit of J-register is reset as data is transferred to R-register
WCLOCK	Write into HSM (R-register)	Data is transferred from J-register to location of R-register specified by LR-register, which is controlled by input address register
TRS170	R/SWRITE, R/SREAD	Request to write into buffer and request to read from the buffer latches are cleared
	DO ⇒ R/JFILL	Initializes latch JFILL
TRS220	RJi8 ⇒ R/Ji8 (i = A through D)	Appropriate byte presence bit of J-register is transferred from that byte position to R-register
TRS250	S/CYCLE	Enables TR delay line
	CYCLE ⇏ R/WRITE	Clears WRITE latch
	DO ZBC0T15 NFLDC FM41A ⇒ S/ENDDIO	If data out, last data has been written into buffer
	DI ED ⇒ S/END1	If data in, ED was set and last data has been written into buffer

Notes:

1. Skips of phases are possible
2. The symbol ⇒ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

Table 3-34. READ Phases Sequence

Phase/Clock	Logical Operation	Comments
READ	(SWRITE + SREAD) CYCLE ⇏ S/TR	Either a request to write (SWRITE) into buffer or a request to read (SREAD) from buffer starts TR delay line
NTRS020	R/RJA8, R/RJB8, R/RJC8, R/RJD8	Latches initialized
TRS030	R/DREAD	Latch initialized
	SREAD N(DO SWRITE) ⇒ S/READ	Buffer read cycle is initiated
NTRS040	R/CYCLE	Cuts off input to delay line so that pulse traveling down line will be 40 ns in duration
	LRX0	LR latches are cleared
TRS060	LRXOA	Enables LR-register to assume state of output address (OA) register
	R/NANOTHER	Latch ANOTHER is initialized
	NSPACE ⇒ R/MAYWRITE	Control of BA and BC counters is inhibited during data out operations if required space is not available in buffer for new data
TRS120	C/OA, C/RK	Clock signal provided for output address register and locations filled (RK) register
	S/KAA, S/KAB, S/KAC, or S/KAD	Least significant byte of K-register into which data will be transferred from buffer
TRS170	R/SREAD, R/SWRITE	Request to read from buffer and request to write into buffer latches are cleared
	S/DREAD	Implies later part of read cycle has been reached
TRS220	S/NANOTHER	If current buffer location contains no more valid data, signal ANOTHER will be set false, indicating that data will not come from same buffer location during next read cycle
RCLOCK	Transfer R to K	Data is transferred from location of R-register specified by LR-register (controlled by output address register) to K-register

Notes:

1. Skips of phases are possible
2. The symbol ⇒ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

(Continued)

Table 3-34. READ Phases Sequence (Cont.)

Phase/Clock	Logical Operation	Comments
TRS250	NANOTHER SPACE ⟹ S/MAYWRITE, R/HFM21O22A	During data out service, if specified buffer locations are empty and next read cycle will not read same location, FA and and BC counters will be enabled by signal MAYWRITE. Signal NHFM21O22A implies core memory accesses will not inhibited by lack of space in buffer
TRS290	S/CYCLE	Enables TR delay line
	CYCLE ⟹̸ R/READ	Clear latch READ
	S/KFILL1	K-register is full and TI delay line may start for FI12 (data out), or TM delay line may start for FM24 (data in)
	NANOTHER DO EMPTY ENDDIO ⟹ S/KFILL2	Signifies during data out that all valid data written into buffer has been emptied, zero byte count has been reached with data chaining not specified, and TI delay line may start for FI12A
	NANOTHER DI EMPTY END1 ⟹ S/END2	Designates during data in that last data was written into and read from buffer
	NANOTHER ⟹ NRRM	Clears R-register modification logic
	ANOTHER ⟹ C/OA, C/RK, RM1SET	Clock signal provided for OA- and RK-registers, and R-register modification logic enabled

Notes:

1. Skips of phases are possible
2. The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⟹̸ indicates a function not controlled by a clock pulse

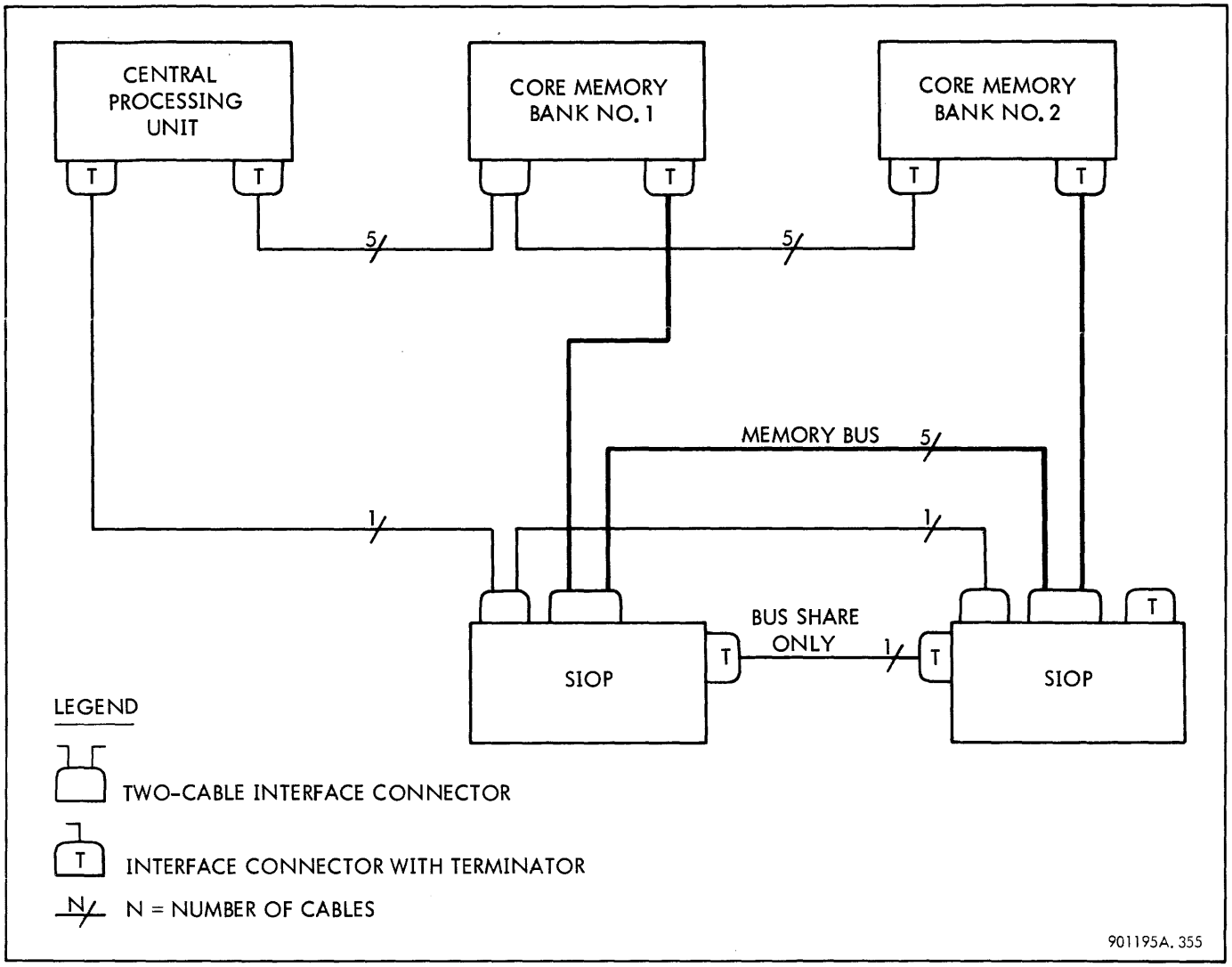


Figure 3-53. Interconnection of Bus-Sharing SIOPs

3-74 SIOP-Bus-Share Interface Signals

A brief description of the bus-sharing interface signals follows (see figure 3-54 and table 3-35):

a. Memory bus request MEMRQ. This line is driven by the low priority (B) SIOP only. The B SIOP drives this line to request the use of the memory bus during those SIOP operations in which SIOP-core memory data exchanges are required. These operations include all CPU requests for service that are normally executed by the SIOP and all D/C service sequences except order in.

