CIP/2000 COMPUTER INTERFACE MANUAL

February 1970

Publication 3 501 0108

Price $ 3.00

Products Division | Cincinnati Milacron Inc.
Cincinnati, Ohio 45209

TABLE OF CONTENTS

1.	**GENERAL INFORMATION**	1-1
1.1	Introduction	1-1
1.2	I/O System Organization	1-1
1.3	Serial I/O Interface	1-3
1.4	Byte I/O Interface	1-3
1.5	Program-Controlled I/O	1-3
1.6	Concurrent I/O	1-3
1.7	External Priority Interrupts	1-3
1.8	Direct Memory Interface	1-5
1.9	Mechanical Considerations	1-5
1.10	Mainframe Chassis	1-5
1.11	Expansion Chassis	1-6
2.	**SERIAL I/O INTERFACE**	2-1
2.1	Introduction	2-1
2.2	Character Assembly and Disassembly	2-1
2.3	Microcommands and Macro Instructions	2-2
2.4	Teletype Interface Connection	2-3
3.	**BYTE I/O INTERFACES**	3-1
3.1	Introduction	3-1
3.2	Byte I/O Bus	3-1
3.3	Input Data Lines	3-1
3.4	Output Data Lines	3-1
3.5	Input Control Lines	3-3
3.6	Output Control Lines	3-3
3.7	Internal Byte I/O Control Lines (Mainframe Chassis)	3-4
3.8	External Byte I/O Control Lines (I/O Cable)	3-5
3.9	Spare Lines	3-5
3.10	Byte I/O Fundamentals	3-5
3.11	Device Addressing	3-5
3.12	Device Orders	3-5
3.13	Status Bytes	3-6
3.14	Byte I/O Operations and Timing	3-6
3.15	Program-Controlled I/O Operations	3-6
3.16	Data Output Timing	3-6
3.17	Typical Data Output Logic	3-6
3.18	Function Output Timing	3-9
3.19	Typical Function Output Logic	3-9
3.20	Data Input Timing	3-9
3.21	Typical Data Input Logic	3-11
3.22	Status Input Timing	3-11
3.23	Typical Status Input Logic	3-11
3.24	Typical Program-Controlled Transfer Sequence	3-11
3.25	Concurrent I/O Operation	3-12
3.26	Concurrent I/O Timing	3-14
3.27	Typical Concurrent I/O Logic	3-14

TABLE OF CONTENTS (Cont.)

3.28	External Interrupt Operation	3-16
3.29	Priority Determination	3-16
3.30	Interrupt Requests	3-16
3.31	Interrupt Sequence and Timing	3-16
4.	**DIRECT MEMORY ACCESS INTERFACE**	4-1
4.1	Introduction	4-1
4.2	DMA Selector Channels	4-1
4.3	DMA External Interface Lines	4-1
4.4	Data Output Lines	4-1
4.5	Data Input Lines	4-2
4.6	Device Status Lines	4-2
4.7	Control Lines	4-2
4.8	DMA Control	4-3
4.9	DMA Channel Operations	4-3
4.10	Simple Block Transfer	4-3
4.11	Continuous Block Transfer	4-6
4.12	Block Transfer with Linked Buffers	4-6
4.13	Interface to DMA Channel	4-6

APPENDIXES

A.	**INTERFACE CONNECTOR SIGNAL LISTS**	A-1
B.	**I/O INTERFACE SIGNAL GLOSSARY**	B-1
C.	**TELETYPE MODIFICATION PROCEDURES**	C-1

ILLUSTRATIONS

1-1	**Typical CIP/2000-Series I/O Configuration**	1-2
1-2	Mainframe Chassis Interface Connectors	1-5
1-3	Interface Circuit Board Dimensional Data	1-6
1-4	I/O Line Driver and Receiver Interface Option	1-7
1-5	I/O Expansion Cabling	1-8
2-1	Serial I/O Interface Circuit	2-2
2-2	Serial I/O Timing	2-3
3-1	Byte I/O Interface, Simplified Logic Diagram	3-2
3-2	Relationship of Control Signals CPH1 and CPH2/	3-4
3-3	Data or Function Output Timing	3-8
3-4	Data Output Logic	3-8
3-5	Function Output Storage and Decoding Logic	3-10
3-6	Data or Status Input Timing	3-10
3-7	Data Input Logic	3-11
3-8	Status Input Logic	3-12
3-9	Typical Program-Controlled Transfer Sequence	3-13
3-10	Concurrent I/O Timing	3-14
3-11	Concurrent I/O Logic	3-15
3-12	Typical Priority Wiring Scheme	3-17
3-13	External Interrupt Timing	3-17
4-1	Relationship of Processor, Memory, and DMA Channels	4-2
4-2	DMA Channel Control Word Memory Locations	4-4
4-3	DMA Channel Block Transfer Timing	4-5
4-4	DMA Status Byte Format	4-6
4-5	DMA Bus Cabling for Multiple Controllers	4-6

TABLES

1-1	I/O Interface Options	1-4
1-2	I/O Cable Specifications	1-7
3-1	I/O Control States	3-4
3-2	Standard I/O Device Addresses	3-6
3-3	Standard I/O Device Orders	3-7
3-4	I/O Device Status Byte Definition	3-7
4-1	**DMA Channel Instructions**	4-5

1. GENERAL INFORMATION

1.1 INTRODUCTION

This manual describes the three primary input/output interfaces of the CIP/2000-series computers and provides guidelines for connecting external equipment to the interfaces. The information in this manual should not be considered the final authority for the I/O interface configuration of any particular machine. The final authority in all cases is the latest revision of all applicable engineering drawings and specifications.

Input/output operations for both the CIP/2000 and 2100 Computers are described in the manual. In both computers, all I/O operations are performed under control of microcommands. In the CIP/2000 Computer the sequence of microcommands (microprogram) can be designed for a wide range of I/O operations. Microprogramming in the 2100 Computer provides a standard set of I/O macro instructions for performing program-controlled and concurrent I/O operations. These standard I/O operations, which may be used with any CIP/2000 configuration, are described in addition to the elementary microlevel functions.

The information in the manual is arranged in four sections and three appendixes: One section provides general information on the I/O interfaces, and three sections provide detailed information on each of the three I/O interfaces. The three appendixes contain reference material in the form of I/O connector signal lists, a signal glossary, and Teletype modification instructions.

1.2 I/O SYSTEM ORGANIZATION

A block diagram of a typical 2000-series computer system is illustrated in figure 1-1. The three primary I/O interfaces shown in figure 1-1 are the serial I/O interface between the Teletype and the processor, the parallel byte I/O interface (both internal and external), and the I/O interface to the direct memory access (DMA) channels.

These three interfaces provide the system designer with the flexibility to structure efficient I/O systems for a wide range of applications. The serial I/O interface, although normally used with a Teletype, can be used for other bit-serial devices as well. The byte I/O interface can be used by controller circuit boards that plug into the mainframe chassis, or it can be extended to an expansion chassis through the I/O Line Driver and Receiver circuit board option. In the configuration shown in figure 1-1, an expansion chassis is used for additional controller and optional Priority Interrupt circuit boards. The Priority Interrupt boards can also be located in the mainframe chassis. The DMA interface provides the

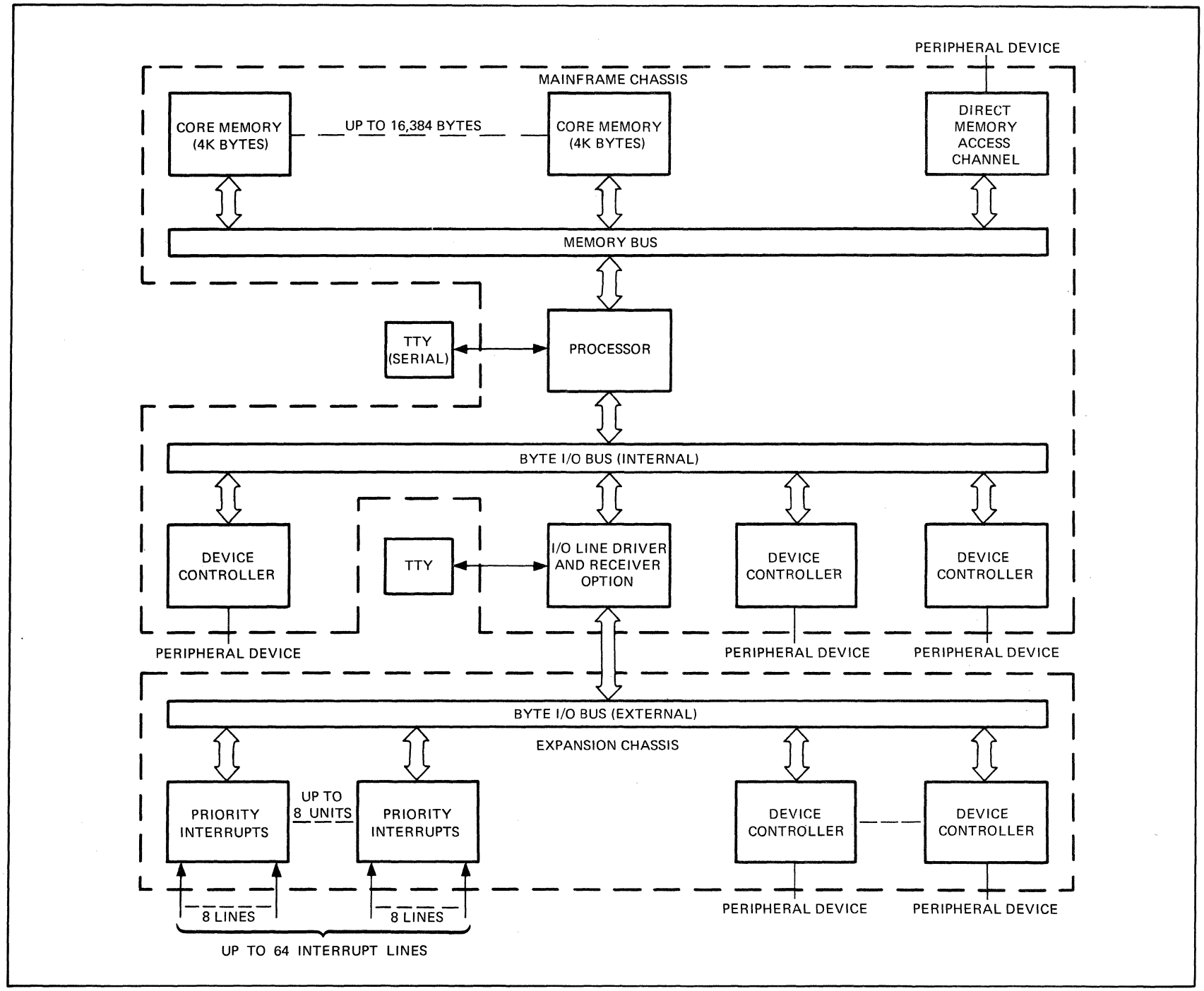


Figure 1-1. Typical CIP/2000-Series I/O Configuration

facility for external I/O devices to communicate directly with core memory through optional DMA channels located in the mainframe chassis.

1.3 SERIAL I/O INTERFACE

The serial I/O interface is designed for communicating with a full duplex Teletype. Character assembly and disassembly, including all timing and synchronization, are performed at the microprogram level in both the 2000 and 2100 Computers. The 2100 Computer has two macro instructions, Input Byte Serially (IBS) and Output Byte Serially (OBS), for communicating with the Teletype. The microprogram for these instructions is normally set for 110-baud, 11-element characters, such as are used by ASR 33 or ASR 35 Teletypes. However, this timing can be easily altered by a simple change in firmware to handle other devices. Descriptions of the timing, logic, and instructions for this interface are given in section 2.

1.4 BYTE I/O INTERFACE

The byte I/O interface provides the facility for transferring data bytes over a partyline I/O bus under microprogram control. The 2100 Computer has programmed I/O and concurrent I/O transfer capability, along with a priority interrupt system.

Data transfers through the byte I/O interface are basically two-phase operations. During the first phase a control byte is placed on the byte I/O bus before the actual transfer of data. The control byte contains a device number specifying the address of one of the I/O devices connected to the bus and a device order signifying the type of operation to be performed during the transfer (data transfer, status/function transfer, and so on). All devices on the bus examine the device number, but only the device with a matching address accepts the control byte and logically connects itself to the bus for the subsequent data byte transfer. During the second phase of the byte I/O operation a single byte is transferred to or from the I/O device. After each byte transfer the device disconnects itself from the bus.

As shown in figure 1-1, I/O devices can be connected to the byte I/O interface at the internal byte I/O bus inside the mainframe chassis or to the external byte I/O bus outside the mainframe chassis in a separate expansion chassis. The external byte I/O bus is an extension of the internal bus brought out of the mainframe chassis through an optional plug-in I/O Line Driver and Receiver board. The I/O Line Driver and Receiver board is one of several optional I/O interface circuit boards available **for use with the 2000-series computers. All of the boards are made to plug into the mainframe chassis, but they can be used as well in an expansion chassis. A list of the optional I/O interface boards available is given in table 1-1.**

Although the external byte I/O bus is an extension of the internal bus, there are a few major differences between them. These differences are as follows:

a. All nine output lines of the external byte I/O interface are buffered by nine type 944 DTL drivers for driving terminated lines.

b. Outputs from the three control flip-flops of the I/O control register are directly available at the internal interface, but are decoded and made available at the external interface on seven individual lines buffered by type 944 DTL drivers.

c. The internal computer clock is used by interface boards connected to the internal bus, but a half-frequency, 50 percent duty cycle clock is used by boards connected to the external bus.

1.5 Program-Controlled I/O

The CIP/2100 Computer has six instructions for transferring data to and from devices on the byte I/O bus under programmed control. These instructions permit transfers between the device and the A-register, B-register, or memory. The type of transfers that can be performed are function output, data output, status input, or data input as determined by the device order in the control byte of the I/O instruction.

1.6 Concurrent I/O

The concurrent I/O feature provides the capability for block transfers between I/O devices connected to the byte I/O interface and memory at a maximum rate of 50,000 bytes per second. Once started, the transfers are fully automatic and proceed without program intervention. Concurrent I/O operations take priority over instruction execution and force a break in the execution of long instructions such as multiply, divide, and shifts to ensure that concurrent I/O servicing delays are not excessive. Concurrent I/O operations make use of pairs of two-byte address control words stored in core memory. One pair of address words is used by each device. The control words, which contain the address of the current byte being transferred and the address of the last byte in the block, are initially set by the software program and thereafter are manipulated automatically by firmware for each byte transferred.

1.7 External Priority Interrupts

The external interrupt system of the 2000-series computers operates through the byte I/O interface (both internal and external). Interrupts can originate from device controllers or from optional Priority Interrupt interface boards connected to the byte I/O bus. Each Priority Interrupt interface board provides control of

Table 1-1. I/O Interface Options

Model No.	Description
2700	Input/Output Line Driver and Receiver Board. Expands internal byte I/O bus to an external bus allowing integration of up to 10 peripheral interfaces under program control or concurrent data transfer with interrupt.
2701	Parallel Teletype Controller. Assembles and disassembles serial information to and from the Teletype for parallel transfer to and from the computer under program control or concurrent block transfer.
2703	Priority Interrupt Board. Allows interfacing of 8 external interrupt lines with expansion capability to 64 lines using 8 boards. Priority, timing, interrupt storage, address generator, and independent enabling or disabling of each interrupt are provided.
2704	Direct Memory Access Selector Channel. Provides for transfer of eight-bit bytes directly between external devices and core memory. Transfer is in block form with continuous cycling of one or more buffers. An internal interrupt indicates end of transfer to the processor program.
2705	Input/Output Expander. Full Word (32 Bits). Expands the byte I/O bus into multiple (4) byte I/O under processor control. The outputs are independently latched. Outputs and inputs are standard DTL or TTL logic levels with the capability for interface with multiplexing analog-to-digital converters, digital-to-analog converters, keyboards, and low-speed peripheral devices such as incremental tape units, buffered line printers, and X-Y plotters.
2801	Synchronous Modem Controller. Provides an interface to a 201/A-3 or equivalent data set. This option operates with point-to-point, multipoint data only, or switched networks with optional automatic calling-answering for either two-wire or four-wire service. Data transfers are on a byte basis under concurrent input/output or programmed control. Interface levels are per EIA Standard RS-232-B.
2901	Card Reader Controller. Provides an interface to Mohawk Data Sciences SCCR 6002 Card Reader. Handles punched card input at rates up to 400 cards per minute.
2903	High-Speed Tape Punch Controller. Provides an interface to Facit 4070 Tape Punch. Allows paper tape punched output at rates up to 75 characters per second. Can be combined with High-Speed Tape Reader Controller.
2902	High-Speed Tape Reader Controller. Provides an interface to Digitronics 2540EP Tape Reader. Handles paper tape input at rates up to 400 characters per second. Can be combined with Tape Punch.

eight external interrupt signals. Up to eight boards can be used in one system to control a maximum of 64 interrupt signals.

The byte I/O interface contains a single interrupt line common to all I/O devices on the byte I/O bus and a hard-wired priority line that is carried through all devices on the bus. Each I/O device receives priority from the preceding device in the priority chain and passes it along to the next device in line if it is not ready to request an interrupt. A device receiving priority and ready to request an interrupt does not pass along the priority signal, but instead activates the interrupt signal.

After receiving acknowledgment of the interrupt request the interrupting device places on the I/O bus an address byte that the processor uses to transfer program control to the proper interrupt servicing routine.

For detailed descriptions of the timing, logic, and instructions associated with the byte I/O interface, refer to section 3 of this manual.

1.8 DIRECT MEMORY INTERFACE

All CIP/2000-series computers are prewired to receive optional direct memory access (DMA) channels. The DMA channels are plug-in, printed circuit board options (one channel per board) that connect to the memory data and address buses of the computer (figure 1-1). The mainframe chassis can accommodate a maximum of two DMA channels. When the maximum amount of mainframe core memory (16K bytes) is used, only one DMA channel is permitted. However, if mainframe core memory is restricted to 12K bytes or less, two DMA channels are permitted. Although the DMA channels can be installed only in the mainframe chassis, additional memory modules can be added in an external expansion chassis.

Each DMA channel is a high-speed port connecting core memory to external I/O devices. The channels are used for transferring single or multiple blocks of data bytes directly between memory and external devices. Several devices can share one DMA channel on a block-at-a-time basis. Used in this way the channel appears much like the byte I/O bus. Each channel requires minimum attention from the microprogram or macroprogram. Instructions are used to set up a channel and I/O devices initially, but thereafter all data transfers occur automatically. Although a DMA channel is most efficient in performing block transfer operations, it can be programmed to perform single byte transfers.

1.9 MECHANICAL CONSIDERATIONS

As shown in figure 1-1, I/O interface units can be physically installed either in the mainframe chassis or in an expansion chassis external to the mainframe. In either case certain basic mechanical characteristics must be considered.

1.10 MAINFRAME CHASSIS

The mainframe chassis has four standard circuit board slots (connectors J18 through J21) prewired for byte I/O interface circuit boards and one standard slot (connector J9) prewired for either a DMA Selector Channel or a Memory Expander circuit board. The locations of these slots are shown in figure 1-2.

