LAL	REV.	DESCRIPTION	CHK	DATE	APPROVED
	A	ENGR. REL.	PMA	8-19-6	PMA
X	B	REVISED & REDRAWN, MFG. RELEASE	BZ	9-14-7	LMK R.Rush
	C	DCR NO. 25351 REVISED: PARA 6.3.4, 7.3.1, 7.3.2.3, 7.3.2.7, 7.3.2.8, 7.3.2.12, 7.4.1, 7.4.3, 7.5 CABLE 2 & 5	CB	4-3-8	RLM LMK CB KDB
X	D	SEE REV. E.O.; REVISED PARA. 6.2.2.6, SH. 29. RETYPE COVER SHEET RLC	BW	6/9/69	
X	E	SEE REV E.O.; REVISED PARA 5.7, SH 19	CB	9/18/69	

REVISIONS — 133991 E

131

NO. REQ.	DRAWING NO.	DESCRIPTION	REFERENCE DESIGNATION	ITEM NO.

MATERIAL LIST

NOTES UNLESS SPECIFIED

1. TOLERANCES .XX ±.030 ANGULAR .XXX ±.010 ±½°
2. BREAK ALL SHARP EDGES .010 APPROX.
3. MACH. SURFACES √
4. ALL DIM. IN INCHES

DRAWN	P. MC AFEE	8-18-66
CHECK	P. Mc Afee	8-19-66
APPR.	L. M.Anderson	8-19-66

SDS
SANTA MONICA, CALIFORNIA

TITLE

PRODUCT DESIGN SPECIFICATION
INPUT/OUTPUT PROCESSOR, SELECTOR

MODEL NO.	SIZE	DWG. NO.	CHANGE LETTER
8485	A	133991	E
NEXT ASSY. 117620	SCALE	DO NOT SCALE DRAWING	SHEET 1 of 70

Table of Contents

1.0 SCOPE

1.1 This specification covers the actions and responses required at the DC/IOP, CPU/IOP, and Memory/IOP interfaces and the design requirements of the free standing selector IOP to be used with the Sigma computer series.

2.0 APPLICABLE DOCUMENTS

2.1 Selector IOP Planning Specification, Sigma Document 182

2.2 I/O System Product Planning Specification, Sigma Document 156

2.3 Specification, Central Processing Unit, Dwg. No. 117336

2.4 Specification, Design 8-Bit Data Path Interface, Dwg. No. 123382

2.5 Specification, Sigma/Delta Core Memory, Dwg. No. 117653

2.6 Specification, Design, Four Byte Data Path Interface, Dwg. No. 137556

2.7 Selector, IOP Specification Addendum, Sigma Document 256

2.8 Revised Selector IOP Product Factoring, Sigma Document 260

3.0 REQUIREMENTS

3.1 IOP Functional Requirements

3.1.1 CPU I/O Instructions

On command from the CPU, the IOP shall perform the required operations associated with the SIO, TIO, HIO, TDV, and AIO instructions.

3.1.2 Device Controller Servicing

The IOP shall perform the operations for the following types of Device Controller service operations:

Service Call

Data Out

Data In

Order Out

Order In

Interrupt

3.1.3 Operating Speed

The Selector IOP shall be capable of exchanging data with a non-interleaved core memory at maximum memory rate (one cycle every 850 nanoseconds). The S/IOP shall be capable of transferring data to and from a device controller at a maximum rate of 1.25 million transfers a second, a transfer may be 1, 2, or 4 bytes wide.

3.2 BASIC FEATURES

The Selector IOP provides for the coupling of the CPU and memories to very high speed peripheral devices. These devices may have data rates that would exceed the bandwidth of the Multiplexing IOP or would use up such a large percentage of that band width as to make it impractical to run any other device concurrently. Capability is built into the IOP which allows it to direct and control I/O operations by executing a command list prepared by the CPU.

The IOP has memory addressing format capability for working with memories as large as 1,048,576 words. However, it will be mechanized to handle memories only as large as 131,072 words (524,288 bytes).

The basic DC/IOP interface has I/O unit addressing capability for up to 136 Device Controllers. However, 32 Device Controllers may be connected and only one Device Controller may be in operation at a time. Concurrent operation of two or more Device Controllers on a single S/IOP is not permitted.

The Selector IOP may operate with Device Controllers having a one-byte, two-byte, or four-byte wide data path.

3.3 OPTIONAL FEATURES

A Selector IOP may be supplied with an optional core memory bus-sharing feature. This optional feature, when installed in the IOP, enables the IOP to time-share a core memory bus with another Selector IOP if the other Selector IOP also has the optional bus-sharing feature.

3.4 IOP INTERNAL OPERATION

The Selector IOP is a passive link between core memory and the Device Controllers connected to the IOP.

Once an I/O operation has been initiated by the CPU, the operation is performed to completion, independently, by the IOP, Device Controller, and Core Memory. The IOP will perform core memory accesses, as required by the Device Controller, and update current

command information. The IOP will also record IOP status for future interrogation by the CPU. During a record, a new command may be loaded by the IOP - making scatter reads and gather writes possible (data chaining). At the end of a record, a new command may be loaded by the IOP and an order sent to the Device Controller - allowing multiple record operations to occur without CPU reinitiation (command chaining). The maximum number of devices that may be attached to a Selector IOP is 32. A Selector IOP allows only one of the several possible devices attached to it to be in transmission operation at a given time. The Selector IOP also allows a Device Controller to exchange a complete record with core memory as the result of a single service request from the Device Controller. The Selector IOP will exchange data with a Device Controller until the Device Controller terminates the exchange or until the entire record has been transmitted as indicated by count done. Limitations on data chaining and number of device controllers that may be connected are functions of device speed, interface width, and cable lengths.

The IOP shall perform the operations described in this section in response to service calls from the CPU or from Device Controllers. Specific interface responses are described in Sections 5.0, 6.0 and 7.0.

3.4.1 CPU Service Calls

The IOP shall respond to service calls from the CPU and perform the following operations: SIO, HIO, TIO, TDV, and AIO. The IOP shall respond in a manner dependent on the state of the IOP such that CPU interference to data transfer between the IOP and core memory is minimized. Each IOP shall have the facility for assignment of the IOP address by the setting of switches on a switch module type LT26 (SDS Part No. 126982), located within the IOP.

SIO

The IOP shall decode and recognize the SIO function and shall decode the IOP address lines. If no address recognition, the IOP's shall pass the Control Strobe along to the last IOP and the Control Strobe shall be returned to the CPU on the Proceed line. If the addressed Selector IOP is busy, (whether or not engaged in service) with a Device Controller as the result of a

previous SIO, the instruction shall be partially executed. (the condition codes shall be sent signifying that the IOP was busy).

If the IOP address is recognized and the IOP is not Busy, the IOP shall access cell 20* for the Device address and the command address. If a memory parity error is detected at this point, the IOP shall terminate the operation and send the Proceed signal with appropriate condition codes. If there is no memory parity error, the IOP transmits the SIO function Indicator and Device address at the Device Controller interface and waits for a response. After receiving the response from the Device Controller the IOP loads cells 20* and 21* if requested by the CPU and sends the appropriate condition codes and Proceed signal to the CPU.

TIO, TDV

These operations are performed in a manner similar to the SIO operation. If the IOP is Busy and currently responding to a Device Controller service request then the IOP partially executes the operation (condition codes are sent signifying the IOP busy state); otherwise, the TIO or TDV operation is normally executed.

HIO

The HIO operation is performed in a manner similar to the SIO operation and is always normally executed. If, when the CPU requests the IOP to perform an HIO operation, the IOP is currently servicing a Device Controller service request then the IOP terminates the Device Controller service and then responds to the CPU request.

AIO

The IOP shall decode and recognize the function. If no interrupt is pending, the IOP's shall pass the Control Strobe along to the last IOP which shall return the strobe to the CPU on the Proceed line with the appropriate condition code settings. The IOP with the highest priority and an interrupt pending shall terminate an ongoing Device Controller service and then pass the AIO function indicator along to the Device Controllers. The IOP shall then wait for the highest-priority Device Controller with interrupt pending to respond with Device/Device Controller address and status. The IOP shall then complete the AIO operation by writing the IOP/Device

*Base 16

Controller/Device address and status in cell 20* and sending condition codes and Proceed signal to the CPU.

