PCP 126

GE-PAC* 30
CONTROL COMPUTER

SYSTEMS INTERFACE MANUAL

GENERAL ELECTRIC

*Registered Trademark of General Electric Company

GE-PAC 30
CONTROL COMPUTER

SYSTEMS INTERFACE MANUAL

GENERAL ELECTRIC

GE 29-003R01

TABLE OF CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

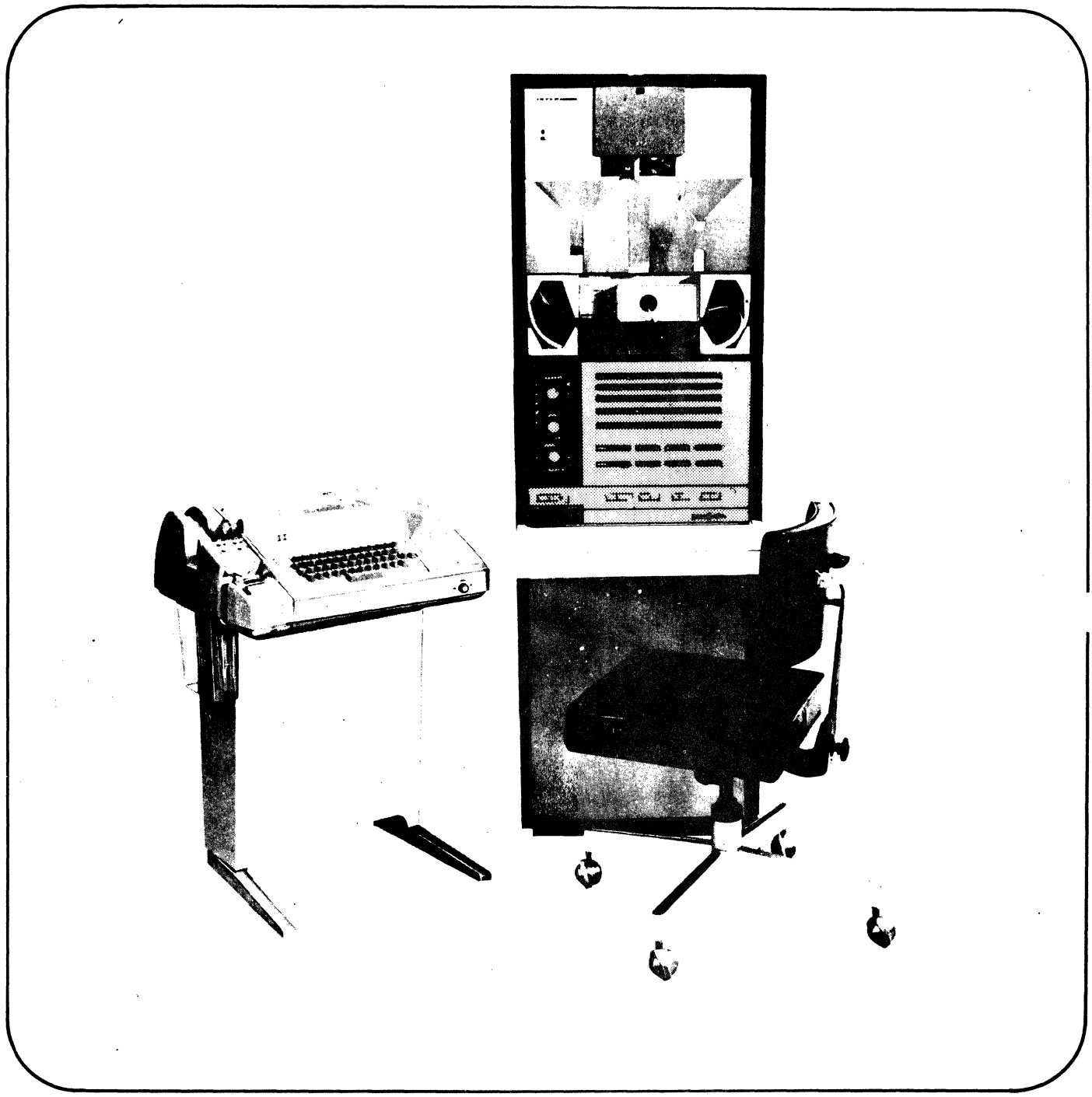


Figure 1-1. Typical GE-PAC 30 Digital System

CHAPTER 1

GENERAL DESCRIPTION

1.1 INTRODUCTION

Figure 1-1 shows a typical GE-PAC 30 Digital System. GE-PAC 30 computers are general purpose, low cost systems, versatile enough to perform a wide range of both industrial control and scientific computation. These third generation computers use integrated circuits for reliability, and feature a modular expansion system which permits an economical approach to the specific requirements of each installation. The Systems Interface provides several methods of communication between the GE-PAC 30 Processor and external devices or systems. The methods vary in speed, sophistication, and the amount of attention from the Processor required. Thus, the System Interface may be tailored to communicate efficiently with the types of peripheral devices presently used with a particular system, and later may be simply expanded in the field to meet changing I/O requirements.

1.2 SCOPE OF MANUAL

There are two primary purposes for this manual: to familiarize the reader with the GE-PAC 30 System Interface, and to provide the data required to effectively interface external equipment to GE-PAC 30 Digital Systems. A functional description of each I/O sub-system is provided later in this Chapter, followed by a physical description of the layout and interconnection of a typical system. Note that the I/O features are described more fully in separate maintenance manuals. Chapter 2 describes the coding and sequence of operation of all I/O instructions. Chapter 3 describes the considerations and specifications in designing device controllers for GE-PAC 30 equipment. Chapter 4 describes a General Purpose GE-PAC 30 to facilitate custom interface design.

1.3 I/O SYSTEM BLOCK DIAGRAM ANALYSIS

Figure 1-2 is a block diagram of a GE-PAC 30 Digital System emphasizing the Systems Interface capability. Note that there are three separate methods of communicating with peripheral devices or systems:

1. The Multiplexor Channel
2. A Selector Channel
3. A Direct Memory Access Channel (DMAC)

Each of the three methods communicates via a bus with device controllers. The device controllers provide data and control interface to the individual devices. The Systems Interface can communicate with up to 256 devices. The following paragraphs describe each of the interface functions.

1.3.1 Multiplexor Channel

Figure 1-3 is a block diagram of the Multiplexor Channel. The Multiplexor Channel is a byte oriented I/O system which communicates directly with up to 256 peripheral devices. The Multiplexor Bus consists of 27 lines; 8 data input, 8 data output, 8 control lines, 2 test lines, and an Initialize line. The two test input lines from the device controllers are Synchronization (SYN) and Attention (ATN). The final line is System Clear (SCLR) to all device controllers.

A typical sequence of operations over the Multiplexor Channel is:

1. The Processor addresses a device controller over the 8 System Data lines. The address appears on the bus to all device controllers.

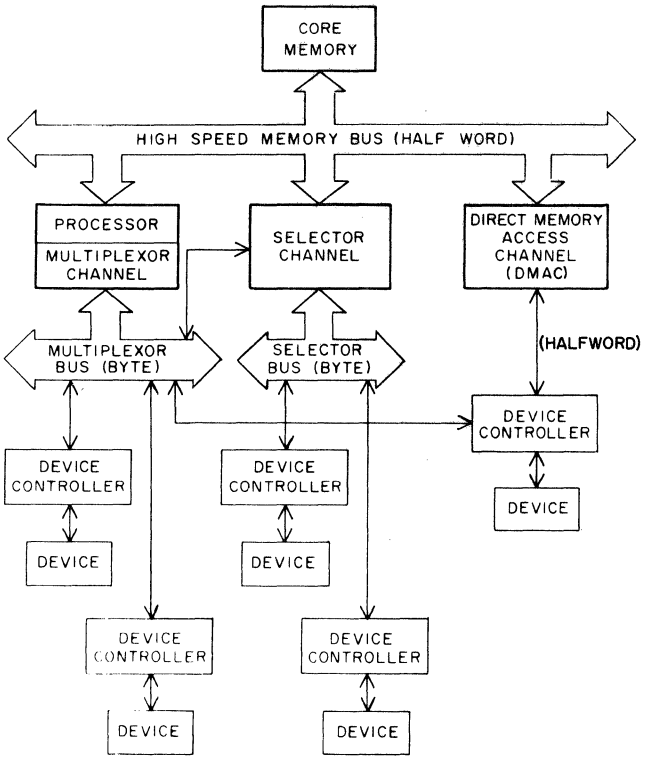


Figure 1-2. Systems Interface, Block Diagram

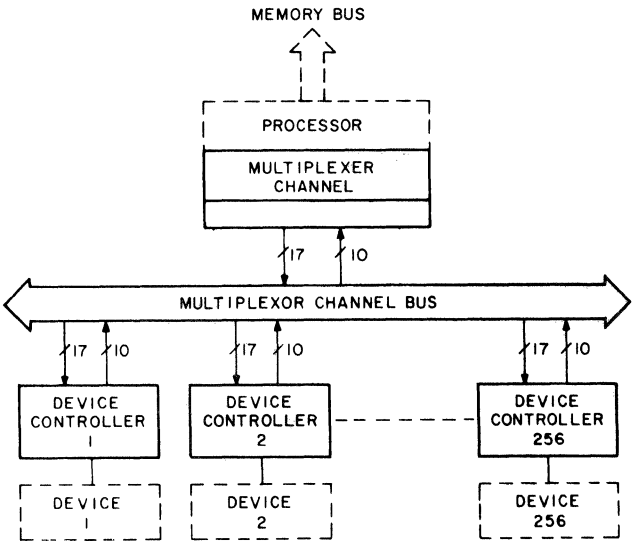


Figure 1-3. Multiplexor Channel, Block Diagram

2. The Processor then activates a System Control line which specifies that the System Data lines now provide an address (rather than data).

3. The device controllers use the System Control line to enable address decoders. Each device controller decodes its own address. Assuming that the System Data lines are providing the address of one of the device controllers tied to the Multiplexor Channel, the device controller decodes its address and responds by sending a SYN signal back to the Multiplexor Channel.

4. The Processor may now change the System Control and System Data lines. The device controller remains addressed until another device controller is addressed or until a System Clear (SCLR) signal is received.

5. The Processor next activates a System Control line which indicates whether this is an input or output operation.

6. If this is an output operation, the Processor provides the byte of data on the System Data lines. The device controller responds with a SYN signal when it has accepted the byte.

7. If this is an input operation, the device controller sends the byte from the device to the Processor via the System Data lines. A SYN signal is sent to indicate that the data is ready.

The sequence provided here is simplified. The entire sequence for each type of instruction is listed in Chapter 2. The final line to be introduced here is the Attention (ATN) line. The Attention line provides a means of interrupting the Processor. Each controller normally has an interrupt Queue flip-flop which may be set by conditions within either the device or the device controller. The output from the Queue flip-flop is sent to the Multiplexor Channel as ATN. Note that ATN may be initiated by any device controller,

whether it is addressed or not. The Processor initiates hardware interrupt resolution to determine which device controller caused the ATN signal. The interrupting device automatically returns its device number to the Processor. This interrupt feature is described in detail in Section 3.6.

Wiring information for the Multiplexor Channel is provided in Appendix 1 of this manual.

1.3.2 Selector Channel

The optional Selector Channel provides block data transfer between 1 of up to 25 I/O devices, and memory. Once initiated, the transfer is independent from the Processor The Processor specifies the device, the type of operation (Read Block or Write Block), the starting address in memory, and the final address in memory. The Selector Channel then completes the transfer without further direction by the Processor. Upon completion of the transfer, or termination of the transfer due to a fault, the Processor is notified via an interrupt.

Figure 1-4 is a block diagram of the Selector Channel. Address lines to, and data lines to and from, the High Speed Memory Bus are shown on the right side of Figure 1-4. The Memory Bus Control Logic (one of several arbitrary functional groupings used only for purposes of this block diagram description) gates an address to the Memory Bus, then gates data to or from the bus depending upon the type of transfer. The Selector Channel Data Register (DR) stores the 16-bit data word to/from memory. The transfer Control Logic gates the data between the Selector Bus (shown on the bottom of Figure 1-4) and the Data Register in 8-bit bytes. The address circuits are shown in the upper right area of Figure 1-4. The 16-bit Final Address Register (FR) is loaded in two 8-bit bytes from the Multiplexor Bus. The Address Register (AR) is loaded with the starting address in two 8-bit bytes. After each byte of data is transferred, the AR is incremented and its contents compared to the contents of FR. If the two are equal, the Block Transfer is complete, and a terminate signal is sent to the Transfer Control Logic. If the two are not equal, the next 16-bit transfer is initiated to/from the next sequential memory address. The Multiplexor Bus is shown on the left side of Figure 1-4. Note that the 8-bit Multiplexor Bus may be gated to any one of six places. The gates are functionally represented by a six position rotary switch. With the gating as shown by the switch position, and assuming the Transfer Control Logic is also as shown, the Multiplexor Bus is gated directly to the Selector Bus. This is the condition which exists when the Selector Channel has not been addressed. Thus, all devices on the Selector Channel may be used via the Multiplexor Channel if the Selector Channel is not in use. (Of course, the device must be capable of operating within the Multiplexor Channel timing constraints.) Four of the remaining five points that the Multiplexor Bus may be gated to, are the Upper and Lower halfs of FR and AR. The sixth point is designated Command and Sense Logic on Figure 1-4. Commands from the Processor are decoded in this block to produce control signals to both the Transfer Control Logic and the Multiplexor Input Control Logic. Status Bytes from the device are returned to the Processor via this block during Sense Status instructions.

The following is a typical sequence of operation for a Selector Channel I/O operation. Figure 1-5 is a flow chart of Selector Channel operation. Circled numbers on Figure 1-5 refer to steps in the following sequence:

1. The device controller is addressed from the Multiplexor Channel and the appropriate command sent to it (for example, Read Tape Forward).
2. The Selector Channel AR and FR are loaded via four byte transfers from the Multiplexor Channel. The initial contents of the AR (known as the starting address - SA) must be even - i.e. a half word boundary. The FR may be odd or even - i.e. byte or half word boundary.

NOTE

Steps 1 and 2 may be reversed.

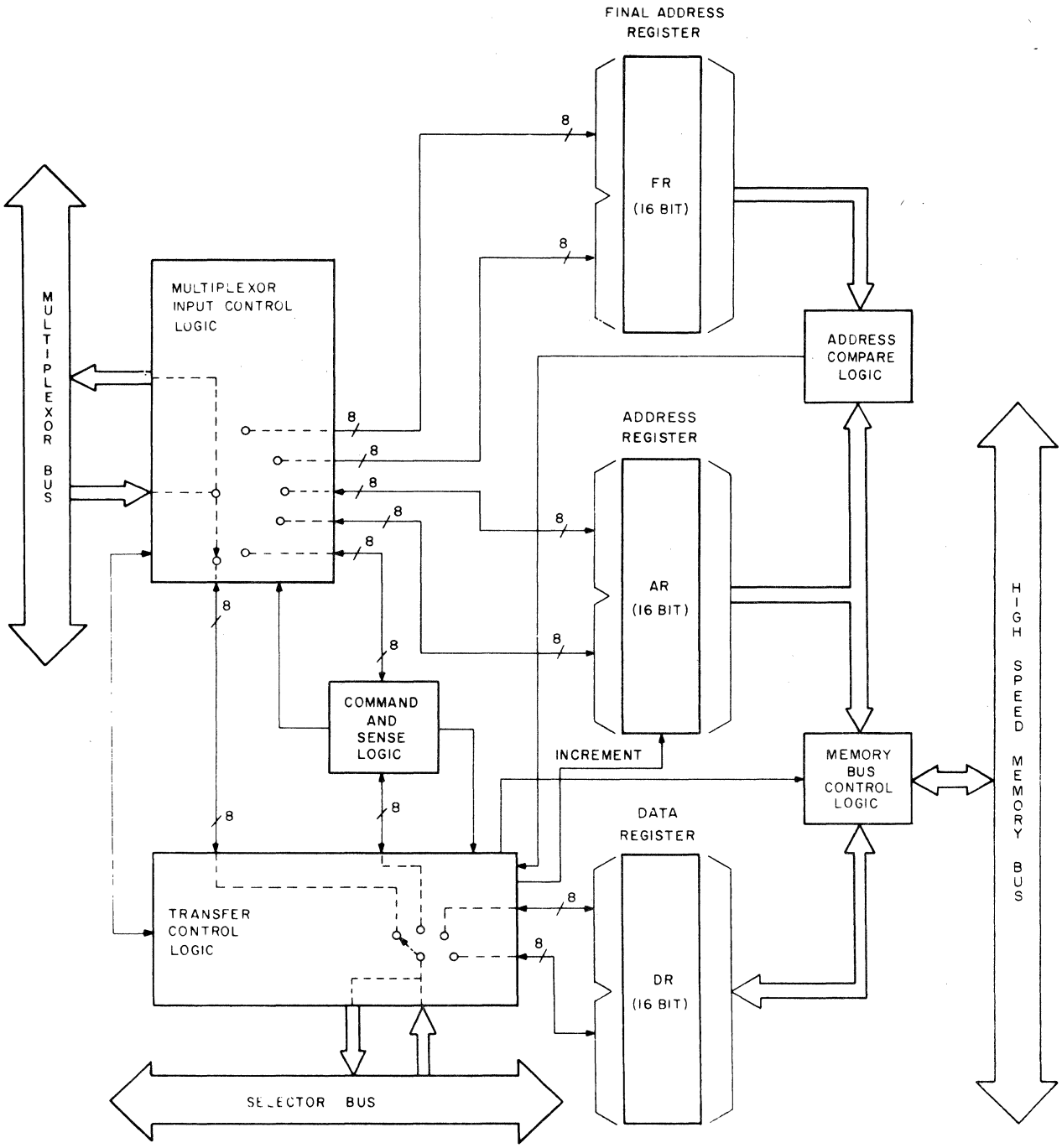


Figure 1-4. Selector Channel, Block Diagram

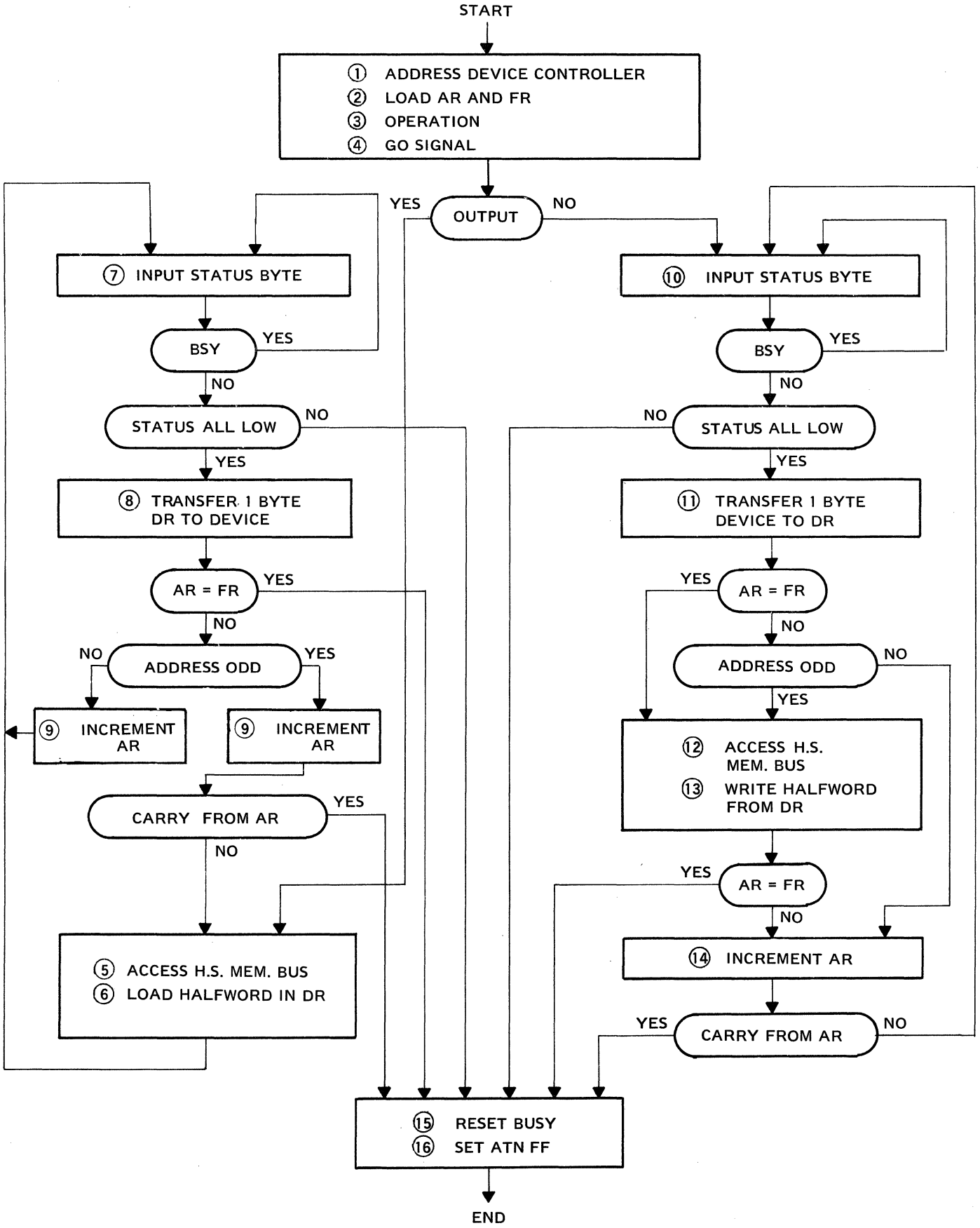


Fig. 1-5. Selector Channel, Flow Chart

3. A command which specifies whether this is an input or an output operation is sent to the Selector Channel from the Multiplexor Channel.

4. A Go Command which starts the transfer operation is sent from the Multiplexor Channel to the Selector Channel.

NOTE

The Processor is now free to continue its program while the block I/O transfer is performed by the Selector Channel. While the transfer is in progress, the Selector Channel ignores all Multiplexor Bus signals except Address, Status Request, and Stop.