In addition to slots J9 and J18 through J21, four other circuit board slots (connectors J16, J17, J5 and J7) can

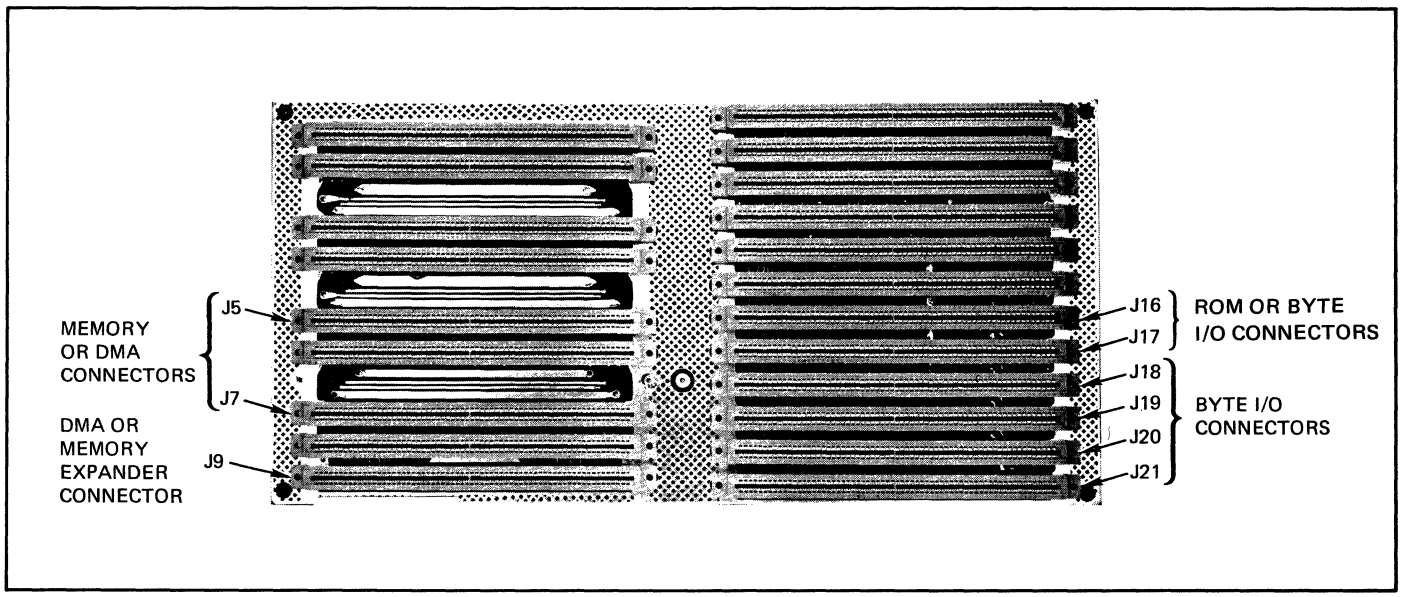


Figure 1-2. Mainframe Chassis Interface Connectors

be used for interface circuit boards if they are not used for their normal functions. Slots J16 and J17 normally contain Read-Only Memory (ROM) circuit boards, but can be used for byte I/O interface option boards if either or both connectors are not used for ROM boards. Slots J5 and J7 normally contain core memory modules, but can be used for DMA Selector Channel boards if less than the maximum mainframe memory is used. Guidelines governing the use of slots J5, J7 and J9 are as follows:

a. If a Memory Expander board is used, it is always installed in connector J9.

b. DMA Selector Channel No. 1 is installed in connector J9 when the Memory Expander board is not used, or in connector J7 when the Memory Expander board is used.

c. DMA Selector Channel No. 2 is installed in connector J7 when the Memory Expander board is not used, or in connector J5 when the Memory Expander board is used.

Mainframe connectors are wired identically for byte I/O interface boards; the only difference between them is their relative position in the hard-wired priority chain. Connector J18 is first in line to receive the priority signal, and connector J21, the last. (J16 can be made first by adding a jumper on the backplane.) Any of the CIP interface goard options listed in table 1-1 can be plugged into these connectors. If CIP interface board options are not used, the connectors can be used for customer-designed interface boards. Slot spacing determines the type and number of customer-designed board that can be used. The six slots can accommodate six printed circuit boards or three wirewrap socket boards having the dimensional specifications shown in figure 1-3. The backplane connector edge of the boards must be designed for plugging into 130-pin (0.1-inch centers), double-row connectors.

1.11 EXPANSION CHASSIS

When more interface circuit boards are required than can be accommodated in the mainframe chassis, they can be housed in a separate expansion chassis. The backplane wiring of the expansion chassis is connected by I/O cable to one of the byte I/O interface connectors in the mainframe chassis through the I/O Line Driver and Receiver interface option. This circuit board option normally plugs into connector J21 in the mainframe chassis, but it can plug into any of the other three byte I/O connectors or two ROM connectors. The connector into which it is plugged determines the position of the expansion chassis (and all its interface units) in the hard-wired priority chain. Plugging the board into connector J21 places the expansion chassis interface units last in the priority chain.

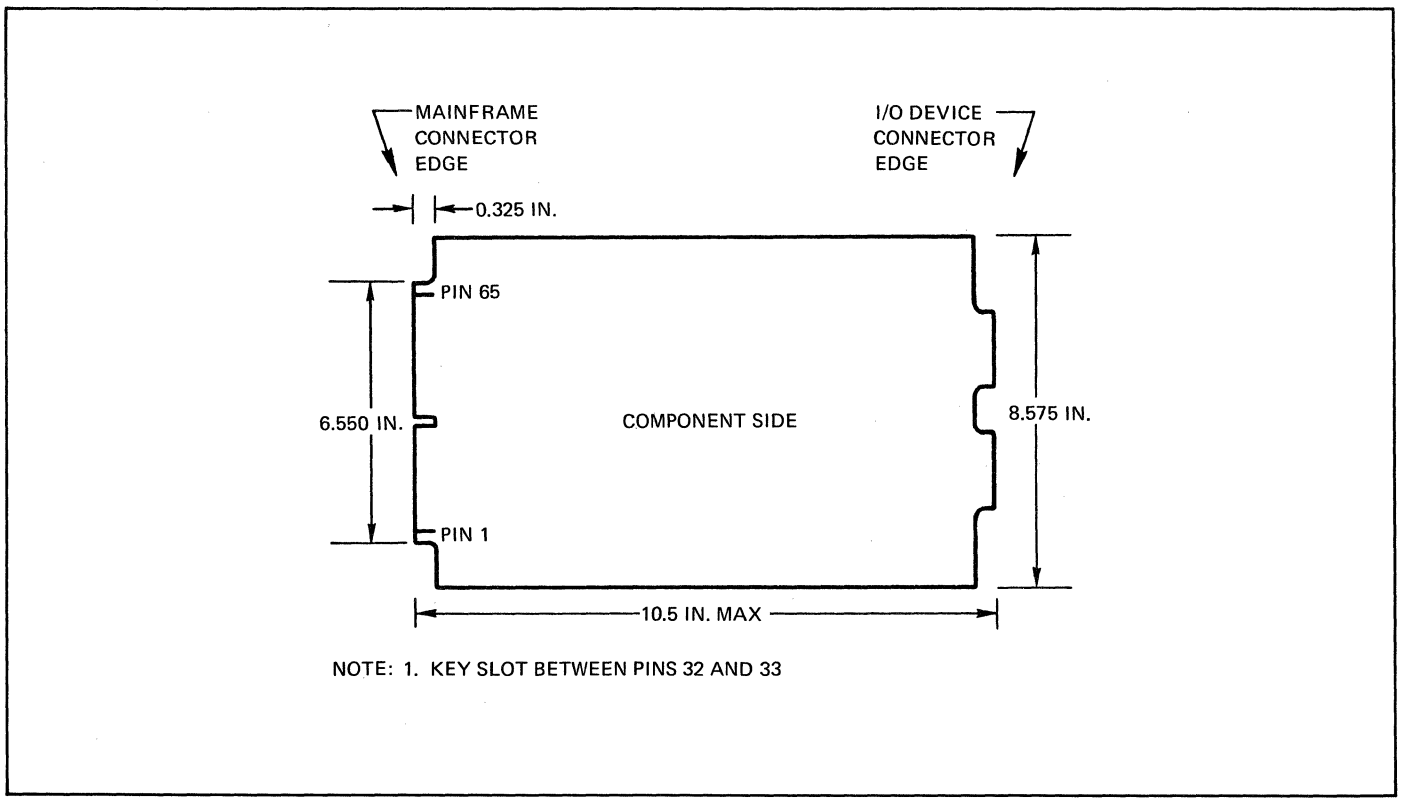


Figure 1-3. Interface Circuit Board Dimensional Data

The I/O Line Driver and Receiver interface option shares one circuit board with the Parallel Teletype Controller interface option. The circuit board, with both options included, is shown in figure 1-4. The I/O cable connecting the expansion chassis to the mainframe chassis attaches to card edge connectors P3 and P4 (figure 1-4) on the I/O Line Driver and Receiver board. The other end of the I/O cable is hard-wired to a terminator board in the expansion chassis as shown in figure 1-5. The terminator board contains termination resistors for the I/O cable and connects the cable to the expansion chassis backplane. Two connectors on the terminator board can be used to extend the I/O bus to another expansion chassis. Detail specifications for the I/O cable are given in table 1-2.

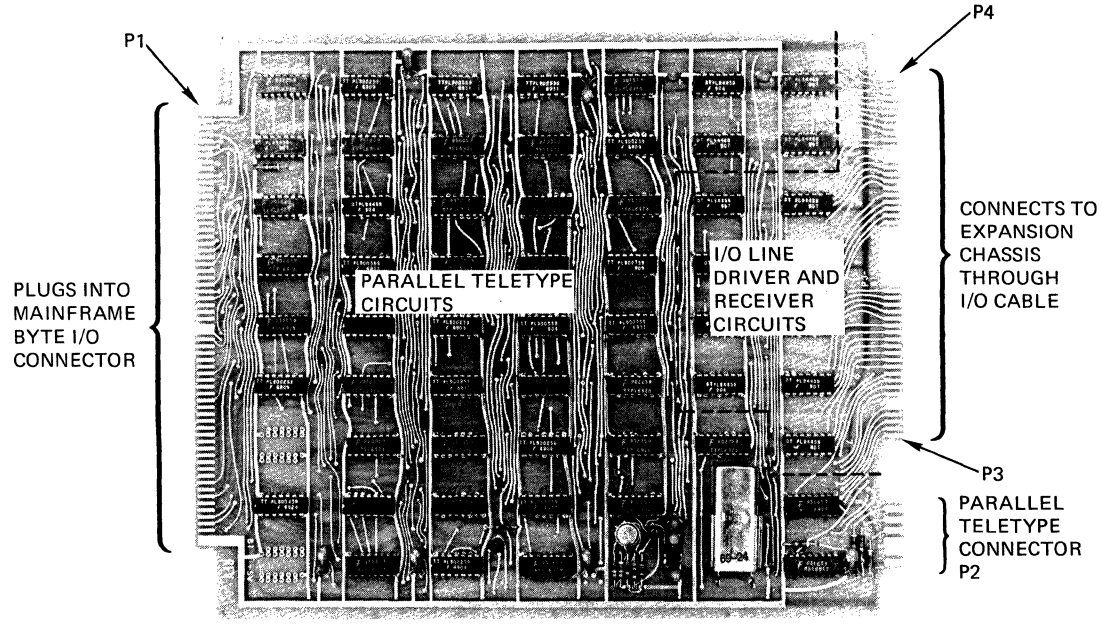


Figure 1-4. I/O Line Driver and Receiver Interface Option

Table 1-2. I/O Cable Specifications

Characteristic	Specification
Number of conductors	88 (44 pairs)
Wire type	Number 24 AWG (MIL-W16878D) twisted pair; 1-1/2 twists per inch
Cable length	30 ft maximum
Characteristic impedance	120 ohms
Capacitance	Less than 15 pF/ft
Propagation delay	Approximately 1.5 ns/ft

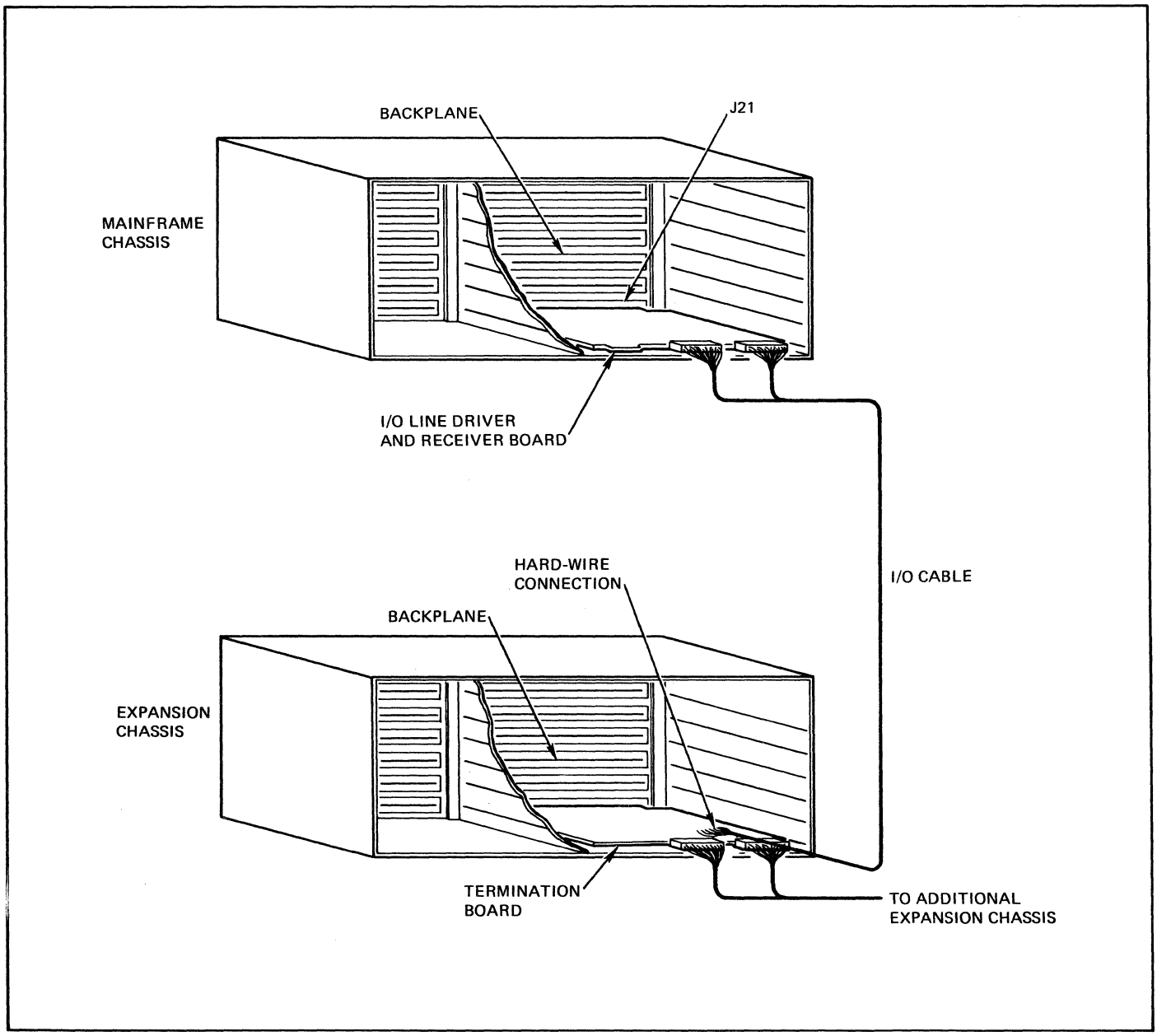


Figure 1-5. I/O Expansion Cabling

2. SERIAL I/O INTERFACE

2.1 INTRODUCTION

The serial Teletype I/O interface is a standard feature of the CIP/2000-series computers. A Teletype wired for four-wire, full duplex, 20-mA operation can be connected directly to the cable provided with the computer.

The four-wire I/O interface circuit is shown in figure 2-1. The transmit portion of the circuit contains a 20-mA current source that can be turned on or off depending on the state of the I/O control register. When the I/O control register is in any mode other than mode 3, the output of gate Z42 is high, emitter follower Q5 conducts, and approximately 20 mA of current flow through resistor R23. This current holds the Teletype in the mark condition. When the I/O control register is set to mode 3 by a microcommand*, the output of gate Z42 is low, emitter follower Q5 cuts off, and no current flows to the Teletype.

The receive portion of the interface circuit contains a low-pass filter network connecting the Teletype distributor to bit 6 of file register 0 where it may be sensed by microcommands. One side of the Teletype distributor is connected to -6V through resistor R62. The other side of the distributor is connected to a TTL gate, which forms bit 6 of file register 0. When the Teletype sends a mark signal, the output of the gate is held low and a 0-bit appears in bit 6 of file register 0. When the Teletype sends a space signal, a 1-bit appears in bit 6 of file register 0.

*See CIP/2000 Reference Manual for a description of I/O control register operation.

2.2 CHARACTER ASSEMBLY AND DISASSEMBLY

Teletype character assembly, disassembly, synchronization, and timing in the CIP/2100 Computer is accomplished by a firmware routine initiated by the macro instructions for the serial I/O interface. Figure 2-2 shows the timing for transmitting or receiving 110-baud Teletype characters.

During an input operation the firmware program searches for the leading edge of the start bit by continuously testing the Teletype input lines. Once a space level is detected the firmware program delays 4.5 milliseconds and then samples the input every 9.09 milliseconds until

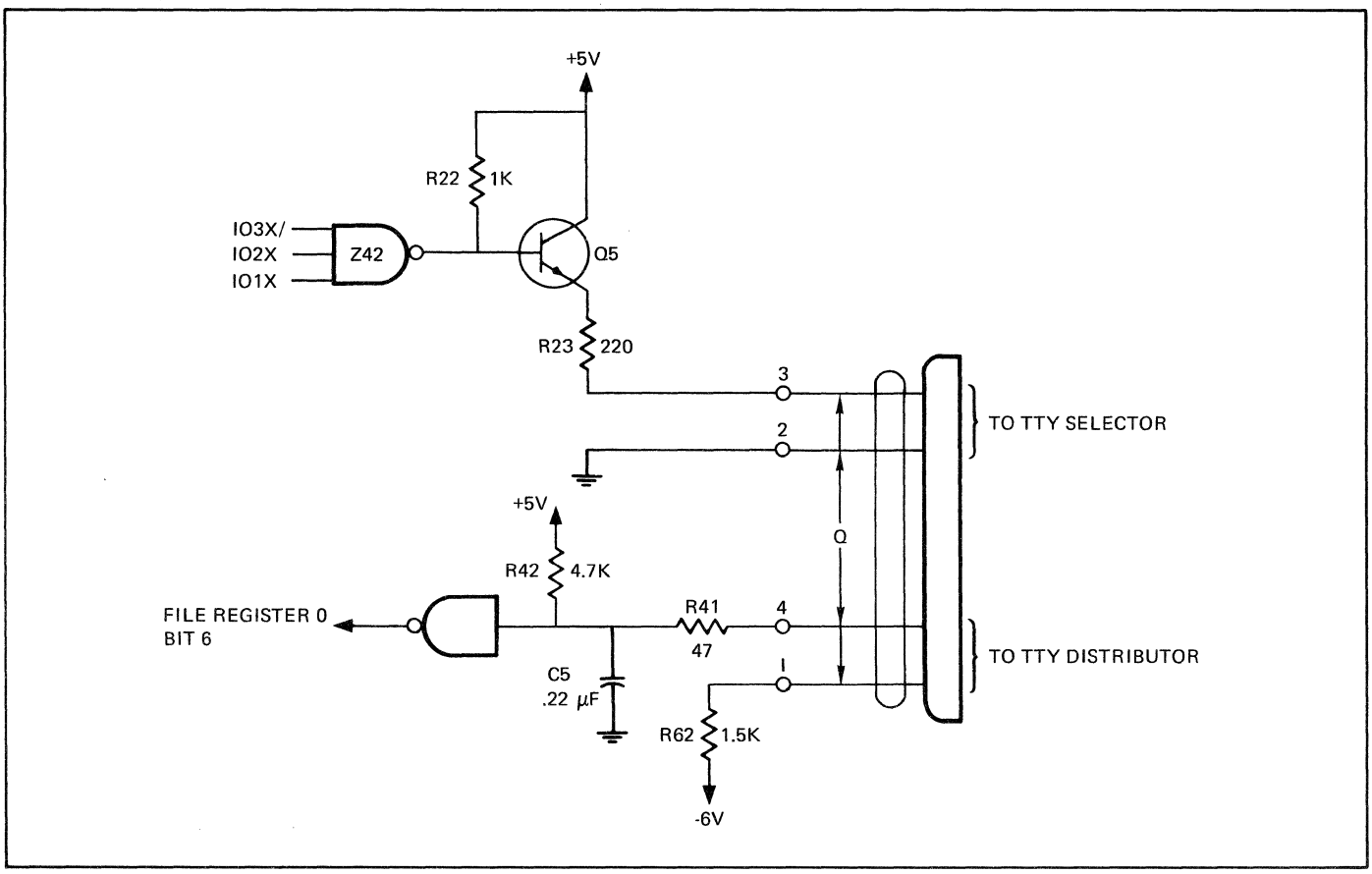


Figure 2-1. Serial I/O Interface Circuit

the eighth information bit is shifted into the assembly register. In the 2100 Computer the character is assembled in the least significant byte of the A-register (file register 4). The initial delay of 4.5 milliseconds after detecting the leading edge of the start bit causes sampling to occur in the middle of each signaling element.