3.1.3.2 Device Controller Service Calls

The Selector IOP shall respond to Device Controller service calls as described in Section 7.0 and perform the following operations:

Order Out, Order In, Data Out, and Data In

Data Exchange

The IOP shall perform exchanges of Data between the Device Controller and memory in response to Data In/Out requests from the Device Controller. Since the Device Controller data path may be one, two, or four bytes wide and the memory data path is four bytes wide and also since the Device Controller may exchange any number of bytes with any starting memory byte address, then the IOP shall be responsible for the proper assembly and disassembly of data during these exchanges and for the control of partial-word or full-word writing in core memory during a Data In operation. Data buffering shall be provided to allow for memory port interference, data alignment within the IOP, data chaining, and delays in the IOP data path.

Data Out

During Data Out operations the IOP shall access the memory location determined by the current byte address and load the data into the data buffer.(lookahead is employed to keep the data buffer filled). In response to Device Controller Request Strobes the IOP shall access the data buffer, align the data as required by the state of the byte address and byte count registers, generate odd parity for a one byte data path, and transmit the data to the Device Controller. As a function of the byte count decrementing to zero and the data chaining flag the IOP shall perform data chaining operations. The data exchange shall be terminated if either the Device Controller or IOP signals end data.

*Base 16

Data In

During Data In operations the IOP shall, in response to Device Controller Request Strobes, perform one-byte odd parity checks if requested by the Device Controller and load the data buffer. Detecting that the data buffer is not empty the IOP shall align the data as a function of the state of the byte address and byte count registers, access the core memory location designated by the current byte address, and control partial or full write core memory operations. The IOP shall increment, if a forward operation, or decrement, if a backward operation, the byte address register and shall decrement the byte count register each time a core memory access is made. When the byte count has decremented to zero the IOP shall perform data chaining if the data chain flag is true. The data exchange shall be terminated if either the Device Controller or IOP signal end data.

Order Out

During Order Out operations the IOP shall access the core memory doubleword designated by the command address register, send the order to the Device Controller, store the flags, byte address, and byte count and terminate the operation.

Order In

During Order In operations the IOP shall accept the Operational Status Byte from the Device Controller. The Device Controller may report any of the following conditions in the Operational Status Byte: Transmission Error, Incorrect Length, Chaining Modifier, Channel End, or Unusual End. The IOP shall take action appropriate to the conditions reported by the Device Controller and terminate the operation.

Terminal Orders

The IOP shall terminate each Device Controller service sequence with a Terminal Order. The IOP may report in the Terminal Order any of the following conditions: Interrupt, Count Done, Command Chain, or IOP Halt.

*Base 16

3.5 SYSTEM INTERCONNECTION

The IOP shall interconnect with the CPU, memories, and Device Controllers as shown in Figure 3.1. IOP's shall be connected by cables to the CPU in a trunk-tail fashion for the purpose of interrupt priority determination. Addressing of eight IOP's is permitted. All IOP's connected to a given CPU shall be connected to the memory bank containing cells 20* and 21* for that CPU. However, all IOP's need not be connected to all memories. Device Controllers shall be connected to the IOP in trunk-tail fashion as shown.

The Selector IOP may be connected to Device Controllers with a one, two, or four-byte data path. Figure 3.2 shows, for example, how this may be done with a one and four-byte path. Figure 3.3 shows how bus-sharing Selector IOP's may be connected to core memory.