Steps 5 thru 9 apply solely to read operations (memory to device). Steps 10 thru 14 apply to write operations (device to memory).

5. If this is a read operation, the Selector Channel requests the High Speed Memory Bus. When the Memory Bus responds, the Selector Channel initiates a memory read cycle.

6. When the memory data becomes available, it is gated to the DR. After the first memory read cycle, the AR will still be even. The Selector Channel proceeds to Step 7 without incrementing the AR. This is done to keep the proper relationship between the contents of AR and the number of bytes moved at the time the AR/FR match check is made. (Consider the case of a single byte transfer where AR=FR when initially loaded.)

7. The Status Byte is input from the device. If the device is Busy, the Selector Channel inputs the Status Byte again. This is continued until the Busy bit is low. If any of the Status bits 5, 6 or 7 are high, the transfer is terminated.

8. A byte is transferred from the DR to the device. If the AR and FR are equal, the transfer is terminated.

9. The AR is incremented. If there is a Carry from the AR, the transfer is terminated. If the AR is even, only one byte has been transferred since the last memory access, and the sequence is repeated from Step 7. If the AR is odd, both bytes have been transferred, and the sequence is repeated from Step 7.

10. If this is an input instruction, the Status Byte is input from the device. If the Busy Bit is high, the Selector Channel inputs the Status Byte again. This process is repeated until the Busy Bit is low. The Selector Channel then checks the other three bits in the Status Code. If any bit is high, the transfer sequence is terminated. If all bits are low, the sequence continues.

11. A byte is transferred from the device to the DR. The contents of AR are compared to the contents of FR. If the addresses are equal, the sequence continues with the memory write operation listed in Step 12. If the addresses are not equal, the AR is checked for contents odd or even. If odd, two bytes have been input and the sequence continues with Step 12. If the address is even, the sequence skips to Step 14.

12. The Selector Channel requests the High Speed Memory Bus. When the Memory Bus responds, the Selector Channel initiates a memory write cycle.

13. The halfword in DR is written into the addressed memory location.

NOTE

If this is the last transfer in an instruction which ends with an even byte address, the previous contents of the second half of DR are written into the right half of the memory location.

The AR/FR match check is made and the sequence is terminated if a match occurs.

14. The AR is incremented. If there is a carry from AR, the transfer is terminated. If there is no carry, the sequence returns to Step 10.

Steps 15 through 17 describe the termination sequence. The conditions which terminate the instruction are:

1. AR equals FR (transfer is complete).
2. AR increments to 0 (carry out of AR).
3. A Status failure from the device (EX, EOM, or DU - See Section 2.4).
4. A Stop command from the Processor.

NOTE

If the Selector Channel is within a Memory Bus cycle when the Stop command is received from the Processor, execution of the Stop command will be delayed until the completion of the memory cycle.

Any one of the above conditions causes the sequence to continue with Step 15.

15. Reset the Selector Channel Busy indication.

16. Set the Selector Channel Attention flip-flop to generate an interrupt to the Processor.

17. After the Processor acknowledges the interrupt and addresses the Selector Channel, it may send a status request to the Selector Channel which will check the Status Code of the device controller. The status byte returned to the Processor will be the device controller status byte with bit 4 (Busy) forced to zero by the Selector Channel. Another option which is normally used after a termination other than the transfer complete termination, is an AR readout. Two Data Requests may be sent to gate first the most significant, then the least significant byte of AR. The programmer may therefore determine at what address the sequence was terminated.

Figure 1-6 illustrates the Selector Channel Output Operation. The examples show both a 3 byte and a 4 byte data transfer between the device controller and memory with termination due to AR/FR match.

Figure 1-7 shows the 3 and 4 byte data transfer for the Input Operation.

The Selector Channel is composed of 3 mother-boards which are mounted in an expansion chassis. Wiring between the boards is by means of the stitch pattern wiring on connector -1 and the top of connector -0. Additional wiring between the mother-boards is provided by 4 cables.

The Selector Bus is created by removing the normal Multiplexor Bus wiring (27 leads) from the I/O Back Panel between the positions occupied by the Selector Register Low board and the Selector Control board. This is shown in Appendix 1 of this manual.

1.3.3 Direct Memory Access Channel

The optional Direct Memory Access Channel permits a 16-bit data transfer between memory and an external device, without transferring the data through the Processor. The Processor simply enables the DMAC device controller. The device specifies the

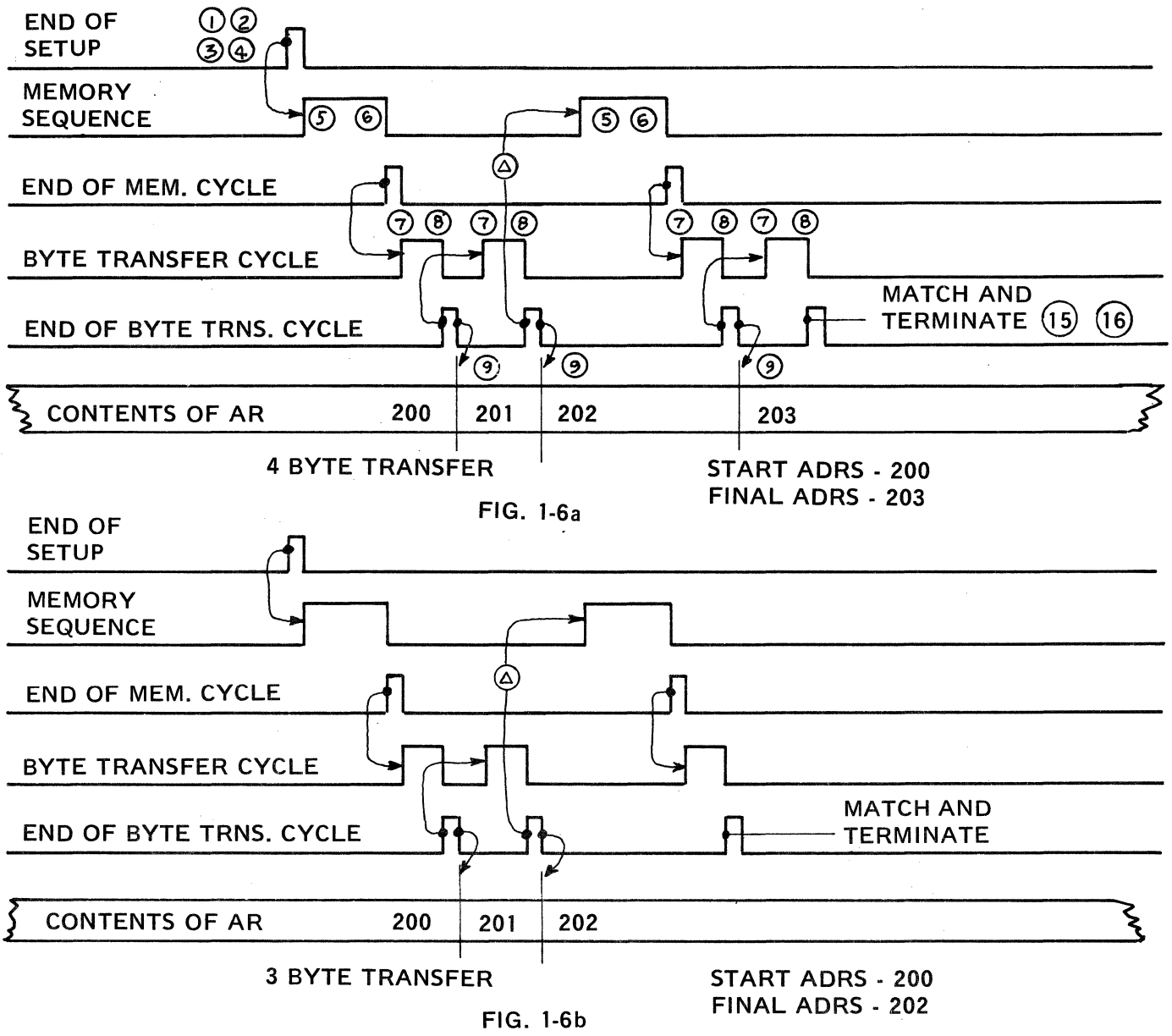


Figure 1-6. Selector Channel Output Timing

memory address involved in the transfer, and the operation to be performed. The DMAC completes the transfer without any manipulation by the Processor. Operations which may be performed via the DMAC are: 1-Read, 2-Write, 3-Read-Increment-Write, and 4-Set AR14. The first two are simply 16-bit halfword I/O operations. The latter two operations are related, and are used in applications which require the counting of external events or signals. Each time the Read-Increment-Write command is received by the DMAC, the data at the specified memory address is accessed, incremented by one, and returned to the same memory location. Thus, that memory address may be used to store the number of times an external event occurs. The Set AR14 (Address Register bit 14) command may be used to increment the address to the next sequential halfword to enable the core registers to count up to 2^{32} events. This feature is used primarily in Pulse Height Analysis applications.

Figure 1-8 is a block diagram of the DMAC. The interface to/from the High Speed Memory Bus is shown on the right side of

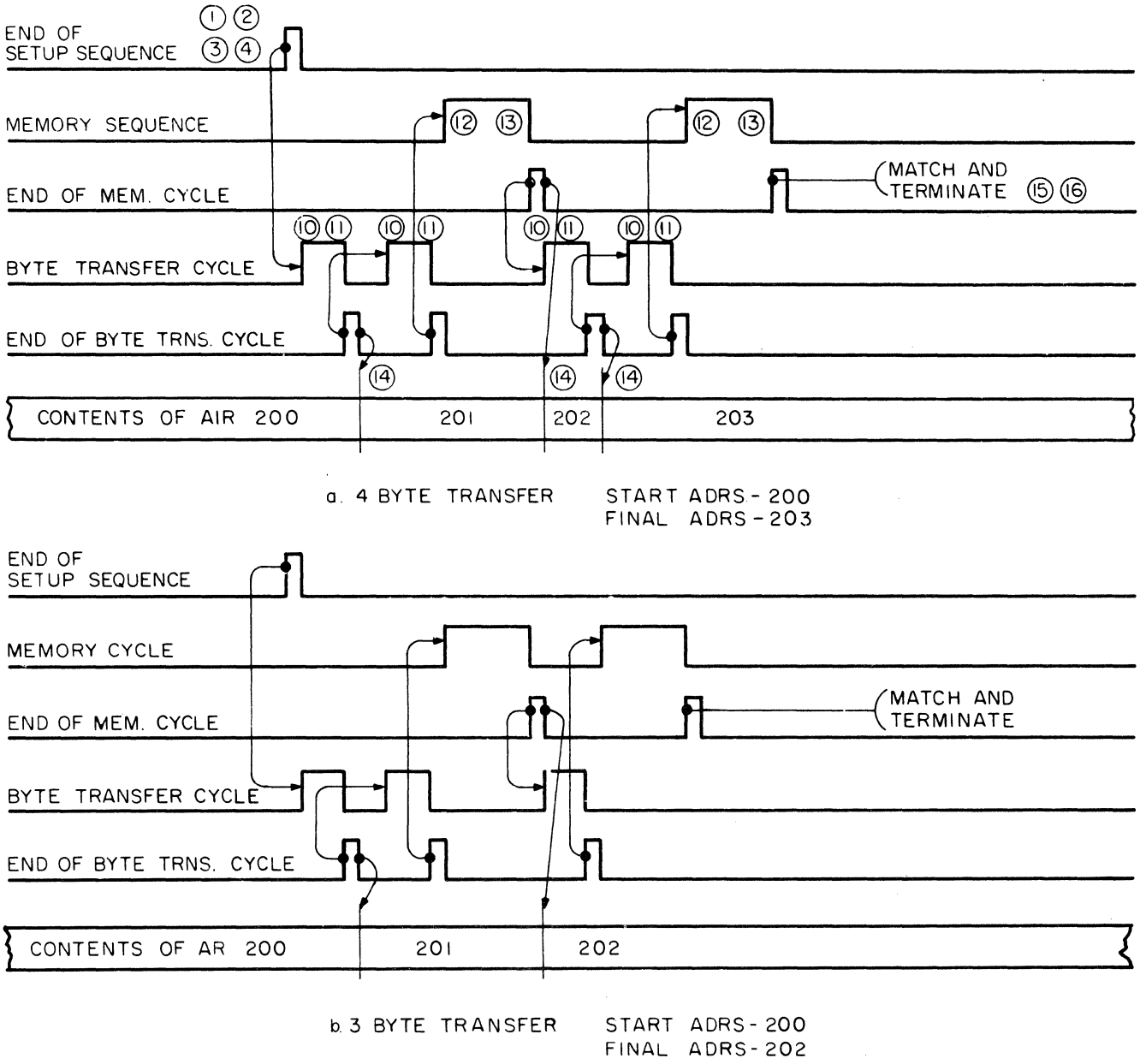


Figure 1-7. Selector Channel Input Operation Timing

Figure 1-8 and is simply the address to memory, and the data to/from memory. The device controller interface is shown on the left side of Figure 1-8. Note the four command inputs to the DMAC: Read, Write, Read-Increment-Write, and Set AR14. When the DMAC receives a command, it replies with the DMAC Busy indication. The 16-bit C Bus provides the address and data interface between the device controller and the Address or Data Register (AR or DR). Gating to/from the C Bus is controlled by the four load/unload and AR/DR select signals: COAR (C Bus Output from AR), CIAR, CIDR, and CODR. The Sync output from the DMAC indicates that a load or unload operation is in progress. The final output is a Carry signal which indicates that the DR is at its maximum count.

The following sequence of operations is a typical example of DMAC I/O operations:

1. The Processor enables the DMAC device controller via the Multiplexor Channel.

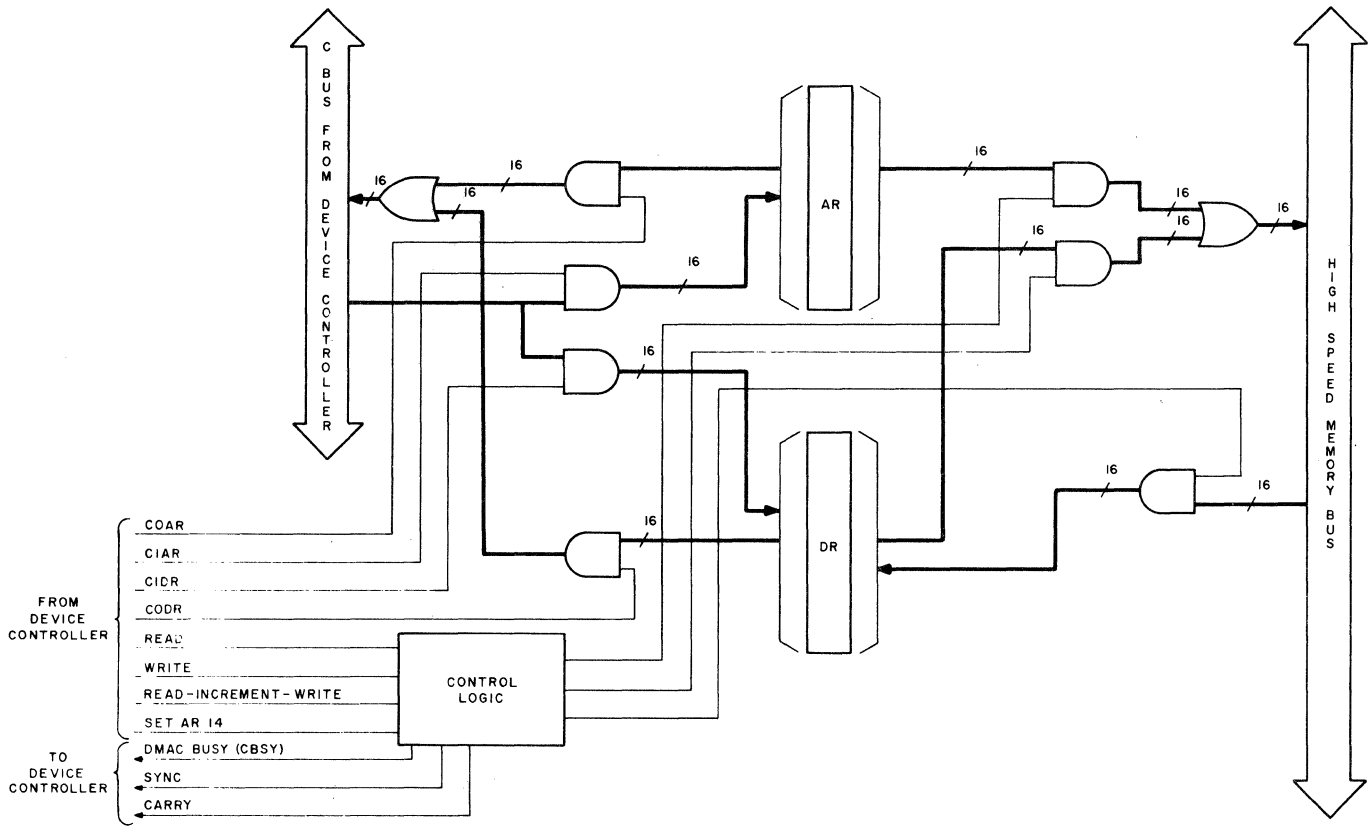


Figure 1-8. DMAC, Block Diagram

NOTE

The Processor is now free to proceed with its program. The I/O operation is executed by the device controller through the DMAC.

2. When the device senses an event, it provides an indication to the device controller.

3. The device controller decodes the indication, sends the address to the DMAC, and sends a command to the DMAC indicating the operation to be performed.

4. If an input operation, the device controller fetches the data from the device and sends it to the DMAC.

   The DMAC then initiates a Write cycle to store the data in memory.

5. If an output operation, the DMAC initiates a Read cycle. When the data from memory becomes available, the device controller gates the data from the DR, through the device controller, to the device.

NOTE

The DMAC always transfers a 16-bit halfword.

6. If a Read-Increment-Write operation, the DMAC initiates the Read cycle. When the data becomes available, the DMAC accepts the data in DR, increments it, and initiates a Write cycle to write the data +1 back into memory. If the data is now all ones, a Carry signal is generated to the device controller. Typically, the device controller generates a Set AR14 command when

Carry is received. The next higher halfword address is then incremented as the lower halfword goes to all zeros. Thus, the registers count up to 2^{32}.

Wiring information for the DMAC is provided in Appendix 2 of this manual.

1.4 MECHANICAL LAYOUT AND WIRING

The GE-PAC 30 Systems Interface employs a unique mechanical layout and wiring configuration which simplifies expansion of the system.

Expansion consists of simply plugging in additional logic boards; no back plane wiring additions are ever required. This section describes the GE-PAC 30 System layout in general, and the Systems Interface layout in detail.

1.4.1 Mechanical Layout

An GE-PAC 30 System consists of a basic card file which may be mounted in a standard 19" RETMA rack, and additional expansion card files as required. Both the basic and expansion card files mount up to 25 9.5" x 10.5" circuit boards designated mother-boards. Each mother-board, in turn, may mount up to 40 smaller component boards designated daughter-boards. Daughter-boards plug into mother-boards via a set of 16 pins. The mother-board is divided into 40 fields as shown on Figure 1-9 to accommodate the daughter boards. Figure 1-10 illustrates a card file, mother-boards, and daughter-boards. A series of daughter-boards which provide a variety of standard logic functions is available from GE-PAC 30. Mother-boards with provisions for mounting potentiometers, relays, indicator lamps, capacitors, and resistors are also available. These general purpose components are described in the Logic Module Handbook, GE-PAC 30 Publication Number 29-005.

Note on Figure 1-10 that there are three sizes of daughter-boards. Figure 1-11 illustrates the size and pin designations for each of the three daughter-board sizes.

Each mother-board may have two 69-pin connectors. The back panel of a card file is shown on Figure 1-12. Note that the connectors are numbered from left to right on the wiring side. The lower row of connectors is designated Field 0; the upper row is designated Field 1. A strip power bus is provided between the two fields. Figure 1-13 shows the pin numbering system. The first digit of the pin number specifies the column number, from 0 to 2. The second two digits specify the row number, from 00 to 22. The dash number at the end specifies the connector field.

1.4.2 Wiring

All card files are completely wire-wrapped at the factory. The wiring is arranged so that the Field 0 connectors on all System Interface mother-boards are used solely for connections to/from the Multiplexor or Selector Channel Bus. The Field 1 connectors are used for communication between boards. The Field 0 (lower) connectors are effectively jumpered to each other to form the bus. The Field 1 (upper) connectors are wired in a different manner. Each pin is wired to the next lower pin (in the same column) on the connector to its right as viewed from the wiring side. This "Stitching" arrangement permits locating I/O Boards in any adjacent positions with no wiring changes. The only constraint is that the boards must be placed correctly relative to each other.

Wiring between card files is via plug in cables which mate with the wiring side of the mother-board connectors. Wiring between a card file and a device is via a plug in cable which mates with one set of the daughter-board connectors on a mother-board. Figure 1-14 illustrates the interconnections between card files and external devices.