This line is dropped by the B SIOP after the high priority (A) SIOP has driven the MEMCYCB line. The A SIOP takes action on this request as follows: if the A SIOP is not currently using the memory bus and has not requested use of the memory bus, the A SIOP grants use of the bus to the B SIOP by raising lines MEMCYCB and MEMCYCB/1.

b. Memory interface clear MEMRESET. This line is driven by the A SIOP as the initial response to a memory bus request by either the A SIOP or the B SIOP. In response to this signal, both SIOPs clear the SIOP-core memory interface data lines, address lines, and write byte lines.

c. Memory cycle request enable MEMCYCB. This line is driven by the A SIOP in response to a memory bus request by the B SIOP via the MEMRQ line. The A SIOP drops this line in response to a false signal received from the B SIOP on the NMEMCYCX0 line. When MEMCYCB line goes true, the B SIOP can make core memory accesses.

d. Memory cycle request disable NMEMCYCX0. This line is normally held true by the B SIOP. After the A SIOP grants use of the core memory bus to the B SIOP by raising the MEMCYCB line and the last memory request of the permitted sequence of requests has been made, the B SIOP causes this line (NMEMCYCX0) to go

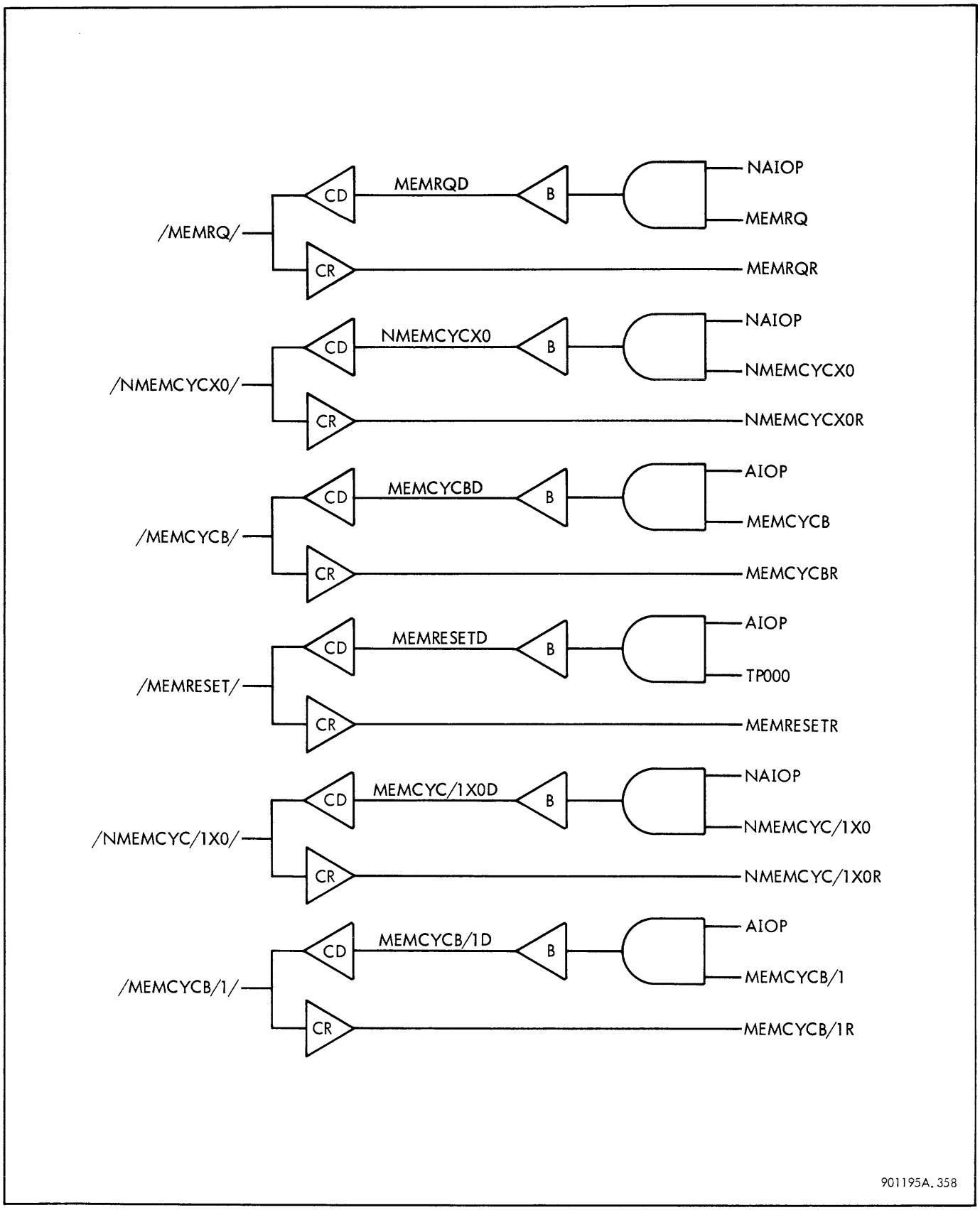


Figure 3-54. Bus Share Interface

Table 3-35. Bus-Sharing Interface Signals

SIGNAL	DESIGNATOR	NO. OF LINES	DIRECTION OF SIGNAL FLOW	
			A SIOP to B SIOP	B SIOP to A SIOP
Memory bus request	MEMRQ	1		X
Memory interface clear	MEMRSET	1	X	X
Memory cycle request enable	MEMCYCB	1	X	
Memory cycle request disable	NMEMCYCX0	1		X
Memory bus request disable	MEMCYCB/1	1	X	
Memory bus request enable	NMEMCYC/1X0	1		X

false. The permitted sequences of requests are as follows:

1. During normally executed CPU service requests. The B SIOP reads core memory location X'20' if the request is not an AIO and writes in locations X'20' and X'21' before relinquishing the core memory bus.

2. During order out and data chaining operations. The B SIOP responds to the transfer-in-channel order (if requested) and accesses the command doubleword before relinquishing the core memory bus.

3. During data out and data in operations. The B SIOP accesses one core memory location and then relinquishes the core memory bus.

e. Memory bus request disable MEMCYCB/1. This line is driven true by the A SIOP at the same time that the MEMCYCB line is driven true. The A SIOP drops this line in response to a false signal received from the B SIOP on the NMEMCYC/1X0 line. While this line is true, the A SIOP can not respond to requests for the memory bus from either the A or the B SIOP.

f. Memory bus request enable NMEMCYC/1X0. This line is normally held true by the B SIOP. After the A SIOP has granted use of the core memory bus to the B SIOP by raising the MEMCYCB line, the B SIOP causes this line to go false after all core memory interface operations of the last request of a permitted sequence have been completed.

3-75 Priority Determination Logic

The priority determination logic consists of the TP delay line (figure 3-55) and memory bus priority latches PRQA (high priority) and PRQB (low priority). Memory bus requests are made by either of the bus-sharing SIOPs by raising signal MEMRQ during any of phases FM01A, FM21A, RM24A, or FM31A. Requests for the bus are directed to the priority determination logic (PRQA and PRQB) in the A SIOP and cause either PRQA or PRQB to be set. Since PRQA has priority with respect to PRQB, PRQB can not be set if PRQA is true.