*Base 16

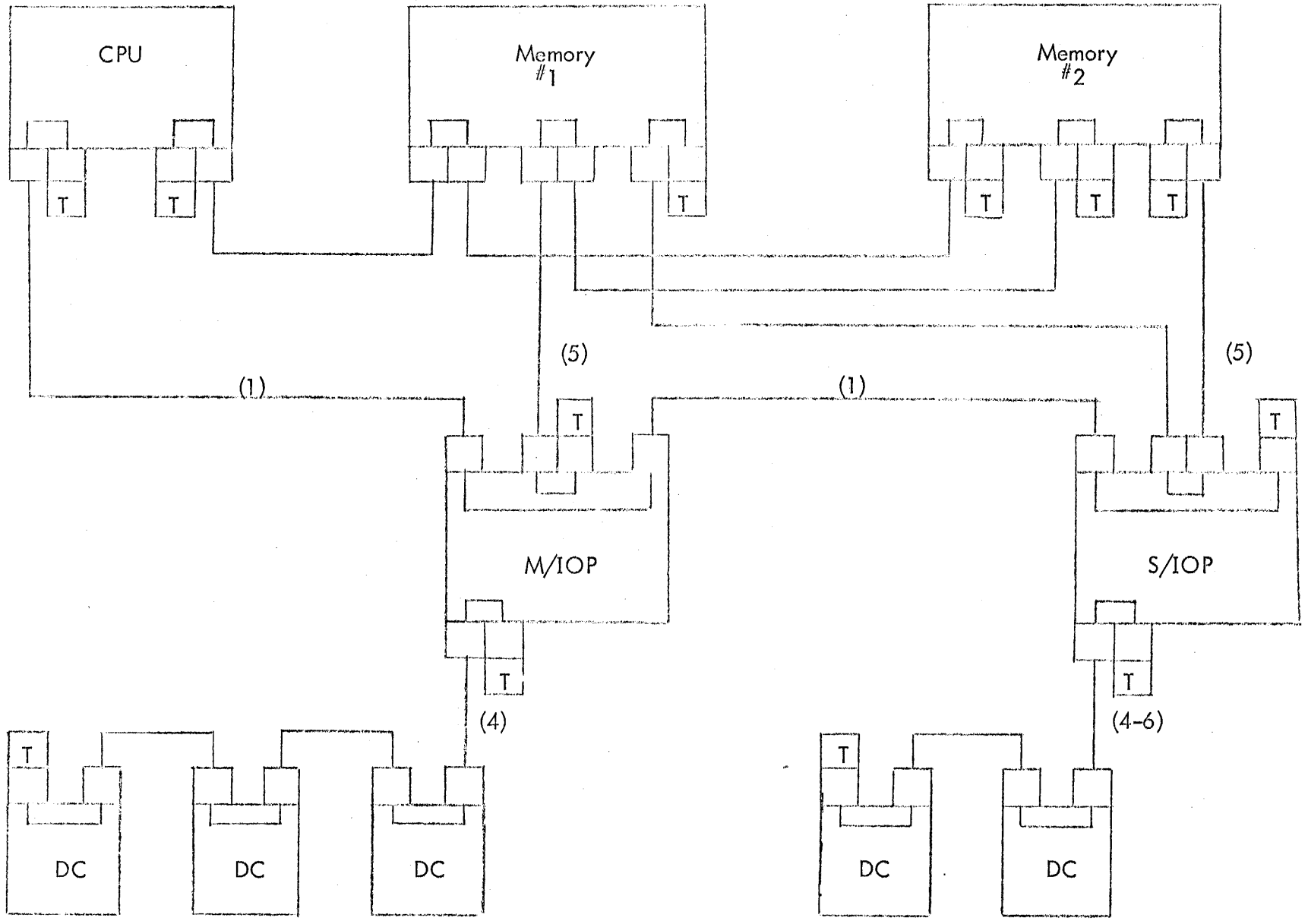


Figure 3.1 SYSTEM INTERCONNECTION

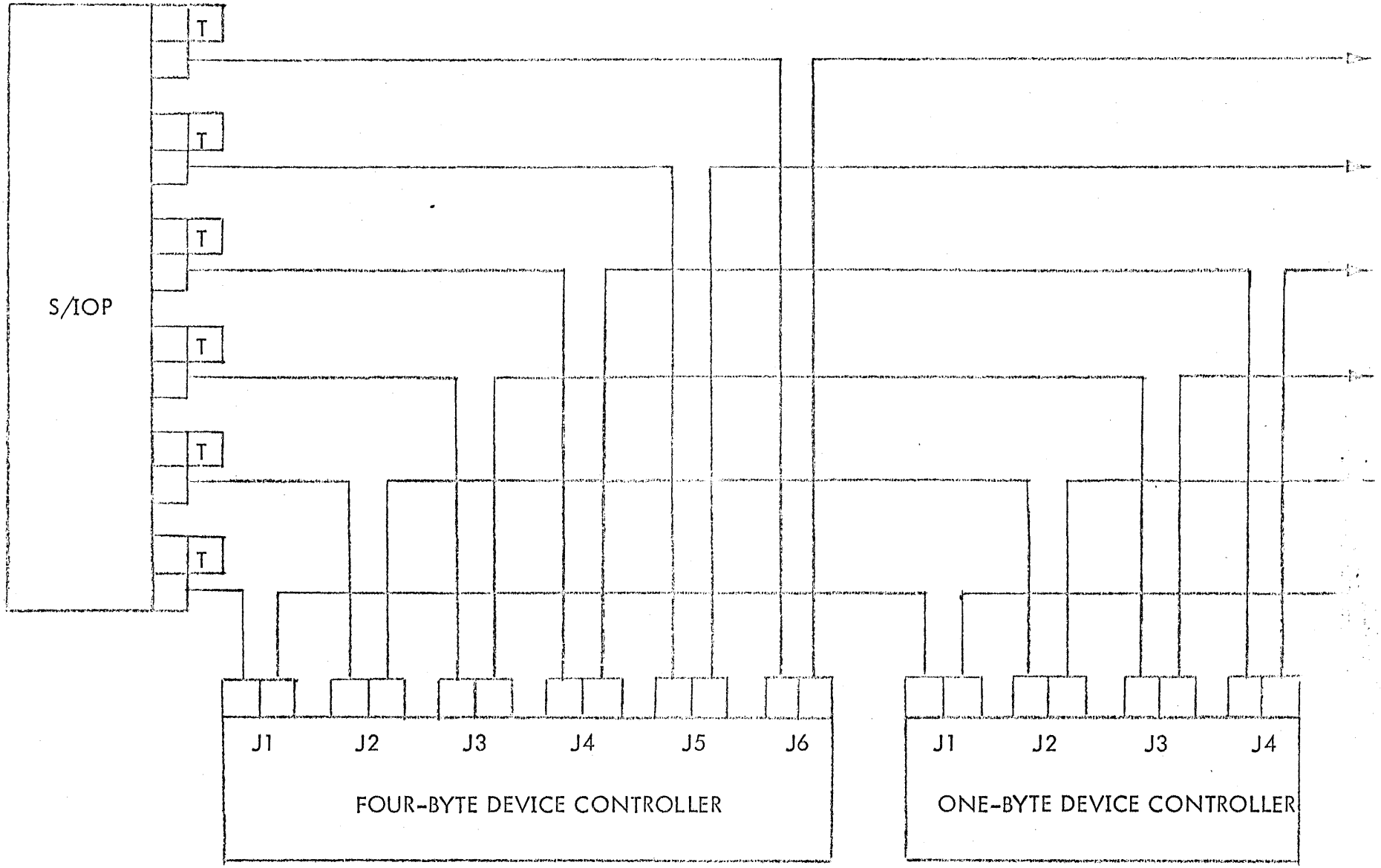


Figure 3.2 INTERCONNECTION OF S/IOP TO DEVICE CONTROLLERS

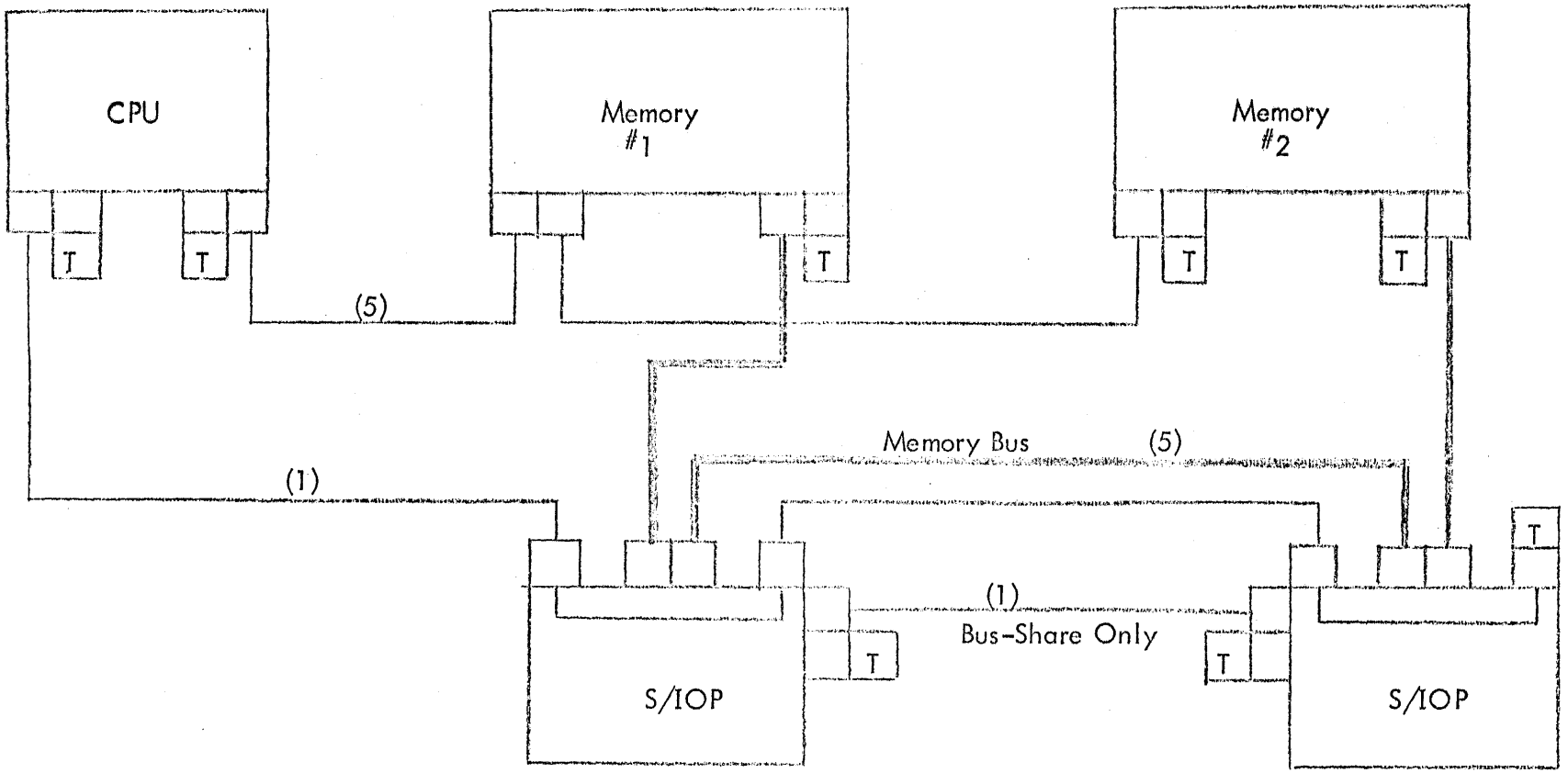


Figure 3.3 INTERCONNECTION OF BUS-SHARING SELECTOR IOP'S

4.0 INTERFACE SIGNAL TRANSMISSION

4.1 GENERAL

All interface signal exchange shall be via AT10 Cable Receiver modules (SDS Part No. 123018), AT11 Cable Driver/Receiver modules (SDS Part No. 123019), AT12 Cable Driver modules (SDS Part No. 124629), or AT13 Cable Driver/Receiver module (SDS Part No. 125260) as used throughout the Sigma system. Communication shall be via terminated shielded wire having a nominal impedance of 33 ohms.

4.2 CABLE CHARACTERISTICS

The cable consists of 14 shielded wires (transmission lines) as specified under SDS Part No. 101787. Nominal cable (wire) characteristics are:

Characteristic Impedance	33 ohms
DC Resistance (Center Conductor)	23 milliohms/foot
DC Resistance (Shield)	10 milliohms/foot
Inductance	50 nanohenry/foot
Capacitance	50 picofd/foot
Signal Delay	1.4 ns/foot

4.3 SIGNAL CHARACTERISTICS

Logical One = +2 volts DC (Driver output = low impedance)

Logical Zero = 0 volts DC (Driver output = high impedance)

4.4 CABLE CONNECTORS

Connectors are SDS Part No. 115833. These connectors provide for mating with 14 printed wiring contacts on cable plug modules. Two cable connectors can connect to one cable plug module to facilitate "Trunk-Tail" type interconnections.

4.5 CABLE TERMINATION

Cables shall be terminated at the extremities with 33 ohm resistors to ground. The terminating loads shall be a part of the terminator plug assembly SDS Part No. 127315 that connects to the Cable Driver/Receiver module.

5.0 MEMORY INTERFACE

5.1 SIGNAL CONVENTION

Signal exchange shall be via AT11 Cable Driver/Receiver modules and AT12 Cable Driver modules. The signals will use a convention of +2 volts as true logic.

5.2 BUSSES

Each Selector IOP shall have a bus for interfacing with core memories. Two Selector IOP's may; however, share a core memory bus if the Bus Share option is present in both

An IOP shall use only a single memory bus and shall not make a memory request before receiving an Address Release signal or an Address Here false signal from the prior request.