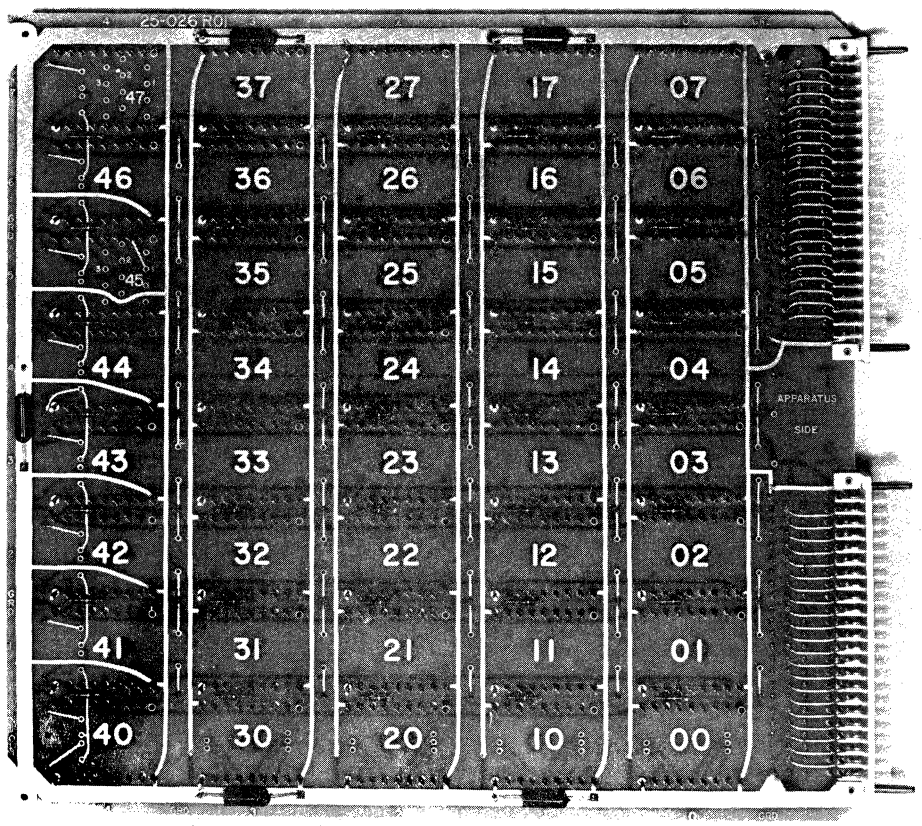
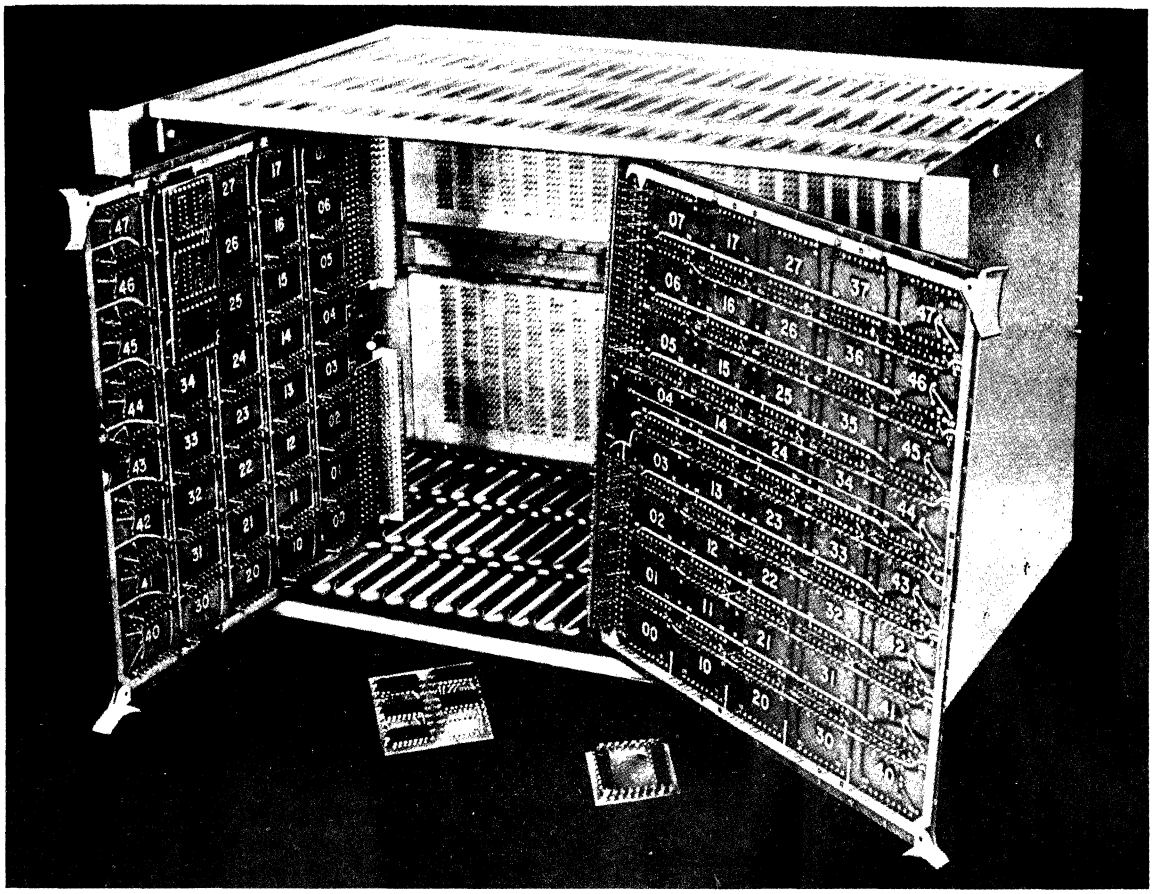


Figure 1-9. Mother-Board Layout



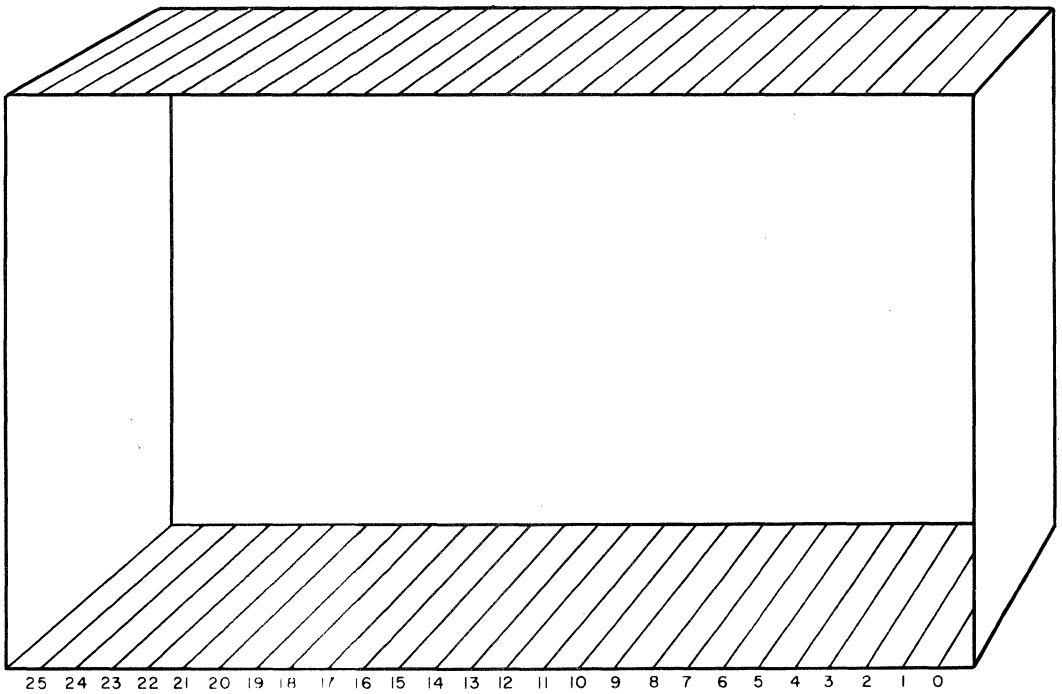


Figure 1-10. Typical GE-PAC 30 Card Cage, Three Quarter Front View

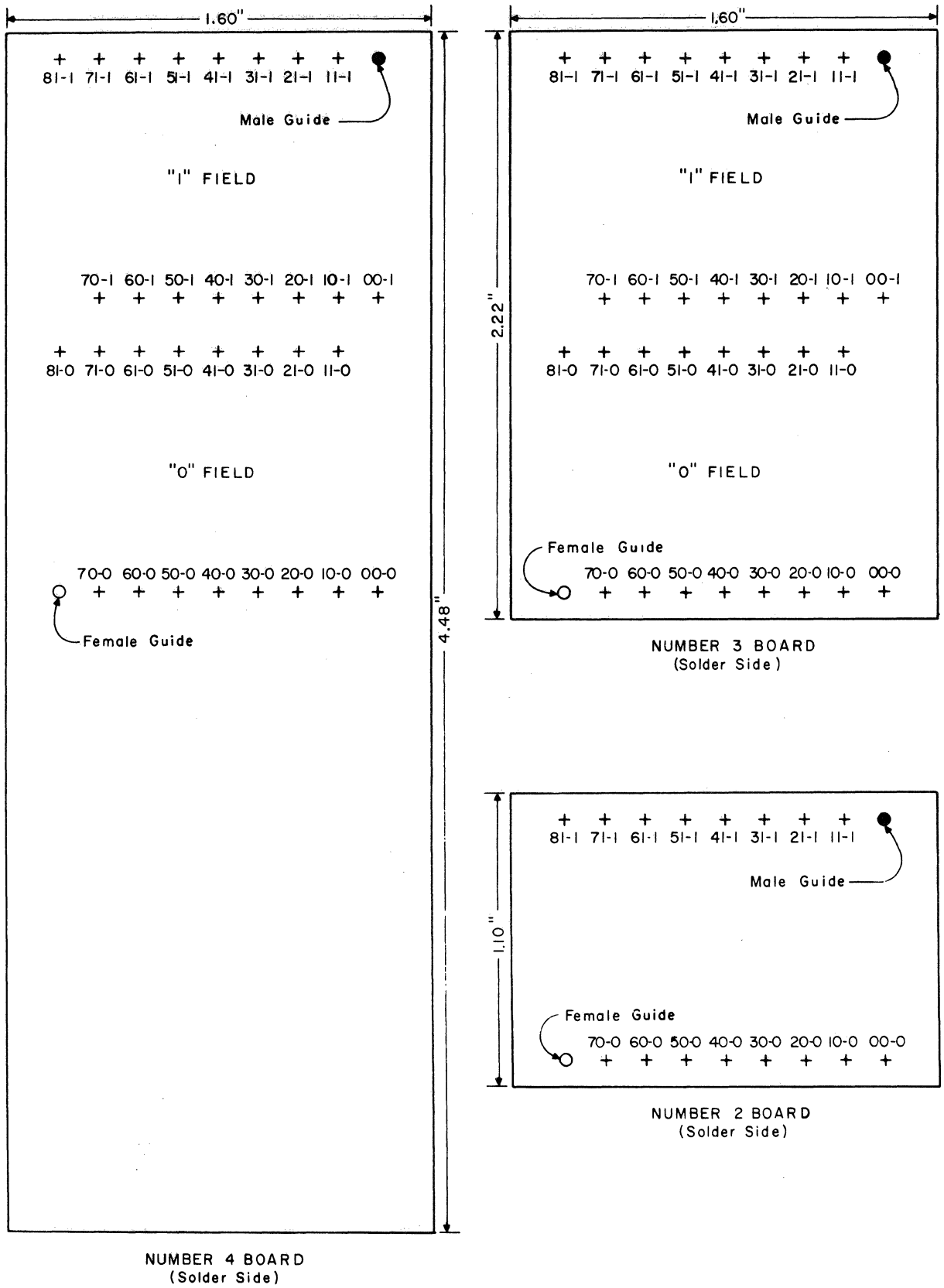


Figure 1-11. Daughter Board Layouts

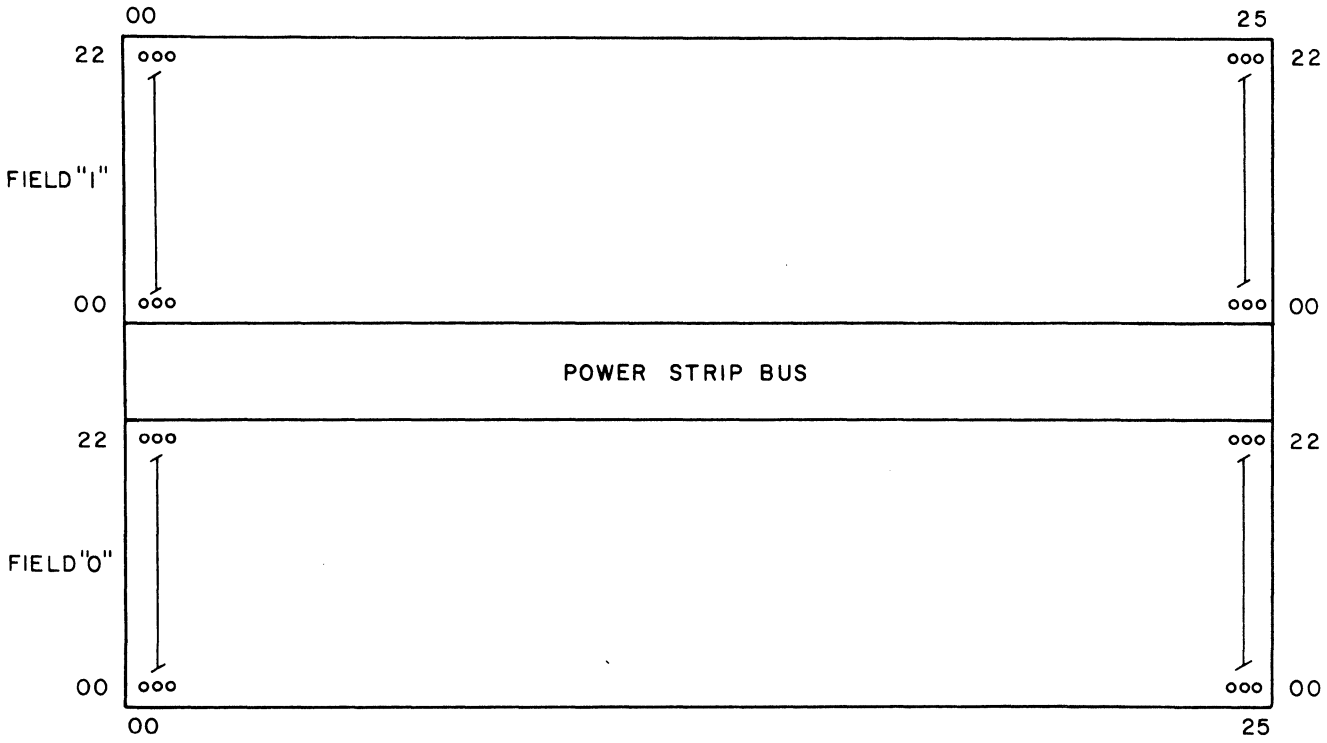


Figure 1-12. Typical GE-PAC 30 Card Cage, Back Panel Layout

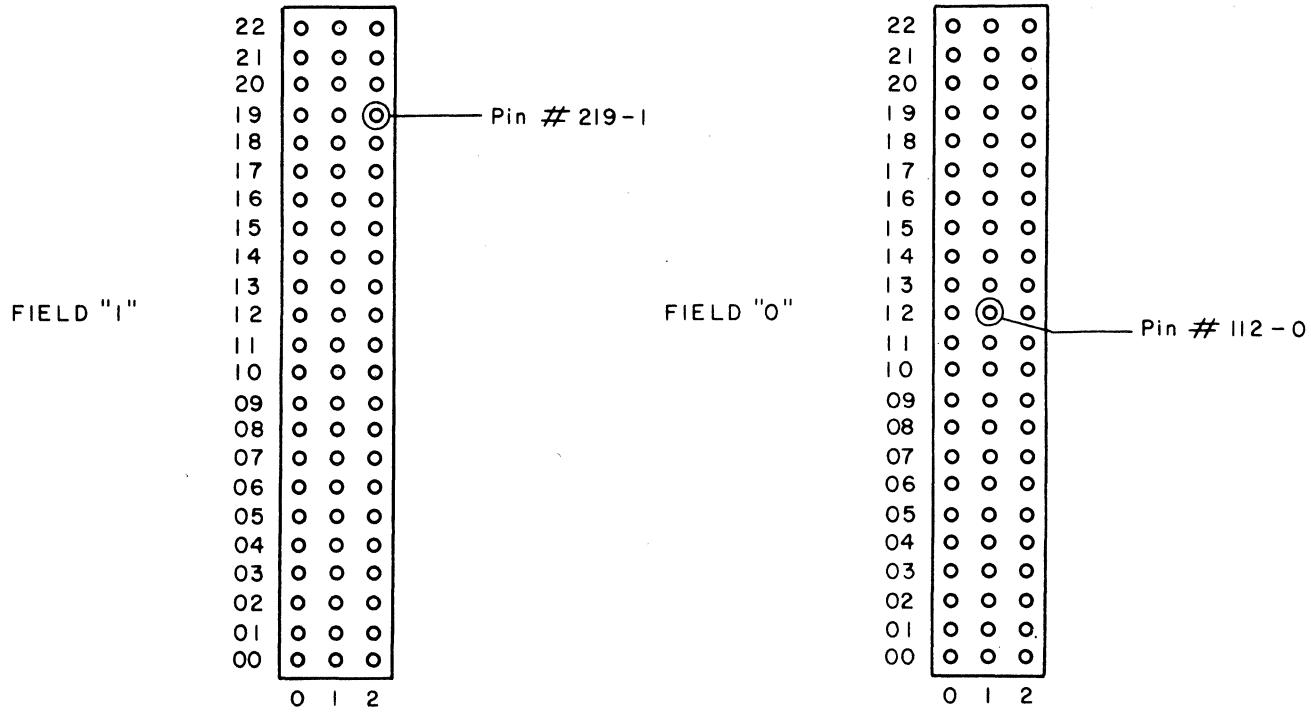


Figure 1-13. Mother Board Connector Layout

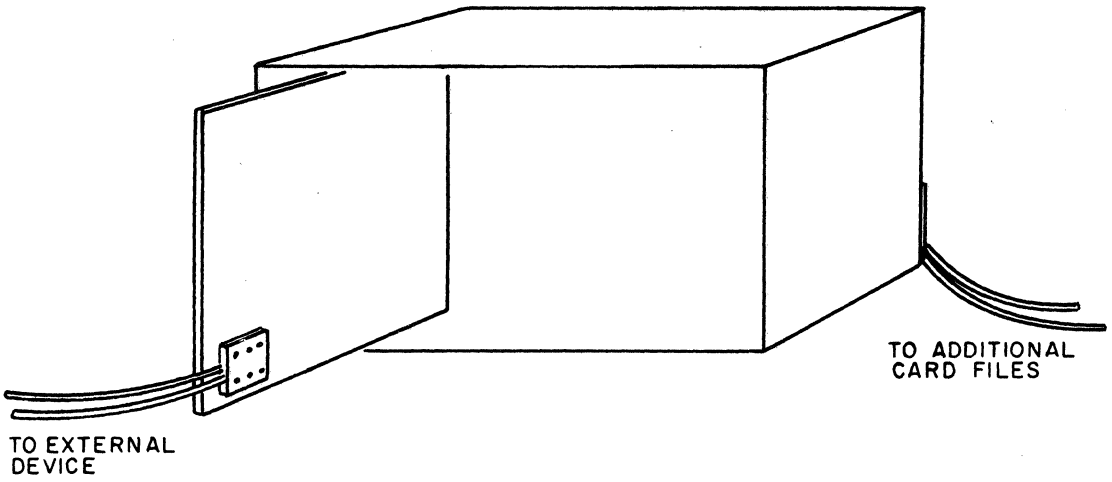


Figure 1-14. Typical GE-PAC 30 Card Cage, Interface Cable Layout

CHAPTER 2

INPUT/OUTPUT INSTRUCTIONS

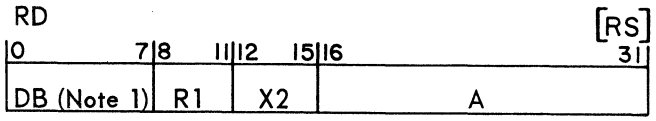
2.1 INTRODUCTION

This Chapter describes the GE-PAC 30 Input/Output (I/O) instructions. Each instruction is implemented by a sequence of operations generated automatically by a micro-coded program. This technique provides a powerful I/O instruction repertoire. For example, a single instruction can transfer a byte of data between any one of 65,536 memory addresses, and any one of 256 external devices. The same instruction can also provide for an automatic indication to the processor when the transfer is completed properly. Each instruction is described separately in the following paragraphs. For more information on instructions, refer to the GE-PAC 30 Reference Manual, Publication Number 29-004.

2.2 READ DATA (RD) INSTRUCTION

Figure 2-1 illustrates the instruction format for the RD instruction. Execution of the RD instruction accesses an 8-bit byte of data from the device specified by the contents of General Register R1. The byte is transferred to the memory byte address specified by A, indexed by the contents of General Register X2. If X2 equals zero, the byte is transferred directly to the memory address specified by A. The transfer takes place via the Multiplexor Channel. Refer to Figures 2-2 and 2-3 during the following description of the RD instruction sequence of operation. The gate and flip-flop designations listed in this Chapter reference the Figures only. They have no hardware significance.

1. The first thing the micro-program does after decoding an RD instruction, is to place the contents of R1 (the device number) on the Data Available Lines (DAL00,0 through DAL07,0).



Note 1: The operation codes in this figure and all similar figures which follow in this chapter are given in hexadecimal notation.

Figure 2-1. Read Data Instruction Format

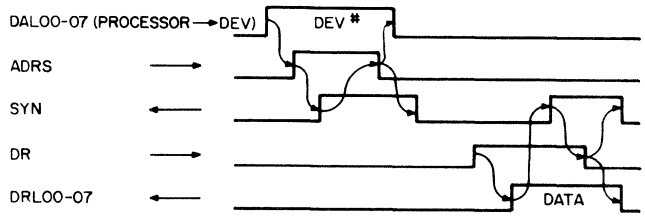


Figure 2-2. Read Data Instruction Timing

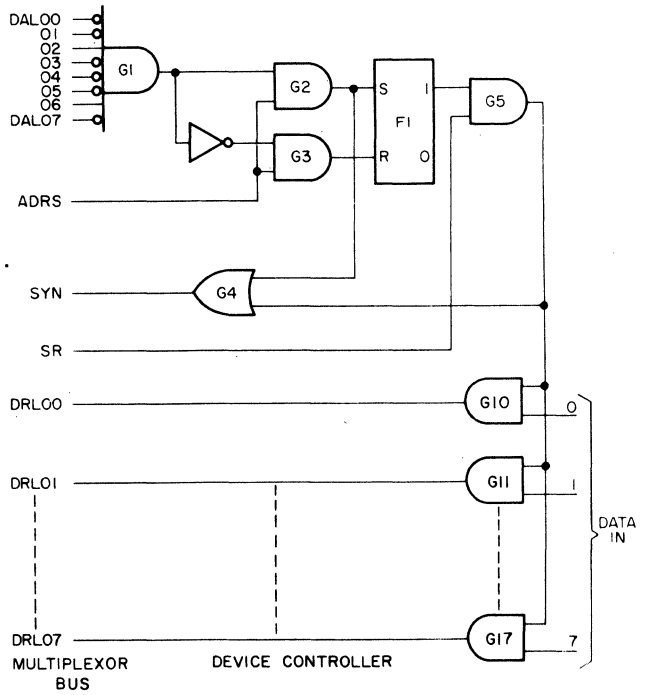


Figure 2-3. Device Controller Logic for Read Data Instruction

NOTE

A 0 or 1 is appended to GE-PAC 30 signal designations to indicate the active state of the signal. Thus, the designation DAL00,0 through DAL07,0 indicates that the 8 DAL lines (DAL00 through DAL07) are low when active; in other words, this is false bus.