3-76 Bus-Sharing Operation

The TP delay line starts in response to a true state of either PRQA or PRQB. A false state of NTP000, TPCYCLE, or NMEMCYC then inhibits the setting of PRQA or PRQB.

At TP000, the core memory interface registers (S, M, and MW) are cleared in both SIOPs by signal MEMRSET. At TP105, either latch MEMCYCA or MEMCYCB is set. This enables signal TMRQASHARE to be true in either the A or the B SIOP (NA SIOP). Signal TMRQASHARE then enables the TM delay line to start for the group of phases in which the memory request signal (MRQ) may be raised. Latch MEMCYCA or MEMCYCB is then cleared by signal NMEMCYCX0 after all permitted memory cycles have been made. Also at TP105, either latch MEMCYCA/1 or MEMCYCB/1 is set to inhibit further memory bus requests until all operations associated with the current memory bus request have been completed. These latches are cleared by signal NMEMCYC/1X0.

See table 3-36 for the logical operations performed when the TP delay line is started.

3-77 GLOSSARY

A glossary of SIOP signals appears in table 3-37, and a glossary of SIOP terms appears in table 3-38.

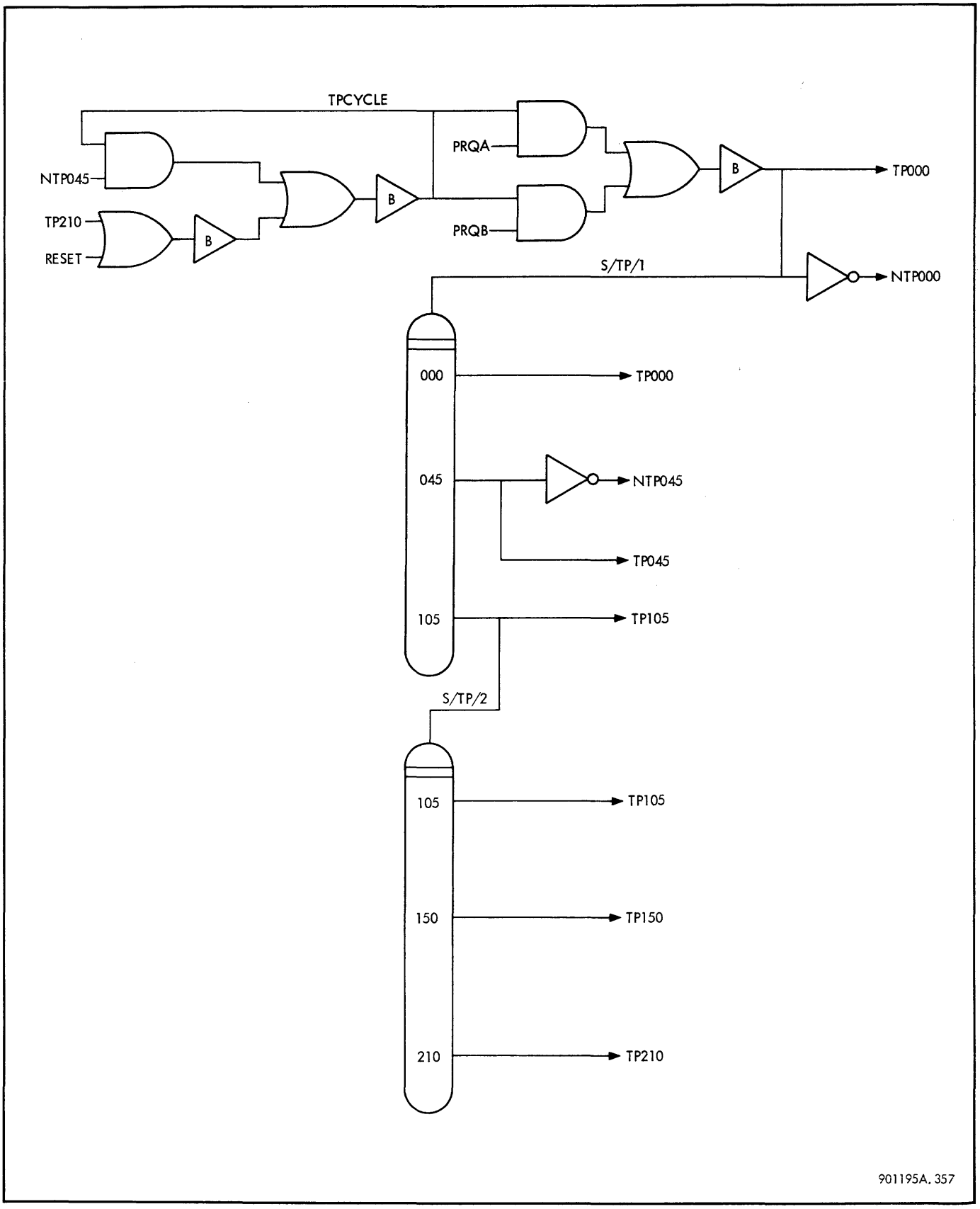


Figure 3-55. TP Delay Line Logic Diagram

Table 3-36. Bus-Sharing Sequences

Phase/Clock	Logical Operation	Comments
Bus-share	MEMRQ NMEMCYC NTP000 TPCYCLE ⟹ S/PRQA MEMRQ NMEMCYC NTP000 TPCYCLE NPRQA ⟹ S/PRQB (PRQA + PRQB) TPCYCLE ⇏ S/TP	If memory bus is not in use (NMEMCYC) when a memory bus-sharing request (MEMRQ) is received by bus-sharing logic, priority latch (PRQA or PRQB) is set and starts TP delay line
TP000	MEMRESET	S-, M-, and MW-registers are cleared
TP045	R/TPCYCLE	Input to delay line cut off. Establishes width (45 ns) of delay line pulse
TP105	PRQA ⟹ S/MEMCYCA, S/MEMCYCA/1 PRQB ⟹ S/MEMCYCB, S/MEMCYB/1	Enables the TM delay line in either A or B SIOP and inhibits further memory bus requests
TP150	R/PRQA, R/PRQB	Initializes priority latches
TP210	S/TPCYCLE	Enables TP delay line

Note:

The symbol ⟹ indicates a function controlled by a clock pulse. The symbol ⇏ indicates a function not controlled by a clock pulse

Table 3-37. Glossary of SIOP Signals

Signal	Definition
A0-A7	Flip-flop register (8-bit). Stores device controller address
AH	Cable receiver output for memory interface. Address here signal
AIOP	Switch contact. With bus-sharing option, designates high priority SIOP
NANOTHER	Buffered latch output. Not another read cycle from same buffer location
AR	Cable receiver output for memory interface. Address release strobe
AVO	Cable receiver output for I/O interface. No address recognition
AXIO	Logic signal for transfer of I/O register byte A into A-register
BA00-BA18	Flip-flop register (19-bit). Stores byte address
BAA0-BAA3	Logic signals for decoding two least significant bits of BA counter
BAXM	Logic signal for transferring contents of M-register into BA counter
BC00-BA15	Flip-flop register (16-bit). Stores byte count
BCC1	Logic signal for byte count equals one
NBCC3	Logic signal for byte count less than 3
BCC4T1	Logic signal for byte count less than 5
BCNT	Buffered latch. Controls updating of byte count and byte address counters
BCXM	Logic signal for transferring contents of M-register into BC counter
BKWD	Buffered latch. Stores read backwards order
CA00-CA15	Flip-flop register (16-bit). Stores command address
CACNTD	Logic signal for CA counter countdown enable
CACNTU	Logic signal for CA counter count up enable
CAXM	Logic signal for transferring contents of M-register to CA counter
CBC1-CBC4	Logic signals for update BC and BA counters by 1, 2, 3, or 4 counts
NCC1	Buffered latch output. Drives NCOND1 line to CPU
NCC2	Buffered latch output. Drives NCOND2 line to CPU
CHAIN	Buffered latch output. True during command or data chaining
CL1	Cable receiver output for CPU interface. 1.024 MHz clock from CPU
CM	Buffered latch output. Stores chaining modifier state
CNST1	Logic signal for control strobe is not for this SIOP
CNSTD	Cable driver input for CPU interface. Control strobe to next lower priority IOP
CNSTR	Cable receiver output for CPU interface. Control strobe from CPU or next higher priority IOP
CNTCK	Clock driver output, clock to BA, BC, and CA counters
NCOND1D	Cable driver input for CPU interface. Inverse of CC1 to CPU
NCOND2D	Cable driver input for CPU interface. Inverse of CC2 to CPU
CSRBM	Buffered latch output. Control set of RBM1 latch
CSRCM	Buffered latch output. Control set of RCM latch
CYCLE	Buffered latch output. TR delay line start enable