During an output operation the firmware program sets the I/O control register to the appropriate mark or space condition every 9.09 milliseconds according to the start and stop bits and the data to be serially transmitted. Before the first information bit is transferred the I/O control register is set to mode 3 to transmit the start bit. The firmware program for transmitting a Teletype character remains active for 11 intervals (100 milliseconds) to assure the proper stop interval before the next character is transmitted.

2.3 MICROCOMMANDS AND MACRO INSTRUCTIONS

Two microcommands (2000 Computer) and two macro instructions (2100 Computer) affect the operation of the serial I/O interface: Load Seven Control (LS), Control (K), Input Byte Serially (IBS), and Output Byte Serially (OBS).

The Load Seven Control Microcommand gates the Teletype input to bit 6 of file register 0 during the next command time if the memory spare bit option is installed. In this case a 1701 command must be executed immediately before testing the Teletype input. If the memory spare bit option is not installed, the serial Teletype input is constantly at bit 6 of file register 0.

The Control microcommand sets the state of the I/O control register, which, in turn, establishes the mark or space of the serial Teletype output. Executing this command with bit 7 set to 1 causes the contents of bits 4 through 6 to be entered into the I/O control register. When the register contains a 3, the serial Teletype interface transmits a space. For all other settings of the register the serial I/O interface transmits a mark. For example, the command 7XOX causes the interface to go to a marking condition, while the command 7XBX causes the output to transmit a space.

The 2100 Computer Input Byte Serially macro instruction transfers an eight-bit character from the Teletype into the eight low-order bits of the A-register. The execution of this instruction terminates when a complete

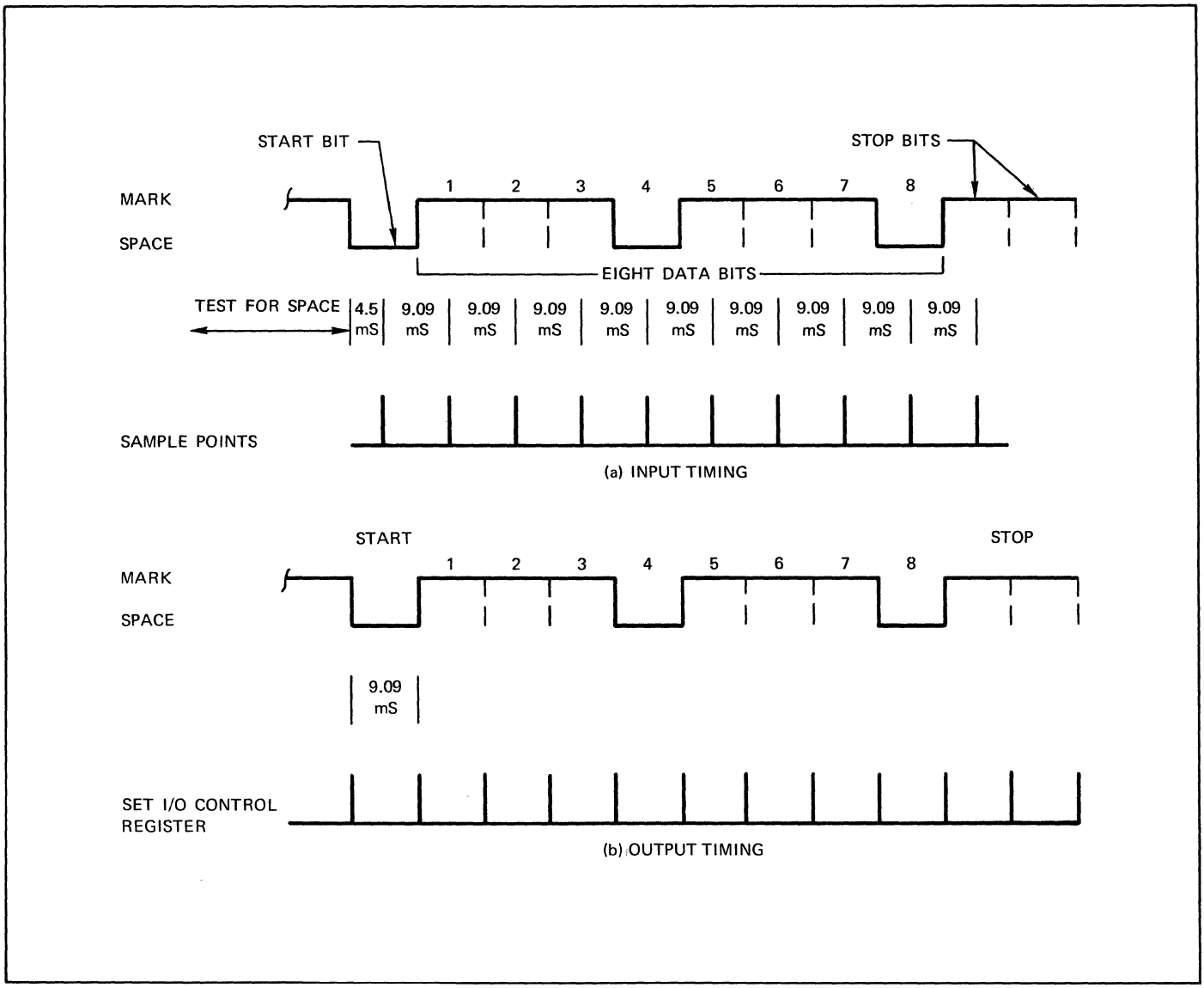


Figure 2-2. Serial I/O Timing

Teletype character has been received. For proper operation, the execution of the instruction must be started before the start of the Teletype character. Once the instruction is started the computer becomes tied up until a Teletype character is received. The execution time of the instruction extends approximately 84 milliseconds after the leading edge of the Teletype character start bit. When the program echoes input characters back to the Teletype, the effective input rate cannot exceed five characters per second, since no input can be handled during the 100 milliseconds required for output.

The 2100 Computer Output Byte Serially macro instruction disassembles the eight low-order bits of the A-register and transfers them serially, as a Teletype character, through the serial I/O interface. During the execution of this instruction the eight low-order bits of the A-register are set to 1's; the eight high-order bits remain unchanged.

2.4 TELETYPE INTERFACE CONNECTION

If an ASR 33 Teletype is purchased from Cincinnati, it has already been modified for use with the 2000-series computers. It can be connected to the computer simply by mating the serial I/O connector with connector S2 of the Teletype (located at the left rear).

Procedures for modifying standard ASR 33 or 35 Teletypes for use with 2000-series computers are given in appendix C.

3. BYTE I/O INTERFACES

3.1 INTRODUCTION

The byte I/O interfaces (both internal and external) comprise nine input data lines, five input control lines, nine output data lines, eight (internal) or ten (external) output control lines, and four spare lines. The points of interface to both the internal and external byte I/O buses and the relationships between both buses are depicted in figure 3-1. This figure shows the point of origin or destination in the processor for all interface lines. It also shows the interface logic on the I/O Line Driver and Receiver interface option (shown plugged into connector J21) and shows how the option is used to extend the byte I/O bus to an external expansion chassis.

3.2 BYTE I/O BUS

The following paragraphs describe the input and output data lines and the control lines of the byte I/O bus. Except where specifically stated otherwise, the descriptions apply to both the internal and external byte I/O buses.

Note

Throughout this manual a virgule (/) appended to a signal or line mnemonic indicates negation; that is, the signal or line is low when the function specified by the signal or line is occurring.

3.3 INPUT DATA LINES

Input data lines ID00/ through ID08/ are continuous from the last interface unit on the I/O cable through the I/O Line Driver and Receiver board, the backplane, and into the processor B-bus. A termination network for each line is located in the processor. The lines are driven by DTL power gates (944 or equivalent) with uncommitted collectors. Because of the termination network, the lines are allowed to swing only between ground and +3V. Each line is at ground potential when any one of the DTL power gates on the line is turned on. Each line is at +3V when all gates on the line are off. When a gate is switched on, the line to which it is connected places a logical 1 onto the B-bus and into the destination register or memory.

The input data lines are handled exactly the same whether a device controller is connected to mainframe connectors J16 through J21 or to the I/O cable in the expansion chassis. However, when the external lines are used, they must be terminated at the remote end as well as the processor end.

3.4 OUTPUT DATA LINES

Output data lines T00X/ through T08X/ and OD00/ through OD08/ connect the processor T-register to external I/O devices. The T-register is the physical source for output

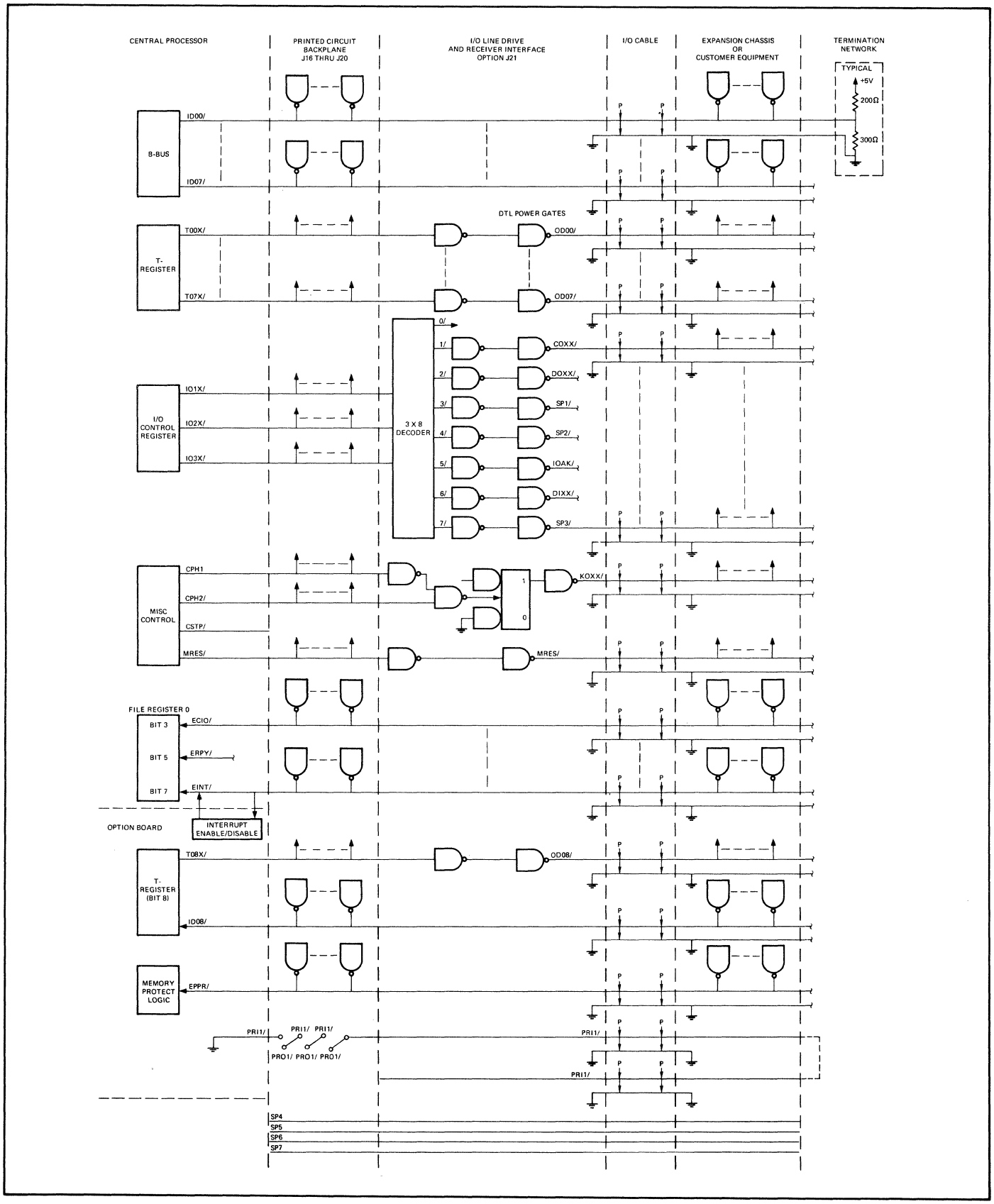


Figure 3-1. Byte I/O Interface, Simplified Logic Diagram

data. Data or address information to be transferred to an I/O device is first fetched from the A-register, the B-register, or memory and then placed in the T-register. From the T-register it is transferred to the I/O device.

Lines T00X/ through T07X/ are available at mainframe byte I/O connectors J16 through J21. (Line T08X/ is available at the same connectors only if the spare bit option is installed in the processor.) Lines OD00/ through OD08/ are the external byte I/O equivalent of lines T00X/ through T08X/. They carry the same signals (slightly delayed) through the I/O cable to the expansion chassis.

To preserve the expansion capability of the output data bus, each device controller on the bus is restricted to a single unit load (one DTL or TTL gate) or 1.6 mA maximum on each output data line. Both the T0XX/ and ODXX/ lines have the following characteristics:

T-Register Content	T0XX/	ODXX/
Binary 1	0V	0V
Binary 0	+4V (nom)	+3V (nom)

3.5 INPUT CONTROL LINES

The five input control lines available at the byte I/O interface are:

a. ECIO/ – concurrent I/O request

b. ERPY/ – I/O reply

c. EINT/ – external interrupt

d. EDPR/ – external device protect

e. PRI1/ – priority return

All five lines are continuous from the last interface unit on the I/O cable, through the I/O Line Driver and Receiver board, the backplane, and into the processor. The lines are driven by DTL power gates (944 or equivalent) with uncommitted collectors. A termination network for each line is included in the processor. Because of the termination networks, the lines are allowed to swing only between ground and +3V. All five lines are active, or indicate assertion, when they are at ground potential. For example, ground potential on line EINT/ causes an external interrupt.

Input control lines ECIO/, ERPY/, and EINT/ provide inputs to file register 0 in the processor. The status of these lines can be determined in the microprogram by testing the associated bits of the file register.

Line ECIO/ becomes bit 3 of file register 0, where it acts as an interrupt to the CIP/2100 processor. It causes the processor to discontinue normal instruction execution and begin a firmware subroutine for handling a concurrent data transfer.

Line ERPY/ becomes bit 5 of file register 0. This bit is not currently used in the CIP/2100 processor. It is included for those system designers who may want to effect their own microprogrammed I/O transfers (since file register 0 bits are testable only at the microprogram level).

Line EINT/ is the external interrupt line, which is shared by all Priority Interrupt boards and device controllers. It becomes bit 7 of file register 0. It causes **the CIP/2100 processor to discontinue normal instruction execution and begin a firmware sequence to determine the address of the interrupting device. (See paragraph 3.28 for a description of external interrupt operation.)**

Line EDPR/ is activated by an I/O device during input transfers to override the optional memory protection error interrupt. It signifies that the device has been programmed to transfer data into a protected area of memory.

Line PRI1/ is the priority input line from the next higher priority controller. It is used by each controller in determining priority before requesting an external interrupt or concurrent I/O request. (See paragraph 3.29 for a description of priority determination.)

3.6 OUTPUT CONTROL LINES

Of all the interface lines only the output control lines are functionally different at the internal and external byte I/O interfaces. The following lines are available at each interface:

Internal Interface	External Interface
IO1X/	COXX/
IO2X/	DOXX/
IO3X/	SP1/
CPH1	SP2/
CPH2/	IOAK/
MRES/	DIXX/
PRO1/	SP3/
	KOXX/
	MRES/
	PRO1/

3.7 Internal Byte I/O Control Lines (Mainframe Chassis)

Internal I/O control lines IO1X/ through IO3X/ originate at the false outputs of the three I/O control flip-flops in the processor. These flip-flops are set and reset at the microcommand level. The eight states that the flip-flops can assume are assigned meanings to indicate various I/O control modes. Definitions of the eight control flip-flop states for the CIP/2100 are given in table 3-1. Other definitions can be applied for specialized CIP/2000 configurations.

As shown in figure 3-1, lines IO1X/ through IO3X/ are decoded on the I/O Line Driver and Receiver board, and seven lines are then available (state 0 is not used) at the external interface. A device controller designed for use in the mainframe chassis must decode these three lines as is done on the I/O Line Driver and Receiver board. Device controllers designed for the CIP/2100 processor are required to decode only states 1 (COXX/), 2 (DOXX/), 5 (IOAK/), and 6 (DIXX/). These states represent control output, data output, I/O acknowledge, and data input functions, respectively.

Table 3-1. I/O Control States

State	Control Definition	Logic Term
0	None	None
1	Control output	COXX/
2	Data output	DOXX/
3	Space serial Teletype	SP1/
4	Spare	SP2/
5	I/O acknowledge	IOAK/
6	Data input	DIXX/
7	Spare	SP3/

It is important to remember that a device controller in the mainframe chassis must decode the I/O control flip-flop lines, but one in the expansion chassis does not, since the external bus is used to connect the two chassis.

Lines CPH1 and CPH2/ provide processor clock signals to device controllers connected to the mainframe byte I/O connectors. Each line can be used independently as a square wave source (4.55 MHz), or they may be used together in a NAND gate to produce a 35-ns clock pulse (CPH1 is inverted and delayed 35 ns from CPH2/). The relationship of the signals on lines CPH1 and CPH2/ is shown in figure 3-2.

Control line CSTP/ of the internal byte I/O interface carries a signal to indicate that the internal processor clock has been stopped for either of the following reasons:

a. The processor is halted.

b. The processor is executing a logical pause (for 1 to 5 clock times) while performing a jump or attempting to overrun a memory operation during a microcommand sequence.

Line CSTP/ is available to gate clocks in device controllers when intimate microlevel control of a device is required. It is not included in standard byte I/O interfaces and must be jumpered to an open pin of an option board connector if its use is required.

Control line MRES/ is the master reset line. It is activated by the RESET switch on the operator's console, or by the optional power fail circuit during power failure or restart. It is used to clear all control flip-flops to their initialized conditions. Ground potential is applied to this line when the RESET switch is pressed.

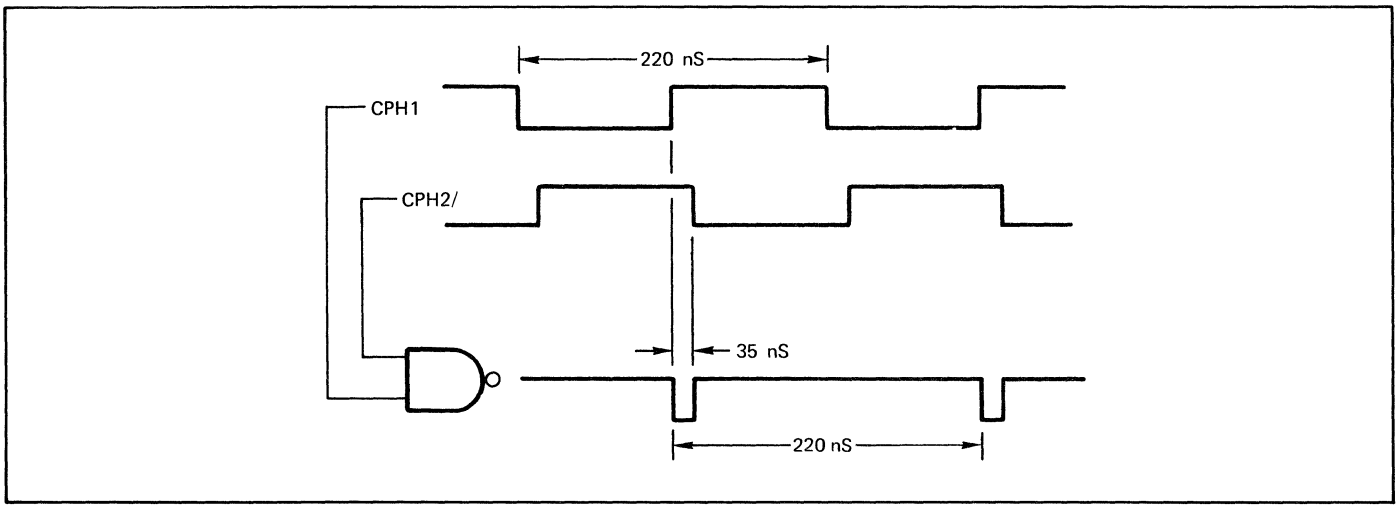


Figure 3-2. Relationship of Control Signals CPH1 and CPH2/

Control line PRO1/ carries the priority signal from controller to controller. It becomes line PRI1/ at the input of the next lower priority controller. Backplane strapping on the mainframe chassis determines the relative priority of devices on the I/O bus. A modification of the strapping permits interface units in the mainframe chassis to have lower priority (for interrupt or concurrent I/O operations) than units in the expansion chassis at the end of the I/O cable. (See paragraph 3.29 for a description of priority determination.)