2. The Address (ADRS) control line is raised.

3. The device controller which decodes its address via Gate G1 (arbitrary hexadecimal address 22 on Figure 2-3) sets the Address flip-flop F1 through Gate G2.

4. The output from Gate G2, via OR Gate G4, also raises the Synchronization (SYN) response from the device controller to the Processor. The SYN signal at this time indicates that the device controller has decoded its address and has received an ADRS command.

5. When the SYN signal is received, the Processor removes the ADRS command and the device number. The device controller, in turn, lowers the SYN signal.

6. The Processor next raises the Data Request (DR) command. This command is ANDed with the Address Flip-flop (F1) in Gate G5. The output from Gate 5 enables the byte of data from the device to the Processor (Gates G10 through G17). Data to the Processor is sent on the Data Request Lines (DRL00,0 through DRL07,0).

7. The output from Gate G5 also raises the SYN line to the Processor to indicate that the data is ready.

8. The Processor gates the data to the designated byte address (location A indexed by the contents of X2, if specified).

9. When the byte has been stored, the Processor lowers the DR command. The device controller then lowers SYN and removes the data from DRL00,0 through DRL070,0 the sequence for one byte is now completed. The Address flip-flop (F1) remains set until another device is addressed, or until a System Clear (SCLR,O) signal is generated. When another device controller is addressed, F1 is reset through Gate G3. Resetting F1 effectively disconnects the device controller from the Multiplexor Bus.

A Time Out feature is provided in the Processor to prevent locking up the computer on a malfunctioning device or a non-existent device. The Time Out signal is generated if the device controller fails to return the SYN response within 50 to 100 microseconds of a request. The Time Out feature also sets the V Flag in the Program Status Word (PSW) condition code. The programmer may therefore branch on the Time Out condition, typically to an error message print-out routine.

The Read Data to Register (RDR) instruction is executed in exactly the same manner as the RD instruction. The only difference is that the data is stored in General Register R2 instead of A. Figure 2-4 shows the format of the RDR instruction.

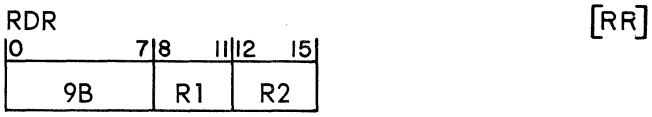


Figure 2-4. Read Data to Register Instruction Format

2.3 WRITE DATA (WD) INSTRUCTION

The format of a WD instruction is shown in Figure 2-5. Execution of the WD instruction transfers a byte from Memory Address A indexed by the contents of General Register X2, to the device number specified by the contents of General Register R1. Refer to Figures 2-6 and 2-7 during the following sequence of operations.

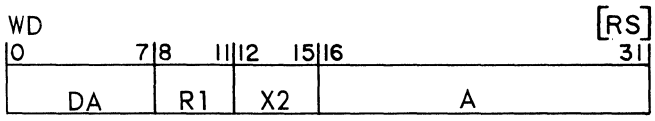


Figure 2-5. Write Data Instruction Format

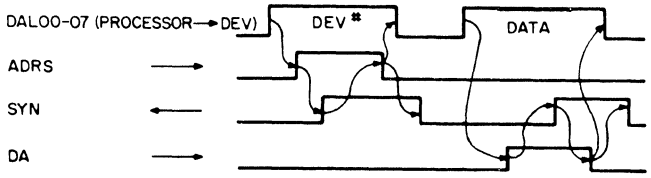


Figure 2-6. Write Data Instruction Timing

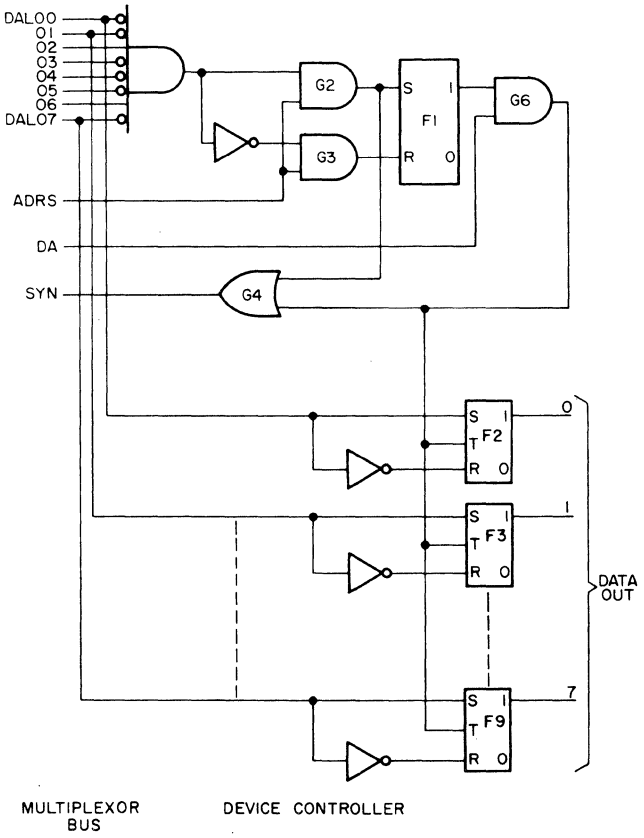


Figure 2-7. Device Controller Logic for Write Data Instruction

1. The device controller is addressed and connected exactly as described in Steps 1 through 5 of the RD sequence (Section 2.2).

2. The contents of the byte address (A indexed by X2) are placed on the Data Available Lines (DAL00,0 through DAL07,0).

3. The Data Available (DA) control line is then raised.

4. The DA signal is ANDed with the Address flip-flop by Gate G6. The G6 output strobes the data into the device controller register (F2 through F9).

5. The G6 output also generates a SYN response to the Processor to indicate that the data has been accepted.

6. When it receives the SYN signal, the Processor lowers the DA line and removes the data from the DAL00,0 through DAL07,0 lines.

7. The device controller then lowers the SYN line.

As described in Section 2.2, the device controller remains selected until another device controller is selected or a System Clear is generated. The Time Out feature is also exactly as described in Section 2.2 for the RD instruction.

Figure 2-8 shows the instruction format for the Write Data from Register (WDR) instruction. The WDR instruction is similar to the WD instruction, except that the byte which is transferred originates in General Register R2 rather than a memory address.

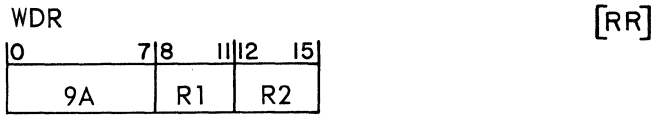


Figure 2-8. Write Data from Register Instruction Format

2.4. SENSE STATUS (SS) INSTRUCTION

Figure 2-9 illustrates the format of the Sense Status (SS) Instruction. Execution of the SS instruction transfers an 8-bit Status Code from the device specified by General Register R1 to memory location A, indexed by the contents of General Register X2. In addition, the four least significant bits are placed in the four bit condition code of the Program Status Word (PSW).

C = Device Busy (BSY) - Indicates that the device is not ready to transfer data.

V = Examine Status (EX) - Indicates that the device has detected a condition which is indicated by the most significant four bits of the Status Condition Code.

G = End of Medium (EOM) - Indicates that the device has reached the end of its data. For example, the Card Reader has reached the end of a card.

L = Device Unavailable (DU) - Indicates that the device is either not connected or not ready.

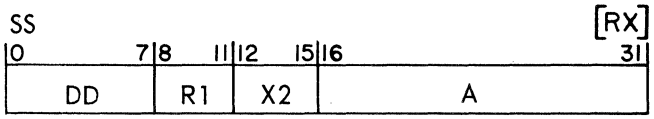


Figure 2-9. Sense Status Instruction Format

Thus the program, after executing an SS instruction, may branch directly on any of the above conditions. Normally if the V Flag is set, the program examines the other four bits of the Status Condition Code. The four bits are stored at memory location A, and may be assigned any significance which is appropriate for the particular device.

Refer to Figures 2-10 and 2-11 during the following sequence of operations for the SS instruction.

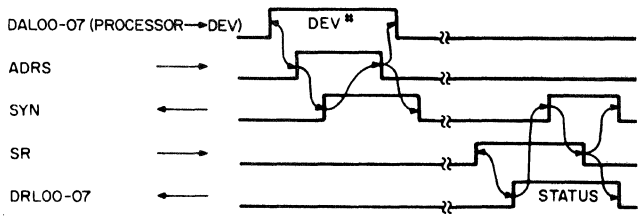


Figure 2-10. Sense Status Instruction Timing

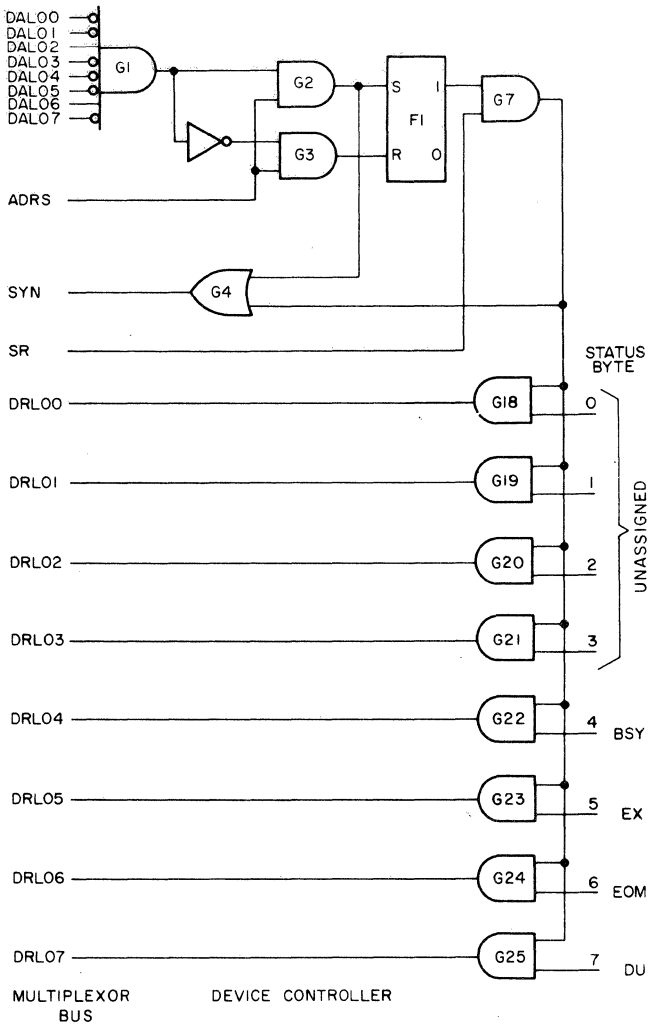


Figure 2-11. Device Controller Logic for Sense Status Instruction

1. The device controller is addressed and connected as described in Steps 1 through 5 of the RD sequence (Section 2.2).

2. The Processor next raises the Status Request (SR) control line, causing a high output from device controller Gate G7.

3. The output from Gate G7 enables the Status Byte from the device to the Processor via **DRL00,0** through **DRL07,0**, and sends a **SYN** response to the Processor to indicate that the data is on the bus.

4. When it receives the SYN signal, the Processor transfers the Status Byte Bits 0 through 7 to Address A, and bits 4 through 7 only to the Condition Code of the PSW.

5. The Processor then lowers SR, which causes the device controller to lower SYN and remove the Status Byte from the bus.

The device controller Address flip-flop and the Time Out feature are as described in Section 2.2 for the RD instruction. Figure 2-12 shows the format of the Sense Status to Register (SSR) instruction. The SSR instruction is similar to the SS instruction except that the Status Byte is stored in General Register R2 instead of memory address A.

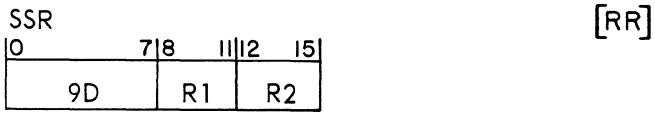


Figure 2-12. Sense Status to Register Instruction Format

2.5 OUTPUT COMMAND (OC) INSTRUCTION

Figure 2-13 illustrates the format of the OC instruction. Execution of the OC instruction transfers an 8-bit Output Command from address A, indexed by the contents of General Register X2, to the device specified by the contents of General Register R1. Command line coding is normally assigned to either device control function or device controller modes of operation. None of the command bits are preassigned a specific function. The OC instruction is therefore a powerful instruction which may be tailored to the specific requirements of a particular system.

Refer to Figures 2-14 and 2-15 during the following sequence of operation description.

1. The device controller is addressed and connected as described in Steps 1 through 5 of the RD sequence (Section 2.2).

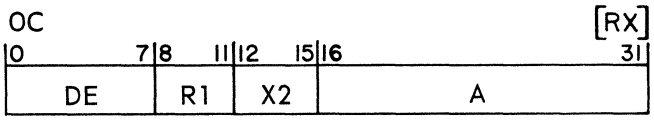


Figure 2-13. Output Command Instruction Format

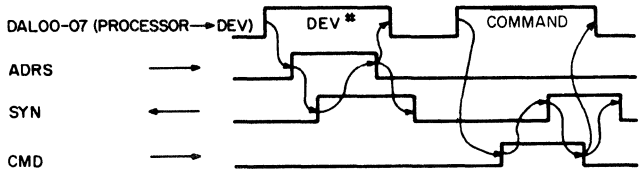


Figure 2-14. Output Command Instruction Timing

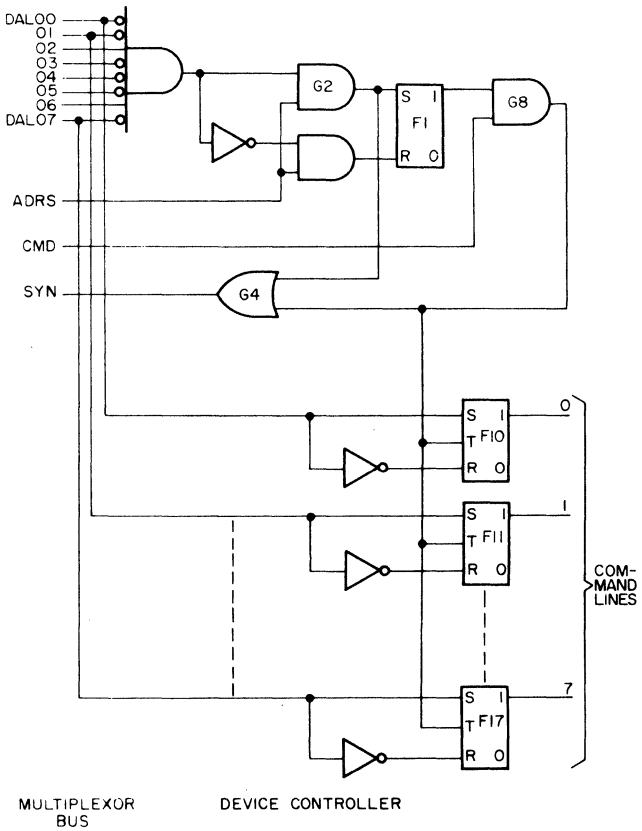


Figure 2-15. Device Controller Logic for Output Command Instruction

2. The Processor next places the contents of the byte address A (the command word) on DAL00,0 through DAL07,0).

3. The Processor then raises the Command (CMD) control line. The CMD signal is ANDed with the Address flip-flop by Gate G8.

4. The output from G8 strobes the Command word (on DAL00,0 through DAL07,0) into the Command Control Register in the device controller (F10 through F17). The output from G8 also sends a SYN response to the Processor to indicate that the device has stored the Command word.

5. The Processor then lowers CMD and removes the Command word from DAL00,0 and DAL07,0.

6. Finally, the device controller lowers its SYN line.

The device controller Address flip-flop and the Time Out feature are as described for the RD instruction in Section 2.2. Figure 2-16 shows the format of the Output Command from Register (OCR) instruction. The OCR instruction is similar to the OC instruction except that the Command Byte is from General Register R2 instead of Address A.

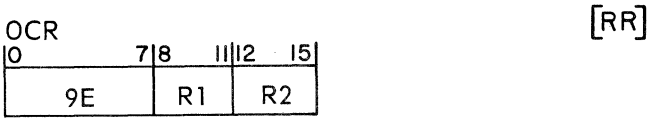


Figure 2-16. Output Command from Register Instruction Format

2.6 ACKNOWLEDGE INTERRUPT (AI) INSTRUCTION

To understand the AI instruction, the reader should have a working knowledge of the GE-PAC 30 Interrupt System. A brief description of the Interrupt System is provided here; for more detail refer to the Reference Manual (Publication Number 29-004). Interrupts are provided to detect events both within, and external to, the computer. The occurrence of the interrupt may cause the program to Branch from its normal sequence to an interrupt subroutine which performs some operation appropriate to the interrupt. For example, an external device may interrupt to indicate that it is ready to transfer another byte of data. The interrupt subroutine would then initiate another transfer. After the interrupt is serviced (the interrupt subroutine is completed), the main program is resumed at the point at which it was interrupted.

Figure 2-17 shows the format of the Acknowledge Interrupt (AI) instruction. The AI instruction is normally the first instruction in the interrupt subroutine which services interrupts from external devices. When executed, the device number of the interrupting device is placed in General Register R1, and the Status Byte of the interrupting device is placed in Address A, indexed by the contents of General Register X2. The least significant four bits of the Status Byte are placed in the Condition code of the Program Status Word (PSW) exactly as described in Section 2.4 for the SS instruction. Thus, for example, with a single AI instruction a magnetic tape reader may be identified, and the fact that it has reached an EOR gap determined (via the EOM Status Flag).

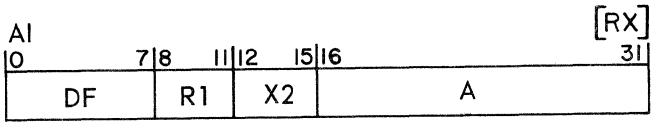


Figure 2-17. Acknowledge Interrupt Instruction Format

Refer to Figures 2-18 and 2-19 during the following sequence of operation:

1. The device interrupts and sets flip-flop F18 in the device controller. The output from F18 generates an Attention (ATN) signal to the Processor.

2. The Processor responds by raising the Acknowledge (ACK) control line.

NOTE

The ACK line is received by the first device controller in the line as Receive Acknowledge (RACK). See figure 2-19. If F18 in the first controller is not set, the RACK signal is gated out of this controller as Transmit Acknowledge (TACK). The next controller receives it as RACK. Thus the ACK signal "daisy chains" through the device controllers until it finds

one with its Interrupt flip-flop (F18) set. The '0' output from F18 inhibits the propagation of TACK to the next device controller.

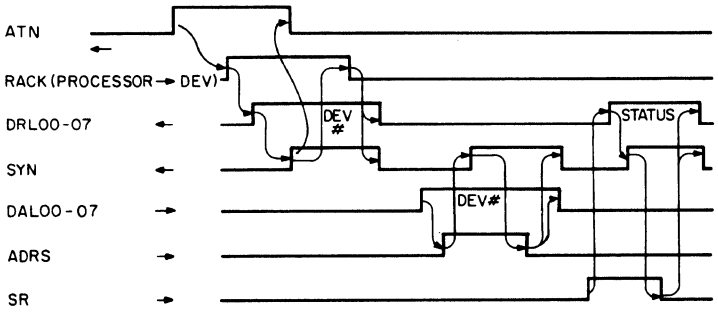


Figure 2-18. Acknowledge Interrupt Instruction Timing

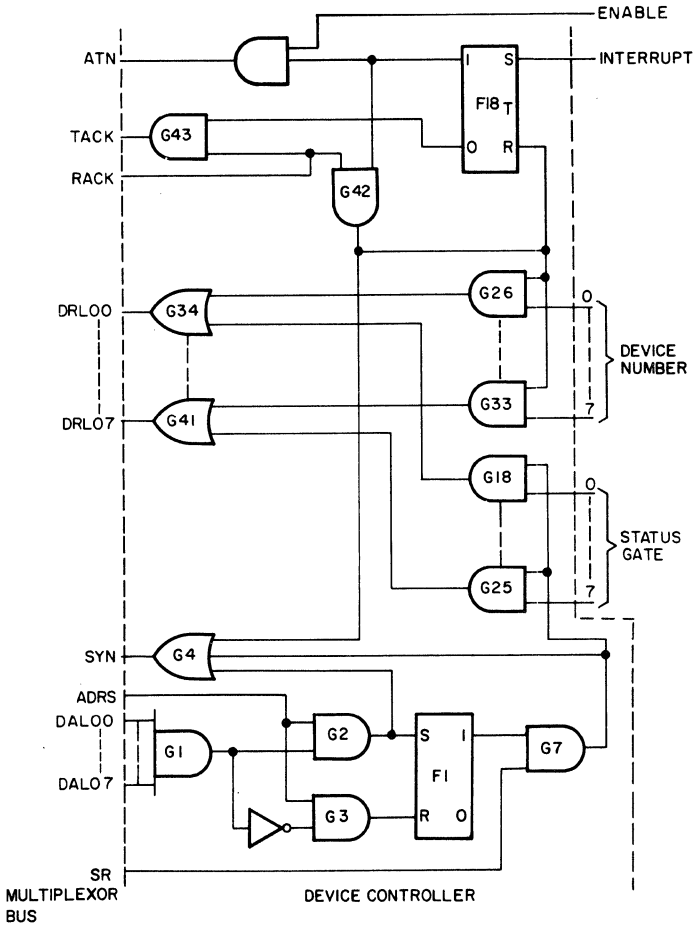


Figure 2-19. Device Controller Logic for Acknowledge Interrupt Instruction

3. The F18 high output and RACK are ANDed to enable the Device Number from the device to the **DRL00,0 through DRL07,0 lines, and to send SYN to the** Processor to indicate that the device number is on the lines. The ATN flip-flop is also reset.