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
DA0R-DA7R	Cable receiver outputs for I/O interface. Byte A data lines
DAPR	Cable receiver output for I/O interface. Byte A parity bit
DATAERR	Buffered latch output. Temporarily stores transmission data error or transmission memory error status
DB0R-DB7R	Cable receiver output for I/O interface. Byte B data lines
DC0R-DC7R	Cable receiver output for I/O interface. Byte C data lines
DD0R-DD7R	Cable receiver output for I/O interface. Byte D data lines
DG	Cable receiver output for memory interface. Data gate strobe
DI	Logic signal. Decodes ORD and OUT latches to designate data in operation
DO	Logic signal. Decodes ORD and OUT latches to designate data out operation
DOR	Cable receiver output for I/O interface. Data order request from device controller
DPATH1	Buffered latch output. Stores PATH1 state
DPATH3	Buffered latch output. Stores PATH3 state
DREAD	Buffered latch output. Defines latter part of buffer read cycle
DX1	Logic signal. I/O interface data path is one byte wide
DX2, DX4	Cable receiver outputs. I/O interface data path is 2 or 4 bytes wide
ED	Cable driver input for I/O interface. End data state
EDI	Cable receiver output for I/O interface. End data state from device controller
EMPTY	Logic signal. Buffer is empty
END1	Buffered latch output. During data out, designates that ED was set during FI12; during data in, designates that ED was set and that this is last data written into buffer
END2	Buffered latch output. During data out, is set if END1 and HFM21O22A true; during data in, designates END1 true and buffer empty
END3	Buffered latch output. Enables termination if ED was set. Is set during FM21A, FM22A or FM24A, if END2 true
ENDDIO	Buffered latch output. During data out, designates that last data is in buffer; during data in, causes termination to send zero byte count interrupt
ENDFI	Buffered latch output. Designates last TI line phase
ES	Cable driver input for I/O interface. Designates end service state
FAIO	Cable driver input for I/O interface. Acknowledge interrupt function indicator
FAST	Cable receiver output for I/O interface. If true, SIOP does not terminate to send zero byte count interrupt during data chaining
FGAIO	Logic signal for AIO decoded from CPU interface function code lines
FGHIO	Logic signal for HIO decoded from CPU interface function code lines
FGSIO	Logic signal for SIO decoded from CPU interface function code lines
FGTDV	Logic signal for TDV decoded from CPU interface function code lines
FGTIO	Logic signal for TIO decoded from CPU interface function code lines

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
FHIO	Cable driver input for I/O interface. Halt I/O function indicator
FIXXA	Buffered latch output. Initial part of TI line phase FIXX. FI01 is initial phase of each phase sequence. FI11 starts with initial RS of each service sequence. FI12 designates I/O interface exchanges during data out only; FI13 designates I/O interface exchanges during data in only
FIXXB	Buffered latch output. Latter part of TI line phase FIXX
NFLCLEAR	Buffered latch output. Designates during phase FM34 of a chaining sequence that phase FM41 is to follow
FLCC	Buffered latch output. Stores command chain flag
FLDC	Buffered latch output. Stores data chain flag
FLHTE	Buffered latch output. Stores halt on transmission error flag
FLICE	Buffered latch output. Stores interrupt at channel end flag
FLIUE	Buffered latch output. Stores interrupt at unusual end flag
FLIZC	Buffered latch output. Stores interrupt at zero byte count flag
FLSIL	Buffered latch output. Stores suppress incorrect length flag
FLSKIP	Buffered latch output. Stores skip flag
FLXM	Logic signal. Transfers M-register byte A into flag register
FMXXA	Buffered latch output. Initial part of TM line phase XX. FM01 is second phase of each phase sequence. Phases FM11-FM14 operate during I/O instruction execution only. Phases FM21-FM22 operate during data out service only. Phase FM24 operates during data in service only. Phases FM31-FM34 operate during order out or data chaining only. Phase FM41 is the terminal order phase. Generally, during FMXX phases, memory interface exchanges are requested
FMXXB	Buffered latch output. Latter part of TM line phase FMXX
FNC0-FNC2	Cable receiver output for CPU interface. Function code lines
FNCAIO	Buffered latch output. Set during execution of AIO instruction
FNCASC	Buffered latch output. Cable driver input for I/O interface. Set during response to service call
FNCF	Buffered latch output. SIOP is responding to CPU service request
FNCG	Buffered latch output. Gates FGXXX signals at I/O interface
FNCSIO	Buffered latch output. Set during execution of SIO instruction
FR0-FR7	Cable receiver outputs for I/O interface. Function response lines
FS	Cable driver input for I/O interface (buffered latch output). Function strobe
FSIO	Cable driver input for I/O interface. Start I/O function indicator
FSL	Cable receiver output for I/O interface. Function strobe acknowledge
FTDV	Cable driver input for I/O interface. Test device function indicator
FTIO	Cable driver input for I/O interface. Test I/O function indicator
HA-HD	Buffered latch register (4-bit). Temporarily stores contents of WB-register during data out
HFM21O22A	Buffered latch output. During data out, inhibits core memory accesses if space not available in buffer

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
HWRITE	Buffered latch output. Inhibits writing in buffer if another read cycle required when buffer is filled
HX0	Logic signal. Clears H-register
HXWB	Logic signal. Transfers contents of WB-register into H-register
IA1-IA3	Flip-flop register (3-bit). Contains buffer input address
IACK	Logic signal, IA-register clock
IAOAX0	Logic signal. Clears IA- and OA-registers
IC	Cable receiver output for I/O interface. Interrupt call line
IOA0-IOA7	Buffered latch register (8-bit), cable driver input, I/O interface data register byte A
IOAEVEN	Parity generator output. Generates and checks odd parity for I/O register byte A
IOAP	Buffered latch output. Parity bit for I/O-register byte A
IOAXJD	Logic signal. Controls transfer of J-register byte D to I/O-register byte A
IOB0-IOB7	Buffered latch register (8-bit), cable driver input, I/O interface data register byte B
IOC0-IOC7	Buffered latch register (8-bit), cable driver input, I/O interface data register byte C
IOD0-IOD7	Buffered latch register (8-bit), cable driver input, I/O interface data register byte D
IOPA0R-IOPA2R	Cable receiver output for CPU interface. IOP address lines
IOR	Cable receiver output for I/O interface. Input/output request line
IOXD	Logic signal. Controls transfer of I/O interface data lines into I/O-register
IOXK	Logic signal. Controls transfer of K-register to I/O-register
IOXM	Logic signal. Controls transfer of M-register byte A into I/O-register byte A
IOXTO	Logic signal. Controls transfer of terminal order into I/O-register byte A
IOX0	Logic signal. Clears I/O-register
IR	Buffered latch output, cable driver input, CPU interface interrupt request to CPU
J8XH	Logic signal. Controls transfer of H-register into JA8, JB8, JC8, and JD8
JA0-JA7, JB0-JB7, JC0-JC7, JD0-JD7	Buffered latch register (32-bit), input to buffer during data operations, input to M-register during I/O instruction execution
JA8, JB8, JC8, JD8	Buffered latch outputs. Provide byte presence indicators to buffer during data out/in service
JAXFR	Logic signal. Controls transfer of function response to J-register byte A
JDXFR	Logic signal. Controls transfer of function response to J-register byte D
JFILL	Buffered latch output. J-register is filled and buffer write cycle may start
JX0	Logic signal. Clears J-register
JXBC	Logic signal. Controls transfer of BC counter to J-register bytes C and D during I/O instructions
JXCA	Logic signal. Controls transfer of CA counter to J-register bytes A and B during I/O instructions
JAXIO, JXIO	Logic signals. Control transfer of I/O-register to J-register