3.8 External Byte I/O Control Lines (I/O Cable)

Figure 3-1 shows how the output control lines are modified in the I/O Line Driver and Receiver board to develop the control lines available in the I/O cable. As explained in the preceding paragraph, lines IO1X/ through IO3X/ are converted into unitary control lines. The main reason for this conversion is the elimination of ambiguity resulting from flip-flop crossover variations. These crossover variations could produce spikes that would cause incorrect operation if the decoded terms were used as clocks. This problem can be overcome in the mainframe chassis by using a synchronous clock created from signals CPH1 and CPH2/. Since this solution is unacceptable at the end of the I/O cable, the outputs of the three I/O control flip-flops are converted into seven unitary lines on the I/O Line Driver and Receiver board after crossover skew has been eliminated.

Of the seven decoded control lines available at the I/O cable, only four are presently used for byte I/O operations: COXX/, DOXX/, IOAK/, and DIXX/. Lines SP1/, SP2/, and SP3/ are spare control lines that are not used in the CIP/2100. They can be used to carry I/O control signals originated by special firmware.

Note

Although control line SP1/ is considered a spare, it represents I/O control state 3, which is the space serial Teletype mode. This line can be used only in the absence of the serial Teletype.

Control line COXX/ carries a low signal to specify that a control byte containing a device number and device order is on the output data lines. The device being addressed accepts the byte and decodes the order.

Control line DOXX/ carries a low signal to specify that an information byte is on the output data lines. The previously addressed device accepts the byte.

Control line IOAK/ carries a low signal to acknowledge a request from an I/O device for an interrupt (paragraph 3.28) or a concurrent I/O transfer (paragraph 3.25). The requesting device uses the signal to transfer an address byte on the input data lines to the processor.

Control line DIXX/ carries a low signal that is used by a previously addressed I/O device to transfer an information byte on the input data lines to the processor.

Control line KOXX/ carries a 2.275-MHz, 50% duty cycle clock signal that can be used by external devices for timing.

Control lines MRES/ and PRO1/ have the same meaning in the I/O cable as they do for the internal byte I/O interface.

3.9 SPARE LINES

Spare lines SP4 through SP7 are continuous from the last interface unit on the I/O cable, through the I/O Line Driver and Receiver board, and onto the backplane. These lines are not terminated in any way and are provided only for special customer requirements.

3.10 BYTE I/O FUNDAMENTALS

Although the flexibility of the byte I/O system lends itself to customizing for individual applications, certain standard conventions have been adopted for byte I/O operations in the CIP/2100 Computer. These conventions are described in the following paragraphs.

3.11 DEVICE ADDRESSING

Each I/O device on the byte I/O bus is assigned a unique five-bit device address or number. On most CIP device controllers, the addresses are selected by the placement of jumper wires on the printed circuit board of the controller.

Each device controller on the I/O bus determines if it is being addressed by comparing its assigned address to the five-bit device number in the control byte sent to all controllers on the output data lines. The device number portion of the control byte appears on data lines T00X/ through T04X/ (OD00/ through OD04/). The assigned device address is also used to identify the I/O device requesting an interrupt or concurrent I/O transfer. The processor acknowledges each request with signal IOAK/. On receiving the acknowledgement signal, the requesting device places its address (times 2) on input data lines ID01/ through ID05/.

Table 3-2 lists the device addresses assigned to CIP/2100 standard interface units. Customer-designed controllers should not use the assigned addresses if the use of standard CIP controllers is planned.

3.12 DEVICE ORDERS

Accompanying the five-bit device address in the control byte sent to all devices before each programmed transfer is a three-bit device order specifying the I/O operation

Table 3-2. Standard I/O Device Addresses

Address (Hexadecimal)	I/O Device
00	Teletype (parallel interface)
01	Low-Speed Asynchronous Modem or Teletype Interface
02	High-Speed Paper Tape Reader
03	High-Speed Paper Tape Punch
04	Card Reader
05	Not assigned
06	Drum/Disc
07	Not assigned
08	32 x 32 Discrete Input/Output Interface option
09 and 10	Not assigned
11	Low-Speed Asynchronous Modem or Teletype Interface
12 thru 15	Not assigned
16	DMA Selector Channel No. 1
17	DMA Selector Channel No. 2
18	Priority Interrupt Group 7
19	Priority Interrupt Group 6
1A	Priority Interrupt Group 5
1B	Priority Interrupt Group 4
1C	Priority Interrupt Group 3
1D	Priority Interrupt Group 2
1E	Priority Interrupt Group 1
1F	Priority Interrupt Group 0

to be performed by the device. The device order portion of the control byte appears on output data lines T05X/ through T07X/ (OD05/ through OD07/). A list of standard device orders is given in table 3-3. Not all device controllers are required to use all the orders listed in the table. Their use is dictated by controller design.

3.13 STATUS BYTES

In response to a status order from the processor, the addressed I/O device places a status byte on input data lines ID00/ through ID07/. Four of the status bits are common to all device controllers; the other four are device dependent and differ from controller to controller. The status byte is transferred into the A-register, the B-register, or memory by an input instruction with device order 1. The significance of each bit in the status byte is described in table 3-4.

3.14 BYTE I/O OPERATIONS AND TIMING

The following paragraphs describe the program-controlled and concurrent I/O operations of the CIP/2100 Computer. Timing diagrams are included for each operation, and typical controller logic for performing the operations is also shown. For a description of the I/O instructions that pertain to the byte I/O operations, consult the CIP/2100 Reference Manual.

3.15 PROGRAM-CONTROLLED I/O OPERATIONS

3.16 Data Output Timing

The timing diagram for a typical data output operation is shown in figure 3-3. When an output byte instruction (OBA, OBB, or OBM) is executed, the second byte of the instruction containing the device address and order is placed on output data lines T00X/ through T07X/ (OD00/ through OD07/). Approximately 220 ns later, control line COXX/ goes low to indicate the presence of a control byte on the output data lines. During the 880 ns that line COXX/ is low, each device controller on the bus examines lines T00X/ through T04X/ (OD00/ through OD04/) to determine if it is the controller being addressed. The controller whose address is on the lines connects itself for service and decodes and stores the device order on output data lines T05X/ through T07X/ (OD05/ through OD07/). When the controller is connected for service, it is susceptible to either data output or data input signals from the processor and transfers data accordingly. Once connected for service, the controller remains connected until a data output or data input signal occurs.

After removing the control output signal and control byte from the lines, the processor places a data byte on the output data lines. Approximately 220 ns later, data output line DOXX/ goes low to indicate the presence of data on the lines. The controller then strobes the data byte from the output data lines into its data register. When line DOXX/ again goes high, the controller disconnects itself from further service.

3.17 Typical Data Output Logic

The timing diagram of figure 3-3 can be related to the typical data output logic shown in figure 3-4. The eight

Table 3-3. Standard I/O Device Orders

Order Number	Operation	Description
0	Data	The data order causes a data byte to be transferred between the processor and the addressed device. The direction of transfer depends on the type of instruction (input or output).
1	Status/function	The status/function order causes a status byte to be transferred from the addressed device to the processor, or a function byte to be transferred from the processor to the device depending on the type of instruction (input or output).
2	Block input	The block input order notifies the device to proceed with a concurrent block input to memory. This order can be sent with either an input or an output instruction.
3	Block input with interrupt	The block input with interrupt order notifies the device to proceed with a concurrent block input to memory and to generate an interrupt at the end of the transfer. This order can be sent with either an input or output instruction.
4	Stop	The stop order causes the block input or output operation in progress to be stopped. An external interrupt is generated if an interrupt would normally have been generated at the end of the block transfer.
5	Protect state	The protect state order allows the device to write into protected areas of memory with concurrent input. The device is taken out of this mode at the end of the transfer.
6	Block output	The block output order notifies the device to proceed with a concurrent block output from memory. This order can be sent with either an input or an output instruction.
7	Block output with interrupt	The block output with interrupt order notifies the device to proceed with a concurrent block output from memory and to generate an interrupt at the end of the transfer. This order can be sent with either an input or an output instruction.

Table 3-4. I/O Device Status Byte Definition

Bit Number	Condition	Description
0	Ready	This bit is set to 1 when the I/O device is in a ready state.
1	Input flag	This bit is set to 1 when the I/O device has a byte ready for input to the processor.
2	Output flag	This bit is set to 1 when the I/O device is ready to receive a byte from the processor.
3	Error	This bit is set to 1 when an error has occurred during a transfer operation. The error may be the result of timing, parity, or a device malfunction. The bit is cleared when the status byte is transferred.
4-7	Undefined	These four bits are unique for each I/O device.

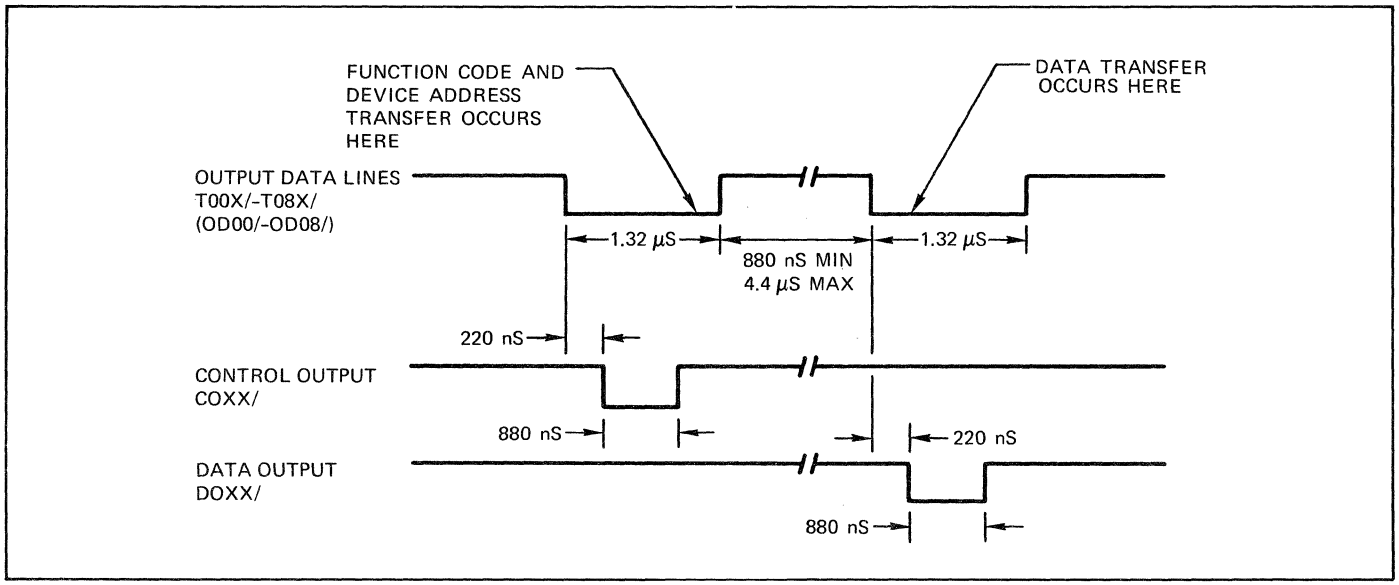


Figure 3-3. Data or Function Output Timing

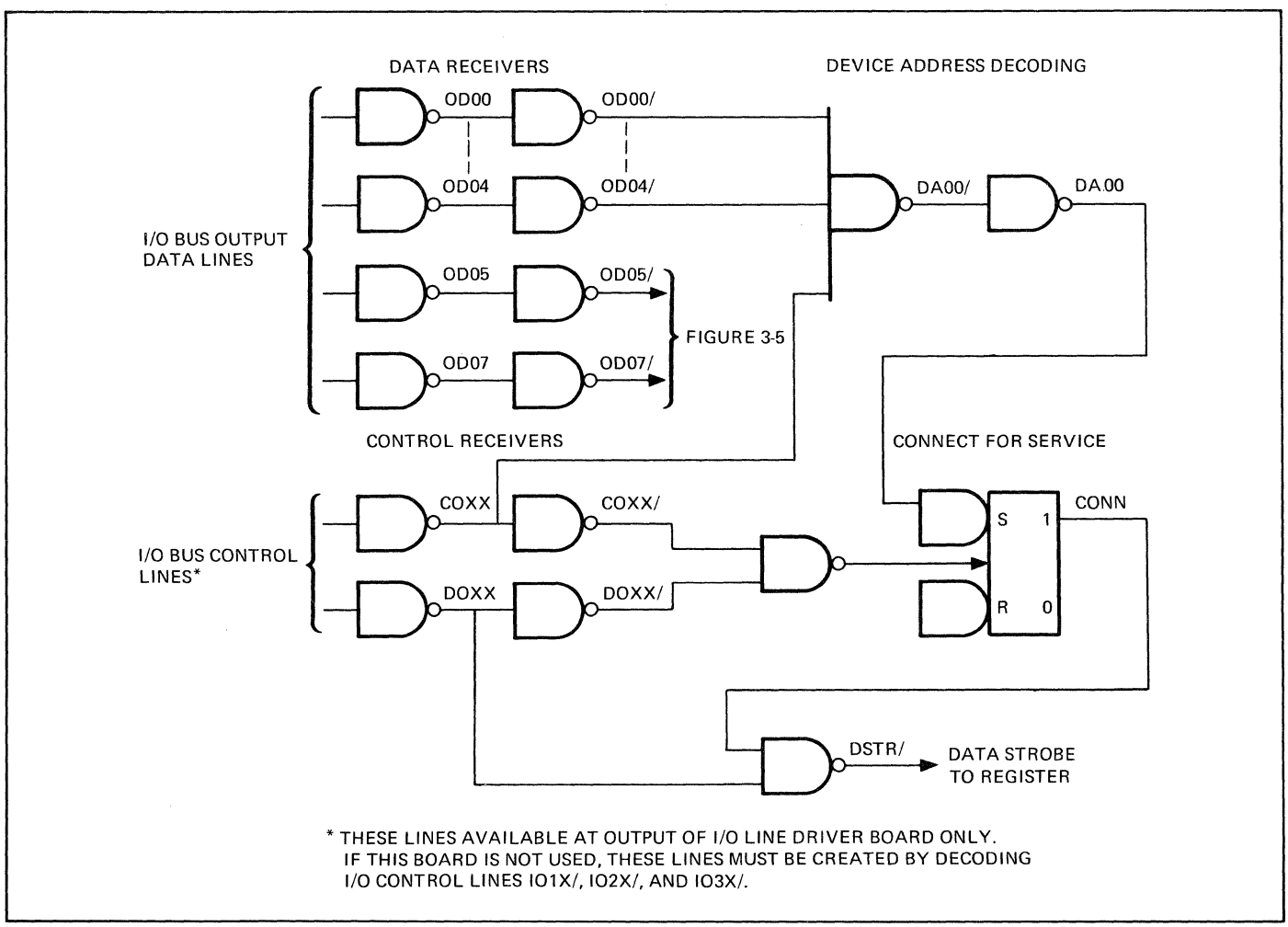


Figure 3-4. Data Output Logic

output data lines are connected to eight TTL or DTL inverters in the device controller. Each output line is buffered by an inverter or gate to minimize the loading on the line and to allow for expansion to the full number of devices. Receiver outputs OD00/ through OD04/ are applied to an address decoding circuit. In the example shown, the circuit is connected to decode a device address of 00 (all false terms are used).

Control lines COXX/ and DOXX/ are used for dual purposes in the example logic. Signal COXX is used as a qualifying signal in the addressing decoding circuit, and its complement COXX/ is used to clock the connect-for-service flip-flop. Signal DOXX/ is also used to clock the connect-for-service flip-flop and its complement (DOXX) generates the signal for strobing data into the controller register.

When the connect-for-service flip-flop sets, the controller is receptive to the actual transfer of data (the second phase of the data output operation). The signal for gating the output data lines into a device register or similar storage device is generated by the connect-for-service signal and control signal DOXX (data output). As figure 3-3 indicates, signal DOXX/ occurs anywhere from 880 ns to 4.4 μs maximum after the device address is removed from the output data lines. The timing varies because the output data can originate in the A-register, the B-register, or memory.

3.18 Function Output Timing

The timing diagram shown in figure 3-3 for a data output operation is also valid for a function output operation. The function output operation is typically used to control a discrete action in an I/O device for which data transfer is not required. Rewinding tape is an example of such an action. The most efficient way to perform this operation is to issue a single instruction containing all the information necessary to alert the device and cause the tape to rewind.

In the CIP/2100 Computer the output byte instructions (OBA, OBB, and OBM) are also used to perform the function output operation. The only difference in the instructions is the assignment of the f-code (bits 5 through 7) in the control byte of the instruction. When an output byte instruction is used for function output, the f-code of the control byte designates the unique function in the I/O device to be controlled. The assignment of f-codes for function operations precludes the use of the same codes for data transfer operations.

The function output operation is executed exactly like the data output operation described in paragraph 3.16. In fact, a data byte is transferred from either the A-register, B-register, or memory depending on the output byte instruction used. This data byte is usually ignored by the device controller, since the f-code of the control byte contains enough information to describe most function operations. However, should a controller require more function definition than is possible in the control byte, the data byte transferred during the function operation could be used to carry additional function information.

3.19 Typical Function Output Logic

The alerting of the device controller by sending the control output signal (COXX/) along with the device address is accomplished exactly as described for data output. When line COXX/ goes low, the controller examines bits 5 through 7 (OD05/ through OD07/) of the control byte and, if necessary, stores them so that it can perform the ordered function. Figure 3-5 shows typical logic for storing and then decoding the function bits. An alternative method would be to decode the function bits first (during DA00 time) and then store the decoded results in unitary flip-flops. This method would be advantageous when several control states are required, each one executing a separate function and each one having an asynchronous reset term dependent on the device.

The important point to remember is that functions needed for longer than 880 ns must be stored either before or after decoding. A suitable clock for storing this information is signal DA00, derived from the device address bits and signal COXX.

3.20 Data Input Timing

The timing diagram for a typical data input operation is shown in figure 3-6. When an input byte instruction (IBA, IBB, or IBM) is executed, a two-phase data input operation, similar to data output, is performed. The first phase is identical to the first phase of the data output operation. The control byte containing the device address and order is placed on output data lines T00X/ through T07X/ (OD00/ through OD07/) by the processor, and then control output line COXX/ goes low. At that time the device controller connects itself for service and prepares to transfer data to the processor on input data lines ID00/ through ID08/. In terms of controller design, the device controller is not required to detect, during the device address phase, whether an input or output operation is to be performed. It can connect itself for service in either case and then allow the data input (DIXX/) or data output (DOXX/) control lines to direct its further activity.

As shown in figure 3-6, approximately 880 ns after the control byte is removed from the output data lines, data input line DIXX/ goes low for 880 ns. Device data to be transferred to the processor must be settled no later than 440 ns after line DIXX/ goes low. The data byte can be applied to the input data lines as early as the beginning of signal COXX/. For interface design freedom, the input data may be applied even during an output byte instruction with no adverse effect.

The input data must be removed from the input data lines no later than 440 ns after line DIXX/ goes high again.