4. The Processor gates the device number into General Register R1.

5. The Processor then lowers the ACK line which, in turn, causes the device controller to lower the SYN line.

6. The Processor then addresses the same device and gates its Status Byte to Address A and the condition code of the PSW exactly as described in Steps 1 through 5 of Section 2.4.

The device controller Address flip-flop and the Time Out feature are as described previously in Section 2.2 for the RD instruction. Figure 2-20 shows the format of the Acknowledge Interrupt to Register (AIR) instruction. The AIR instruction transfers the Status Byte to General Register R2, rather than to memory location A, as in the AI instruction.

AIR [RR]

0 7	8 11	12 15
9F	R1	R2

Figure 2-20. Acknowledge Interrupt to Register Instruction Format

2.7 READ BLOCK (RB) INSTRUCTION

Figure 2-21 shows the instruction format of the Read Block (RB) instruction. General Register R1 specifies the device number, the indexed address A contains the starting address for the block transfer. The next sequential halfword contains the ending address. When an RB instruction is executed, a block of data is transferred from an external

device to sequential byte locations in memory. Between each byte transfer the device status is checked. The four least significant bits of the Status Byte are scanned. If the BSY bit (C Flag) is set, the Processor assumes that the transfer is still in progress. The Processor initiates another Status Byte input sequence each time the BSY bit is set. When the BSY bit is low (indicating that the byte transfer is complete), the Processor scans the remaining three least significant bits. If any of the bits are set, the transfer sequence is terminated. If all bits are reset, the next byte transfer is initiated. At the end of the transfer sequence (when the present address equals the ending address), the Status Condition Code should contain all zeros. The program may therefore branch conditionally on the Status Condition Code following the RB (or RBR) instruction. Condition Code assignments are as described in Section 2.4. Refer to Figures 2-22 and 2-23 during the following sequence of operation description. The sequence is essentially a combination of RD and SS instructions. However, the device is only addressed once, the DR and SR control lines are then raised alternately until the transfer is terminated.

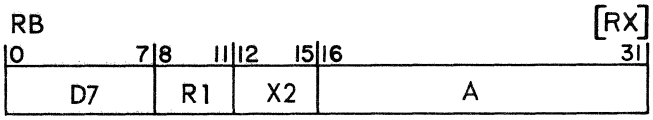


Figure 2-21. Read Block Instruction Format

1. The device controller is addressed and connected as described in Steps 1 through 5 of the RD sequence (Section 2.2).
2. A byte is transferred from the device to the Processor as described in Steps 6 through 9 of the RD sequence (Section 2.2).
3. The Status Byte from the device is then transferred to the Processor as described in Steps 2 through 5 of the SS instruction sequence (Section 2.4).
4. If the address does not match the final address, and the Condition Code of the PSW is all zeros, the Processor increments the address and repeats Steps 2 through 4 of this sequence.

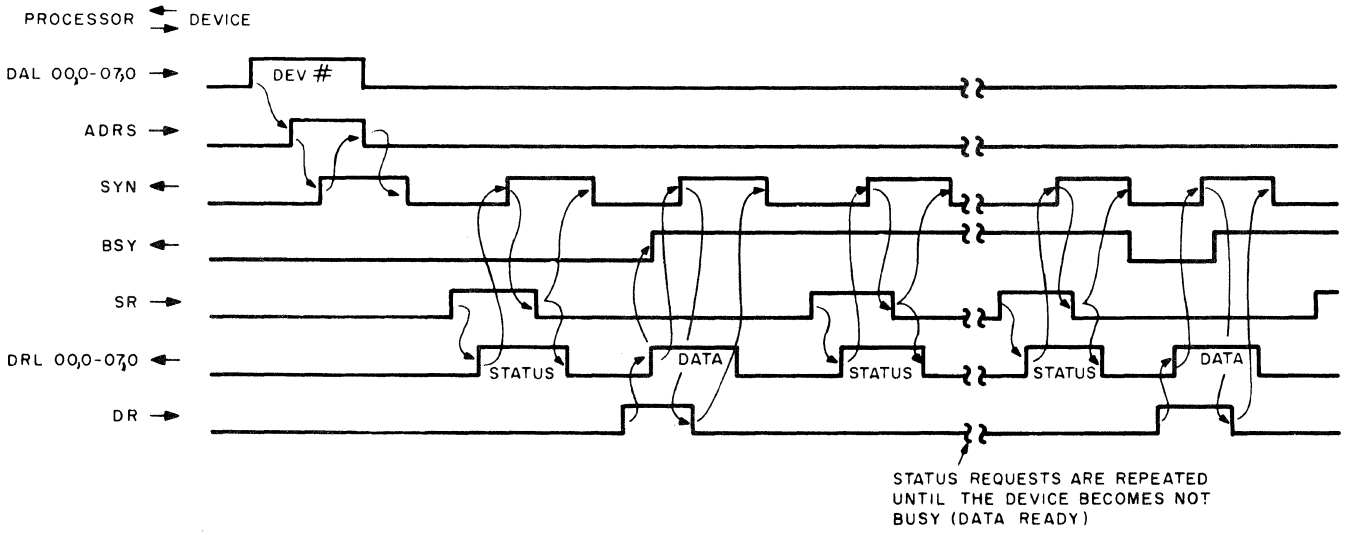


Figure 2-22. Read Block Instruction Timing

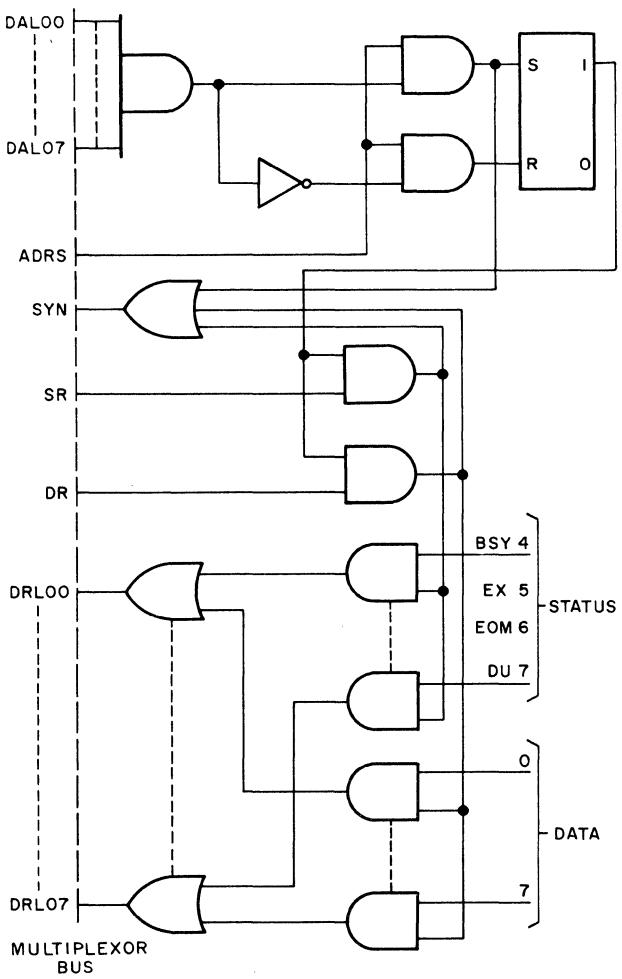


Figure 2-23. Device Controller Logic Read Block Instruction

5. The instruction terminates normally when the address equals the final address. If bit 5, 6, or 7 of the Condition Code in the PSW is set after a transfer, the instruction is terminated. As described previously, the program may then branch Conditionally on the Condition Code. If bit 4 (BSY) is set, Step 3 is repeated.

The device controller Address flip-flop and the Time Out feature are as described previously in Section 2.2 for the RD instruction. Figure 2-24 shows the format for the Read Block to Register (RBR) instruction. The RBR instruction differs from the RB instruction only in that the starting address is specified by the contents of General Register R2.

RBR [RR]

0 7	8 11	12 15
97	R1	R2

Figure 2-24. Read Block to Register Instruction Format

2.8 WRITE BLOCK (WB) INSTRUCTION

Figure 2-25 shows the instruction format of the optional Write Block (WB) instruction. The General Register specified by R1 contains the device number. The indexed address contains the starting address for the block transfer. The next sequential halfword contains the ending address. Execution of this instruction transfers bytes from sequential locations in memory to the external device specified. The Status Byte is checked between each byte transfer. The four least significant bits of the Status Byte are scanned. If the BSY bit (C Flag) is set, the Processor assumes that the transfer is still in progress. The Processor initiates another Status Byte input sequence each time the BSY bit is set. When the BSY bit is low (indicating that the byte transfer is complete), the Processor scans the remaining three least significant bits. If any of the bits are set, the transfer sequence is terminated. If all bits are reset, the next byte transfer is initiated. At the end of the transfer sequence (when the present address equals the ending address), the Status Condition Code should contain all zeros. The program may therefore branch conditionally on the Status Condition Code following the WB (or WBR) instruction. Condition Code assignments are as described for the Sense Status instruction in Section 2.4. Refer to Figures 2-26 and 2-27 during the following sequence of operation description.

WB [RX]

0 7	8 11	12 15	16 31
D6	R1	X2	A

Figure 2-25. Write Block Instruction Format

1. The device controller is addressed and connected as described in Steps 1 through 5 of the RD sequence (Section 2.2).

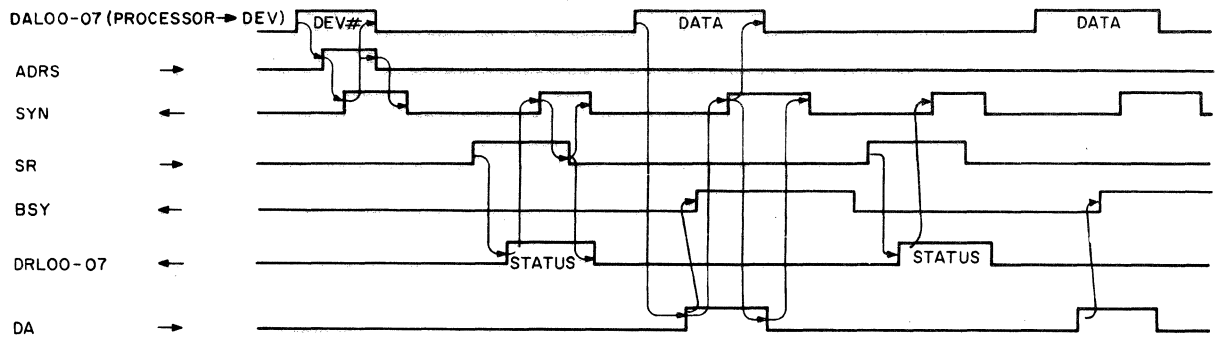


Figure 2-26. Write Block Instruction Timing

2. A byte is transferred from the Processor to the device as described in Steps 2 through 7 of the WD sequence (Section 2.3).

3. The Processor reads in the Status Byte as described in Steps 2 through 5 of the SS sequence (Section 2.4).

4. If the BSY indication is set, Step 3 is repeated to input the Status Byte again.

5. If the BSY indication is low, and any of the three least significant bits are set, the transfer is terminated.

6. If all four least significant Status Byte bits are reset, the Processor compares the memory address with the ending address. If they are equal, the transfer is terminated. If the addresses are not equal, the address is incremented and Steps 2 through 6 of this sequence are repeated.

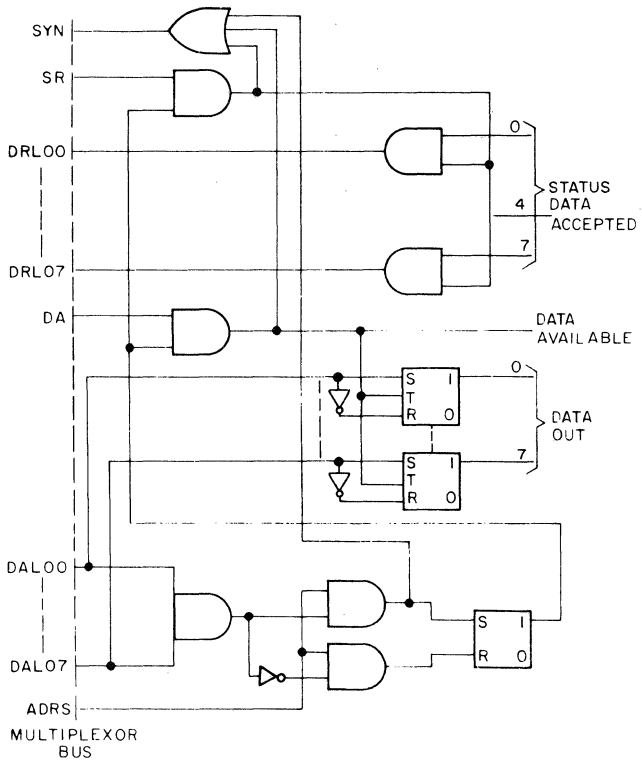


Figure 2-27. Device Controller Logic for Write Block Instruction

Again, the device controller Address flip-flop and the Time Out feature are as described previously for the RD instruction (Section 2.2). Figure 2-28 shows the format of the WBR instruction. The WBR instruction is the same as the WB instruction except that the starting address is specified by the contents of General Register R2 rather than by the effective address.

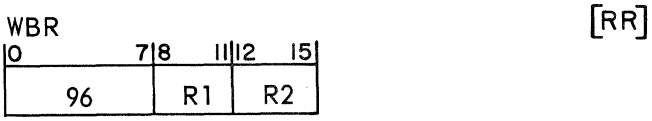


Figure 2-28. Write Block from Register Instruction Format

CHAPTER 3

DEVICE CONTROLLER LOGIC DESIGN

3.1 INTRODUCTION

This Chapter describes the procedure to follow in designing I/O device controllers. While it would be impossible to describe all possible controllers, this Chapter explains representative circuits in enough detail to permit design of most controllers. Note that Chapter 4 describes a General Purpose Interface Controller which simplifies device controller design.

3.2 MULTIPLEXOR CHANNEL

The I/O bus system (either Multiplexer or Selector Channel Bus) consists of 27 shared, unidirectional leads which may be divided into four groups:

1. Data Available Lines (DALs) form a group of eight lines from the Processor and carry address, command, or data bits to the Device Controller circuits.

2. Data Request Lines (DRLs) form a group of eight lines which carry status, device address, or data from the Device Controller circuits to the Processor. In the Processor the lines are gated into registers.

3. Control Lines (CLs) form a group of eight lines from the Processor. Control Lines are energized on a one-out-of-eight basis. These lines control the use and intent of the DALs and DRLs. One of these lines, CL05,0, carries the interrupt acknowledge (ACK) signal and is not a shared line, but breaks up into a series of short lines to form the daisy-chain priority system. The Device Controller closest to the Processor has the highest priority since the ACK signal must pass through it first.

4. System Synchronize (SYN), Interrupt Attention (ATN) and System Clear (SCLR) lines form the last group. The SYN and ATN lines carry signals to the Processor where they are used in the timing and control of the I/O bus system. A SYN signal indicates that the Device Controller circuit has received a signal on one of the Control Lines. The ATN line is energized when any of the Device Controller circuits cause an interrupt. Access to the ATN line is under control of an Enable (EBL) flip-flop in each Device Controller. The SCLR,O line provides a relay contact closure to ground which is used to set up initial or preferred states in each Device Controller.

All buses are of the false type, i.e. zero active. The Device Controller circuits used to communicate with the I/O bus system are shown in Figure 3-1. In a typical case, the DALS and Control Lines are buffered by standard gates to drive the Address, Command, Control and ATN/ACK circuits. The signals back to the Processor on the DRLs are gated by power gates whose outputs are OR tied within the Device Controller and on the bus. The load resistors for the DRLs are located in the Processor. The paragraphs which follow list the conditions affecting bus usage, and provide a set of design rules. Standard circuits for ATN/ACK and address decoding are also shown.

The Systems Interface uses Diode-Transistor Logic (DTL) power gates for bus drivers on the unbuffered I/O Bus System. On the DAL and Control Lines, the

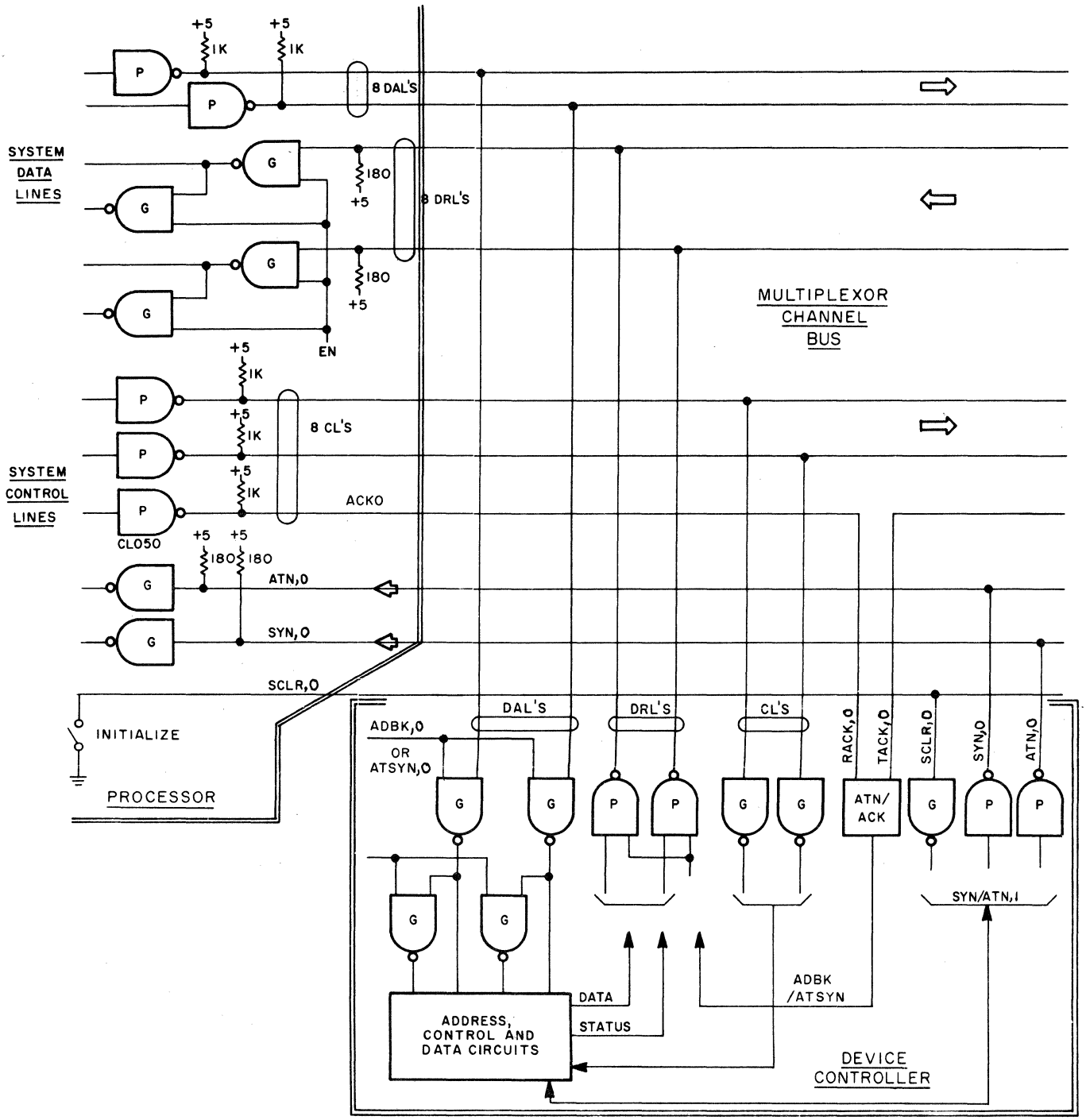


Figure 3-1. I/O Bus Communication Circuits, Logic Diagram

line drivers located in the Processor are capable of handling 25 DTL loads in addition to a 1K pull-up resistor. The ATN,O, SYN,O and DRL bus lines are driven by power gates distributed throughout the device controller boards.

On each line, the gate collectors are OR tied and share a common load resistor (located in the Processor) as shown on Figure 3-1. The value of the load resistor, and the number of OR ties, is determined by the total OFF leakage current of the power gates, considered with the maximum ON current of a single gate whose saturation voltage is still below the logical zero level.

Calculations for the bus load resistor (RL on Figure 3-1) and for the allowable fan-in, show that for worst case conditions and a 0.7 volt noise margin, the fan-in is about 50 gates. The basic rules for device controllers tying to the I/O buses are:

1. Only one DTL load should be placed on each device controller input line from an I/O bus (DAL Control Lines, and SCLR,O line).