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
JXM	Logic signal. Controls transfer from M-register to J-register during data out operations
JXST	Logic signal. Controls transfer from status register to J-register during I/O instructions
NK81SET	Logic signal. Initializes NKA8, NKB8, NKC8, NKD8, and KFILL1 for data in operations
NK82SET	Logic signal. Initializes NKA8, NKB8, NKC8, NKD8, and KFILL1 for data out operations
K8X1	Logic signal. Latches presence bits of K-register
KA0-KA7, KB0-KB7, KC0-KC7, KD0-KD7	Buffered latch register (32-bit). Intermediate register between buffer and I/O-register during data out operations or between buffer and M-register during data in operations
NKA8, NKB8, NKC8, NKD8	Buffered latch outputs. If true during a buffer read cycle, associated K-register byte does not contain valid data. Inverse provide byte presence indicators for K-register
KAA, KAB, KAC, KAD	Buffered latch outputs. Only one of these may be true during a buffer read cycle to designate least significant byte in K-register into which data will be transferred from buffer. A logical matrix of KA(X) and SA(X) signals designates path from buffer to K-register
KFILL1	Buffered latch output. K-register is filled. FI12A may start if data out; FM24A may start if data in
KFILL2	Buffered latch output. During data out operations, zero byte count was reached with no data chaining. All data written into buffer has been emptied from buffer and FI12A may start
LAST	Switch contact. If true, this is last IOP in priority chain attached to CPU
LR1, LR2, LR3	Buffered latch register (3-bit). Selects buffer address
MA0-MA7, MB0-MB7, MC0-MC7, MD0-MD7	Buffered latch register (32-bit), cable driver input. Memory interface register bytes A, B, C, and D
MA0R-MA7R, MB0R-MB7R, MC0R-MC7R, MD0R-MD7R	Cable receiver outputs for memory interface. Data lines
MATCH	Logic signal. If contents of IOA-register equal contents of A-register, this signal is true
MAYREAD	Buffered latch output. Enables buffer read cycle during data in operations
MAYWRITE	Buffered latch output. Enables setting of BCNT during data out operations
MAYWRITEX0	Logic signal. Clears MAYWRITE latch
ME	Logic signal. CPU service request accepted by SIOP
NMEMCYC	Logic signal. In bus-sharing option, designates that memory bus is not in use. Enables S/PRQA or S/PRQB
NMEMCYC/1X0	Logic signal. In bus-sharing option, clears MEMCYCA/1 and MEMCYCB/1 latches
NMEMCYC/1X0D	Cable driver intput for bus-sharing interface. NMEMCYC/1X0 generated by NAIOP
MEMCYCA	Buffered latch output. In bus-sharing option enables AIOP to start FMXX phases
MEMCYCB	Buffered latch output. In bus-sharing option enables NAIOP to start FMXX phases
MEMCYCBD	Cable driver input for bus-sharing interface. MEMCYCB signal to NAIOP from AIOP
MEMCYCBR	Cable receiver output for bus-sharing interface. MEMCYCB signal received by NAIOP
MEMCYCA/1	Buffered latch output. In bus-sharing options, AIOP has not completed memory bus operations and inhibits new requests for bus

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
MEMCYCB/1	Buffered latch output. In bus-sharing option, NAIOP has not completed memory bus operations and inhibits new requests for bus
MEMCYCB/1D	Cable driver input for bus-sharing interface. MEMCYCB/1 signal to NAIOP from AIOP
MEMCYCB/1R	Cable receiver output for bus-sharing interface. MEMCYCB/1 signal received by NAIOP
NMEMCYCX0	Logic signal. In bus-sharing option, clears MEMCYCA and MEMCYCB latches
NMEMCYCX0D	Cable driver input for bus-sharing interface. NMEMCYCX0 signal from NAIOP to AIOP
MEMRESETD	Cable driver input for bus-sharing interface. Signal generated by AIOP to clear memory interface M-register, S-register, and MW-register in AIOP and NAIOP
MEMRESETR	Cable receiver output for bus-sharing interface. MEMRESET signal received by AIOP and NAIOP
MEMRQ	Logic signal. In bus-sharing option request for memory bus
MEMRQD	Cable driver input for bus-sharing interface. MEMRQ from NAIOP to AIOP
MEMRQR	Cable receiver output for bus-sharing interface. MEMRQ from NAIOP received by AIOP
MLX0	Logic signal. Clears M-register bytes C and D
MLXJ	Logic signal. Controls transfer of J-register bytes C and D to M-register bytes C and D
MNUM0-MNUM2	Switch contacts. Designate SIOP address
MODE0-MODE6	Logic signals. Control transfer from buffer to K-register
MPE	Buffered latch output. Set by PE strobe
MPEX0	Logic signal. Clears MPE latch
MPESHARE	Logic signal. In bus-sharing option, controls data transfer from memory interface receivers to M-register
MPR	Buffered latch output. Signifies memory parity release
MPRX0	Logic signal. Clears MPR latch
MRQ	Buffered latch output, cable driver input. Memory cycle request signal
MRQ1	Buffered latch output, MRQ was raised in FM01A, FM12A, FM14A, FM24A, FM31A, or FM33A or BCNT was true during FM21A (BA and BC updated)
MRQ1X0	Logic signal. Clears MRQ1 latch
MRQ2	Buffered latch output. MRQ was raised in FM31A or BCNT was true in FM22A (BA and BC updated)
MRQ2X0	Logic signal. Clears MRQ2 latch
MUX0	Logic signal. Clears M-register bytes A and B
MUXJ0	Logic signal. Controls transfer of JA and JB to MA and MB
MWA-MWD	Buffered latch outputs, cable driver inputs. Memory interface byte presence indicators
MXKB	Logic signal. Transfers contents of K-register to M-register during read backwards
MXKF	Logic signal. Transfers contents of K-register to M-register during read forward
MXM	Logic signal. Transfers data from memory data receivers to M-register when DG true
MYAIO	Logic signal. FGAIO signal is true and this SIOP has a pending interrupt
MYNUM	Inverting switch comparator output. If true, IOPA0-2 and MNUM0-2 signals compare