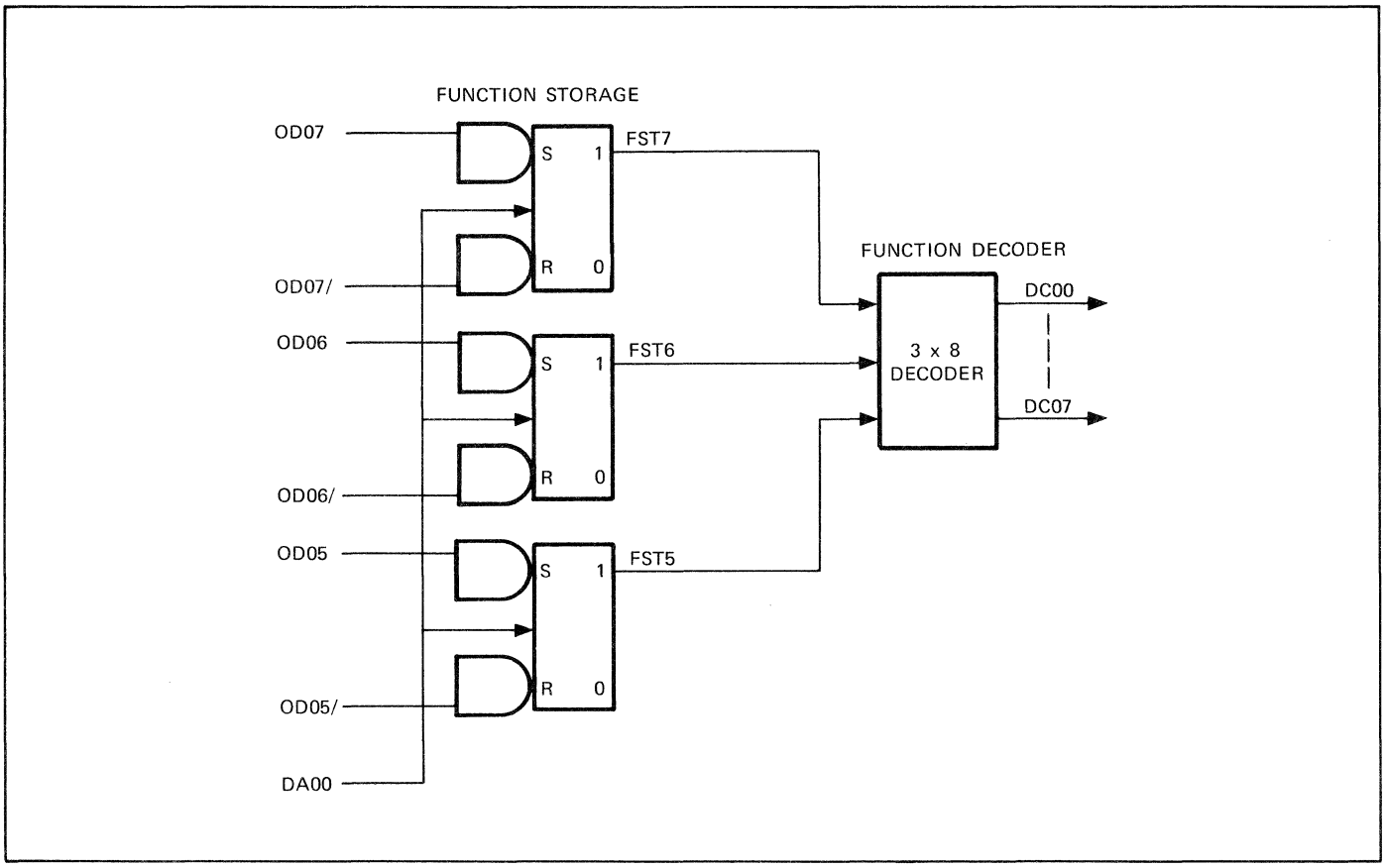


Figure 3-5. Function Output Storage and Decoding Logic

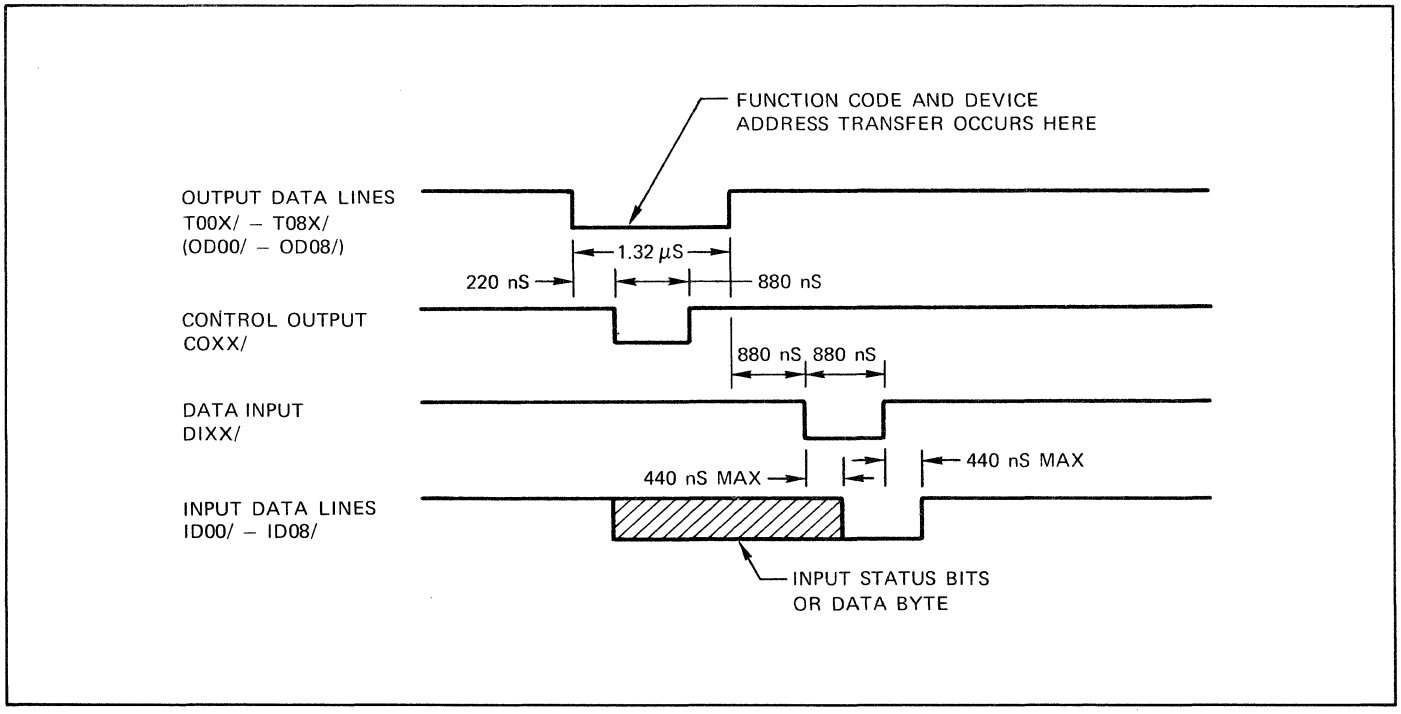


Figure 3-6. Data or Status Input Timing

For normal operation on the external byte I/O bus with less than a 30-foot twisted pair cable, it is feasible to use the DIXX/ signal itself for gating or qualifying the application of input data to the input data lines.

3.21 Typical Data Input Logic

Figure 3-7 shows the additional logic required for a data input operation. Eight power drivers are used to drive the eight input data lines (ID00/ through ID07/). The power drivers are type 944 DTL (or equivalent) power gates with open collectors. A single pull-up resistor is contained in the processor. The power drivers must be held off when the device is not being addressed, and should be turned on only when the device is connected for service.

3.22 Status Input Timing

The timing diagram shown in figure 3-6 for a data input operation is also valid for a status input operation. A similar relationship exists between a status input and data input operation as existed between a function output and data output operation. In the CIP/2100 Computer the input byte instructions (IBA, IBB, and IBM) are used both for data input and status input operations. To differentiate between the two operations, an f-code of 000 is used in the control byte for data transfer, but a code of 001 is used for status transfer. Otherwise, the operations are identical except that a status byte (table 3-4) is placed on input data lines ID00/ through ID07/ instead of a data byte.

3.23 Typical Status Input Logic

Figure 3-8 shows the change in data input logic required to accommodate a status input operation. As shown in the example, the false terms of both data and status bits are used as inputs to the drivers. The function code stored during the control output phase of the operation (DC00 for data or DC01 for status) determines whether a data byte or a status byte is transferred to the processor over the input data lines.

The combination of logic shown in figures 3-4, 3-5, and 3-8 produces a device controller than can perform program-controlled data output, data input, function output, and status input operations.

3.24 TYPICAL PROGRAM-CONTROLLED TRANSFER SEQUENCE

Figure 3-9 is a flow chart showing the sequence and timing for a typical program-controlled block transfer operation that could be handled using the typical device controller logic described in paragraphs 3.16 through 3.23. The sequence consists of:

a. Sampling and determining the status of the device controller

b. Transferring data from memory to the controller when the controller is ready to accept it

c. Updating the index register (used as a data pointer) and checking for the end of the block

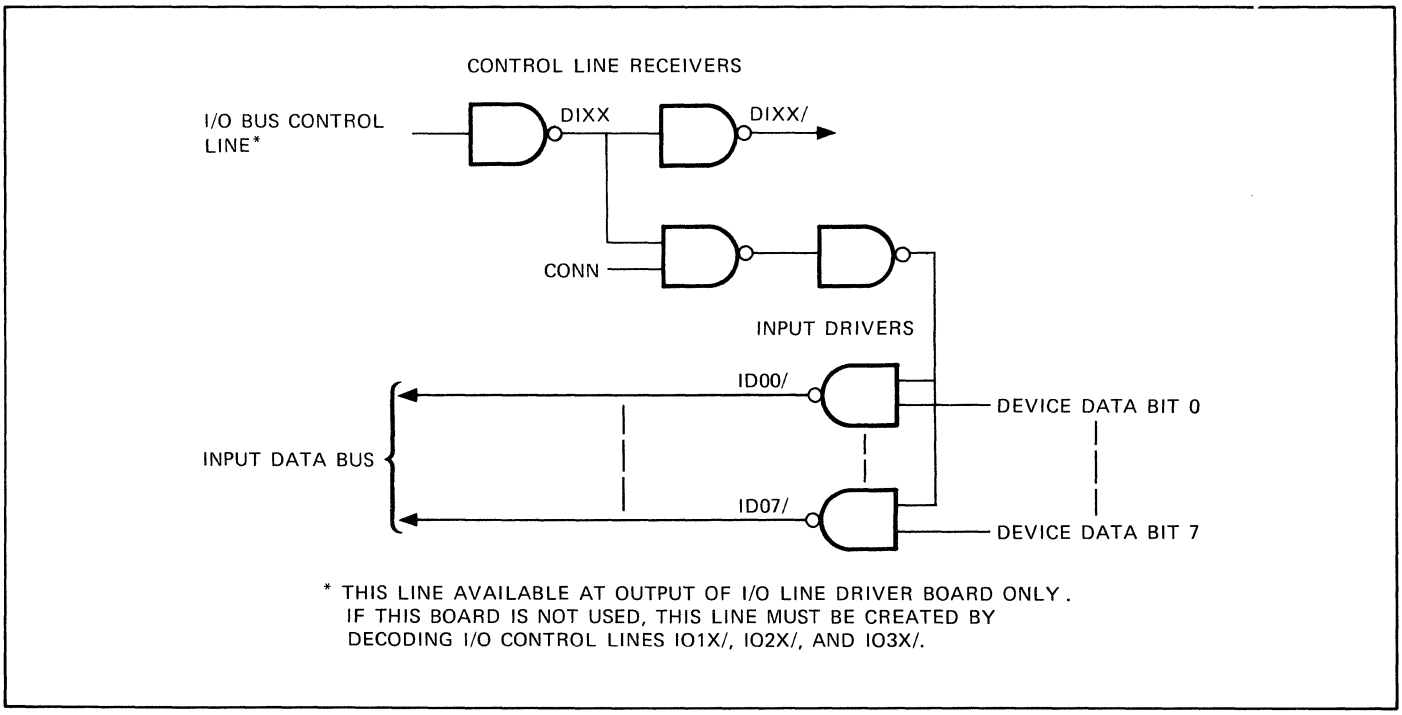


Figure 3-7. Data Input Logic

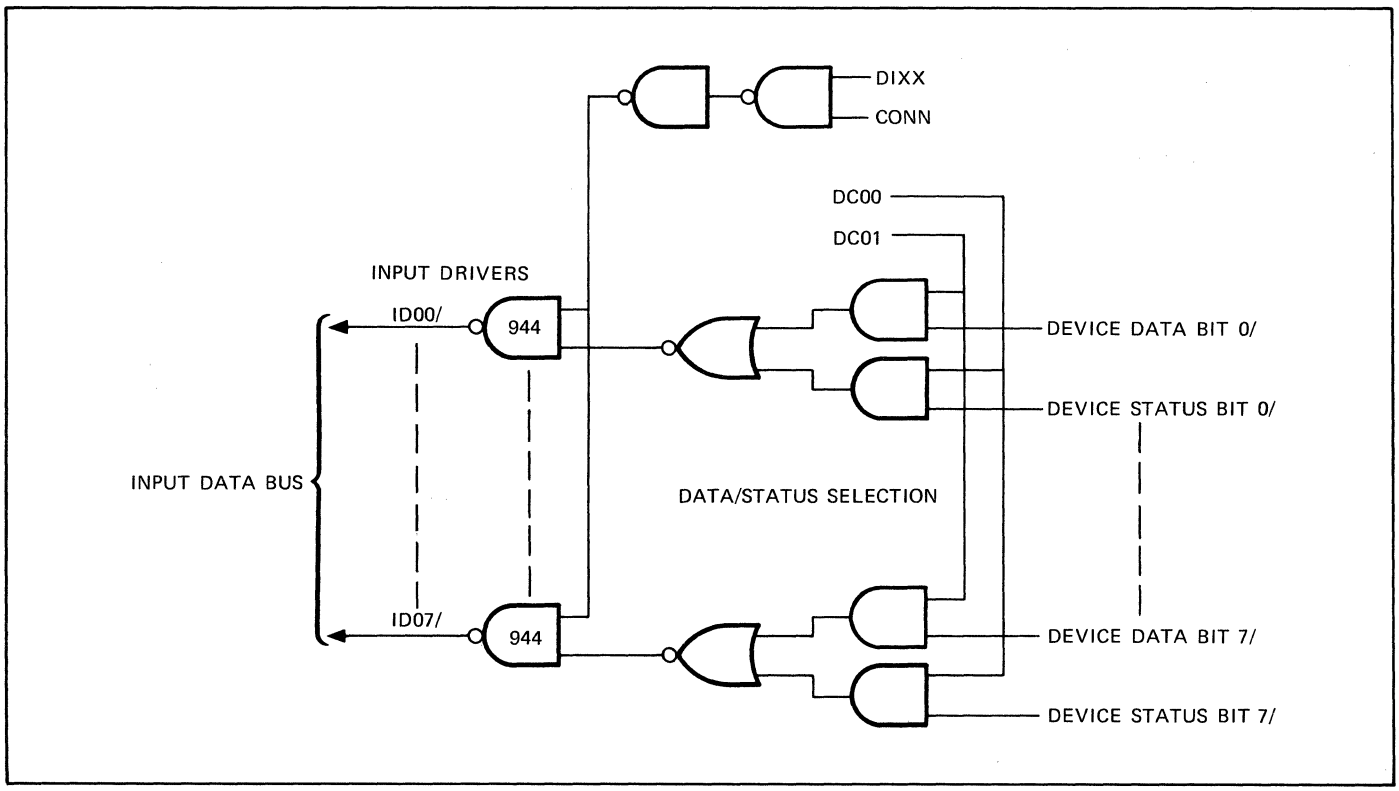


Figure 3-8. Status Input Logic

The typical transfer rate using the sequence of instructions shown in figure 3-9 is approximately 19 kilobytes per second. This rate is determined from the sum of execution times of all instructions (52.8 microseconds). The rate is valid only if the device is ready when the first status check is made.

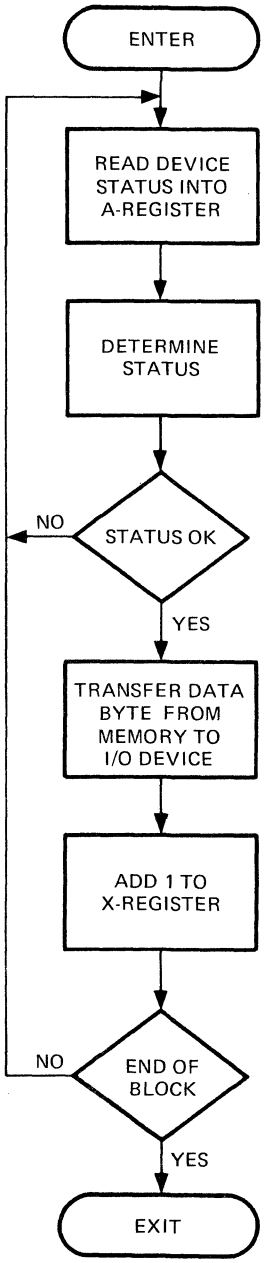
The first operation performed in the sequence is status input to the A-register. An Input Byte to A (IBA) instruction with a suitable device address is used for this purpose. Next, the status bits are checked using AND and JAZ instructions to determine if the device is ready for the data transfer. This check is important in any program-controlled data transfer. Device data transfers typically cannot be performed without reference to some condition of the device (a status bit) that indicates its readiness to transfer or accept data.

Status checking continues until the device indicates it is ready to receive data. A data byte is then transferred to the device by an Output Byte from Memory (OBM) instruction. Following the transfer the X-register, acting as a data pointer, is updated and tested for zero by INX and NXZ instructions. In this example the X-register initially holds a negative count specifying one more than the number of bytes to be transferred. As each byte is transferred one count is added to the register. When the last byte has been transferred, the X-register contains zero and the subroutine exits to the main program.

The data rate of this sequence can be increased if the device status checking steps are eliminated (assuming the device is always ready to accept data). The maximum data transfer rate for a sequence of this kind is 34 kilobytes per second (based on instruction execution times totalling 29.4 microseconds). The instruction sequence would consist of OBM, INX, and JMP instructions.

3.25 CONCURRENT I/O OPERATION

Concurrent I/O operation is the name given to the block transfer technique used in the CIP/2100 Computer. The software program sets up starting and stopping addresses for the block transfer in dedicated memory locations; thereafter, firmware controls the data transfer operation automatically on demand from the device controller. In terms of controller design, the additional logic required for concurrent I/O operations can be thought of as an overlay to the program-controlled logic discussed earlier. All that is required is additional logic to recognize a data ready condition and assert a concurrent I/O request at that time instead of waiting for status sampling as is the case with program-controlled operations. Another way of saying this is: A device performing a concurrent I/O operation initiates its own data transfers when it is ready.



INSTRUCTION	EXECUTION TIME
INPUT BYTE TO A (IBA)	8.36 μS
AND (ANA)	5.50 μS
JUMP IF A EQUAL TO ZERO (JAZ)	8.80 μS (CONDITION MET) 7.70 μS (CONDITION NOT MET)
OUTPUT BYTE FROM MEMORY (OBM) (INDEXED)	15.84 μS
INCREMENT X (INX)	7.04 μS
JUMP IF X NOT EQUAL TO ZERO (NXZ)	8.36 μS (CONDITION MET) 7.48 μS (CONDITION NOT MET)

Figure 3-9. Typical Program-Controlled Transfer Sequence

3.26 Concurrent I/O Timing

The timing diagram for a typical concurrent I/O operation is shown in figure 3-10. When the device controller is ready to transfer data, it causes the concurrent I/O request line (ECIO/) to go low. After recognizing the request, firmware in the processor causes it to respond by lowering the I/O acknowledge line (IOAK/). While line IOAK/ is low, the device controller applies an address byte containing its own address times 2 and a bit indicating the direction of transfer (input or output) to input data lines ID00/ through ID07/.

Note

For proper concurrent I/O operation, the controller must supply its own address times 2. For example:

Actual Device Address (Hex)	Address Supplied During IOAK/ (Hex)
00	00
01	02
02	04
.	.
.	.
.	.
1F	3E

The most significant bit of the address byte (ID07/) indicates the direction of transfer: Bit 07 is turned on (ID07/ is grounded) for output and off (high) for input.

The address byte is used by the firmware routine to proceed with a normal data input or data output operation. If an input operation was specified the processor drops data input line DIXX/, and the controller responds by applying a data byte to input data lines ID00/ through ID07/. If an output operation was specified the processor applies a data byte to output data lines T00X/ through T07X/ (OD00/ through OD07/) and drops data output line DOXX/. The controller responds by capturing the data byte.

3.27 Typical Concurrent I/O Logic

Since there is no transfer of control bytes to a controller during concurrent I/O operations, logic must be included in the controller to force it into the connected state. Once the forced connection is made, the second phase of the data transfer operation occurs normally.