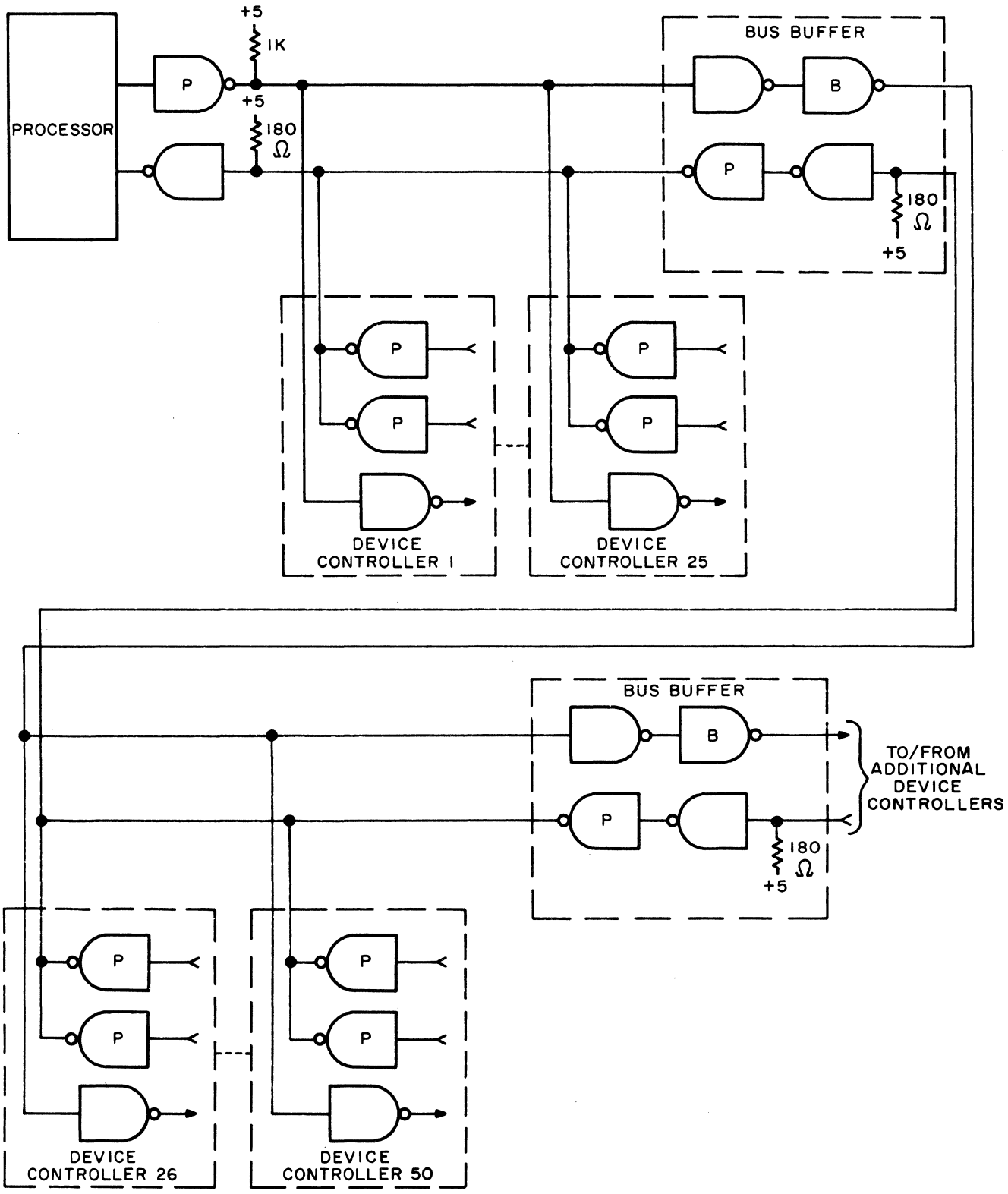


Figure 3-2. Multiplexor Channel Bus Buffers

2. Not more than two DTL power gates should be OR to a device controller output line to an I/O bus (ATN, SYN, and DRLs).

The previous two rules give the Processor a basic I/O drive capability of 25 device controllers. Additional buffering of the bus, within the computer is provided by expansions as shown on Figure 3-2.

3.3 DEVICE CONTROLLER ADDRESSING

Refer to Figure 3-3 during the following description. The dotted lines around the groups of logic functions represent GE-PAC 30 standard logic packs (daughter boards). Further details on the logic packs may be found in the GE-PAC 30 Logic Module Handbook, Publication Number 29-005. When a device controller is addressed, the 8 bit address code is placed on the Data Available Lines (DAL00,0 thru DAL07,0). The two Model 35-040 packs buffer the lines and provide the true and false DAL lines. The Model 35-038 boards are wired with the desired address code, and the 8 coded outputs are applied to an 8 input gate, Model 35-022. Thus, the Decoded Device output (DD,1) goes true. The address control line, ADRS,1 then strobes the DD,1 line into the address Flip Flop (Model 35-001).

The Synchronize signal is returned to the processor, during the presence of ADRS,1, via the address Sync line ADSY,O. The Model 35-022 gate is used here as an OR gate for returning the other device command Sync lines. The set output of the address flip-flop called Device Enable (DENB,1), is used to gate all other I/O control lines to the device controller. When another device is addressed, the decoded device line, DD,1, is low, causing the ADRS,1 strobe line to reset the address flip-flop, and disabling the controller. Capacitor C1 on the SYNC return is used to generate a delay of approximately 200 nanoseconds to allow gate conditions to settle on the lines.

3.4 DATA AND STATUS INPUT

Figure 3-4 shows how a byte of data and status may be read into the Processor. When the device is addressed, DENB,1 is high, enabling the Status Request (SR) or Data Request (DR) control line. The SR or DR in turn, enables the status or data bytes onto the Data Request Lines (DRL00,0 through DAL07,0). The Model 35-020 logic pack is a convenient means of OR tying multiple data sources onto the DRL lines. Each of the control lines automatically generates a return sync signal SRSYO or DRSYO. The device controller logic should place a high on BSY1 until the data is ready and settled on the Data Request lines (DR01,0 through DR07,0). The Processor may now be synchronized to the device data rate by executing Sense Status instructions and branch looping on Busy until the Busy bit is low. Then, when the Busy bit is low, the program may execute a Read Data instruction. Device synchronization can also be achieved by generating an interrupt when the data is ready.

The End of Medium (EOM) bit is normally placed high at the termination of the device medium, such as End of Card. The Device Unavailable (DU) bit typically signifies that device power is not turned on.

The Examine Status (EX) bit is used to signify other appropriate device conditions. In this case the user assigns S01 through S31 to appropriate conditions, such as Parity Error, etc.

3.5 DATA AND COMMAND OUTPUT

Figure 3-5 shows how a byte of data and command may be output from the Processor. The buffered true and false Data Available Lines DAL00,1 through DAL07,1 and DAL00,0 through DAL07,0 from Figure 3-3 feed to the set and reset inputs of the Data Register. When the device is addressed, DENB,1 is high, enabling the control line DAG,1 to

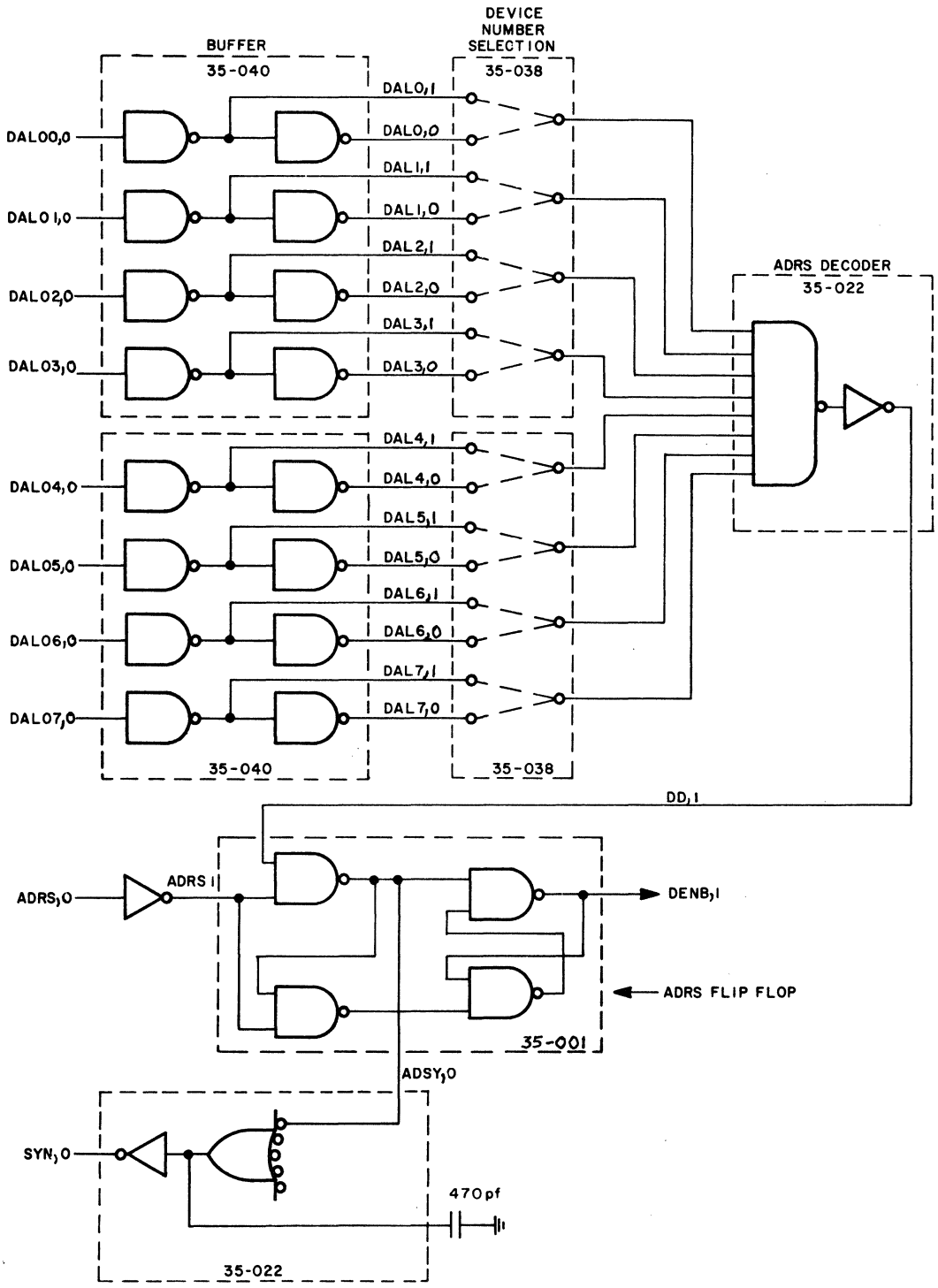


Figure 3-3. Device Addressing, Logic Diagram

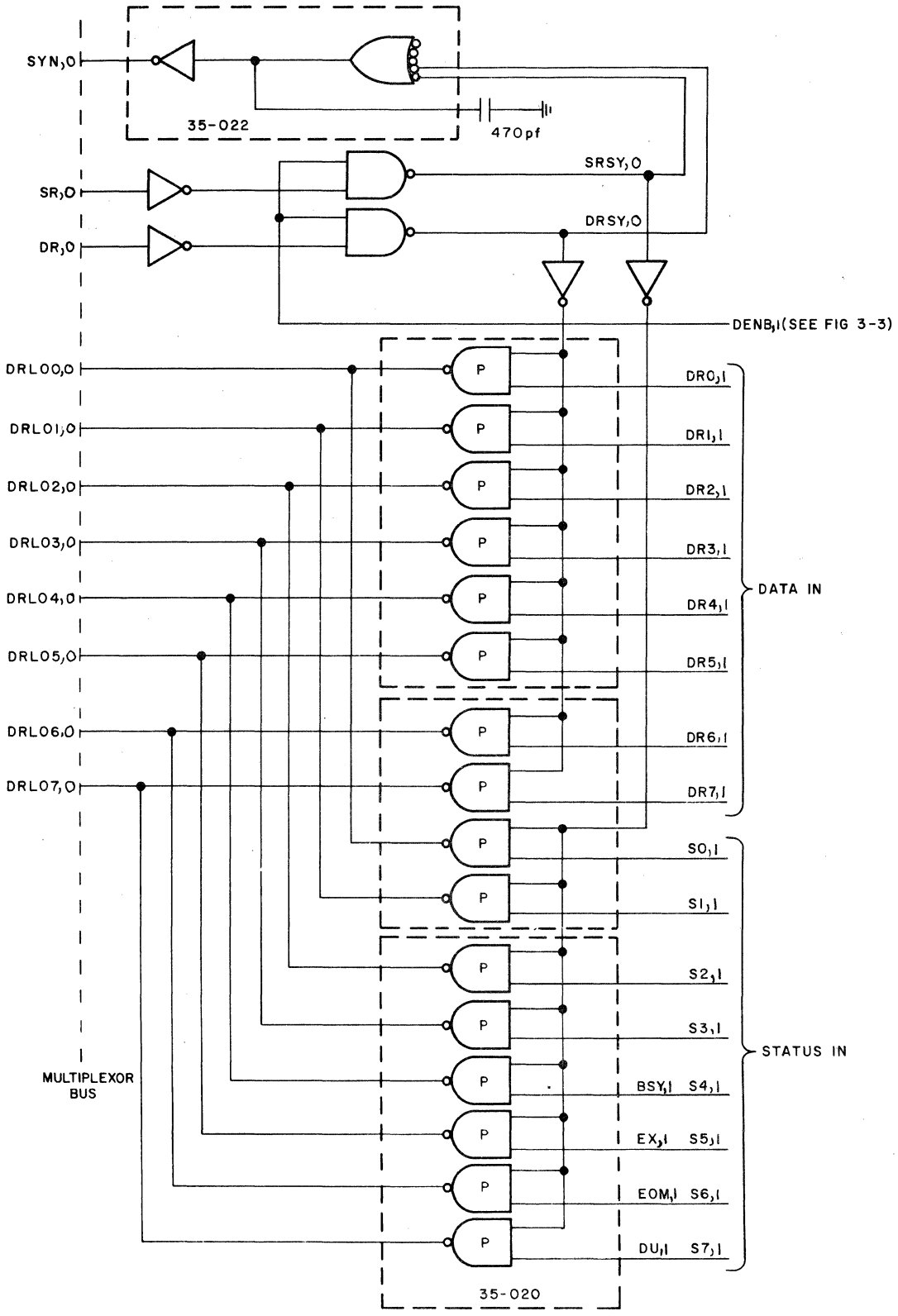


Figure 3-4. Data and Status Input, Logic Diagram

strobe the data condition into the J-K flip-flop Data Register. The **DASY,O line also** returns the sync signal to the Processor. Either Model 35-015 or 35-016 logic packs may be used. The Model 35-015 contains four J-K flip-flops and the Model 35-016 contains two J-K flip-flops.

The Command lines are shown on Figure 3-5 as being used in the toggle mode. For example, a high on bit 0 **(DAL00,1) sets a** control relay when **CMG,1 goes high. A** high on bit 1 **(DAL01,1) resets the relay.** Bits 6 and 7 are shown operating an indicator. Other pairs of bits may be used to enable/disable interrupts, etc. The toggle flip-flops may use the Model 35-001 pack which contains 4 2-input inverters.

3.6 INTERRUPT CONTROL

Figure 3-6 shows a complete general purpose interrupt and interrupt acknowledge logic system. When an interrupt is generated, the Queue flip-flop is DC set via a differentiated negative going pulse. The output from the Queue flip-flop generates an Attention signal **(ATN,O) to the Processor.** The Processor responds with an Acknowledge control line which is received by the controller as Receive acknowledge (RACK). Since the Queue flop was set prior to receiving the RACK, the Gate G1 output disables G9, holding the G9 output high. The high output from G9 stops **TACK,0 from sending the acknowledge to** the next device. Thus **RACK,1 and the G2** output generate **ATSY,0 via G3. ATSY,0** sends a SYNC back to the Processor, and also forces all inputs **(DAL00,0 through DAL07,0) to zero. This causes the device** number wired in by the address strap board to appear on the inputs of G10 through G17. Thus, the **ATSY,1 output** from G4 enables the device number onto **DRL00,0 through DRL07,0.**

Capacitor C2 removes a 30nS pulse which appears if the Queue flop is set at the same instant that **RACK,0 is received in response** to another device interrupt. This pulse

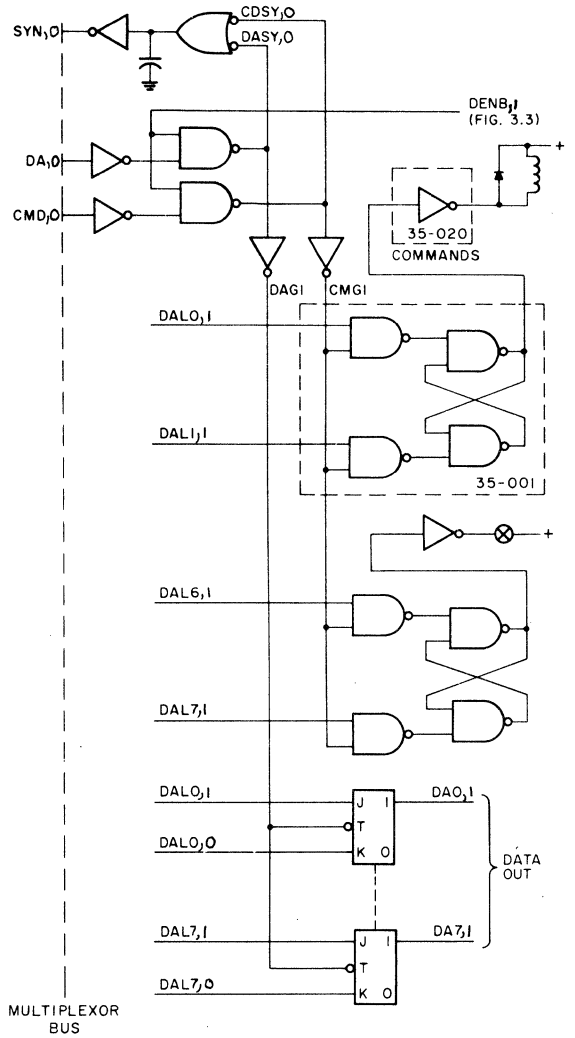


Figure 3-5. Data and Command Output, Logic Diagram

might otherwise reset the Queue flip-flop before the interrupt is serviced. The output from G4 also raises the acknowledge signal to the device. On receiving the **SYN,O, the Processor lowers RACK,1, causing** the output of G4 to drop. This, in turn, causes the Queue flip-flop to reset.

NOTE

If the interrupt has not set the Queue flip-flop, the **RACK,1 signal passes** through G2 to **TACK,0, and on to the** next device.

If **RACK,1 is high in response to an**other device, the output of G2 is low,

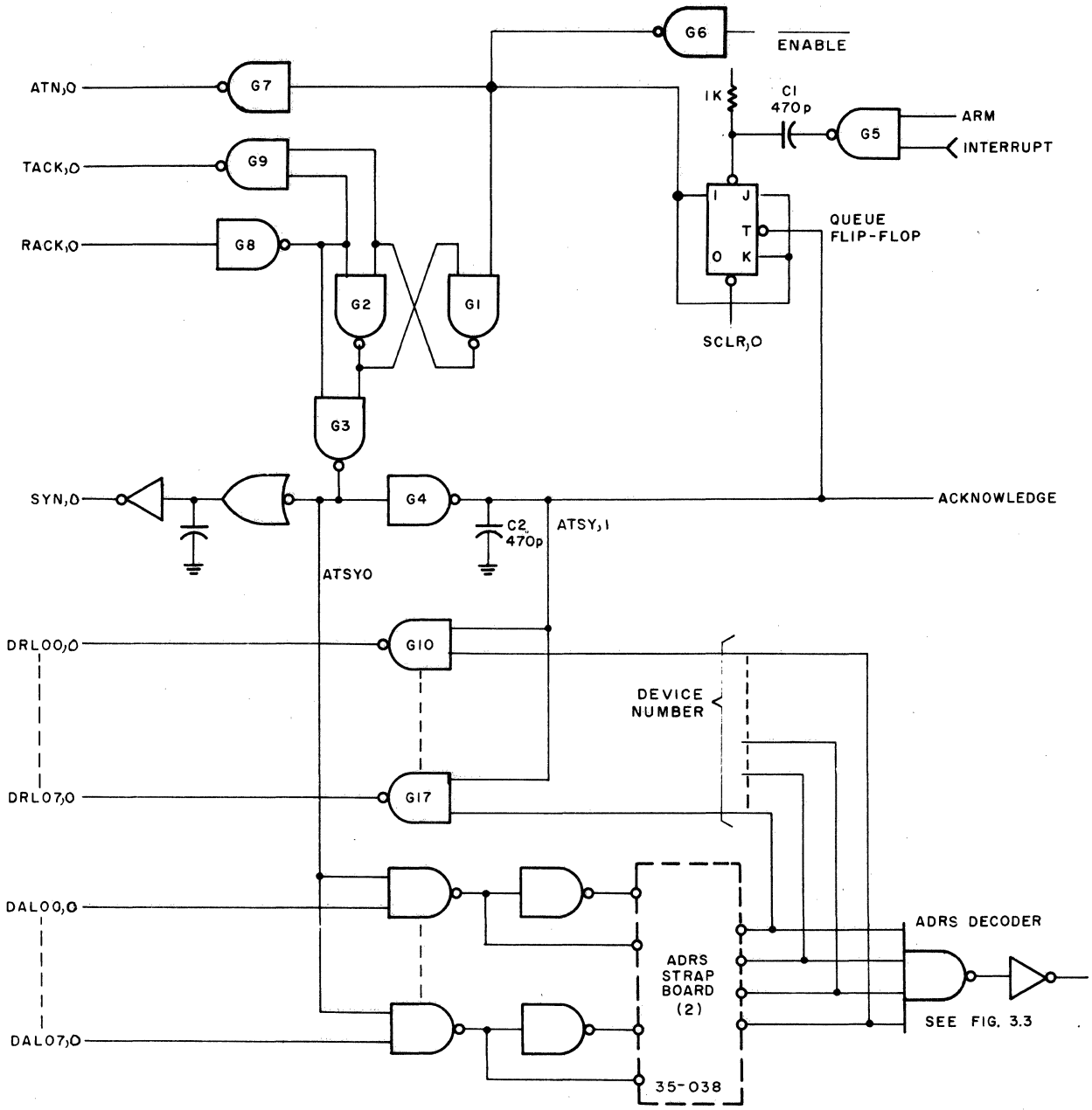


Figure 3-6. Interrupt Control, Logic Diagram

thus disabling the interrupt from affecting G1. However, the interrupt remains in the Queue flip-flop, and is serviced after completion of the previous interrupt service.