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
MYSIO	Logic signal. An SIO operation is directed to this SIOP
MYTEST	Logic signal. A TIO or TDV operation is directed to this SIOP
OA1-OA3	Flip-flop register (3-bit), buffer output address register
OACK	Logic signal. Clock for OA1-OA3
OI	Logic signal. Decodes ORD and OUT latches to designate order in operation
OO	Logic signal. Decodes ORD and OUT latches to designate order out operation
ORD	Buffered latch output. During I/O instructions, SIOP will write into one memory location X'21'. During service calls, stores state of DOR line
OUT	Buffered latch output. During service calls, stores state of IOR line
PATH0-PATH6	Logic signals. Control path from buffer to K-register
(PC DX1)	Logic signal. If DX1 is true, IOP will check parity at I/O-register byte A during data in
PE	Cable receiver output for memory interface. Parity error strobe
POK	Cable receiver output for memory interface. Parity OK strobe
PR1, PR2	Buffered latch outputs. When PR1 and NPR2 are true, proceed line (PRD) is driven
PRD	Cable driver input for CPU interface. Proceed line
PRQA	Buffered latch output. In bus-sharing option, receives memory bus request from AIOP
PRQB	Buffered latch output. In bus-sharing option, receives memory bus request from NAIOP
PRQX0	Logic signal. Clears PRQA and PRQB latches
C/RA, C/RB, C/RC, C/RD	High speed memory write clock signals, buffer bytes A, B, C, D
L/RA/1-L/RA/3, L/RB/1-L/RB/3, L/RC/1-L/RC/3, L/RD/1-L/RD/3	High speed memory address lines, buffer bytes A, B, C, D
W/RA0-W/RA8, W/RB0-W/RB8, W/RC0-W/RC8, W/RD0-W/RD8	High speed memory data input lines, buffer bytes A, B, C, D
RA0-RA8, RB0-RB8, RC0-RC8, RD0-RD8	High speed memory output lines, buffer bytes A, B, C, D. RA8, RB8, RC8, RD8 are buffer byte presence indicators
RAM	Buffered latch output. Modifies state of buffer bit RA8
RAXJA, RAXJB, RAXJC, RAXJD	Logic signals. Only one may be true at one time. Control transfer of J-register to buffer byte A
RBM1, RBM2	Buffered latch outputs. Modify state of buffer bit RB8
RBXJB, RBXJD	Logic signals. Only one may be true at one time. Control transfer of J-register to buffer byte B
RCLOCK	Buffered delay line sensor output for TR delay line. Buffer read clock
RCM	Buffered latch output. Modifies state of buffer bit RC8
READ	Buffered latch output. Designates that current buffer cycle is a read cycle
RESET	Logic signal. True if either RIO or TESTRIO signal is true. Initializes certain register in the SIOP

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
RIO	Cable receiver output for CPU interface. Reset I/O signal
RJA8, RJB8, RJC8, RJD8	Buffered latch outputs. During data out, if NDX4, resets appropriate Ji8 bit as data is transferred to HSM. Also is used to clear JFILL
RJSET	Logic signal. Controls setting of RJ(X)8
RK0-RK3	Flip-flop register (4-bit). Designates number of buffer locations containing data
RKCK	Logic signal, clock for RK0-RK3
RKD	Logic signal. Controls down counting of RK0-RK3 during WRITE phase
RKU	Logic signal. Controls up counting of RK0-RK3 during READ phase
RM1SET	Logic signal. Controls setting of RAM, RBM1, RBM2, and RCM during data out or during data in if byte count greater than 3
RM2SET	Logic signal. Controls setting of RAM, RBM1, RBM2, and RCM during data in if byte count is less than 4
NRRM	Logic signal. Clears RAM RBM1, RBM2, and RCM
RS	Cable receiver output for I/O interface. Request strobe
RSA	Cable driver input for I/O interface. Request strobe acknowledge
RSTRB	Clock driver output, read strobe. During buffer read cycle, strobes buffer output into K-register
S15-S31	Buffered latch register (17-bit), cable driver input for memory interface, memory address register
SAA, SAB, SAC, SAD	Logic signals. Only one may be true during a buffer read cycle to designate location of least significant data byte in buffer from which data will be transferred. A logical matrix of KA(X) and SA(X) signals designates path from buffer to K-register
SC	Cable receiver output for I/O interface. Service call
NSERVICE1	Buffered latch output. When true, enables TIO or TDV to be executed normally
SERVICE2	Buffered latch output. Set when ED RSA true to enable TIO or TDV to be executed normally if NSERVICE1 true
SHARE	Switch contact. If true, designates that bus-sharing option is installed
SPACE	Logic signal. During data out operations at least two buffer addresses are empty if NDX1, and at least four buffer addresses are empty if DX1
SRDWRTRESET	Logic signal. Resets SREAD and SWRITE during order out
SREAD	Buffered latch output. If true during data out or data in, designates a buffer read cycle request
ST2-ST7	Logic signals. Enable status bits to J-register as follows:

Signal	Status Bit Enabled If NAIO	Status Bit Enabled If AIO
ST2	STTME	STZBCI
ST3	STMAE	STCEI
ST4	STIOPME	STUEI
ST5	STIOPCE	0
ST6	STIOPH	0
ST7	STIOPB1	0

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
NSTAX0	Logic signal. Clears interrupt status bits
STIL	Buffered latch output. Incorrect length status bit
STTDE	Buffered latch output. Transmission data error status bit
STTME	Buffered latch output. Transmission memory error status bit
STMAE	Buffered latch output. Memory address error status bit
STIOPME	Buffered latch output. IOP memory error status bit
STIOPCE	Buffered latch output. IOP control error status bit
STIOPH	Buffered latch output. IOP halt status bit
STIOPB1	Buffered latch output. IOP busy status bit
STIOPB2	Buffered latch output. When true, order in was received and command chain flag was false
NSTX0	Logic signal. Clears certain status bits, flag register, and chaining modifier flag
STZBCI	Buffered latch output. Zero byte count interrupt status bit
STCEI	Buffered latch output. Channel end interrupt status bit
STUEI	Buffered latch output. Unusual end interrupt status bit
NSTI	Logic signal. True if all interrupt status bits are false
SUCCESS	Logic signal. True if FNCSIO NCC1 NCC2 is true. Designates a successful SIO
SWRITE	Buffered latch output. If true during data out or data in, a buffer write cycle request
SX0/1	Buffered latch output. Enables clearing of S-register
SXBA	Logic signal. Controls transfer of byte address to S-register
SXCA	Logic signal. Controls transfer of command address to S-register
TESTRIO	Cable receiver output. Provides I/O reset signal from JX58 when connected
S/TI	Starts TI delay line
TICYCLE	Buffered latch output. Controls pulse width on TI delay line
TIL000-TIL560	Delay line tap outputs for TI delay line
TIS000-TIS560	Delay line sensor outputs for TI delay line
TIRQ	Logic signal. Requests TI delay line to start during FIXXA
S/TM	Starts TM delay line
TMCYCLE	Buffered latch output. Controls width of pulse on TM delay line
TMRQA	Logic signal. Requests TM delay line to start during FMXXA phases
TMRQB	Logic signal. Requests TM delay line to start during FMXXB phases
TML000-TML300	Delay line tap outputs for TM delay line
TMS000-TMS280	Delay line sensor outputs for TM delay line
TMA000-TMA240	Clock pulses derived from TM delay line during FMXXA phases
TMB000-TMB240	Clock pulses derived from TM delay line during FMXXB phases
TMATRIPA	Buffered latch output. Gates TM line sensor outputs to form clock pulse TMA000 during FMXXA phases

(Continued)

Table 3-37. Glossary of SIOP Signals (Cont.)