5.3 CELLS 20 AND 21

The Selector IOP uses memory cells 20* and 21* when responding to CPU requests for service. It shall be a requirement that the memory (memories if interleaving is used) that possesses IOP memory cells 20* and 21*, must also possess CPU memory cells 20* and 21*. This requirement shall also be true for an IOP which is to provide the CPU with its initial information during a bootstrap load operation.

5.4 PORTS

The IOP shall normally be connected either to memory port A or memory port B (memory port A having highest priority). However, there shall be no special form of implementation which would preclude the IOP from using port C.

5.5 SIGNAL LINES INTERCONNECTING IOP AND MEMORY

The lines tabulated below shall interconnect the IOP and memory: (In the table, X refers to one of memory ports A, B, or C).

5.5 (cont'd)

Description	Designation	Number of Lines	Direction of Signal Flow	
			IOP to Memory	Memory to IOP
Data Signals	/MX0/ thru /MX31/	32	x	x
Address Signals	/LX15/ thru /LX31/	17	x	
Address Here	/AHX/	1		x
Address Release	/ARX/	1		x
Data Gate	/DGX/	1		x
Parity Error	/PEX/	1		x
Parity O. K.	/POKX/	1		x
Write Byte Signals	/MW0X/ thru /MW3X/	4	x	
Memory Request	/MQX/	1	x	

5.6 SIGNAL TIMING

Timing of the signals described in 5.5 shall conform to the following: (All times as measured at IOP/Cable Interface).

5.6.1 DATA LINES

5.6.1.1 Data Input to Memory

During a core memory write operation signals on the data lines (MX0 thru MX31) shall be present and stable within 120 nanoseconds after the Memory Request (MQX) is given. The data lines shall remain stable for a minimum of 160 nanoseconds after the Address Release signal is received by the IOP for a full write operation or the Parity OK signal is received by the IOP for a partial write operation.

5.6.1.2 Data Output from Memory

During a core memory read operation, signals on the data lines are strobed by the Data Gate signal into the IOP memory interface register. The data signals shall not change during the interval 80 nanoseconds before Data Gate falls until 65 nanoseconds after Data Gate falls.

5.6.2 ADDRESS LINES

Signals on the Address Lines (LX15 thru LX31) shall be stable not less than 90 nanoseconds before the Memory Request signal rises and shall remain stable until 20 nanoseconds after the Memory Request signal has been dropped by the IOP.

5.6.3 DATA GATE SIGNAL

The Data Gate signal strobes data into the IOP memory interface register during a core memory read operation. The Data Gate signal shall be a positive-going pulse not less than 80 nanoseconds in duration.

5.6.4 PARITY ERROR AND PARITY OK

The Parity Error (PEX) and the Parity OK (POKX) signals shall be positive-going pulses not less than 70 nanoseconds wide and shall indicate the result of the parity check performed during a memory read or partial write operation by the memory.

5.6.5 WRITE BYTE SIGNALS

During a Memory full-write or partial write operation, the Write Byte (MW0X thru MX3X) signals shall designate the memory byte or bytes into which data is to be written. These signals shall be present and stable within 120 nanoseconds after the Memory Request signal. They shall be present until the Address Release Signal is received at the IOP.

5.6.6 MEMORY REQUEST SIGNAL

The Memory Request signal shall come true when the IOP wishes to make a memory access. This signal shall remain true until the IOP receives the Address Release signal or the Address Here signal is false when strobed.

5.6.7 ADDRESS RELEASE SIGNAL

The Address Release shall be a positive-going pulse not less than 80 nanoseconds in duration that designates to the IOP that the address lines may be changed and that the Memory Request line may go false.

5.6.8 ADDRESS HERE

The Address Here line when true shall designate to the IOP that the address coded on linesLA15 through LA31 is an address implemented in the memory to which the IOP is attached. The Address Here line shall be strobed no sooner than 260 nanoseconds after the Address Lines have settled.

5.7 CABLE LENGTHS

The total IOP/Memory cable length shall not exceed 40 feet to the furthest Memory port. The IOP may drive cables in two directions in order to minimize the cable length. All cables in the bus interconnecting and IOP with a particular Memory port shall be of the same nominal length.

5.8 CABLE CONNECTOR PIN ASSIGNMENT

Five cables are required to interconnect the IOP and memory. The pin assignment of signals in those cables at the Cable Driver/Receiver and Cable connector interface shall be as specified below:

Connector Pin	Signal Desig. at Connector	Signal Functional Description
2	/MX0/	Data Line, Port X
1	/MX1/	"
4	/MX2/	"
3	/MX3/	"
5	/MX4/	"
6	/MX5/	"
7	/MX6/	"
8	/MX7/	"
9	/MX8/	"
10	/MX9/	"
11	/MX10/	"
12	/MX11/	"
13	/MX12/	"
14	/MX13/	"

Connector Pin	Signal Desig. at Connector	Signal Functional Description
2	/MX14/	Data Line, Port X
1	/MX15/	"
4.	/MX16/	"
3	/MX17/	"
5	/MX18/	"
6	/MX19/	"
7	/MX20/	"
8	/MX21/	"
9	/MX22/	"
10	/MX23/	"
11	/MX24/	"
12	/MX25/	"
13	/MX26/	"
14	/MX27/	"

Cable 3

AT11 Cable Driver/Receiver Module

Connector Pin	Signal Desig. at Connector		Signal Functional Description
2	/MX28/		Data Line, Port X
1	/MX29/		"
4	/MX30/		"
3	/MX31/		"
5	/LX29/		Address Line, Port X
6	/LX30/		Address Line, Port X
7	/LX31/		Address Line, Port X
8	/MW0X/		Memory Write Byte Line, Port X
9	/MW1X/		Memory Write Byte Line, Port X
10	/MW2X/		Memory Write Byte Line, Port X
11	/MW3X/		Memory Write Byte Line, Port X
12	/DGX/		Data Gate Line, Port X
13	/EDRX/	*	Early Data Release Signal
14	Not Assigned		

*Not used by IOP

Cable 4

AT12 Cable Driver Module

Connector Pin	Signal Desig. at Connector	Signal Functional Description
2	/LX15/	Address Line, Port X
1	/LX16/	"
4	/LX17/	"
3	/LX18/	"
5	/LX19/	"
6	/LX20/	"
7	/LX21/	"
8	/LX22/	"
9	/LX23/	"
10	/LX24/	"
11	/LX25/	"
12	/LX26/	"
13	/LX27/	"
14	/LX28/	"

Connector Pin	Signal Desig. at Connector		Signal Functional Description
2	/MQX/		Request for Memory Access, Port X
1	/AHX/		Address Here, Port X
4	/ARX/		Address Release, Port X
3	/DRX/	*	Data Release, Port X
5	/PEX/		Parity Error Signal, Port X
6	/SRAX/	*	Second Request Allowed
7	Not Assigned		
8	/ABOA/	*	Abort Memory Lockout Signal
9	/POKX/		Parity OK Signal, Port X
10	/MR/	*	(1) Memory Reset
11	/ORX/	*	Override Port
12	/ORIL/	*	(1) Override Interleave
13	/HOF/	*	(1) Halt on Fault
14	/MFR/	*	(1) Memory Fault Reset

*Not used with Selector IOP

(1) Port C only

6.0 CPU INTERFACE

6.1 SIGNAL CONVENTION

Signals shall be exchanged via the AT13 Cable Driver/Receiver module, SDS Part No. 125260. Unless otherwise stated, the signals shall use a convention of +2 volts as true logic.

6.2 SIGNAL LINES INTERCONNECTING IOP AND CPU

IOP's shall be connected in a tail-trunk fashion to a CPU as shown in Figure 3.1. The IOP's shall be arranged in a priority sequence such that during the execution of an AIO instruction the IOP nearest to the CPU with an Interrupt Request pending shall respond to the CPU request for service.

6.2.1 The lines tabulated below shall interconnect the IOP and CPU.

Description	Designation	Number of Lines	Direction of Signal Flow	
			IOP to CPU	CPU to IOP
Function Code	/FNC0/ thru /FNC2/	3		x
IOP Address	/IOPA0/ thru /IOPA2/	3		x
Not Condition Code 1	/NCOND1/	1	x	
Not Condition Code 2	/NCOND2/	1	x	
Clock 1 - MHz, Source	/CL1S/	1		x
Reset I/O	/RIO/	1		x
Interrupt Request	/IR/	1	x	
Proceed	/PR/	1	x	
Control Stobe	/CNST/	1		x

6.2.2 The utilization of the lines interconnecting the IOP and CPU shall be as follows: (Timing of signals is as measured at the IOP/Cable interface).