Figure 3-11 shows the additional logic required for concurrent I/O operations. The concurrent I/O request line (ECIO/) and the I/O acknowledge line (IOAK/) were described in the preceding paragraphs.

The addition of priority input (PRI1/) and priority output (PRO1/) lines are shown in the logic. Signal PRI1 is used to condition the concurrent I/O request along with the I/O acknowledgement. The priority input signal is used to ensure that the device controller has priority before requesting a concurrent I/O operation and connecting for service. A description of priority determination is given in paragraph 3.29.

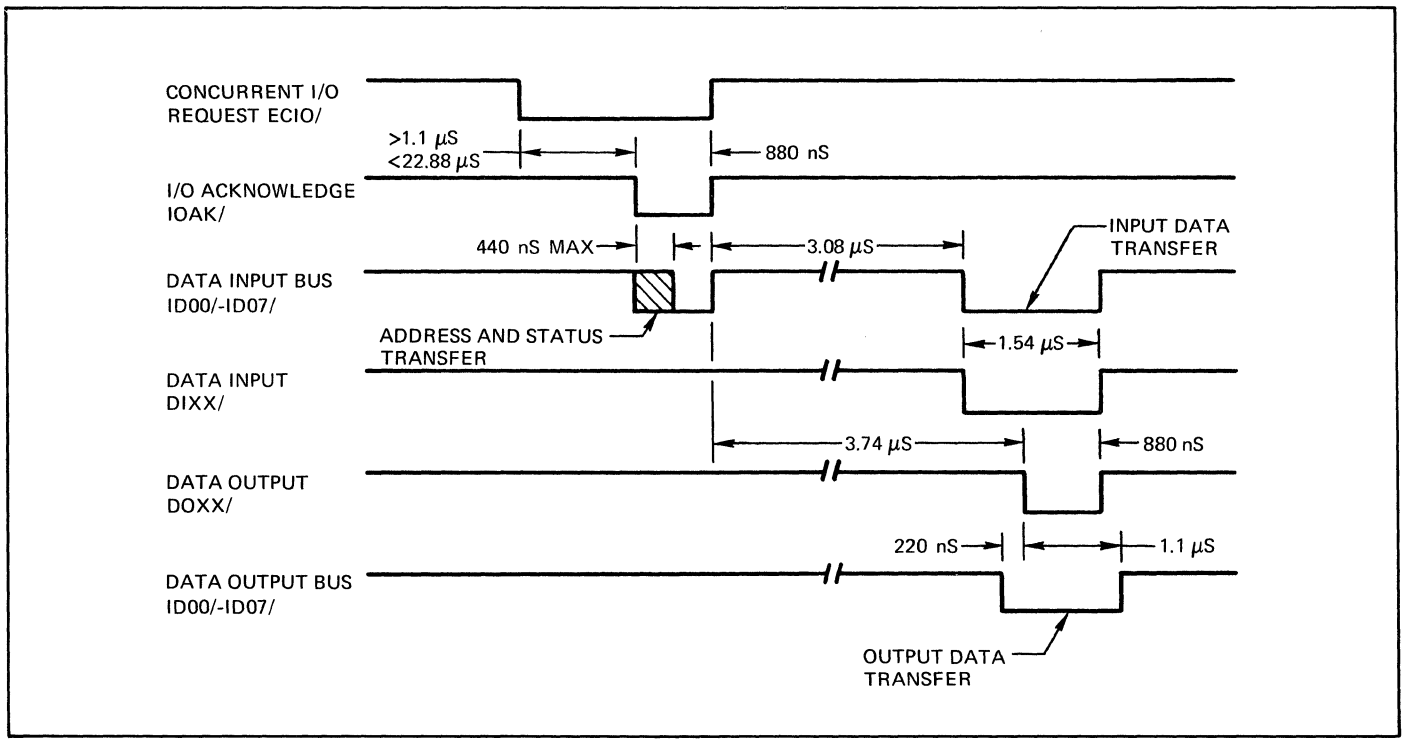


Figure 3-10. Concurrent I/O Timing

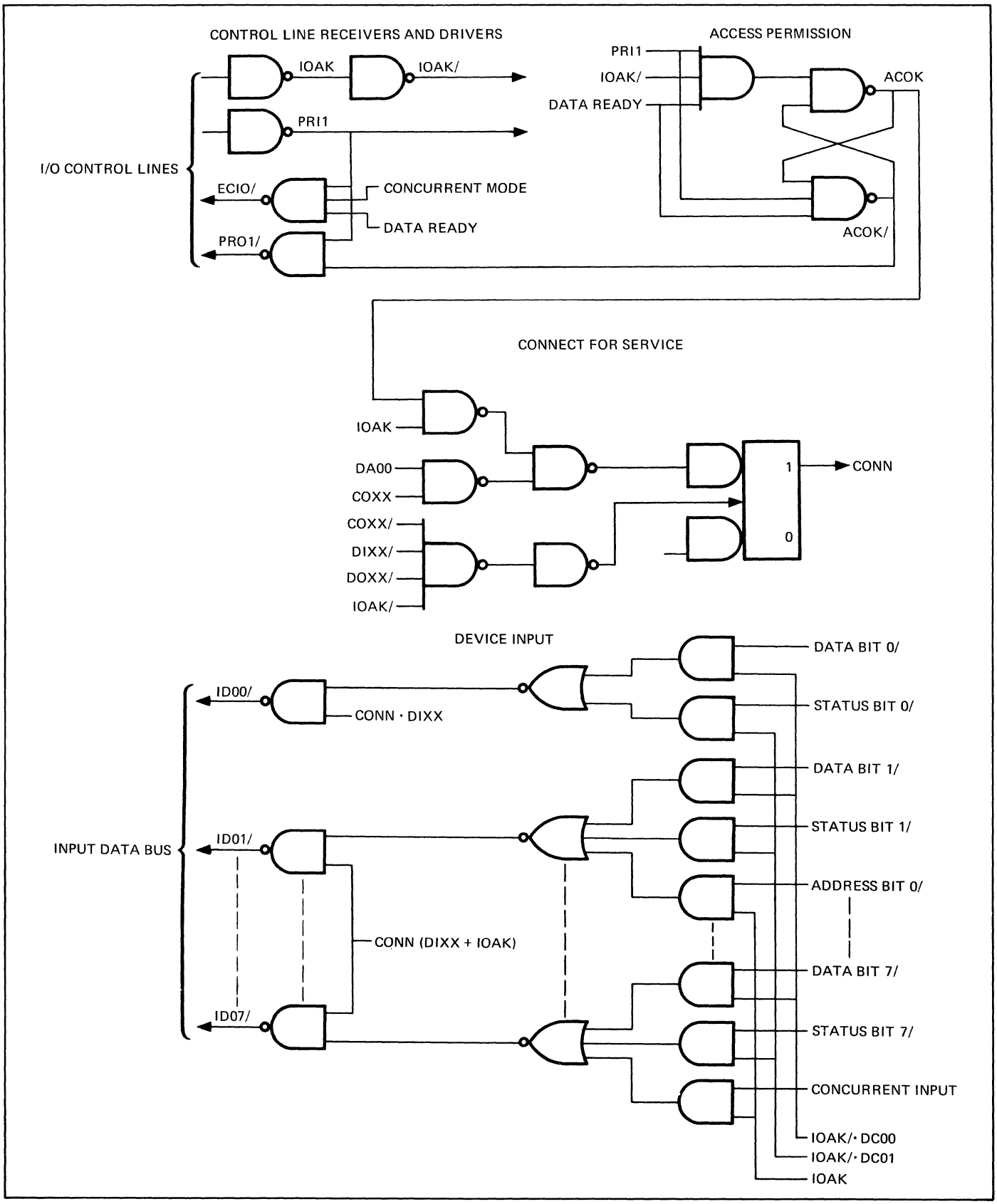


Figure 3-11. Concurrent I/O Logic

Latch ACOK defines the state when concurrent access is allowed. It is set with PRI1, IOAK/, and data ready signals to ensure that higher priority controllers do not interfere with a lower priority controller that has begun an access operation. Signal ACOK must condition address gating to the processor and all state advances in the controller; for example, advancing to the connect-for-service state.

The driving logic for the input data lines has been expanded in the example to include address transfer capability and a flag (ID07/) for indicating direction of transfer.

3.28 EXTERNAL INTERRUPT OPERATION

Interface lines PRO1, PRI1/, EINT/, IOAK/, and ID00/ through ID07/ are used by device controllers or optional Priority Interrupt interface boards on the byte I/O bus for external interrupt operations. Lines PRO1/ (paragraph 3.7) and PRI1/ (paragraph 3.5) make up the hard-wired priority chain that determines the relative priority of each controller and Priority Interrupt board on the byte I/O bus. These lines determine priority for both interrupt and concurrent I/O operations. Lines EINT/ (paragraph 3.5) and IOAK/ (paragraph 3.8) carry the interrupt request and acknowledgement, respectively, between the interface units and the processor. Input data lines ID00/ through ID07/ carry an interrupt address byte from the interrupting interface unit to the processor in response to the I/O acknowledgement signal on line IOAK/. The interrupt address byte is used by the processor to locate the entry address in core memory page 1 of the interrupt servicing subroutine.

3.29 Priority Determination

Interface units on the byte I/O bus are assigned priority for control of external interrupt and concurrent I/O operations. The priority is achieved in the way that lines PRO1/ and PRI1/ are used to link the interface units together. A typical example of priority wiring is shown in figure 3-12. In this example, three controllers in the mainframe chassis and four controllers in the expansion chassis are connected in the priority chain. The I/O Line Driver and Receiver board serves only to pass priority from the mainframe chassis to the expansion chassis and is not a functional part of the priority chain. As shown in the figure, the priority of an interface unit in the chain is not necessarily the same as the physical order of the unit on the I/O bus.

Before a controller can make an interrupt or concurrent I/O request, it must receive priority from the next higher priority controller on the chain in the form of a ground signal on the priority line. The ground signal indicates that the controller now has priority to make a request. If a controller has no request to make, it places a ground signal on the priority line to the next lower priority controller on the chain. A controller never passes priority along to a lower priority controller while making a request. By the same token, a controller must not remove priority from a lower priority controller (make the priority line high) during I/O acknowledgement (IOAK/). Failure to follow these guidelines will result in improper operation.

3.30 Interrupt Requests

External interrupt requests from interface units are carried on line EINT/ to the processor where they appear in bit 7 of file register 0. The internal microprogram recognizes the presence of an external interrupt request by sensing bit 7 and then responds as dictated by interrupt handling firmware.

External interrupt line EINT/ can be used both by device controllers and by optional Priority Interrupt interface boards. The Priority Interrupt option provides the proper interface to the I/O bus, contains priority logic for each interrupt level, and permits processor control over the handling of interrupts. This standard option thus provides, on one circuit board, convenient hardware for eight levels of system interrupts. Because the basic interrupt facility makes use of the byte I/O bus, all device controllers have access to the interrupt request line and can react to the firmware interrupt handling sequences in the processor, provided they operate according to the design guidelines given in the following paragraph.

Note

Requesting an interrupt removes priority from all controllers lower on the priority chain for both interrupt and concurrent I/O operations.

3.31 Interrupt Sequence and Timing

Figure 3-13 shows the timing for a typical external interrupt sequence. The firmware, processor, and I/O device operation during the sequence is as follows:

a. The I/O device controller lowers line EINT/ to signal a request for microprogram attention.

b. At the end of the macro instruction currently being executed, the microprogram senses the interrupt request and jumps to a firmware subroutine to handle the request.

c. The microprogram lowers line IOAK/ to acknowledge the request.

d. In response to the acknowledgement the device controller places an eight-bit interrupt address on input data lines ID00/ through ID07/. The interrupt address specifies the location in core memory page 1 of the two-byte entry address for the interrupt servicing subroutine.

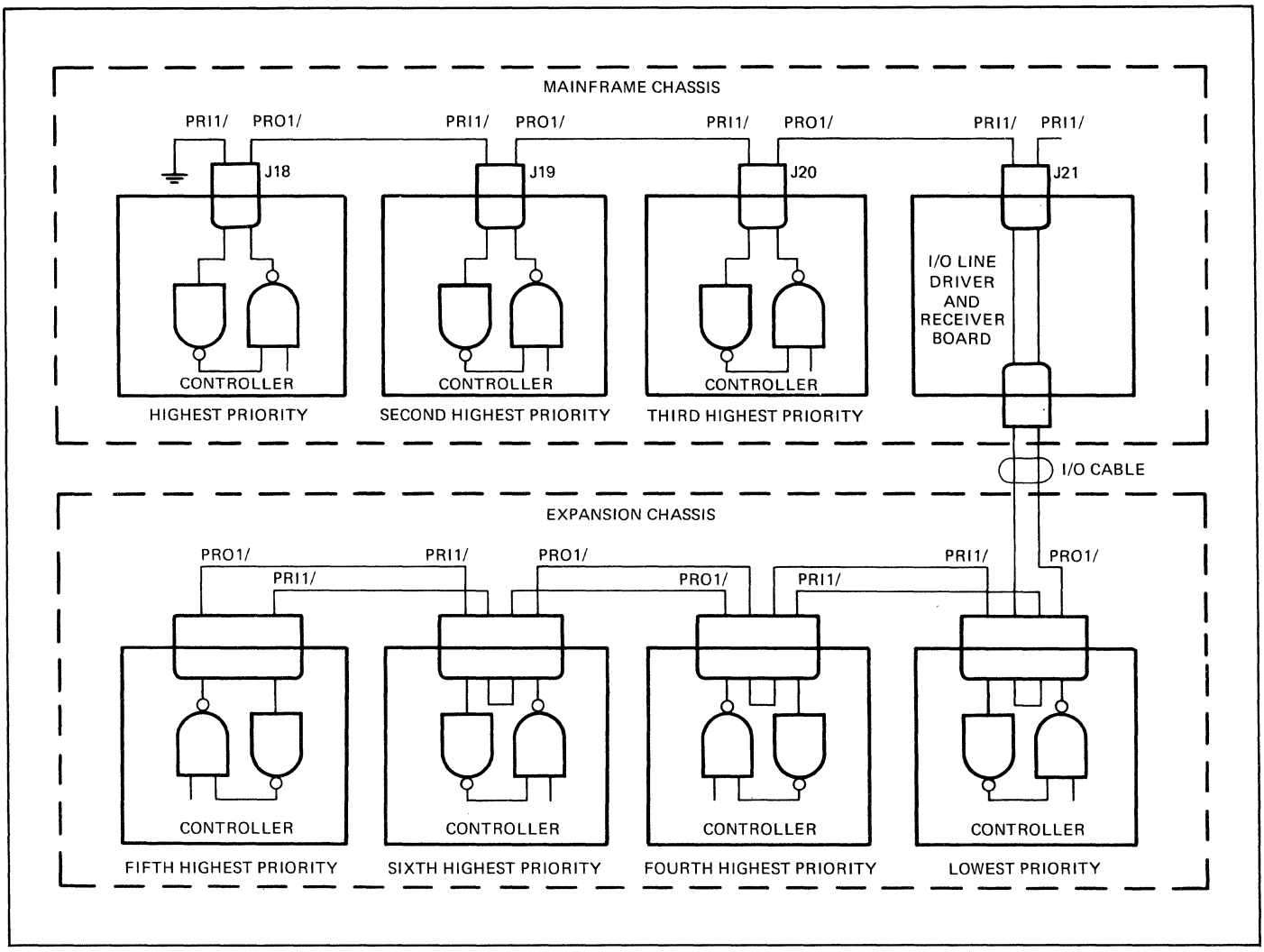


Figure 3-12. Typical Priority Wiring Scheme

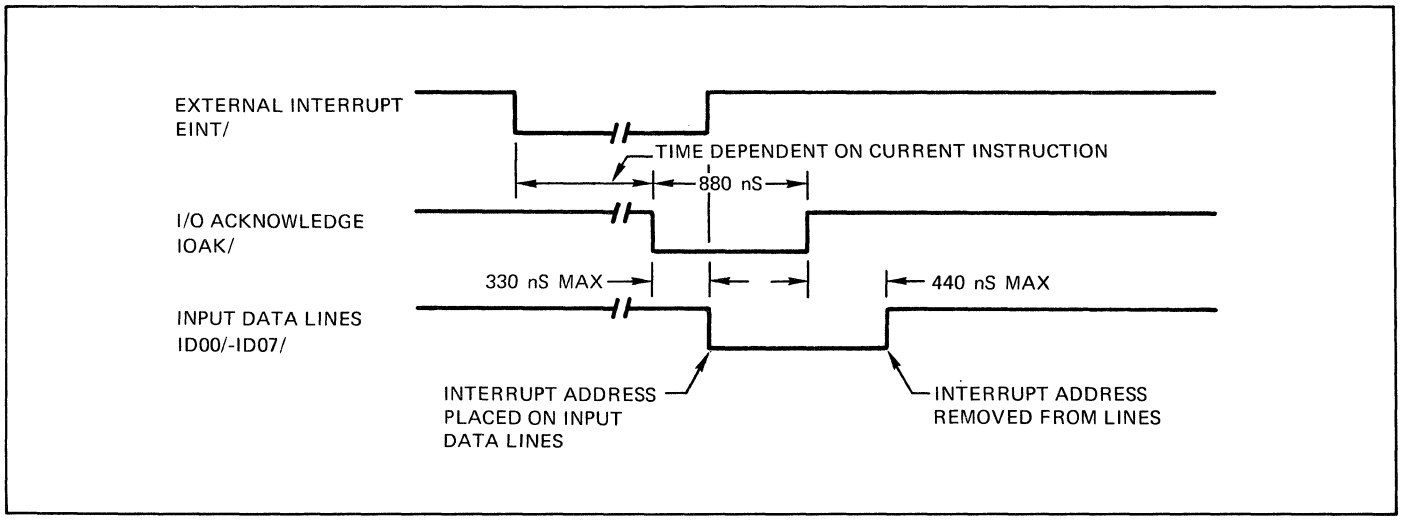


Figure 3-13. External Interrupt Timing

e. The microprogram accepts the eight-bit interrupt address and raises line IOAK/.

f. The microprogram fetches the two-byte interrupt subroutine entry address from memory using the interrupt address byte supplied by the controller as the lower eight bits of an address with the upper address bits set to 01 (page 1).

g. Using the two-byte entry address, the microprogram executes a pseudo return jump to the interrupt servicing routine (the macroprogram counter in the file is stored in the first two bytes of the interrupt subroutine).

h. The interrupt servicing subroutine then proceeds to service the interrupt according to the macroprogram.

4. DIRECT MEMORY ACCESS INTERFACE

4.1 INTRODUCTION

The direct memory access (DMA) interface provides the facility for connecting external I/O device controllers to the memory address, data, and control buses through the DMA Selector Channel interface option. Figure 4-1 is a simplified block diagram showing the points of interface between the processor, memory, DMA channels, and I/O devices. The remainder of this section describes the interface between the DMA Selector Channel and external devices; the mainframe interface between the DMA Selector Channels, the processor, and memory is not described in detail. However, a list of the signals available at the mainframe DMA connector (J5, J7, or J9) is provided for reference in table A-3 of appendix A.

4.2 DMA SELECTOR CHANNELS

Each DMA channel is physically contained on a single plug-in circuit board, which can be inserted into mainframe connector J9, J7, or J5. Connector J9 is wired to accept either a DMA Selector Channel or a Memory Expander board (when additional memory is added in an expansion chassis). Connectors J7 and J5 are also wired to accept DMA boards so that two completely independent DMA channels can be added to the computer even though connector J9 is used for memory expansion. However, when connectors J7 and J5 are used for the DMA option, the maximum memory size that can be accommodated in the mainframe chassis is 8K bytes.

As shown in figure 4-1, DMA Selector Channels can be inserted in mainframe connector J5, J7, or J9, but the Memory Expander board can be inserted only in connector J9.

4.3 DMA EXTERNAL INTERFACE LINES

The external DMA interface (between the DMA channel and I/O devices) comprises eight data output lines, eight data input lines, six device status lines, and eight control lines. These lines are described in the following paragraphs.