Signal	Definition
TMATRIPB	Buffered latch output. Gates TM line sensor outputs to form clock pulses TMA080-TMA240 during FMXXA phases
TMBTRIPA	Buffered latch output. Gates TM line sensor outputs to form clock pulses TMB000-TMB040 during FMXXB phases
TMBTRIPB	Buffered latch output. Gates TM line sensor outputs to form clock pulses TMB080-TMB240 during FMXXB phases
TMRQASHARE	Logic signal. Enables TM line to start with bus-sharing option
TO0	Logic signal. Provides input to IOA0 during terminal order phase
TO1	Logic signal. Provides input to IOA1 during terminal order phase
S/TP	Starts TP delay line. TP delay line resolves conflict between simultaneous requests during memory bus sharing
TPCYCLE	Buffered latch output. Controls width of pulse on TP delay line
TP000-TP210	Delay line tap outputs for TP delay line
S/TR	Designator shows input to TR delay line. TR line controls buffer write and read cycles
TRL000-TRL300	Delay line tap outputs for TR delay line
TRS000-TRS290	Delay line sensor outputs for TR delay line
TRA	Logic signal. Decodes transfer order during chaining
TRA1	Buffered latch output. Transfer order was received during chaining
USEFULHIO	Logic signal. Designates during an HIO service that address of last device started by SIO (and stored in A-register) agrees with device addressed during HIO
WBA, WBB, WBC, WBD	Buffered latch outputs. Provide input to the MW-register during core memory write operations and input to H-register during data out service
WBSET	Logic signal. Enables setting of WB-register
WBX0	Logic signal. Clears WB-register
WCLOCK	Delay line sensor output. Output from TR line tap TRL150. Provides buffer clock signal during buffer write cycles
WP	Logic signal. Designates core memory partial write operation
WRITE	Buffered latch output. Designates that current buffer cycle is a write cycle
ZBC0T13	Logic signal. Byte count register bits BC00-BC13 are false (byte count less than 4)
ZBC0T15	Logic signal. Byte count register bits BC00-BC15 are false (byte count = 0)
ZBC1	Buffered latch output. ZBC0T15 was true during FM21 or FM24
ZBC2	Buffered latch output. ZBC0T15 was true during FM22

Table 3-38. Glossary of SIOP Terms

Term	Definition
AIOP	Higher priority SIOP of two sharing memory bus
Alignment	(With respect to data buffer operation.) Process of transferring data in a particular byte position of R-register to other than corresponding byte position of K-register. For example, transfer of RA and RB to KC and KD respectively
Buffer	(See data buffer entry)
Burst mode	Complete transmission of a record of any number of bytes without relinquishing and regaining access to SIOP-device controller interface, as opposed to exchange of a maximum of four bytes per service cycle with MIOP operation. SIOP does not raise end data line to device controller until entire byte count has been reduced to zero, or until an HIO or AIO causes service call to be interrupted
Bus share	An option that permits two SIOPs to share same core memory bus
Busy state	State entered upon acceptance of an SIO by D/C. (See normal and partial execution entries)
CC	Command chain flag
CEI	Channel end interrupt
Channel end	An indicator set by D/C during an order in service cycle that causes SIOP to (1) set CEI status bit if ICE flag is true and (2) signal interrupt in ensuing terminal order
CM	Chaining modifier. An indicator set by D/C during an order in service cycle to cause SIOP to modify a command chaining sequence
Command list	A sequence of steps generated by CPU pertaining to performance of an I/O operation
Commutation	(With respect to data buffer operation.) Process of transferring bytes of data from successive byte positions of one register to a particular byte position of another register. For example, when SIOP-device controller interface is one byte wide, data is transferred from byte positions JA, JB, JC, and JD of J-register to successive locations of byte position RA of R-register
Count done	A condition that exists when byte count decremented to zero. Signaled by SIOP to D/C when all data associated with current operation has been transferred
Data buffer	A group of registers and control logic that provides a constant rate of flow of data at D/C interface undisturbed by core memory accesses or realignment of data boundaries within SIOP
DC	Data chain flag
D/C	Device controller
Flag	An indicator that specifies to SIOP how to handle various SIOP functions
HTE	Halt on transmission error flag
ICE	Interrupt at channel end flag

(Continued)

Table 3-38. Glossary of SIOP Terms (Cont.)

Term	Definition
IL	Incorrect length
IUE	Interrupt at unusual end flag
IZC	Interrupt at zero byte count flag
IOPB	Designates IOP busy
IOPCE	Designates IOP control error
IOPH	Designates IOP halt
IOPME	Designates IOP memory error
MAE	Memory address error flip-flop
Normally executed instruction	An instruction whose execution involves accessing core memory. Therefore, delay line associated with memory interface is triggered
Operational status byte	Data byte received from a D/C during an order in service cycle
Order	An operational step executed by an I/O device
Partially executed instruction	An instruction whose execution occurs when SIOP is in busy state when certain instructions are received. Condition codes are sent to CPU but no core memory accesses are made. Delay line associated with memory interface is not started
Presence indicator	When true, indicates that associated byte position of a register contains valid data
S	Skip flag
Selector IOP	An input/output processor that performs bidirectional data transfers between core memory and high speed peripheral devices
SIL	Suppress incorrect length flag
TDE	Transmission data error
TME	Transmission memory error
T.O.	Terminal order
UEI	Unusual end interrupt
Valid data	Data associated with a particular I/O operation. Valid data not cleared from a register during a particular I/O operation is considered invalid data during next I/O operation. For example, if data in first two bytes of a word in core memory is not part of record currently being processed, but is read into SIOP M-register during data service, these two bytes are considered invalid data

SECTION IV
MAINTENANCE AND PARTS LIST

4-1 GENERAL

This section contains preventive maintenance procedures and a parts list for the Sigma 5 and Sigma 7 SIOP. Also included is a list of signals most often observed on an oscilloscope when troubleshooting the equipment.

4-2 PREVENTIVE MAINTENANCE

All the documents listed on the Assembly 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 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 should be replaced periodically.

4-5 SIOP TEST PROGRAMS

The diagnostic programs that exercise the units attached to the SIOP should be run while varying the voltage margins ±10%.

4-6 TROUBLESHOOTING SIGNAL SUMMARY

Table 4-1 provides the location of various signals that are most often observed on an oscilloscope when troubleshooting the SIOP.

4-7 PARTS LIST TABLES

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

a. Figure number that shows module location

b. Module name

c. Reference designator (slot and chassis letter of the module

d. Name of the company that manufactures the module

e. Assembly part number of each module

f. Quantity of each module required for each SIOP

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

Table 4-1. Troubleshooting Signal Summary

SERVICE REQUEST AND STROBES			
Signal	Location	Signal	Location
CNSTR	10F20	PRD	10F45
SC	21E40		
IC	14E40		
FS	11E35	FSL	14E36 (AVO 09E08)
RS	14E27	RSA	11E03
MRQ	13D01	AR	13D08

(Continued)

Table 4-1. Troubleshooting Signal Summary (Cont.)

SIOP STATE (PHASE)			
Signal	Location	Signal	Location
FI01A	13F34	ZBC0T15	16E06
FM14B	03B21	ZBC1	15C44
FM41B	04A39	ZBC2	15C45
ENDFI	09B14		
OUT	09F39		
FM01B	03B26		

FLAGS		STATUS		SYNC SIGNALS (D/C SERVICE)	
Signal	Location	Signal	Location	Signal	Location
DC	09D21	IL	04A26	FI11B OO	02A14
IZC	09D46	TDE	04A27	FI11B OI	02A15
CC	09D09	TME	04A46	FI11B DO	02A27
ICE	09D07	MAE	04A07	FI11B DI	02A28
HTE	09D02	IOPME	04A21		
IUE	09D01	IOPH	04A09		
SIL	09D39	IOPB1	06A24		
SKIP	09D42	IOPB2	06A36		
CM	16C47	ZBCI	04A02		
		CEI	04A01		
		UEI	04A14		

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
DT 14	HT 15	BT 22	FT 24	IT 26	LT 13	XT 10	BT 10	XT 10	AT 24	FT 37	IT 13	BT 16	FT 38	FT 38	FT 38	FT 38	AT 16	BT 10	LT 20	IT 16	AT 11	IT 25	LT 21		XT 10	LT 13	LT 13	FT 37	BT 18	BT 15	IT 18