6.2.2.1 Function Code Lines: The CPU shall place on these lines signals representing the I/O instruction being executed. During a CPU request for service, signals on these lines shall be interpreted as follows by the S/IOP:

	SIO	TIO	TDV	HIO	AIO
/FNC0/	0	0	0	0	1
/FNC1/	0	0	1	1	1
/FNC2/	0	1	0	1	0

The signals on these lines shall be settled 250 nanoseconds before the Control Strobe (CNST) is driven true and shall remain settled for 250nanoseconds after the Control Strobe is released. These times as measured at the IOP/Cable interface.

6.2.2.2 IOP Address Lines: The CPU shall place on these lines coded signals representing the address of the IOP from which service is requested during the execution of SIO, TIO, TDV, or HIO instructions. When the Function Code lines are coded representing SIO, TIO, TDV, or HIO operations, the IOP shall compare the state of the IOP Address lines with the state of switch contacts within the IOP that designates the IOP address. If the two sets of signals compare (IOP address recognition) then the IOP shall respond to the Control Strobe signal.

When the Function Code lines are coded representing the AIO operation then the IOP Address lines shall have no meaning. The signals on these lines shall be settled 250 nanoseconds before the Control Strobe is driven true and shall remain settled for 250nanoseconds after the Control Strobe is released. These times are measured at the IOP/Cable Interface.

6.2.2.3 Control Strobe: The CPU shall drive this line during the execution of an I/O instruction to request service from an IOP. The IOP shall respond to the Control Strobe as follows:

If the Function Code lines designate an SIO, HIO, TIO, or TDV operation and IOP address recognition exists or if the Function Code lines designate an AIO operation and the Interrupt Request line is true then the IOP shall respond to the Control Strobe and shall not pass the Control Strobe along to the next lower priority IOP. Response of the IOP to specific function codes will be described in later paragraphs. If an SIO, HIO, TIO, or TDV operation is designated and no IOP address recognition exists or if an AIO operation is designated and the Interrupt Request line is false then the IOP shall not respond to the Control Strobe but shall pass the Control Strobe along to the next lowest priority IOP. If, however, the IOP is designated as the last IOP on the tail-trunk cable by the state of a switch contact in the IOP, then the IOP shall cause the Proceed signal to be driven true and shall drive no other lines.

6.2.2.4 Not Condition Code Lines: The S/IOP may normally or partially execute the response to a CPU request for service. The SIO, TIO, or TDV operations may either be normally or partially executed. The HIO and AIO operations shall always be normally executed -- this may involve prematurely terminating a Device Controller requested service by the IOP. Normal execution of the operation involves accessing data from core memory (except for AIO), communication with a device controller (if no memory parity error while reading memory), possible storing of data in core memory, and control of the NCOND1 and NCOND2 lines. Partial execution of the operation involves only controlling the NCOND1 and NCOND2 lines. If during normal execution of an SIO, TIO, TDV, or HIO operation the IOP detects a memory parity error while accessing cell 20* then the IOP shall not communicate with any Device Controller and shall hold NCOND1 and NCOND2 false. During normal execution of an SIO, TIO, TDV, or HIO operation if no memory parity error is detected while reading cell 20* the IOP shall drive the NCOND1 and NCOND2 lines with the same state that the Device Controller dictated on the DOR and IOR lines respectively.

*Base 16

6.2.2.4 During the execution of an AIO operation the IOP shall drive the NCOND1 and NCOND2 lines as dictated (con't)
by the Device Controller; however, if the responding Device Controller has the address stored by the IOP and an Unusual End interrupt is pending then the S/IOP shall hold the NCOND2 line false.

During partial execution of an operation the S/IOP shall hold the NCOND1 line false and drive the NCOND2 line true. The signals on these lines shall be settled before the Proceed signal is driven true and shall remain settled until the release of the Control Strobe.

6.2.2.5 Proceed Line: In response to a CPU request for service on the Control Strobe line the IOP shall drive the Proceed line to indicate termination of the service. The Proceed line shall operate in a closed-loop manner with respect to the Control Strobe line; i.e., the IOP shall raise the Proceed line in response to the CPU raising the Control Strobe line and shall drop the Proceed line in response to the CPU dropping the Control Strobe line.

Partial Execution:
During partial execution of an SIO, TIO, or TDV operation the S/IOP shall respond to the Control Strobe by driving the NCOND2 and Proceed lines true immediately.

No Address Recognition:
During the response to an SIO, TIO, TDV, or HIO operation with no address recognition or to an AIO operation with the Interrupt Request line false the S/IOP shall not respond to the Control Strobe but shall pass the Control Strobe along to the next lower priority IOP. However, if the S/IOP is designated as the last IOP on the tail-trunk cable interconnecting IOP's with the CPU then the IOP shall respond to the Control Strobe by holding NCOND1 and NCOND2 false and driving the Proceed line true immediately.

Normal Execution:

If during normal execution of an SIO, TIO, TDV, or HIO operation the IOP detects a memory parity error while accessing cell 20*then the IOP shall hold the NCOND1 and NCOND2 lines false and shall drive the Proceed line true immediately.

If during normal execution of an SIO, TIO, TDV, or HIO operation the IOP does not write in memory cells 20* or 21* then the IOP shall respond to the Control Strobe after receiving the Device Controller condition code response on the DOR and IOR lines.

If the IOP writes in memory cells 20* or 21* then the IOP shall raise the Proceed line after receiving an Address Release signal from core memory in response to the last Memory Request.

6.2.2.6 Interrupt Request Line: The IOP shall drive this line true when making a standard I/O interrupt request. All interrupt requests from Device Controllers attached to an IOP are funnelled to this line via the Interrupt Call line at the Device Controller interface. This line shall go false during the execution of all I/O instructions. If an interrupt is then still pending this line shall again be driven true.

6.2.2.7 Reset I/O Line: The CPU shall drive this line true when the I/O system is to be initialized. The IOP in response to this signal shall be initialized in readiness for a service request from the CPU and shall also pass the signal along at the Device Controller interface on the I/O Reset line.

6.2.2.8 Clock, Source Line: The CPU shall drive this line with a periodic timing signal for use by Device Controllers. The IOP shall pass this signal along to the Device Controller at the Device Controller interface on the Clock line.

*Page 16

6.3 IOP OPERATIONS DURING CPU REQUESTS FOR SERVICE

As previously described, the S/IOP may normally or partially execute the response to a CPU request for service. The SIO, TIO, or TDV operations may either be normally executed or partially executed. The HIO or AIO operations shall always be normally executed.

During execution of a CPU request for service the IOP and CPU shall use the lines described in Section 6.2 for direct communication. During normal execution of a CPU request for service the IOP and CPU shall also communicate indirectly via core memory cells 20* and 21*. The IOP shall also communicate at the Device Controller interface to obtain Condition Code information and Status response.

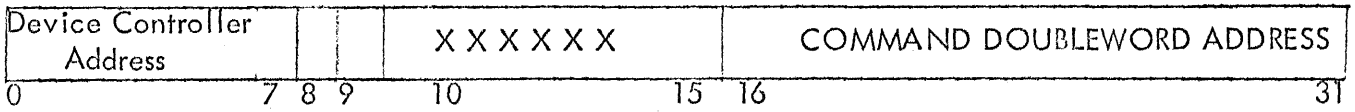
Specific actions of the IOP during CPU requests for service are described in the paragraphs below.

6.3.1 SIO

If the IOP Busy status bit is true (IOP Busy state) or a Zero Count or Channel End or Unusual End interrupt is pending then the IOP shall partially execute the SIO operation. The S/IOP responses for this case have been previously described in Section 6.2.

If the IOP Busy status bit is false and a Zero Count or Channel End or Unusual End interrupt is not pending then the IOP shall normally execute the SIO operation. The IOP shall make a Memory Request to read memory cell 20*. If a memory parity error is detected the IOP shall terminate the operation immediately as previously defined in Section 6.2.2.5. If a memory parity error is not detected then the S/IOP shall interpret the contents of cell 20* as follows:

Cell 20*



*Base 16

Bits 0 - 7: Contain the Device Controller Address. These bits are transmitted at the Device Controller Interface on Data Lines DA0-7. They shall also be stored in the S/IOP if the IOP goes to the Busy state during the SIO operation.