The ENABLE and ARM lines are two useful interrupt control devices which may be set/reset via Command line flip-flops. The Arm/Disarm flip-flop, if reset, disarms the device such that it cannot interrupt the computer. The ENABLE flip-flop, if reset, masks the interrupt from interrupting the Processor. However, it does not stop the Queue flip-flop from recording that an interrupt has occurred. The ENABLE flip-flop thus enables the priorities of interrupts to be programmed. If these two controls are not used, G5 and G6 can be eliminated.

As described previously in Section 3.2, the Control Line **CL05,0 from the Processor** carries the interrupt acknowledge (ACK) signal. This line breaks up into a series of short lines to form the daisy-chain priority system. The ACK signal must pass through every controller that is equipped with Interrupt Control circuits. This includes all device controllers except a few special cases.

Back Panel wiring for interrupt control is shown in the Appendix 1 of this manual on Figure A1-1. At a given position, the received ACK **(RACK,0) appears at pin #114-0** and the transmitted ACK **(TACK,0) at pin** #214-0. The daisy-chain bus is formed by a series of isolated lines which connect terminal 214-0 of a given position to terminal 114-0 of the next position (lower priority). On unequipped positions, a jumper shorts 114-0 and 214-0 of the same connector to complete the bus. Back Panels are wired with jumpers on all positions. Whenever a card chassis position is equipped with a controller, the jumper from 114-0 to 214-0 must be removed from the back panel at that position.

For controllers that occupy several positions, the jumper(s) is removed only at the position where the controller board has ATN/ACK circuits.

3.7 MULTIPLEXOR CHANNEL TIMING

Both the Input and Output Operation on the Multiplexor Channel make use of "request-response" signalling. This allows the system to run at its maximum speed whenever possible but permits a graceful slowdown if the length of the bus or the characteristics of a particular device controller require signals of longer duration. Device controller designs should keep Multiplexor Channel usage as fast as possible consistent with practical circuit margins. This will give the greatest computer throughput when a system is configured with a number of peripheral devices.

Typical operations are shown on Figure 3-7 for Input and Output. On the Output Operation, the Processor places a signal on the Data Available Lines followed by an appropriate Control Line signal. This stagger (T1) will vary depending upon which model Processor is in use, but it is guaranteed to be at least 100 ns. When the device controller has received the output byte, the SYN signal is returned to the Processor which then terminates the Control Line signal. Realizing that T5 is 100 ns minimum, the SYN delay T2 should be only long enough to guarantee proper reception of the output byte. The Control Line/DAL removal time (T3) is important where single-rail to double-rail operation is used - e.g. the ADRS flip-flop on Figure 3-3. A minimum of 100 ns is guaranteed for T3. For SYN generation as per Figures 3-3 and 3-4, the Control Line signal is DC coupled through the gates to form the SYN signal. The SYN removal time (T4) will be the delay through 4 DTL gates.

It should be emphasized that the times shown on Figure 3-7 are defined for signals on the Multiplexor Channel. Within a given controller, one signal may flow through more gates than another signal and these delays must be considered.

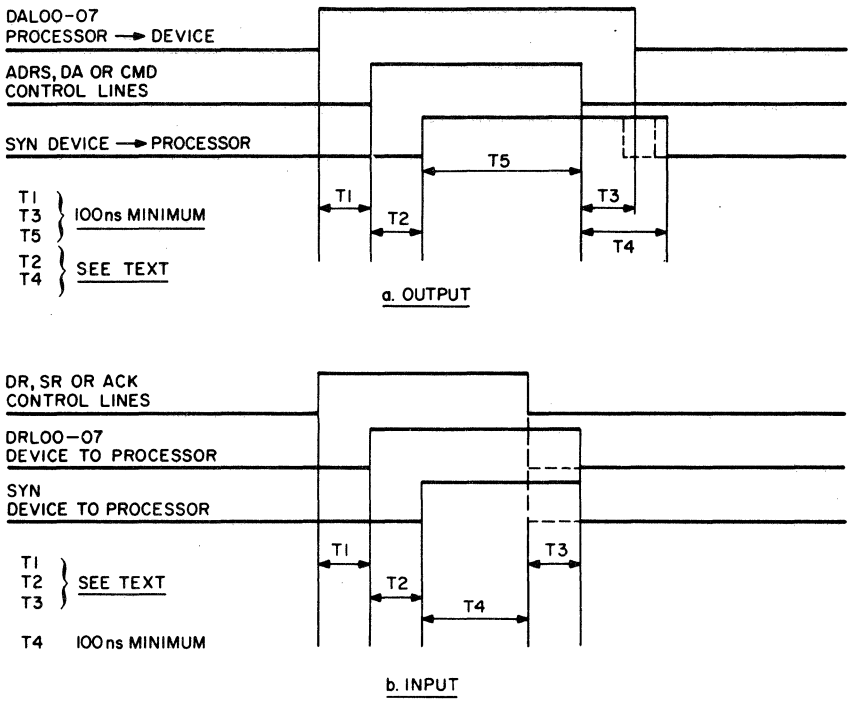


Figure 3-7. Multiplexor Channel Timing

For the Input Operation, the Processor places a signal on a Control Line. The device controller should gate signals to the DRL's as soon as possible to keep T1 minimum. The SYN delay (T2) must guarantee that the input byte is on the DRL's considering the slowest byte gates and the fastest SYN gates. The Processor will remove the Control Line signal when SYN is received with a minimum delay (T4) of 100 ns. With SYN and the byte gate DC coupled to the Control Line, the removal delay (T3) will be the sum of the corresponding gate delays.

When the Control Signal is ACK, the Delay T1 will include the cumulative G8/G9 delays (see Figure 3-6) for all the controllers between the responding controller and the Processor. This will be less than the Processor Time Out even with the maximum limit of 256 controllers.

CHAPTER 4

STANDARD I/O BOARD

4.1 INTRODUCTION

This Chapter describes a basic bus communications board which is available from GE-PAC 30 to facilitate customer controller design by the customer. The Standard I/O Board is essentially a mother-board which contains the bus communications and interrupt circuits used on most all device controllers. These circuits could be implemented using the daughter-board approach described in Chapter 3. However, several daughter-board positions and some design time is saved by using a standard circuit with the integrated circuits mounted directly on the mother-board.

4.2 GENERAL DESCRIPTION

The Standard I/O Board has its IC logic mounted directly on the mother-board in a field near the lower connector (connector-0). This positioning keeps the signal paths to and from the bus as short as possible. See Figures 4-1 and 4-2. The remainder of the board provides 28 connectors for standard GE-PAC 30 daughter-boards which are described in the Logic Module Handbook, Publication Number 29-005. The same Standard I/O Board may be used with either the Multiplexor Channel or the Selector Channel.

4.3 COMMUNICATIONS LOGIC

The Address and SYN circuits on Figure 4-3 perform the same functions as those shown on Figure 3-3. The Address straps for device number selection are wired in a field at location 20 on the mother-board. The lettered pins also provide the true and false states for an 8-bit byte which can be wired to data and/or command registers in the wire wrap portion of the board.

Data and Status bytes returned to the Processor originate in the customer designed wire wrap portion of the board and share the bus driver circuits as shown in Figure 4-4. A group of pins the E field provide this connection.

The Interrupt Control circuit on Figure 4-5 is logically identical to that on Figure 3-6. The Data Output and Command Output circuit shown on Figure 4-6 is logically identical to that on Figure 3-5.

4.4 WIRE WRAP FACILITIES

The Standard I/O Board provides 28 daughter-board locations, 23 mother-board pull up resistors (1K to +5V), space for 18 components (resistor, capacitors or diodes) mounted between pull up resistors, space for 24 Reed relays form 1A, space for 8 PC Trim potentiometers, and 16 test points on the outer end of the mother-board. Cable connectors to external devices or circuits are mounted in the daughter-board locations along the outer edge of the board in the same manner as with standard GE-PAC 30 controllers.

Wire wrap pins for the upper connector pins (connector -1) are available. These are used when more than one mother-board is required for the controller. The back panel "stitch-pattern" shown in Appendix 1 on Figure A1-2 provides straps between adjacent boards. A multi-board controller will usually require only one Standard I/O Board (35-104) to provide the communications circuits. Wire wrap mother-boards (35-050) with 40 daughter-board positions each are used for the additional logic.

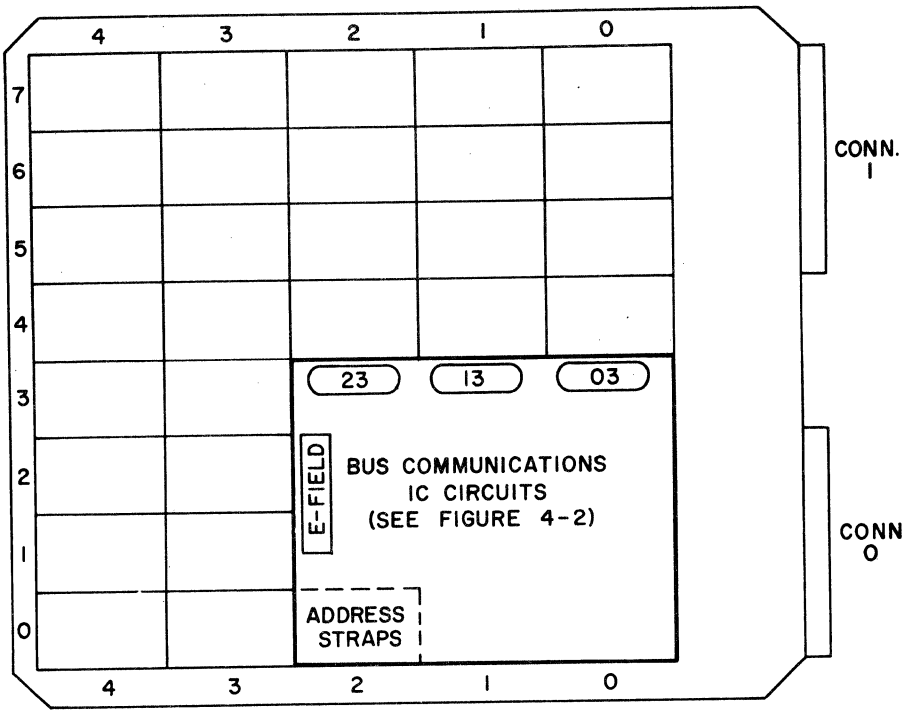


Figure 4-1. Standard I/O Board Layout (35-104)