CHASSIS A

AT 16	IT 25	IT 24	BT 22	IT 25	LT 21	FT 26	BT 16	IT 16	FT 39	FT 39	FT 27	FT 39	FT 26	FT 39	XT 10	LT 13	IT 16	IT 15	IT 11	IT 11	IT 11	BT 18	FT 37	XT 10	XT 10	IT 15	FT 37	IT 16	FT 37	LT 13	IT 25

CHASSIS B

FT 18	FT 18	LT 13	FT 27	XT 10	ZT 25	IT 15	IT 25	BT 25	XT 10	ZT 25	BT 10	BT 25	IT 16	IT 25	ZT 25	BT 10	BT 10	LT 21	BT 15	ZT 25	BT 18	IT 18	LT 13	XT 10	ZT 25	LT 29	DT 14	HT 15	BT 16	ZT 25	BT 22

CHASSIS C

BT 11	FT 18	FT 18	FT 18	FT 18	ZT 25	FT 38	FT 38	FT 38	FT 38	ZT 25	FT 38	AT 11	FT 38	AT 11	ZT 25	FT 38	AT 11	FT 38	AT 11	ZT 25	AT 12	FT 37	FT 37	LT 21	ZT 25	XT 10	DT 11	HT 15	BT 15	ZT 25	BT 22

CHASSIS D

GT 11	FT 18	FT 18	BT 11	FT 18	FT 18	FT 18	FT 18	FT 18	XT 10	FT 38	AT 11	FT 37	AT 11	FT 37	AT 11	BT 16	XT 10	AT 10	BT 18	BT 25	AT 12	LT 21	AT 11	LT 21	XT 10		FT 24	FT 37	LT 20	BT 10	LT 21

CHASSIS E

		BT 22	BT 16	XT 10	BT 11	BT 15	FT 18	FT 18	GT 11	BT 22	BT 10	FT 22	BT 24	IT 16	LT 65	IT 15	IT 25	BT 18	IT 16	IT 25	BT 16	AT 13	LT 64	LT 26	XT 10	IT 16	IT 25				

CHASSIS F

901195A.356

Figure 4-1. SIOP Module Location Chart

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

Fig. No.	Description	Reference Designator	Manufacturer	Part No.	Qty
4-1	Selector Input/Output Processor Models 8285 and 8485		SDS	117620	1
	. Cable Receiver AT10	14E		123018	1
	. Cable Driver-Receiver AT11	9E, 17E, 19E, 21E, 13D, 15D, 18D, 20D	SDS	123019	8
	. Cable Driver AT12	11D, 11E	SDS	124629	2
	. Cable Driver-Receiver AT13	10F	SDS	125260	1
	. Rejection Gate AT16	32B	SDS	126611	1
	. Clock Driver No. 2 AT24	23A	SDS	128168	1
	. Buffered AND/OR Gate BT10	25A, 15C, 16C, 21C, 2E, 21F	SDS	116056	6
	. BAND Gate BT11	32D, 29E, 27F	SDS	116029	3
	. Gated Buffer No. 1 BT15	2A, 13C, 3D, 26F	SDS	117389	4
	. Gated Buffer BT16	20A, 25B, 3C, 16E, 11F, 29F	SDS	125262	6
	. BAND Gate BT18	3A, 10B, 11C, 13E, 14F	SDS	126613	5
	. Fast Buffer BT22	30A, 29B, 1C, 1D, 22F, 30F	SDS	127393	6
	. Buffered AND/OR Gate BT24	19F	SDS	130967	1
	. BAND Gate BT25	24C, 12E, 20C	SDS	130947	3
	. Delay Line DT11	5D	SDS	126963	1
	. Delay Line DT14	32A, 5C	SDS	127319	2
	. Counter Flip-Flop FT18	31C, 32C, 28D-31D, 24E-28E, 30E, 31E, 24F, 25F	SDS	124634	15
	. Universal Flip-Flop FT22	20F	SDS	124713	1
	. Buffered Latch No. 1 FT24	29A, 5E	SDS	126745	2
	. Buffered Latch No. 3 FT26	19B, 26B	SDS	126856	2
	. Buffered Latch No. 2 FT27	21B, 29C	SDS	126986	2

(Continued)

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

Fig. & Index No.	Description	Reference Designator	Manufacturer	Part No.	Qty
4-1	Selector Input/Output Processor Models 8285 and 8485 (Cont.)				
	. Buffered Latch No. 2a FT37	4A, 22A, 3B, 5B, 9B, 9D, 10D, 4E, 18E, 20E	SDS	130942	10
	. Buffered Latch No. 3a FT38	16A-19A, 14D, 16D, 19D, 21D, 23D-26D, 22E	SDS	130952	13
	. Fast Access Memory 8 x 9 FT39	18B, 20B, 22B, 23B	SDS	131072	4
	. Gate Expander No. 1 GT11	32E, 23F	SDS	124715	2
	. Delay Line Sensors HT15	31A, 4C, 4D	SDS	127391	3
	. NAND Gate IT11	11B-13B	SDS	116994	3
	. Inverter Matrix IT13	21A	SDS	117000	1
	. Gated Inverter IT15	6B, 14B, 26C, 16F	SDS	117375	4
	. Gated Inverter IT16	4B, 15B, 24B, 19C, 6F, 13F, 18F,	SDS	125264	7
	. NAND Gate IT18	1A, 10C	SDS	126372	2
	. NAND/NOR Gate IT24	30B	SDS	128188	1
	. NAND Gate IT25	10A, 1B, 28B, 31B, 18C, 25C, 5F, 12F, 15F	SDS	128190	9
	. NAND Gate IT26	28A	SDS	128192	1
	. Buffer Inverter No. 1 LT13	5A, 6A, 27A, 2B, 16B, 9C, 30C	SDS	123016	7
	. Logic Element LT20	3E	SDS	124717	1
	. Logic Element LT21	9A, 27B, 14C, 8D, 1E, 8E, 10E	SDS	126615	7
	. Switch Comparator LT26	8F	SDS	126982	1
	. Clock Logic LT29	6C	SDS	127643	1
	. Logic Control LT64	9F	SDS	135496	1
	. Parity Generator LT65	17F	SDS	135574	1

(Continued)

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

Fig. & Index No.	Description	Reference Designator	Manufacturer	Part No.	Qty
4-1	Selector Input/Output Processor Models 8285 and 8485 (Cont.)				
	. Terminator Module XT10	7A, 24A, 26A, 7B, 8B, 17B, 8C, 23C, 28C, 6D, 7E, 15E, 23E, 7F, 28F	SDS	116257	15
	. Ribbon Cable Frame ZT25	2C, 7C, 12C, 17C, 22C, 27C	SDS	128060	6
	. Selector Input/Output Processor Bus-Sharing Assembly*		SDS	137304	1
	. Cable Driver-Receiver AT11	11A	SDS	123019	1
	. Rejection Gate AT16†	15A	SDS	126611	1
	. Buffered AND/OR Gate BT10	14A	SDS	116056	1
	. Gated Inverter IT16	12A	SDS	125264	1
	. Logic Element LT20	13A	SDS	124717	1

*Optional

†Used in first selector IOP only

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: _____

1495(8/67)

STAPLE

STAPLE

FOLD

FIRST CLASS
PERMIT NO. 1026
SANTA MONICA, CALIF.

BUSINESS REPLY MAIL

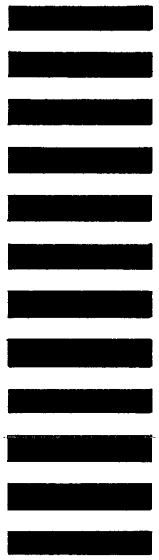
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY

SCIENTIFIC DATA SYSTEMS

701 So. Aviation Boulevard
El Segundo, California 90245

ATTN: TECHNICAL PUBLICATIONS DEPT.



CUT ALONG LINE

FOLD