Bits 8 and 9: Specify the IOP response as follows:

Bit 8	Bit 9	S/IOP Response
0	0	IOP shall respond on NCOND1 and NCOND2 only
0	1	IOP shall respond on NCOND1 and NCOND2 and write in cell 21*
1	1	IOP shall respond on NCOND1 and NCOND2 and write in cells 20* and 21*

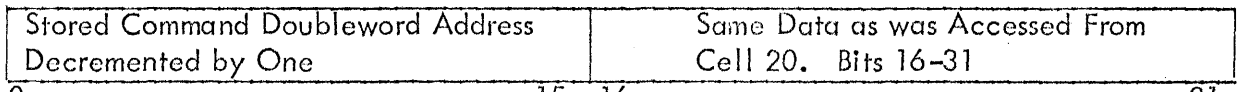
Bits 10 - 15: Ignored by the IOP

Bits 16 - 31: Contain the Command Doubleword Address that shall subsequently be stored in the S/IOP if the IOP goes to the Busy state during the SIO operation.

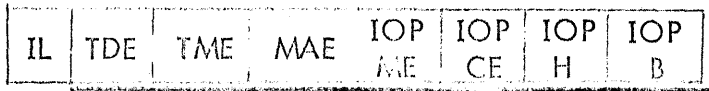
The IOP shall communicate at the Device Controller Interface with the addressed Device Controller and shall complete the response of the CPU service request by writing into cells 20* and 21* as follows:

Cell 20*

Cell 21*



IOP Status:



*Base 16

The information stored in cells 20* and 21* shall have the following meaning:

Cell 20* Bits 0 - 15: Shall contain the Command Doubleword Address stored in the IOP prior to the normal execution of an SIO and decremented by one.

Cell 20* Bits 16 - 21: Shall contain the same data as was read from core memory cell 20* bits 16 - 31:

Cell 21* Bits 0 - 7: Shall contain the Device Controller status information obtained from the Function Response lines 0 - 7 at the Device Controller interface.

Cell 21* Bits 8 - 15: Shall contain IOP status information stored in the S/IOP. Bits 8 - 14 shall be cleared prior to termination of the CPU service if the S/IOP goes to the Busy state. The IOP status bits are defined in greater detail below.

Cell 21* Bits 16 - 31: Shall contain the Byte Count currently stored in the S/IOP.

IOP Status Bits:

Bit 8: Incorrect Length: This bit shall be set true if Incorrect Length was signalled by the Device Controller via the Order In Operational Status Byte or if the IOP detected during a Data In operation that the buffer contained data at Count Done.

Bit 9: Transmission Data Error: This bit shall be set true if Transmission Error was signalled by the Device Controller via the Order In Operational Status Byte or if the IOP detected a byte parity error during a Data In operation.

Bit 10: Transmission Memory Error: This bit shall be set true if a Memory Parity Error was detected by the IOP during a Data Out/In operation (memory partial-word write or read operations).

Bit 11: Memory Address Error: This bit shall be set true if a non-implemented memory address was detected by the IOP during a chaining operation or a Data Out/In operation.

Bit 12: IOP Memory Error: This bit shall be set true if the IOP detected a memory parity error while fetching a command.

*Base 16

Bit 13: IOP Control Error: This bit shall be set true if the IOP detected two successive Transfer In Channel commands.

Bit 14: IOP Halt: This bit shall be set true if the IOP detected an IOP Halt condition.

Bit 15: IOP Busy: This bit shall be set true during the current SIO operation after the IOP status has been written in cell 21* if the Device Controller responded by driving both the DOR and IOR lines true (CC1 = CC2 = 0). This bit shall be cleared under the following conditions:

(1) The Reset I/O line was driven true.

(2) During an HIO operation the Device Controller address stored in the IOP compared identically with the Device Controller address transmitted at the Device Controller interface.

(3) During an Order In operation the Device Controller signalled Unusual End via the Operational Status Byte.

(4) During an Order In operation the Device Controller signalled Channel End via the Operational Status Byte and the Command Chain flag was false or the IOP Halt status bit was true.

The S/IOP shall not report IOP Busy during a normally executed SIO operation.

6.3.2 TIO, TDV

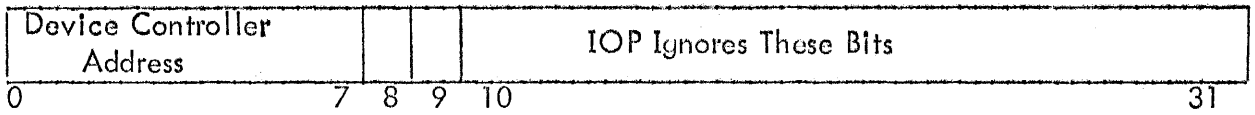
If the S/IOP is in the Busy state and is currently responding to a Device Controller service request and has not signalled End Data and Request Strobe Acknowledge to the Device Controller prior to the receipt of the TIO or TDV function code from the CPU, then the TIO or TDV operation shall be partially executed. The S/IOP responses for this case have been previously described in Section 6.2

If the S/IOP is not in the Busy state or if in the Busy state and not currently responding to a Device Controller service request or if Busy and currently responding to a Device Controller service request and has signalled End Data and Request

*Page 16

Strobe Acknowledge to the Device Controller prior to the receipt of the TIO or TDV function code from the CPU then the IOP shall normally execute the TIO or TDV operation. The IOP shall make a Memory Request to read cell 20*. If a memory parity error was detected then the IOP shall terminate the operation immediately as previously described in Section 6.2.2.5. If a memory parity error was not detected then the IOP shall interpret the contents of cell 20* as follows:

Cell 20*



Bits 0 - 7: Contain the Device Controller Address. These bits are transmitted at the Device Controller interface on Data Lines DA0-7.

Bits 8 and 9: The IOP shall respond to these bits as defined in Section 6.3.1 for the SIO operation.

Bits 10 - 31: The IOP shall ignore these bits.

The IOP shall then communicate at the Device Controller interface with the addressed Device Controller and shall complete the response to the CPU service request by writing - if requested - into cells 20* and 21*. The format of these responses shall be the same as the format of responses defined in Section 6.3.1 for an SIO operation. The definition of the individual bits shall be the same as defined in Section 6.3.1 for an SIO operation.

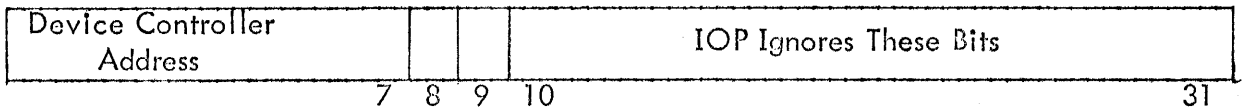
6.3.3 HIO

The IOP shall always normally execute an HIO operation. If the IOP is currently responding to a Device Controller service request the IOP shall cause End Data to be set and then respond to the CPU request for service as soon as the current service is terminated.

The IOP shall make a Memory Request to read cell 20*. If a memory parity error was detected then the IOP shall terminate the operation immediately as previously described in Section 6.2.2.5. If a memory parity error was not detected, then the

IOP shall interpret the contents of cell 20* as follows:

Cell 20*



Bits 0 - 7: Contain the Device Controller Address. The IOP shall transmit these bits at the Device Controller Interface on Data Lines DA0-7. These bits shall also be compared with the Device Controller address currently stored in the S/IOP. If the addresses compare the IOP shall do the following:

(1) Clear the IOP Busy status bit.

(2) Clear the Zero Count interrupt, Channel End interrupt, and Unusual End interrupt status bits.

Bits 8 and 9: The IOP shall respond to these bits as defined in Section 6.3.1 for the SIO operation.

Bits 10 - 31: The IOP shall ignore these bits.

The IOP shall then communicate at the Device Controller interface with the addressed Device Controller and shall complete the response to the CPU request for service by writing - if requested - into cells 20* and 21*. The format of these reponses shall be the same as the format of responses defined in Section 6.3.1 for an SIO operation. The definition of the individual bits shall be the same as defined in Section 6.3.1 for an SIO operation.

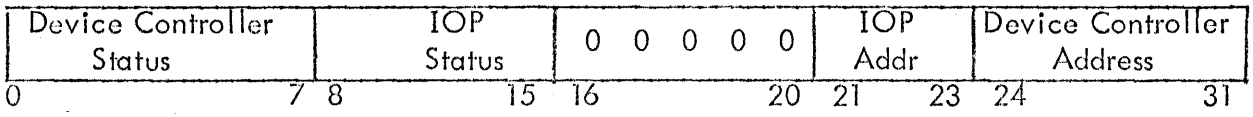
6.3.4 AIO

The IOP shall always normally execute an AIO operation if an interrupt is pending. If the IOP is currently responding to a Device Controller service request then the IOP shall cause End Data to be set, wait for the current service to terminate, and then respond to the CPU request for service.