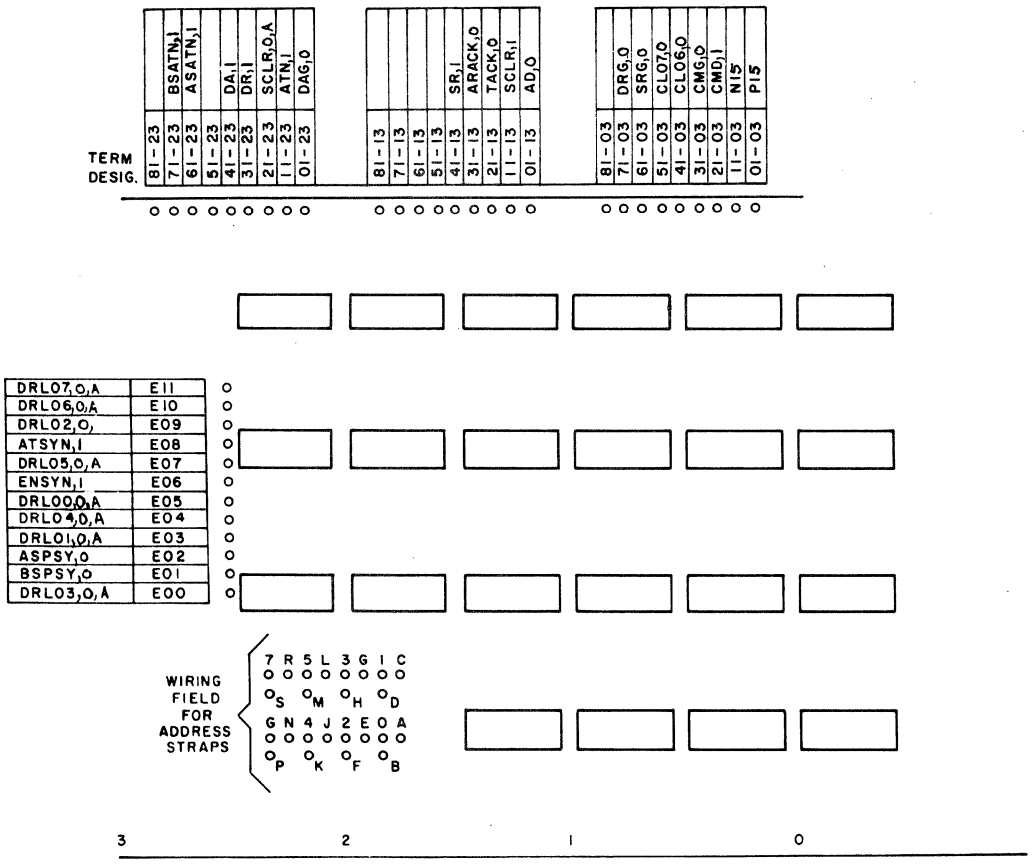


Figure 4-2. Standard I/O Board Field Layout

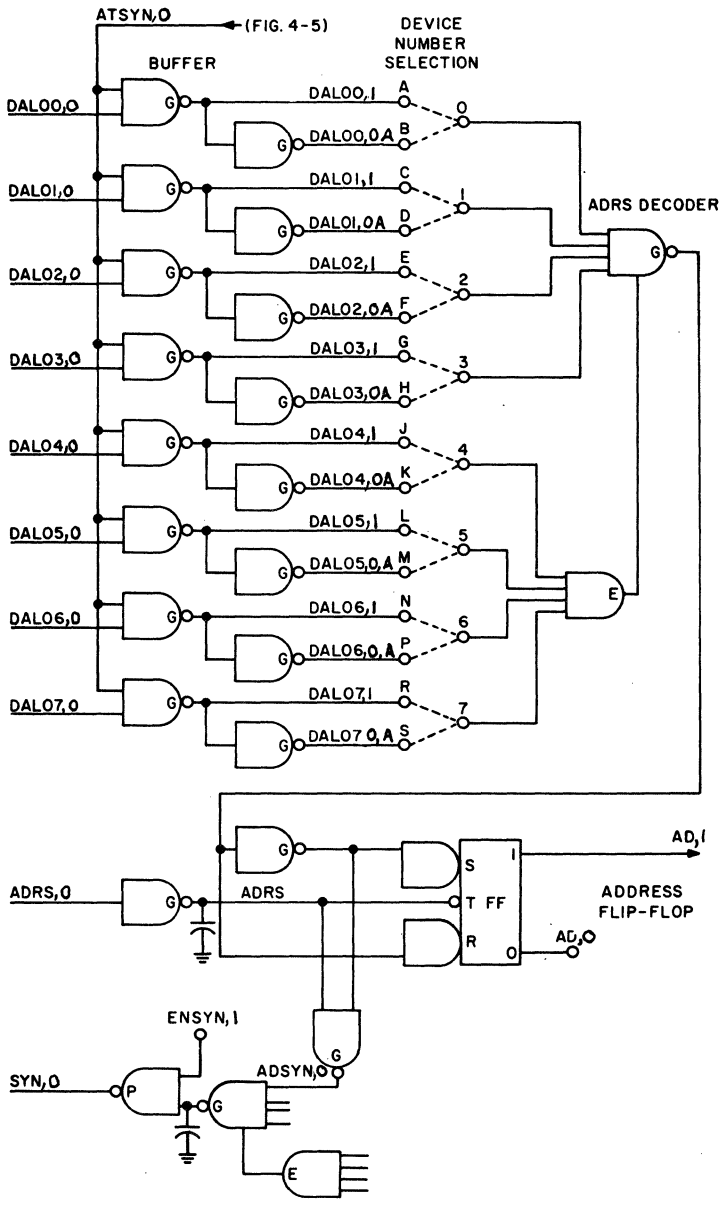


Figure 4-3. Device Addressing, Logic Diagram

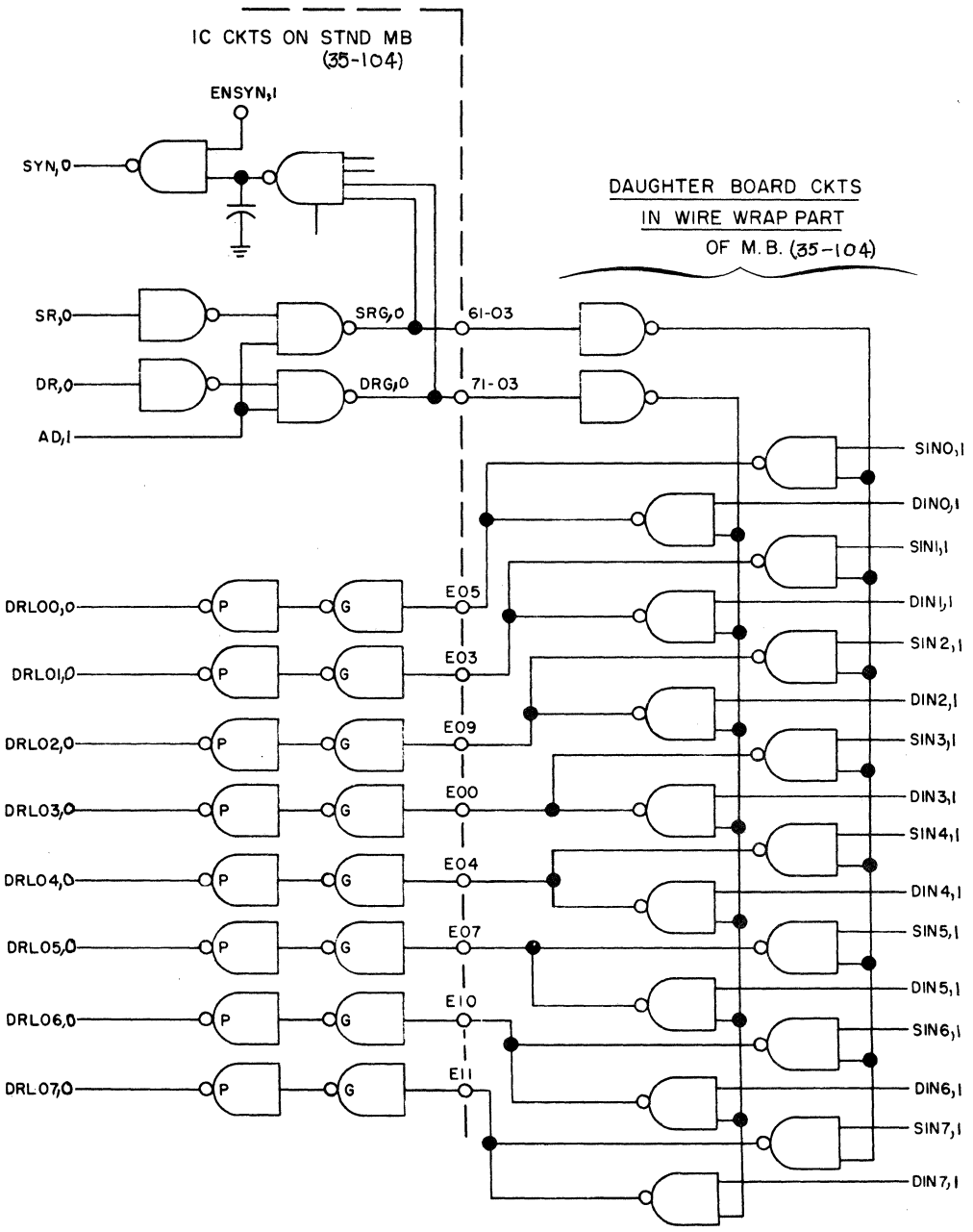


Figure 4-4. Data and Status Request, Logic Diagram

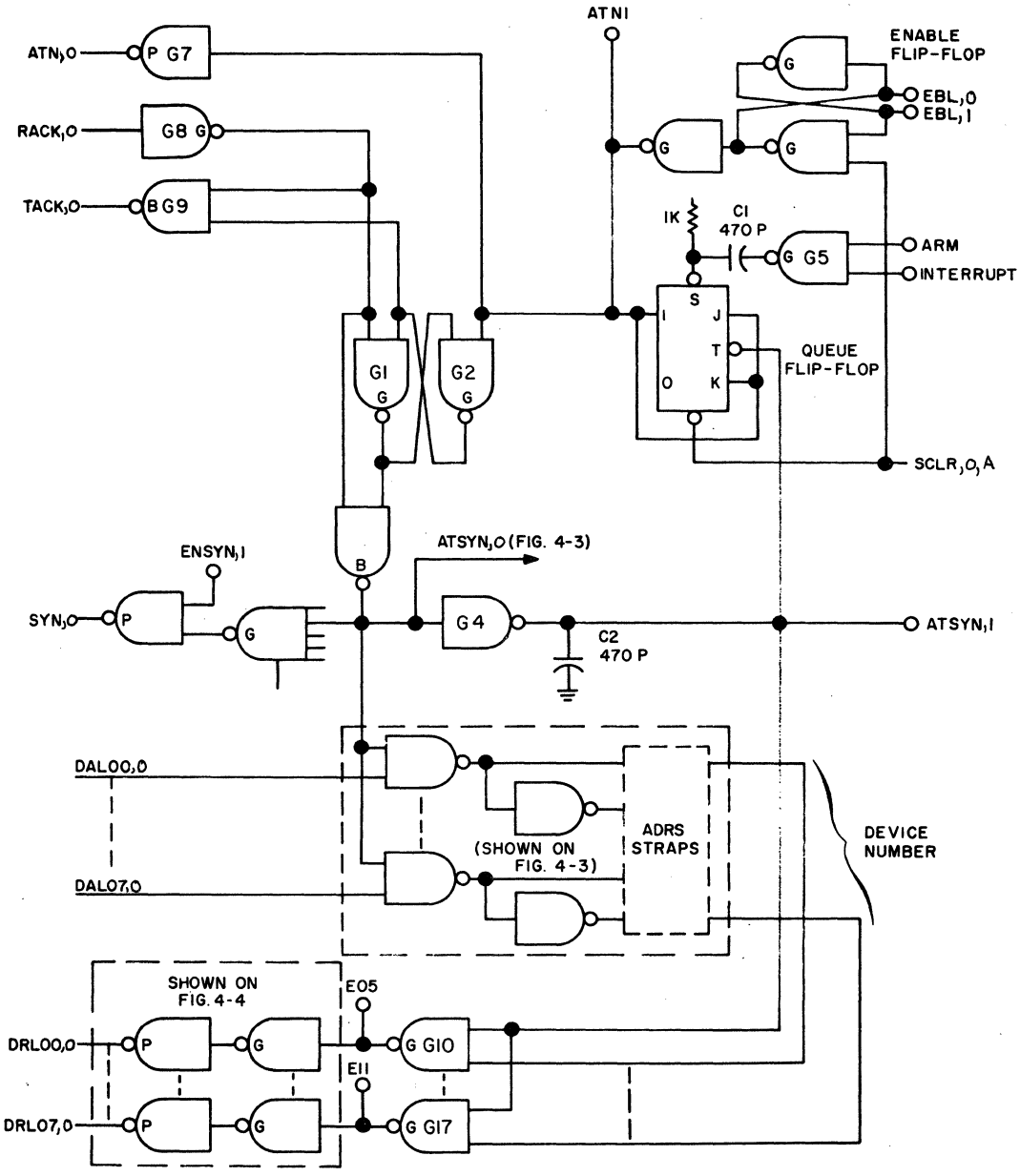
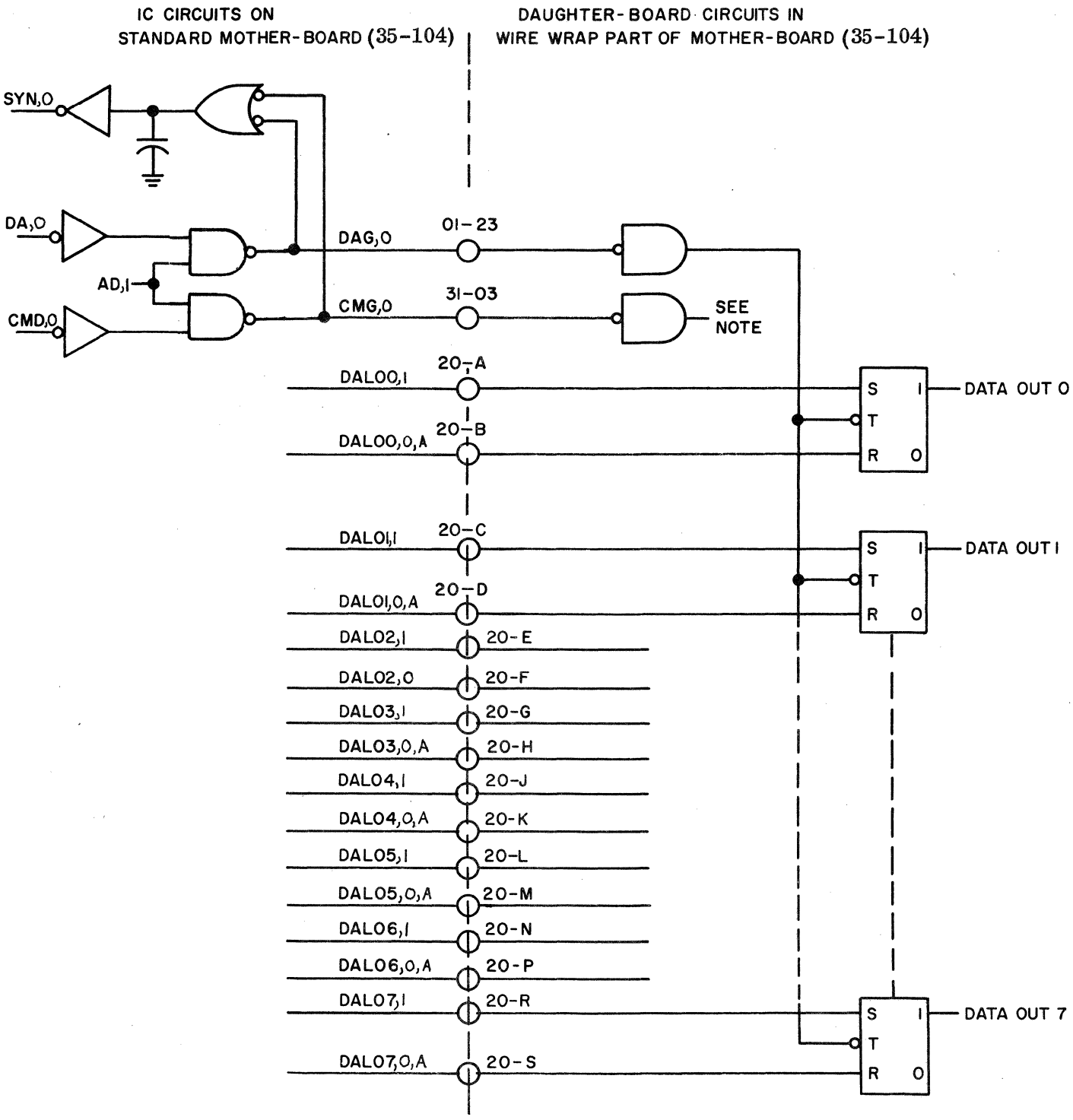


Figure 4-5. Interrupt Control, Logic Diagram



NOTE:

COMMANDS MAY BE IMPLEMENTED IN A MANNER SIMILAR TO THE DATA OUTPUT, OR COMMANDS MAY BE SIMPLY ANDED WITH THE DAL LINE TO PROVIDE A COMMAND PULSE.

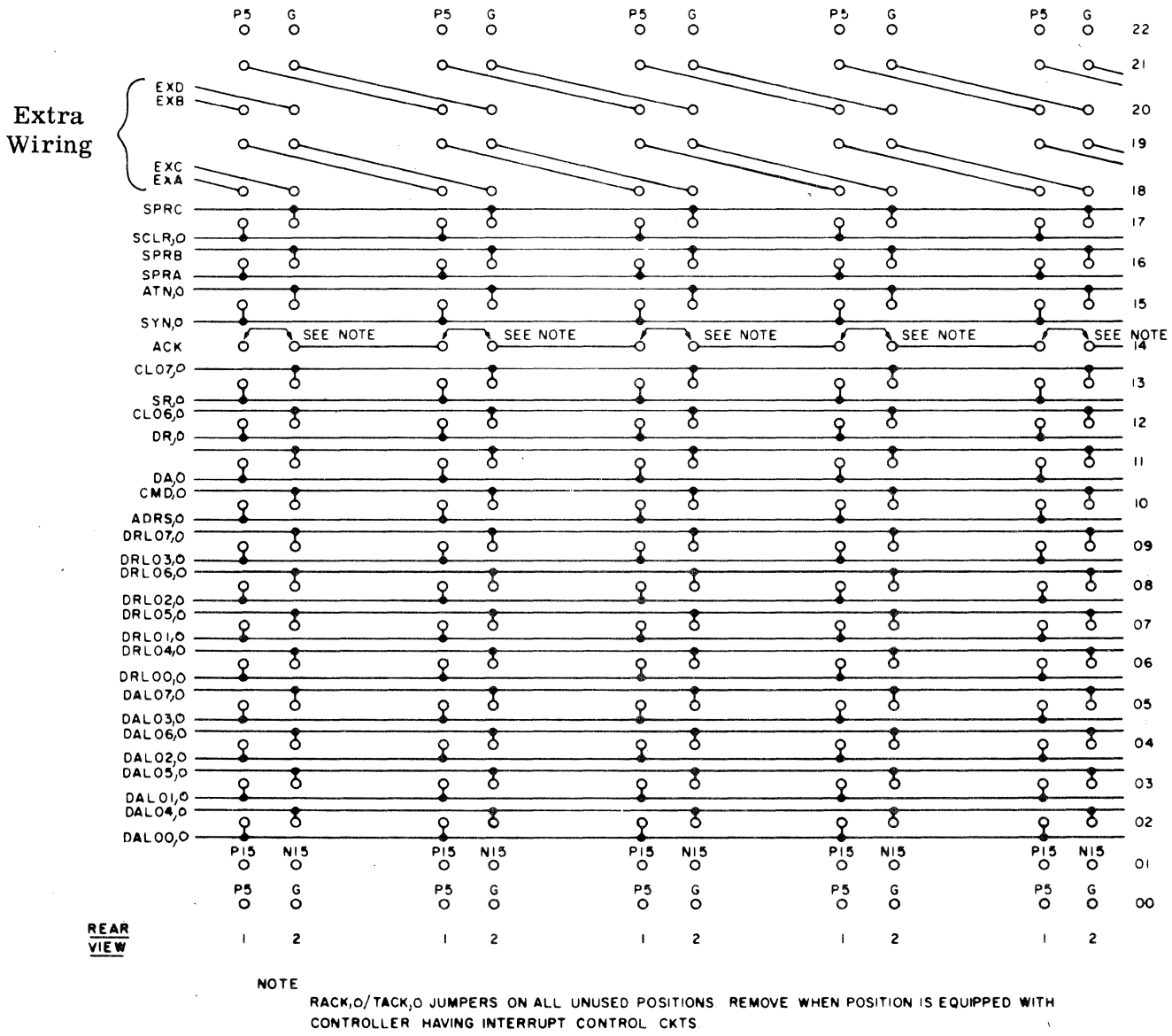
Figure 4-6. Data Output and Command Output, Logic Diagram

APPENDIX 1

MULTIPLEXOR AND SELECTOR CHANNEL WIRING DATA

Wiring for the Multiplexor Channel and the Selector Channel is identical. The bus connections are via the bottom connector (Field 0) and are shown on Figure A1-1. Note that the top four rows of pins are stitched for use in communicating between mother-boards in the same device controller. Signal designations are shown on the right side of Figure A1-1.

The top connector (Field 1) is stitched as shown on Figure A1-2. The Field 1 connector is used solely for communication between mother-boards.



NOTE

RACK,0/TACK,0 JUMPERS ON ALL UNUSED POSITIONS. REMOVE WHEN POSITION IS EQUIPPED WITH CONTROLLER HAVING INTERRUPT CONTROL CKTS.

Figure A1-1. I/O Back Panel - Connector "0" (Bottom)

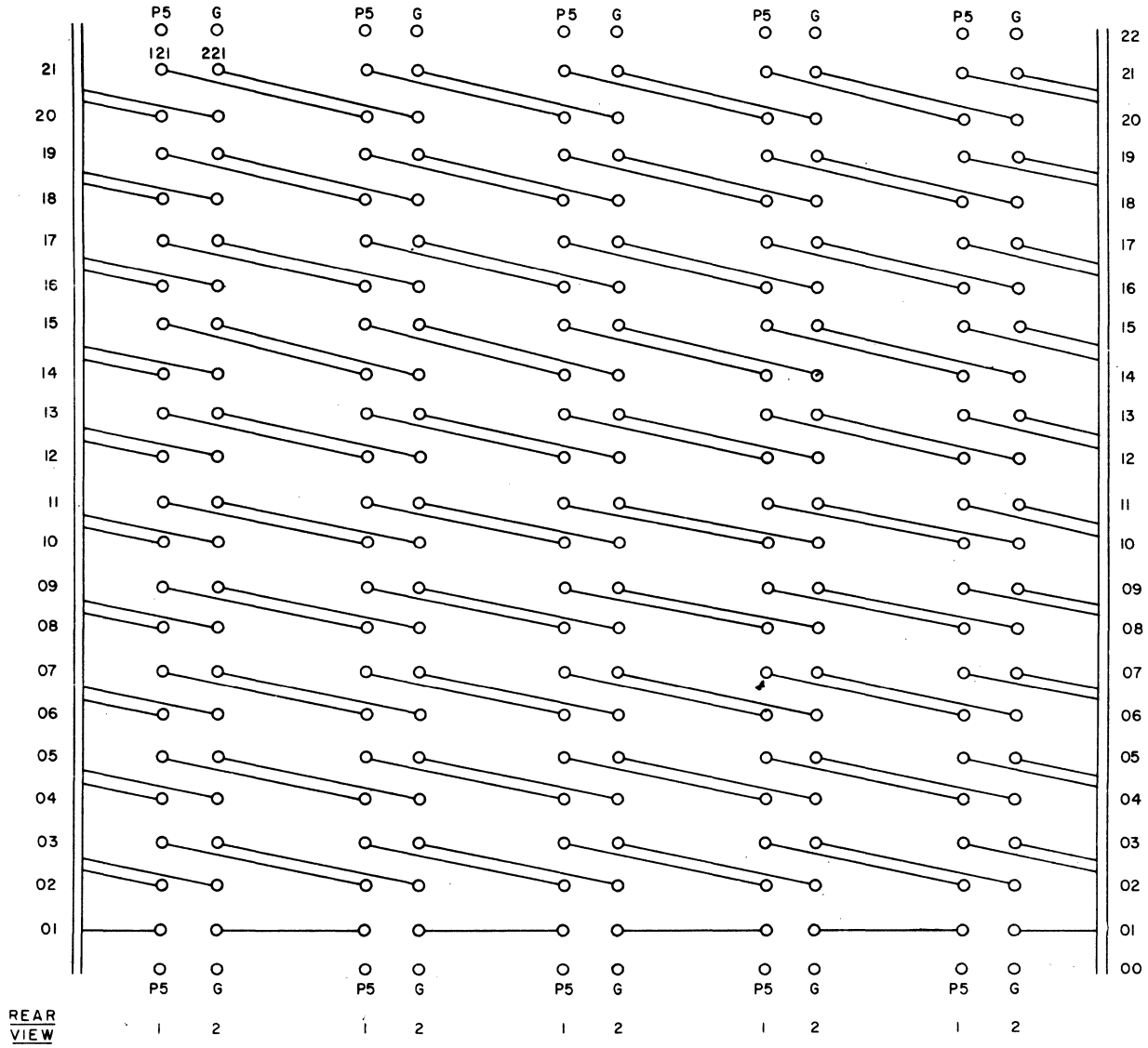


Figure A1-2. I/O Back Panel - Connector "1" (Top)

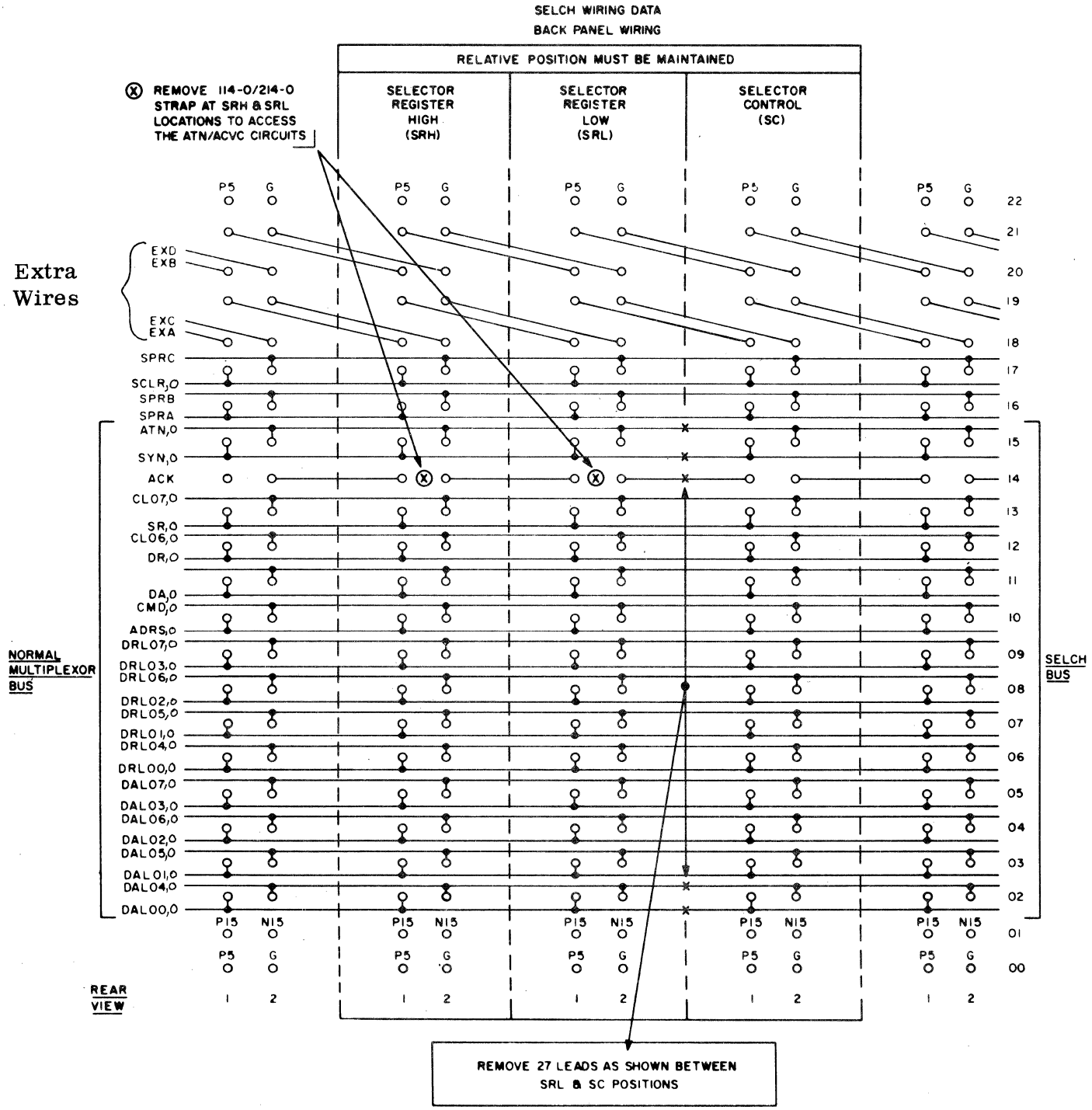


Figure A1-3. Selector Channel Wiring

APPENDIX 2

DMAC WIRING DATA

BACK PANEL WIRING

Conn Pos Moth. Bd. Vert. Pos		Relative Position Must Be Maintained DMACH Horiz Pos			DMACL Horiz Pos		
		0	1	2	0	1	2
FIELD 1	22		+5V				
	21		RDRA,O	RARA,O			
	20					RDRA,O	RARA,O
	19		LCDR,I	LCAR,I			
	18					LCDR,I	LCAR,I
	17		UCDR,I	UCAR,I			
	16					UCDR,I	UCAR,I
	15		UDR,I	UAR,I			
	14					UDR,I	UAR,I
	13		STDT,O	LDR,I			
	12					STDT,O	LDR,I
	11		WT,I	WT,O			
	10					WT,I	WT,O
	09		URS,I	URS,O			
	08					URS,I	URS,O
	07		DA,I	EA,I			
	06					DA,I	EA,I
	05		G0,I	G0,O			
	04					G0,I	G0,O
	03		AR07,I	RIW,I			
	02					AR07,I	RIW,I
	01						
	00		+5 V				
FIELD 0	22						
	21						
	20						
	19						
	18						
	17						
	16						
	15						
	14						
	13						
	12						
	11			NO CONNECTIONS			
	10						
	09						
	08						
	07						
	06						
	05						
	04						
	03						
	02						
	01						

CABLE INFORMATION

There is one cable from Field 40 of the DMACH mother-board and Field 40 of the DMACL mother-board to the memory bus.

There are cables from Field 47 of each board to the device controller (C Bus). The following tables provide wiring information for these cables.

DMACH BOARD

Field 47 C Bus	Pin	Field40 M Bus
GND	00	GND
C00	10	D00
C01	20	D01
C02	30	D02
C03	40	D03
C04	50	D04
C05	60	D05
C06	70	D06
C07	11	D07
CBSY	21	RMA
CRD	31	TMA
CRIW	41	REQ,0
CWT	51	EA,0
CARY	61	DA,0
	71	
	81	

DMACL BOARD

Field 47 C bus	Pin	Field 40 M Bus
GND	00	GND
C08	10	D08
C09	20	D09
C10	30	D10
C11	40	D11
C12	50	D12
C13	60	D13
C14	70	D14
C15	11	D15
CSYN	21	
CIDR	31	
CIAR	41	DSYN
CODR	51	
COAR	61	WT,0
CS14	71	ST,0
	81	

READER COMMENTS

The General Electric Company solicits your comments on publications covering Process Computer equipment. Please explain any "no" responses in the COMMENTS section. Your comments and suggestions become the property of the General Electric Company.

- Name of Manual: _____
- What is your computer application: _____

  _____
- How is this publication used:

  Familiarization ☐ Reference ☐

  Training ☐ Maintenance ☐

  Other (Explain) _____

	YES	NO
• Does this publication meet your requirements	☐	☐
• Is the material:	☐	☐
1) Presented in clear text	☐	☐
2) Conveniently organized	☐	☐
3) Adequately detailed	☐	☐
4) Adequately illustrated	☐	☐
5) Presented at appropriate technical level	☐	☐

- Please provide specific text references (page number, line, etc.) with your comments.

NAME _____ DATE _____

TITLE _____

COMPANY NAME _____

AND ADDRESS _____

COMMENTS:

No postage necessary if mailed in the U. S. A.

Staple

Communications concerning Technical Publications should be directed to:

Manager, Technical Publications
GE Process Computer Department
2255 West Desert Cove Road
Phoenix, Arizona 85029

Fold Fold

FIRST CLASS
Permit No. 4091
Phoenix, Arizona

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY...

GENERAL ELECTRIC COMPANY
PROCESS COMPUTER DEPARTMENT
2255 West Desert Cove Road
Phoenix, Arizona 85029

Attention: Technical Publications

Cut Along Line

Fold Fold

Additional Comments:

Progress Is Our Most Important Product

GENERAL ELECTRIC

PROCESS COMPUTER DEPARTMENT

PHOENIX, ARIZONA