4.4 DATA OUTPUT LINES

Data output lines MD00/ through MD07/ connect directly to the memory data bus. They are used for programmed control of the devices connected to the DMA channel as well as for transferring data from memory to the devices. Because these lines reflect the operation of the byte I/O interface lines, they permit the use of normal output macro instructions to set up both a DMA channel and the device controllers attached to it. During a direct

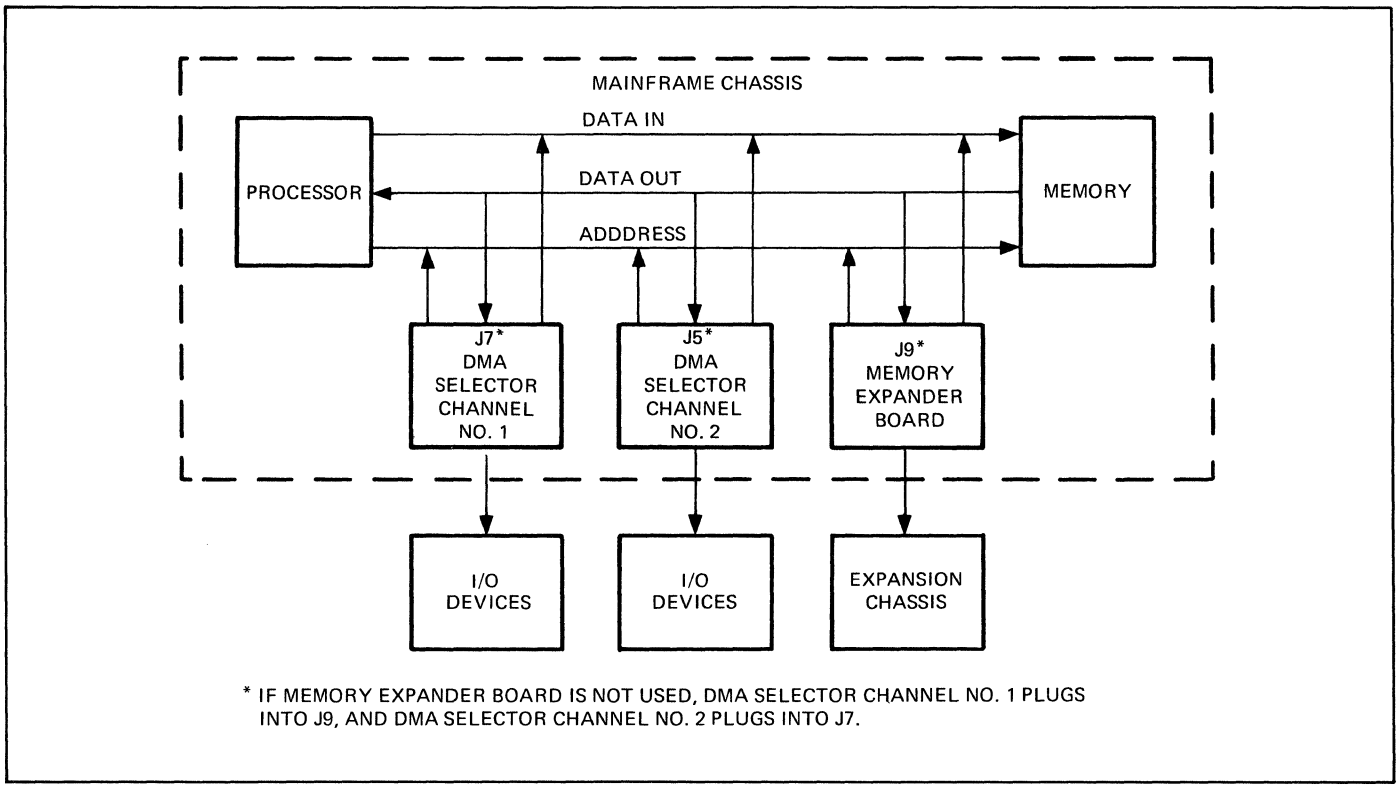


Figure 4-1. Relationship of Processor, Memory, and DMA Channels

memory output operation the lines are sampled by the controller when control line DRDY/ goes low.

4.5 DATA INPUT LINES

Data input lines DD00/ through DD07/ connect to the memory data register in the DMA channel. The data applied to these lines by the controller must be settled when line CREQ/ goes low and must not be removed or changed until line DRDY/ goes low. The DMA channel accepts the data on these lines some time after line EREQ/ goes low.

4.6 DEVICE STATUS LINES

Device status lines DST1 through DST6 are connected through the DMA channel to the memory data bus. The six bits of status information applied to these lines by the device controller are transferred to dedicated memory locations under macroprogram control. The controller applies appropriate status to these lines all during the time it is connected for service and executing a block transfer operation.

4.7 CONTROL LINES

The following eight control lines are available at the external DMA interface:

a. COXX/ – control output

b. DOXX/ – data output

c. MRST/ – master reset

d. EREQ/ – controller access request

e. EWRT/ – external write

f. CRDY/ – channel ready

g. DRDY/ – data ready

h. CSTB/ – start of buffer

Control output line COXX/ carries a low signal from the DMA channel to the controller to specify that a device address is on the data output lines. This line is exactly the same as line COXX/ on the byte I/O bus.

Data output line DOXX/ carries a low signal from the DMA channel to the controller to specify that an information byte is on the data output lines as the result of an output byte operation. This line is identical to the DOXX/ line on the byte I/O interface with the following exception: The data output phase of an output byte operation is subject to alteration if performed during the time either DMA channel is connected and transferring data. For this reason, only function type I/O commands (those that use the COXX phase of the I/O operation) should be sent to device controllers on the DMA channel when the channel is active. This restriction applies only to device controllers connected to the DMA channels.

Master reset line MRST/ carries a low signal from the DMA channel to the controller to clear all control flip-flops to their initialized conditions. This line is identical to control line MRES/ on the byte I/O interface.

Controller access request line EREQ/ carries a low signal from the controller to the DMA channel whenever a direct memory input or output operation is requested by the controller. The line must be held low for a minimum of 250 ns and must be released when line DRDY/ goes low earlier in order to prevent a second memory access immediately after the one requested.

External write line EWRT/ carries a signal from the controller to the DMA channel to indicate the direction of data transfer (input or output). A low signal on the line specifies an input to memory; a high signal specifies an output from memory. The line must be in the required state when line EREQ/ goes low and must remain in that state until line EREQ/ goes low again. As long as this rule is followed, the line can be changed during the transfer of one block of data in order to intermix input and output operations.

Channel ready line CRDY/ carries a low signal from the DMA channel to the controller to indicate that the channel is not in use. The high state of the line indicates that the channel is busy, has been commanded, and is operating in the block mode with some other device. The line stays high during the transfer of multiple blocks of data when a continuous block transfer operation is performed (see paragraphs 4.11 and 4.12).

Data ready line DRDY/ carries a low signal from the DMA channel to the controller to indicate that data is present on the data output lines during a memory output operation, or to indicate that the channel has captured the device data during a memory write operation. The signal is used as a strobe by the controller to capture the data on the data output lines when a memory output operation is being performed. When the signal occurs during a memory input operation, the controller can begin removing or altering data on the data input lines for subsequent accesses. In either case the signal lasts for 440 ns.

Start of buffer line CSTB/ carries a low signal to indicate the start of each new buffer transfer by the DMA channel. This signal is useful for identifying the starting point of a cyclic buffer transfer (paragraph 4.11). Line CRDY/ stays continuously high as long as the DMA channel is operating in the cyclic mode. Line CSTB/ then identifies the start of each block. The line goes low for approximately 4.4 μs at the address initialization time of each block.

4.8 DMA CONTROL

Each DMA channel uses pairs of control words to define the starting and ending addresses for each block of data to be transferred during a direct memory access operation. Each pair of control words identifies a buffer area in memory into which or from which a direct memory transfer operation is performed. One DMA channel can have up to four pairs of control words stored in memory. Normally only one pair is required for a simple (one block) transfer, but the four pairs can be used to link up to four buffer areas for a continuous block transfer.

Each control word occupies two bytes (16 bits) of core memory. Fifteen bits of each word contain address data. The sixteenth bit of each word is used as a link or interrupt flag. The flag bit in the starting address word is set if a block transfer with linked buffers is to be performed. The flag bit in the ending address word is set if an interrupt is to be generated at the end of the block transfer. This interrupt is an internal one that transfers control to a service routine whose address is stored in memory locations 82 and 83 (hexadecimal). The first instruction executed in the interrupt service subroutine must be Input Status, which resets the interrupt signal.

The four pairs of control words for each channel are stored in dedicated locations in memory by the software program before the start of a block transfer operation. After being initialized by the software program, the channel automatically fetches its control words from memory. Thereafter, the block transfer operation proceeds automatically under control of the channel and device controller.

Figure 4-2 shows the dedicated memory locations allocated to the two DMA channels for storing the control words. The figure also shows the positions of the link and interrupt flag bits in the control words.

Each DMA channel is assigned a device address exactly as an I/O device: DMA channel 1 is assigned I/O address hexadecimal 16, and channel 2 is assigned address hexadecimal 17. In the CIP/2100 Computer, output byte (OBA, OBB, or OBM) instructions are used to set up and communicate with the DMA channels. In the CIP/2000 Computer, microcommands must be executed to simulate the control functions of the output byte macro instructions used in the 2100 Computer. Table 4-1 lists and describes the macro instructions that are used in the CIP/2100 Computer to communicate with the DMA channels.

4.9 DMA CHANNEL OPERATIONS

4.10 SIMPLE BLOCK TRANSFER

The timing diagram for a simple block transfer through the DMA channel is shown in figure 4-3. Line CRDY/ goes high when a device controller is connected and communicating with the channel. Each time the controller is ready to send or receive a data byte it causes line EREQ/ to go low. The controller holds the line low for a minimum of 250 ns and must return the line to the high state by the time line DRDY/ goes low.

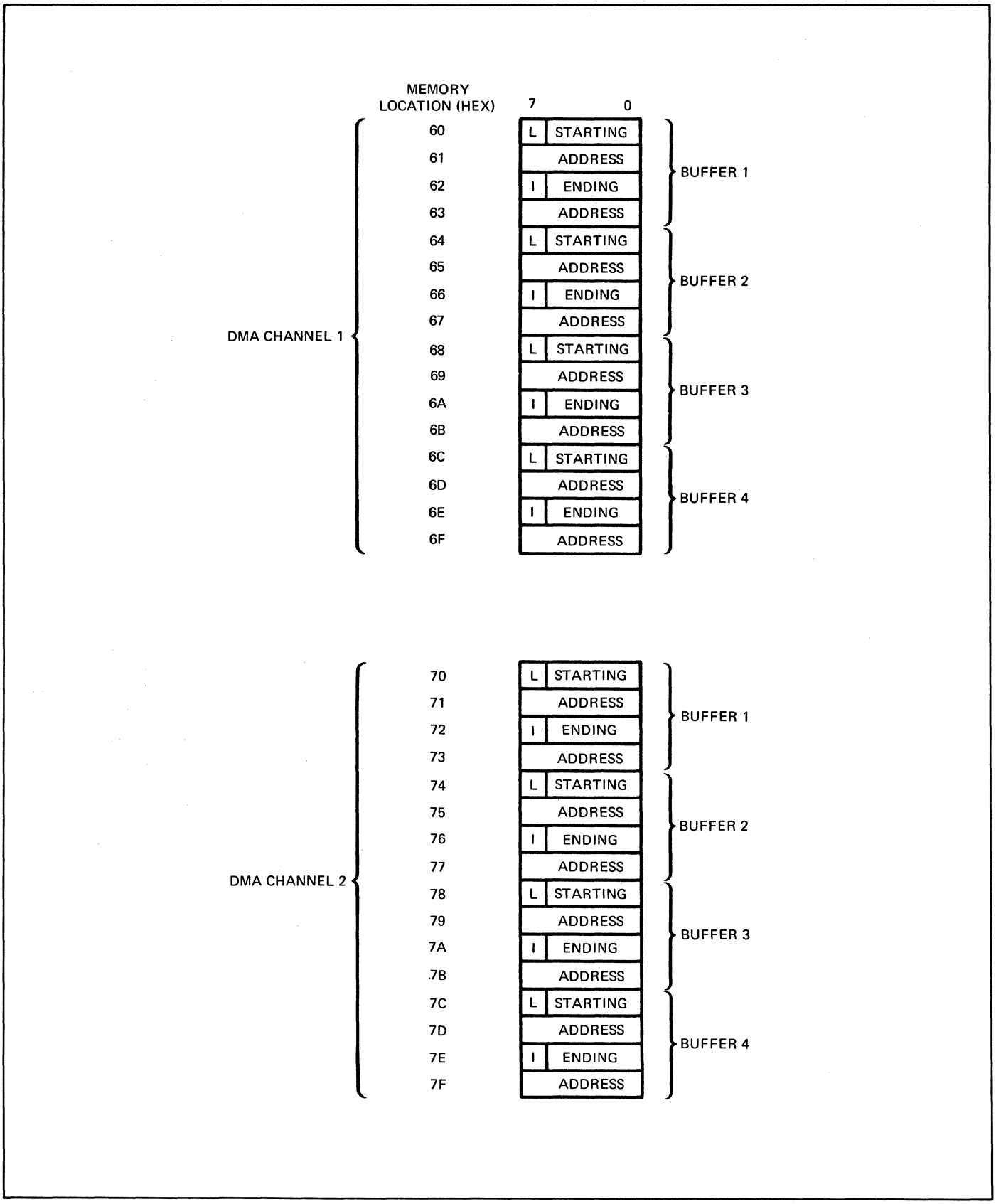


Figure 4-2. DMA Channel Control Word Memory Locations

Table 4-1. CIP/2100 DMA Channel Instructions

Instruction*	Description
3976	Start Channel and Stop at End of Block. The starting and ending address control words are transferred from dedicated memory to the channel. The channel then waits for a device request.
3936	Start Channel – Cyclic Mode. The same as the previous instruction except that the channel continuously transfers from one block or links with up to four blocks.
3956	Stop Channel at Block End. The channel terminates a continuous block transfer at the end of the current block.
3996	Disconnect Channel Immediately. The channel terminates the current block transfer unconditionally.
3916	Input Device Status Directly to Assigned Memory Location. The channel transfers eight status bits directly to dedicated memory locations. Channel 1 status is stored in location 0058_{16}; channel 2 status is stored in location $005C_{16}$. The format of the status byte is shown in figure 4-4.

*Instructions shown are Output Byte from A (OBA) with I/O address of 16_{16}.

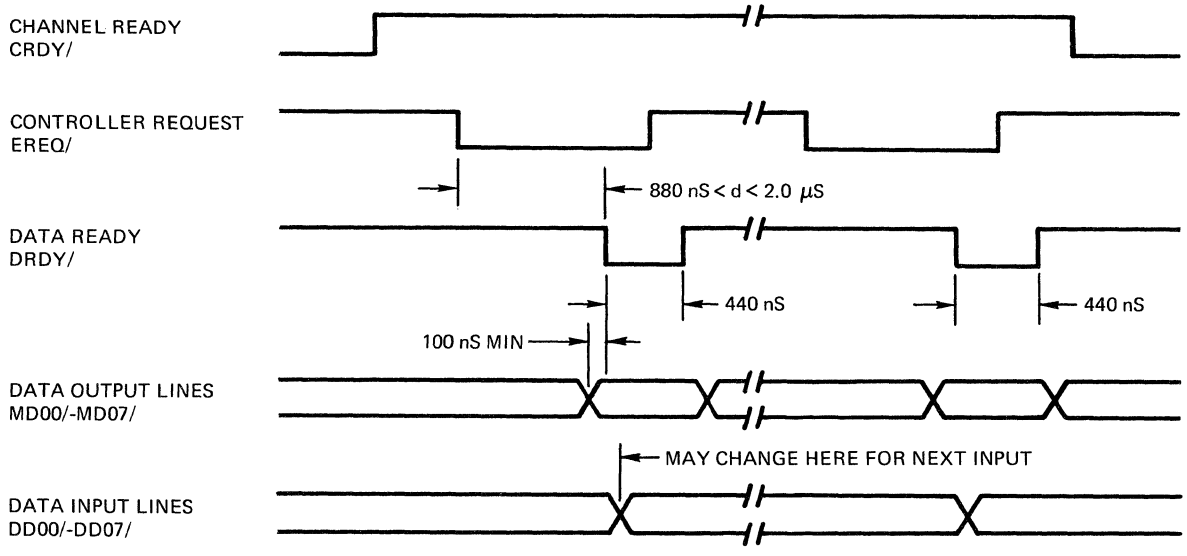


Figure 4-3. DMA Channel Block Transfer Timing

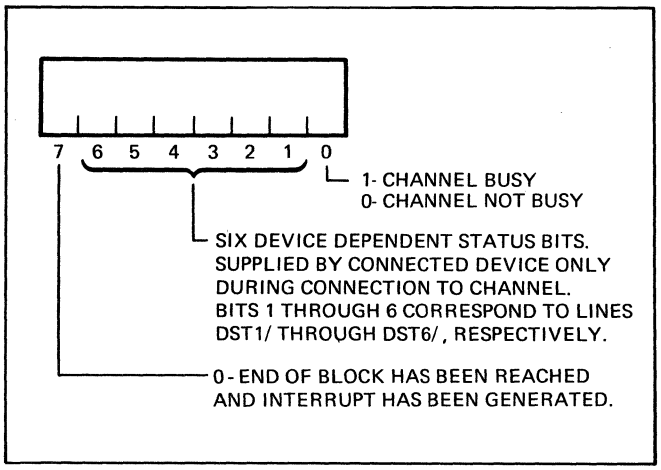


Figure 4-4. DMA Status Byte Format

Line DRDY/ goes low when a data byte is present on the data output lines or when the channel has sampled the data byte on the data input lines. As shown in the diagram, data on the output data lines is valid from approximately 100 ns before line DRDY/ goes low until 200 ns after it goes high again. During this period the data is captured by the device controller. When the controller is transferring data to the channel, it must maintain the data input lines in the appropriate state from the time line EREQ/ goes low until line DRDY/ goes low. The termination of CRDY/ indicates that the block transfer operation is complete. All DMA channel devices must be ready to terminate all operations when line CRDY/ goes to the low state.

4.11 CONTINUOUS BLOCK TRANSFER

The timing diagram shown in figure 4-3 for a simple block transfer is also valid for a continuous (cyclic) block transfer. The only difference is that line CRDY/ does not return to the low state at the end of a block, since the transfer continuously cycles through the same block. Instead, the processor must initiate a disconnect operation. Examples of devices that operate in this mode are CRT controllers with continuous output to some recording media, and other analog-to-digital converters.

4.12 BLOCK TRANSFER WITH LINKED BUFFERS

The timing diagram shown in figure 4-3 applies also to the linked buffer operation, whether the buffers are being handled on a start-stop basis or cyclically. If they are being cyclically transferred, line CRDY/ continues high indefinitely until the processor disconnects. During start-stop operation, the timing for one buffer transfer is equivalent to that shown in the diagram, with the next buffer transfer occurring as directed by macro instruction from the processor. The only difference between linked and unlinked operation is that the block control words are fetched from different memory locations when a linked operation is performed.

4.13 INTERFACE TO DMA CHANNEL

The output from the DMA channel is a multiplexed bus, that is, the drivers can accommodate multiple loads. The input channel is in the form of collector-ORed drivers. All of the input lines described in paragraphs 4.4 through 4.7 must be driven from the controller with DTL power gates (type 944 or equivalent). To maintain channel expandability, the output lines from the DMA channel should be received by DTL or TTL gates with only one load per device.

A single 44-pin connector (P3) on the DMA channel circuit board is used for connecting to I/O device controllers. The signal list for this connector is given in table A-4 of appendix A. If multiple device controllers are to share one DMA channel, they must be connected in a daisy-chain fashion as shown in figure 4-5.