* Base 16

The IOP shall communicate at the Device Controller interface to obtain the address and status of the interrupting Device Controller. The IOP shall then assemble the status and Device Controller address and make a Memory Request to write into cell 20* . The format of the information written into cell 20* shall be as follows:

Cell 20*



IOP Status:

Bits 0 - 7: Shall contain the Device Controller status information received on the Data Lines DA0-7.

Bits 8 - 15: Shall contain IOP status information defined below in further detail.

Bits 16 - 20: Shall always be false.

Bits 21 - 23: Shall contain the address of the responding IOP.

Bits 24 - 31: Shall contain the address of the responding Device Controller. The IOP shall compare this address with the Device Controller address stored in the IOP and shall if the Device Controller also raises the DOR line (CC1 = 0) write the IOP status into bits 8 - 15 of cell 20* and shall then clear the interrupt status bits.

IOP Status Bits:

Bit 8: Incorrect Length: As defined for SIO in Section 6.3.1.

Bit 9: Transmission Data Error: As defined for SIO in Section 6.3.1.

Bit 10: Zero Byte Count Interrupt: This bit shall be set true if the Interrupt on Zero Count flag is true and Zero Byte Count is detected during a Data In operation or Zero Byte Count is detected during a Data Out operation and Data Chaining ensues or Count Done is reached.

Bit 11: Channel End Interrupt: This bit shall be set true if the Interrupt at Channel End flag is true and Channel End is signalled during an Order In via the Operational Status Byte.

Bit 12: Unusual End Interrupt: This bit shall be set true if the Interrupt at Unusual End flag is true and Unusual End is signalled during an Order In via the Operational Status Byte or if IOP Halt is signalled to the Device Controller in the Order In Terminal Order.

Bit 13: This bit shall be false.

Bit 14: This bit shall be false.

Bit 15: This bit shall be false.

6.4 Cables

As specified in 3.1, IOP's shall be connected to a CPU by cables in a trunk-tail fashion. The total length of cables interconnecting IOP's with a CPU shall not exceed 64 feet. All IOP's shall interconnect with a CPU on a single cable. The pin assignment for signals in that cable at the Cable Driver/Receiver and Cable Connector Interface shall be as specified below:

Connector Pin	Signal Desig. at Connector	Signal Functional Description
1	/FNC0/	Function Code
2	/FNC1/	" "
3	/FNC2/	" "
4	/IOPA0/	IOP Address
5	/IOPA1/	"
6	/IOPA2/	"
7	/CNST/	Control Strobe
8	Not Assigned	
9	/NCOND1/	Not Condition Code 1
10	/NCOND2/	Not Condition Code 2
11	/CL1S/	Timing Signal (1 MHz)
12	/RIO/	I/O Reset
13	/IR/	Interrupt Request
14	/PR/	Proceed

7.0 DEVICE CONTROLLER INTERFACE

7.1 Cable Connections – Six cables shall interconnect the IOP and Device Controllers. All Device Controllers shall interconnect in a trunk-tail fashion to the IOP. Device Controllers with the One-Byte Data Path shall use only four of the six cables.

7.2 Signal Convention – Signal exchange shall be via AT10 Cable Receiver Modules, AT11 Cable Driver/Receiver Modules, and AT12 Cable Driver Modules. Unless otherwise stated, signals shall use a convention of +2 volts for true logic.

7.3 Signal-Lines Interconnecting IOP and Device Controllers

7.3.1 The lines tabulated below shall interconnect the IOP and Device Controllers.

DESCRIPTION	DESIGNATION	NO. OF LINES	DIRECTION OF SIGNAL FLOW	
			IOP TO DEVICE CONTROLLER	DEVICE CONTROLLER TO IOP
Data Input/Output (1st byte)	/DA0/thru / DA7 /	8	x	x
Data Input/Output (2nd byte)	/ DB0/ thru /DB7/	8	x	x
Data Input/Output (3rd and 4th bytes)	/DC0/ thru /DC7/, /DD0/thru/DD7/	16	x	x
Data Parity	/DAP/	1	x	x
2 Byte Interface Request	/DX2/	1		x
4 Byte Interface Request	/DX4/	1		x
End Data	/ED/	1	x	x
Parity Check	/PC/	1		x
Data/Order Request	/DOR/	1		x
Input/Output Request	/IOR/	1		x
Function Response	/FR0/thru/FR7/	8		x
Request Strobe	/RS/	1		x
Function Acknowledge Strobe	/FSL/	1		x
Device Controller Available Output	/AV0/	1		x
Interrupt Call	/IC/	1		x
Service Call	/SC/	1		x
I/O Reset	/RST/	1	x	
Clock, 1 MHz	/CL1/	1	x	

DESCRIPTION	DESIGNATION	NO. OF LINES	DIRECTION OF SIGNAL FLOW	
			IOP TO DEVICE CONTROLLER	DEVICE CONTROLLER TO IOP
End Service	/ES/	1	x	
Request Strobe Acknowledge	/RSA/	1	x	
Start I/O Function Indicator	/SIO/	1	x	
Halt I/O Function Indicator	/HIO/	1	x	
Test I/O Function Indicator	/TIO/	1	x	
Test Device Function Indicator	/TDV/	1	x	
Acknowledge Interrupt Function Indicator	/AIO/	1	x	
Acknowledge Service Call Indicator	/ASC/	1	x	
Function Strobe	/FS/	1	x	
Zero Byte Count Interrupt	/ZBCI/	1	x	
Fast Device Controller	/FAST/	1		x
AVI from IOP	/TRUE-1/, /TRUE-2/	2	x	
Reset I/O from JX58	/TEST RIO/	1		x
Enable Two-byte Interface	/EDX2/	1	x	
Enable Four-byte Interface	/EDX4/	1	x	

7.3.2 Definition of Device Controller Interface Lines. All times as measured at IOP/Cable interface.

7.3.2.1 Service Call (SC) Line:

The SC line shall be driven by the Device Controller to request service from the IOP. The IOP shall respond to the request by initiating one of several possible operation sequence, identified as one of the following: Order Out, Order In, Data Out, Data In, or an aborted service request. The Device Controller shall drive the SC line at least until the IOP has driven the ASC function indicator line; otherwise, the IOP shall abort the service request and wait for the next service request from either the CPU or the Device Controller.

CPU service requests shall have priority over Device Controller service requests.

7.3.2.2 Function Indicator Lines (SIO, HIO, TIO, TDV, AIO, ASC):

During the normal execution of a CPU request to perform an SIO, HIO, TIO, TDV, or AIO operation the IOP shall drive the appropriate Function Indicator line. Similarly, during the response to a Device Controller request for service via the SC line the IOP shall drive the ASC line. The driven line shall be stable for at least 185 nanoseconds before the Function Strobe line is driven and shall not be dropped less than 185 nanoseconds after the Function Strobe line is dropped.

7.3.2.3 Function Strobe (FS), Function Strobe Acknowledge (FSL), and Available Out (AVO) lines:

The Function Strobe line shall operate in a closed-loop fashion with either the FSL or AVO lines; that is, after the IOP has driven the FS line true the Device Controller shall drive either the FSL line or the AVO line and shall drop the FSL or AVO line in response to the IOP dropping the FS line.

During the normal execution of an SIO, HIO, TIO, or TDV operation (if no memory parity error is detected while reading cell 20*) the IOP shall place the Device Controller address on the DA0-7 lines, drive the appropriate Function Indicator line, delay, and drive the FS line. If the Device Controller address was not recognized or if the addressed Device Controller was off-line then the IOP shall receive AVO in response to the FS signal. Otherwise, the addressed Device Controller shall respond by placing condition code information on the DOR and IOR lines, Device Controller status on the Function Response lines, and driving the FSL line.

During the execution of an AIO operation the IOP shall drive the AIO line, delay, and drive the FS line. If there was no Device Controller interrupt pending then the IOP shall receive AVO in response to the FS signal. Otherwise, the responding Device Controller shall place its address on the Function Response lines, condition code data on the DOR and IOR lines, Device Controller status on the DAO-7 lines, and raise the FSL line.