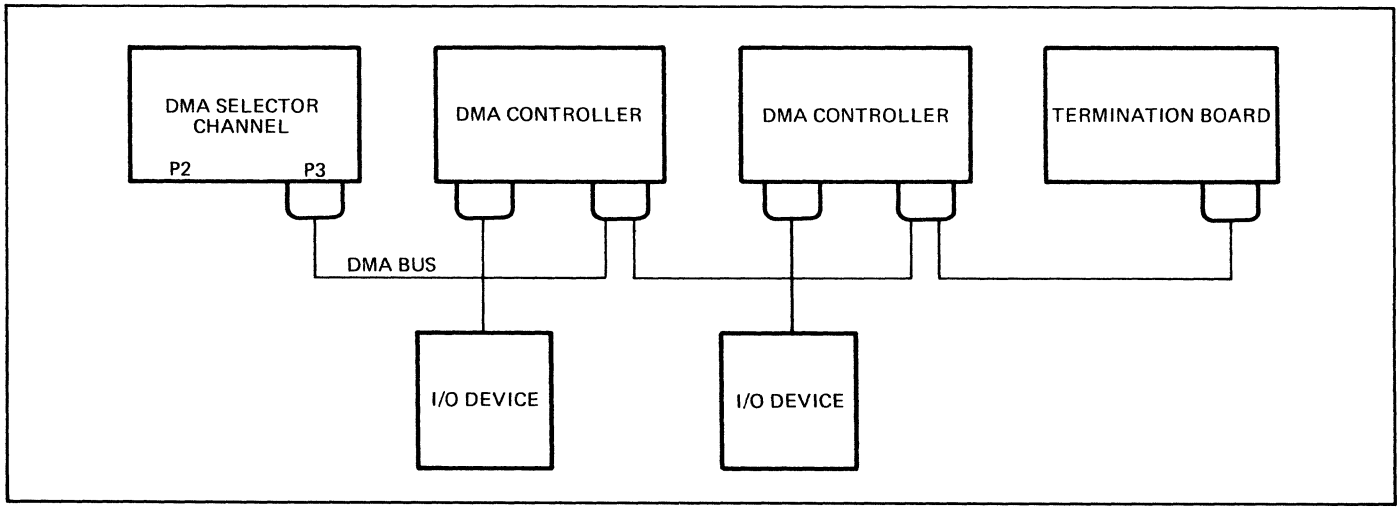


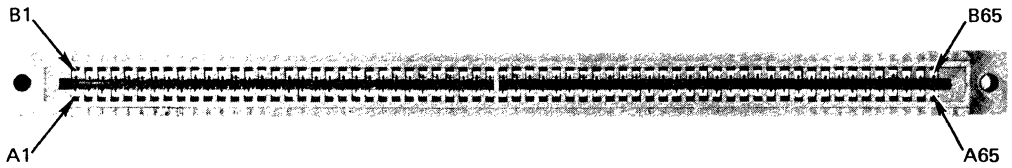
Figure 4-5. DMA Bus Cabling for Multiple Controllers

APPENDIX A
INTERFACE CONNECTOR SIGNAL LISTS

This appendix contains signal lists for the byte I/O and DMA interface connectors. The following tables are included:

Table	Title
A-1	Mainframe Byte I/O Connector Signal List
A-2	External Byte I/O Connector Signal List
A-3	Mainframe DMA Connector Signal List
A-4	External DMA Connector Signal List

Table A-1. Mainframe Byte I/O Connector Signal List



Connectors J18, J19, and J20 – Masterite Part No. 00216-0115

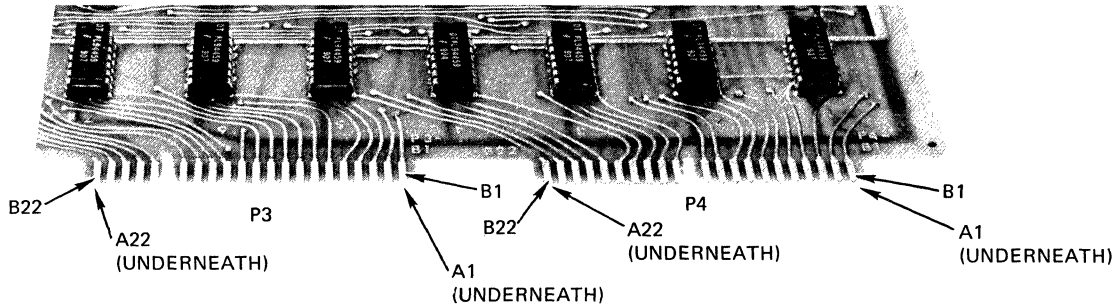
Connector J21 – Masterite Part No. D008AL65-DR-H-BS

Pin	Signal	Pin	Signal
A1	GRD	A12	Not used
A2	GRD	A13	SP5
A3	COXX/ (Reserved – Do not use)	A14	SP4
A4	-6V	A15-A25	Not used
A5	+12V	A26	T02X/
A6	CPH1	A27-A30	Not used
A7	DOXX/ (Reserved – Do not use)	A31	IO2X/
A8	-16.75V/-8.5V	A32	ID04/
A9	Not used	A33-A37	Not used
A10	T05X/	A38	EINT/
A11	Not used	A39-A44	Not used

Table A-1. Mainframe Byte I/O Connector Signal List (Cont.)

Pin	Signal	Pin	Signal
A45	T08X/	B26	T06X/
A46-A54	Not used	B27-B30	Not used
A55	PRO1/	B31	IO1X/
A56	Not used	B32	ID00/
A57	Not used	B33-B36	Not used
A58	T07X/	B37	T04X/
A59	Not used	B38	EDPR/
A60	ID01/	B39	T00X/
A61	ID06/	B40	Not used
A62	ID03/	B41	Not used
A63	Not used	B42	ECIO/
A64	GRD	B43	Not used
A65	GRD	B44	MRST/
B1	+5V	B45	ID08/
B2	+5V	B46-B49	Not used
B3	Not used	B50	ERPY/
B4	IOAK/ (Reserved – Do not use)	B51-B53	Not used
B5	DIXX/ (Reserved – Do not use)	B54	PRI1/
B6	Not used	B55	PROT/ (Reserved – Do not use)
B7	LRXX	B56	Not used
B8	KOXX/ (Reserved – Do not use)	B57	PRIN/ (Reserved – Do not use)
B9	Not used	B58	T03X/
B10	T01X/	B59	ID05/
B11	SP7	B60	ID07/
B12	SP6	B61	IO3X/
B13-B21	Not used	B62	ID02/
B22	CPH2/	B63	Not used
B23	Not used	B64	+5V
B24	CGOX/	B65	+5V
B25	Not used		

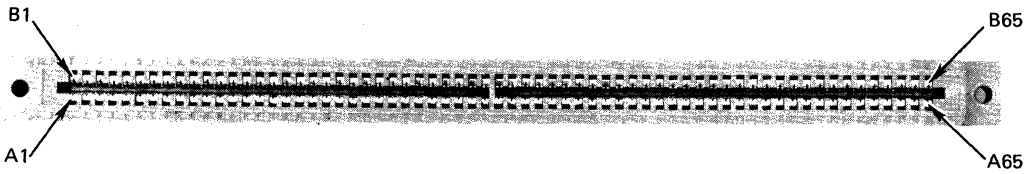
Table A-2. External Byte I/O Connector Signal List



Mating Connectors for P3 and P4 – Masterite Part No. D008GR22-DR-H-X

P3				P4			
Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1-A12	GRD	B7	IOAK/	A1-A11	GRD	B11	ECIO/
A13	Not used	B8	DIXX/	A12	Not used	B12	Key slot
A14	SP4	B9	EINT/	A13-A22	GRD	B13	MRES/
A15	SP5	B10	COXX/	B1	ID08/	B14	OD07/
A16	SP6	B11	DOXX/	B2	ID03/	B15	OD04/
A17	Not used	B12	PRI1/	B3	ID02/	B16	OD03/
A18	BRHF	B13	TSFT	B4	ID06/	B17	OD00/
A19	SRTE	B14	GRD	B5	ID01/	B18	OD08/
A20	RRDY	B15	GRD	B6	ID07/	B19	OD02/
A21	RERR	B16	GRD	B7	ID05/	B20	OD06/
A22	SP7	B17	Key slot	B8	ID04/	B21	OD01/
B1	ERPY/	B18	RR04	B9	ID00/	B22	OD05/
B2	EDPR/	B19	RS02	B10	PRO1/		
B3	SP2/	B20	PUL1				
B4	SP3/	B21	PUL2				
B5	KOXX/	B22	WSFT/				
B6	SP1/						

Table A-3. Mainframe DMA Connector Signal List

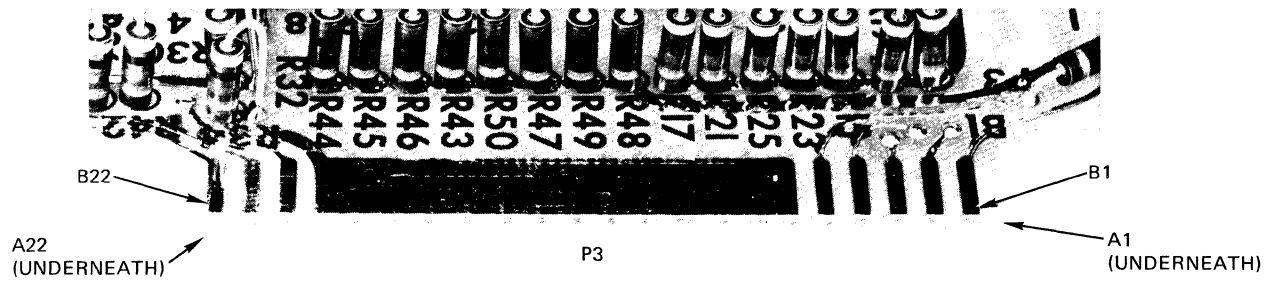


Connector J7 – Masterite Part No. 00216-0115

Connector J9 – Masterite Part No. D008AL65-DR-H-BS

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	GRD	A36	MD04	B1	+5V	B33	GRD
A2	GRD	A37	GRD	B2	+5V	B34	MD00
A3	-16.75V/-8.5V	A38	MD08	B3	-16.75V/-8.5V	B35	GRD
A4	-16.75V/-8.5V	A39	GRD	B4	-16.75V/-8.5V	B36	MD04
A5	+12V	A40	MD06	B5	+12V	B37	GRD
A6	-6V (J7 only)	A41	GRD	B6	-6V (J7 only)	B38	MD08
A7	Priority In	A42	MD02	B7	PRI Out (S1/)	B39	GRD
A8-A10	GRD	A43	CPH1	B8-B10	GRD	B40	MD06
A11	Not used	A44	MPAF/	B11	M01A/	B41	GRD
A12	M02A/	A45	IO1X/	B12	Not used	B42	MD02
A13	Not used	A46	IO2X/	B13	M03A/	B43	MRST/
A14	M00A/	A47	IO3X/	B14	Not used	B44-B47	Not used
A15	Not used	A48	DMAW/	B15	N06A/	B48	DMAR/
A16	N07A/	A49	CPH2/	B16	Not used	B49	DMAS/
A17-A19	Not used	A50	N03A/	B17	M06A/	B50	DMAH/
A20	WTXX/	A51	MBSY	B18	M05A/	B51	N00A/
A21-A23	Not used	A52	N04A/	B19	M04A/	B52	Not used
A24	READ	A53	SPIT/	B20	RTXX/	B53	N01A/
A25	GRD	A54	N05A/	B21-B23	Not used	B54	Not used
A26	MD07	A55	AENI	B24	MD03	B55	N02A/
A27	GRD	A56-A58	GRD (J7 only)	B25	GRD	B56-B58	GRD
A28	MD09	A59	+12V	B26	MD07	B59	+12V
A29	GRD	A60	-6V (J7 only)	B27	GRD	B60	-6V
A30	MD05	A61	Not used	B28	MD09	B61	Not used
A31	GRD	A62	-16.75V/-8.5V	B29	GRD	B62	-16.75V/-8.5V
A32	MD01	A63	-16.75V/-8.5V	B30	MD05	B63	-16.75V/-8.5V
A33	GRD	A64	GRD	B31	GRD	B64	+5V
A34	MD00	A65	GRD	B32	MD01	B65	+5V
A35	GRD						

Table A-4. External DMA Connector Signal List



Mating Connector for P3 – Masterite Part No. D008GR22-DR-H-X

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
A1	CRDY/	A8	DST5/	A14	MD00/	A20	DD00/	B4	MRST/
A2	COXX/	A9	MD06/	A15	MD01/	A21	DD04/	B5	CSTB/
A3	DST4/	A10	MD05/	A16	MD02/	A22	DD07/	B6-B19	GRD
A4	DST6/	A11	MD07/	A17	DD03/	B1	DOXX/	B20	EREQ/
A5	DST2/	A12	MD04/	A18	DD02/	B2	DRDY/	B21	DD06/
A6	DST1/	A13	MD03/	A19	DD01/	B3	EWRT/	B22	DD05/
A7	DST3/								

APPENDIX B
I/O INTERFACE SIGNAL GLOSSARY

This appendix contains a glossary of interface signals for the byte I/O and DMA interfaces. The list is arranged alphabetically by signal mnemonic. All interface signals are listed in one table. The entry for each signal specifies the interface at which the signal is available, its point of origin, its direction of transfer, and its use.

Note

In table B-1 a virgule (/) appended to a signal mnemonic indicates negation; that is, the signal is low when the function specified by the signal name is occurring.

Table B-1. I/O Interface Signal Glossary

Signal Mnemonic	Interface	Description
COXX/	Byte I/O (external) to I/O device	Control output. Low signal specifying that a control byte containing a five-bit device address and a three-bit device order is on output data lines OD00/ through OD07/. The addressed device accepts the byte. This signal results from decoding I/O control register state 1.
COXX/	DMA channel to I/O device	Same as byte I/O interface except that control byte is available on DMA output lines MD00/ through MD07/.
CPH1	Byte I/O (internal) to I/O device	Processor clock. 4.55-MHz square wave.
CPH2/	Byte I/O (internal) to I/O device	Processor clock. Inverted version of CPH1 delayed by 35 ns.
CRDY/	DMA channel to I/O device	Channel ready. Low signal indicating that DMA channel is not in use. A high signal indicates that the channel is operating in the block mode with another device.
CSTB/	DMA channel to I/O device	Start of buffer. Low signal specifying the start of each buffer transfer by the DMA channel.
DD00/ thru DD07/	I/O device to DMA channel	Data input bits 0 through 7. Data byte transferred from I/O device to DMA channel during direct memory access input operation.

Table B-1. I/O Interface Signal Glossary (Cont.)

Signal Mnemonic	Interface	Description
DIXX/	Byte I/O (external) to I/O device	Data input. Low signal used by a previously addressed I/O device to transfer a data or status byte to the processor on input data lines ID00/ through ID07/. This signal results from decoding I/O control register state 6.
DOXX/	Byte I/O (external) to I/O device	Data output. Low signal specifying that a data byte is on output data lines OD00/ through OD07/. The previously addressed device accepts the byte. This signal results from decoding I/O control register state 2.
DOXX/	DMA channel to I/O device	Same as byte I/O interface except that data byte is available on DMA output data lines MD00/ through MD07/.
DRDY/	DMA channel to I/O device	Data ready. Low signal indicating that a data byte is present on DMA output data lines MD00/ through MD07/ during an output operation, or that the channel has captured the data byte on DMA input data lines DD00/ through DD07/ during an input operation.
DST1/ thru DST6/	I/O device to DMA channel	Device status bits 1 through 6. Six device dependent status bits transferred from I/O device to dedicated memory locations under macroprogram control. These bits reflect the status of the device all during the time it is connected for service and executing a block transfer operation.
ECIO/	I/O device to byte I/O	Concurrent I/O request. Low signal from I/O device requesting a concurrent data transfer. This signal appears in the processor as bit 3 of file register 0 where it acts as an interrupt to the macroprogram. It causes the processor to begin a firmware subroutine for handling a concurrent data transfer.
EDPR/	I/O device to byte I/O	Device protect. Low signal from I/O device during input transfer indicating that device has been programmed to transfer data into a protected area of memory. This signal overrides the optional memory protection error interrupt.
EINT/	I/O device to byte I/O	External interrupt. Low signal from I/O device requesting interruption of the macroprogram. This signal appears in the processor as bit 7 of file register 0 where it initiates a microprogram subroutine for transferring control to a macroprogram interrupt servicing subroutine.

Table B-1. I/O Interface Signal Glossary (Cont.)

Signal Mnemonic	Interface	Description
EREQ/	I/O device to DMA channel	Controller access request. Low signal from I/O device to DMA channel requesting a direct memory input or output operation.
ERPY/	I/O device to byte I/O	I/O reply. Low response signal from I/O device when closed loop I/O operation is required. This signal appears in the processor as bit 5 of file register 0. The signal is not currently used in the CIP/2100.
EWRT/	I/O device to DMA channel	External write. Low signal from I/O device to DMA channel specifying a direct memory input operation. A high signal specifies a direct memory output operation.
ID00/ thru ID08/	I/O device to byte I/O	Data input bits 0 through 8. Nine bits of data, status, or address information transferred from the device to a destination register or memory through the processor B-bus gating (bit 8 is part of spare bit option and is applied to T8 gating).
IOAK/	Byte I/O (external) to I/O device	I/O acknowledge. Low signal from processor in response to an interrupt or concurrent I/O request. The device uses this signal to transfer an address byte to the processor on data input lines ID00/ through ID07/. This signal results from decoding I/O control register state 5.
IO1X/ thru IO3X/	Byte I/O (internal) to I/O device	I/O control bits 1 through 3. Three-bit output from processor I/O control register. States 1, 2, 5, and 6 are decoded to provide signals COXX/, DOXX/, IOAK/, and DIXX/, respectively. State 3 is used for serial Teletype control, and the other states are not currently used.
KOXX/	Byte I/O (external) to I/O device	I/O clock. 2.275-MHz, 50% duty cycle clock derived from clock signals CPH1 and CPH2/.
MD00/ thru MD07/	DMA channel to I/O device	DMA output bits 0 through 7. Data byte transferred from DMA channel to I/O device during a direct memory output operation. Control or data byte transferred to I/O device during program-controlled I/O operation.
MRES/	Byte I/O to I/O device	Master reset. Low signal used by I/O device to clear all control flip-flops to initialized conditions.
MRST/	DMA channel to I/O device	Master reset. Same as signal MRES/ on byte I/O interface.

Table B-1. I/O Interface Signal Glossary (Cont.)

Signal Mnemonic	Interface	Description
OD00/ thru OD08/	Byte I/O (external) to I/O device	Output data bits 0 through 8. Control or data byte transferred to I/O device during program-controlled I/O operation.
PRI1/	Byte I/O to I/O device	Priority in. Low signal indicating that I/O device has priority for interrupt or concurrent I/O operation.
PRO1/	I/O device to byte I/O	Priority out. Low signal indicating that I/O device is passing priority along to next lower priority device.
T00X/ thru T08X/	Byte I/O (internal) to I/O device	Same as OD00/ through OD08/ on external byte I/O interface.

APPENDIX C
TELETYPE MODIFICATION PROCEDURES

C.1 ASR 33 TELETYPE MODIFICATION

If a modified ASR 33 Teletype is not purchased from Cincinnati, a standard ASR 33 Teletype or equivalent* can be modified for compatibility with 2000-series computers using the following procedure:

Note

Teletype Corporation wiring diagrams 6353 WD (sheet 1) or 7833 WD (sheet 1) are helpful, but not mandatory, for the following changes.

a. Remove cover of Teletype.

b. Move purple wire from terminal 8 to terminal 9 of terminal strip 151411 located at lower left of Teletype (viewed from rear).

c. Move white-blue wire from terminal 4 to terminal 5 of the same terminal strip.

d. Move brown-yellow wire from terminal 3 to terminal 5 of the same terminal strip.

e. Move blue wire from terminal 3 to terminal 4 of power resistor 181816. This resistor is located at the center of the right side of the Teletype (viewed from the front with the cover removed). When the cover is in place, the resistor can be seen centered under the removable plate on the right side.

*Assembling programs with the 2100 Computer requires an ASR 33 Teletype with automatic reader-punch controls.

C.2 ASR 35 TELETYPE MODIFICATION

To modify a standard ASR 35 Teletype* for use with 2000-series computers use the following procedure:

a. Remove the top cover from the Teletype.

b. Remove the strap between T6 and T7 of terminal block 151415. This terminal block is located at the right lower rear (viewed from the front) of the cabinet behind the printing mechanism.

c. Modify the computer serial I/O cable as follows:

1. Cut off the connector.

2. Connect the four wires to terminal block 151415 as follows:

Wire	Connect to Terminal Pin No.
White	7
Black	8
Brown	6
Red	5

*The ASR 35 Teletype, as shipped from Teletype Corporation, cannot be used for assembling programs with the 2100 Computer. Automatic reader-punch controls must be installed as modification kits.

10268543