During the response of the IOP to a Device Controller request for service on the SC line the IOP shall raise the ASC line, delay, and raise the FS line. If the SC line has dropped prior to the time that the IOP raised the ASC line then the IOP shall receive a response on the AVO line. Otherwise, the highest priority service requesting Device Controller shall raise the FSL line. Refer to the One-Byte or Four-Byte Data Path Specification for the description of the Device Controller actions when connecting for service to the IOP.

7.3.2.4 Data/Order Request (DOR) and Input/Output Request (IOR) Lines:

The DOR and IOR lines shall be driven by the Device Controller. The IOP shall interpret the state of these lines as follows:

(a) During the normal execution of a CPU request for service, if the IOP has raised the FS line, then the Device Controller shall place condition code data on these lines. The IOP shall wait for the FSL line to be driven, delay 185

*Base 16

nanoseconds*, and then strobe these lines and store the state for later transmission to the CPU on the NCOND 1 and NCOND 2 lines. The Device Controller shall not drop these lines until the IOP has dropped the FS line.

(b) During a Device Controller service request, if not aborted, the Device Controller shall connect for service at the time that the IOP drops the FS line. The Device Controller shall then drive the DOR and IOR lines with coded signals, drive the Data Lines if IOR is false, and raise the Initial Request Strobe (RS). The IOP shall delay 185 nanoseconds* after receiving the Initial RS of the service sequence and then strobe the DOR and IOR lines. The IOP shall interpret the state of these lines as follows:

DOR	IOR	
1	1	Order Out
1	0	Order In
0	1	Data Out
0	0	Data In

The IOP shall then perform the sequence of operations dictated by the above coding. These operations will be defined in greater detail in later paragraphs.

The Device Controller shall not change the state of the DOR and IOR lines before the IOP has initially driven the Request Strobe Acknowledge (RSA) lines.

*Minimum

7.3.2.5 Function Response Lines (FR0-7):

The FR0-7 lines shall be driven only by the Device Controller. During the normal execution of an SIO, HIO, TIO, or TDV operation the Device Controller shall supply Device Controller status on these lines. During an AIO service cycle the Device Controller address shall be transmitted on these lines. The IOP shall ignore the state of these lines during a Device Controller service sequence.

The IOP shall delay 185 nanoseconds* after the FSL line has been driven true and then strobe the Function Response lines for later transmission of the data on the lines to Core Memory. The Device Controller shall not change the state of the lines until the IOP has dropped the FS line.

7.3.2.6 Request Strobe (RS) and Request Strobe Acknowledge (RSA) lines:

The RS line shall be driven by the Device Controller and the RSA line shall be driven by the IOP. These lines shall operate in a closed-loop fashion; that is, upon sensing that the RS line has been driven true by the Device Controller the IOP shall delay and then drive the RSA line true. Upon sensing that the RS line has dropped the IOP shall immediately drop the RSA line. The Device Controller, after connecting for service, shall send a succession of RS signals to the IOP until the IOP signals RSA and End Service.

The IOP, after receiving the initial RS of a service sequence, shall delay 185 nanoseconds* and then strobe the DOR and IOR lines to determine the type of service requested by the Device Controller. If the service is an output operation the IOP shall immediately access core memory for the data or order, place the data or order on the Data Lines, delay 185 nanoseconds,* and raise the RSA line. If the service is an input operation then the IOP shall delay 185 nanoseconds* and then strobe the Data Lines. This sequence shall continue until either the IOP or the Device Controller signals End Data. The IOP shall

*Minimum

then wait for the final RS for the Terminal Order, place the Terminal Order on the Data Lines, set End Service, delay 185 nanoseconds,* and then raise RSA to terminate the service sequence.

7.3.2.7 Data Lines (DA0-7, DAP, DB0-7, DC0-7, DD0-7):

The Data Lines shall be driven by either the IOP or the Device Controller.

The DA0-7 lines are multi-purpose lines and shall be used as follows:

(a) During the normal execution of an SIO, HIO, TIO, or TDV operation the IOP shall place the Device Controller address on the DA0-7 lines, delay 185 nanoseconds,* and then drive the FS line. The IOP shall not change the state of the DA0-7 lines until either FSL or AVO have been received.

(b) During the execution of an AIO operation the Device Controller shall place the Device Controller status on the DA0-7 lines. The IOP shall delay 185 nanoseconds* after receiving the FSL signal and then strobe the lines. The Device Controller shall not change the state of the lines until FS has dropped.

(c) During an Order Out service, in response to the initial RS, the IOP shall place the Order on the DA0-7 lines, delay 185 nanoseconds,* and raise the RSA line. The IOP shall not change the state of the Data Lines until the next RS is received.

(d) During an Order in service, associated with the initial RS, the Device Controller shall place the Operational Status Byte on the DA0-7 lines. The IOP shall delay 185 nanoseconds* after receiving RS, then strobe the DA0-7 lines. The IOP shall interpret the Data lines as follows:

DA0	Transmission Error
DA1	Incorrect Length

*Minimum

DA2 Chaining Modifier

DA3 Channel End

DA4 Unusual End

The response of the IOP to the Operational Status Byte will be defined in later paragraphs. The Device Controller shall not change the state of the DA0-7 lines until RSA has been raised.

(e) During a Data Out service, if End Service is false, the IOP shall place the data on the DA0-7 lines, delay 185 nanoseconds,* and raise RSA. The IOP shall not change the state of the Data Lines until the next RS is received.

(f) During a Data In service, if End Service is false, the Device Controller shall place the data on the DA0-7 lines and raise the RS line. The IOP shall delay 185 nanoseconds* and strobe the lines. The Device Controller shall not change the state of the Data lines until RSA has been received.

(g) During any service sequence, in response to the final RS, the IOP shall raise End Service, place the Terminal Order on the DA0-7 lines, delay 185 nanoseconds,* and raise RSA. The definition of the DA0-7 lines during the Terminal Order exchange shall be as follows:

DA0 Interrupt

DA1 Count Done

DA2 Command Chain

DA3 IOP Halt

*Minimum

The conditions under which these lines are raised will be discussed in later paragraphs. The IOP shall not change the state of the Data lines until FS is to be raised during the next service cycle.

The DAP line shall be used to provide an odd parity bit for the data byte on lines DA0-7 during Data Out or Data in service. The IOP shall provide an odd parity bit for the data byte on the DA0-7 lines during all output operations. If the PC line is true and the DX2 and DX4 lines are false then the IOP shall check for odd parity if Data In service. If a parity error is detected during a Data In service, the IOP shall set the Transmission Data Error status bit. The timing for the DAP line shall be the same as the timing for the DA0-7 lines during Data Out/In service.

The DB0-7, DC0-7, and DD0-7 Data lines shall be used only for the exchange of data during Data Out/In service. The timing for these lines shall be the same as the timing for the DA0-7 lines during Data Out/In service.

If the DX2 and DX4 lines are false during Data Out/In service sequences then the IOP shall exchange data with the Device Controller on the DA0-7 lines only. If the DX4 line is true during Data Out/In service sequences then the IOP shall exchange data with the Device Controller on the DA0-7, DB0-7, DC0-7, and DD0-7 lines where DA0-7 shall be the most significant byte and DD0-7 shall be the least significant byte. If the DX2 line is true during Data Out/In operations then the IOP shall exchange data with the Device Controller on the DA0-7 and DB0-7 lines. The DA0-7 lines shall be most significant.

7.3.2.8 End Data (ED) and End Service (ES) Lines:

The ED and ES lines shall designate action to be taken by the IOP and Device Controller during order and data exchanges.

The ED line may be driven true by either the IOP or the Device Controller. The ED line shall always be driven true by the Device Controller with the initial RS of an Order Out/In sequence. If, during a Data Out/In exchange, either the IOP

or the Device Controller drive the ED line; then, this shall signify that no more data be exchanged. The ED line shall be driven true by the IOP under the following conditions:

(a) Whenever the ED line has been driven by the Device Controller.

(b) During Data Out/In sequences if the CPU requests the IOP to perform an HIO operation or if the CPU requests that an AIO operation be performed and the IOP has an Interrupt pending.

(c) During a Data Out sequence the IOP detected an IOP Halt condition, or the IOP detected Zero Byte Count and the Data Chain flag was false and the data buffer was empty (Count Done), or the IOP detected Zero Byte Count and the Data Chain flag was true and the Interrupt on Zero Byte Count flag was true and the FAST line was false., and the data buffer was empty.

(d) During a Data In sequence the IOP detected a Memory Address Error or the IOP detected Zero Byte Count and the Data Chain flag was false (Count Done) or the IOP detected Zero Byte Count and the Data Chain flag was true and the Interrupt on Zero Byte Count flag was true and the Fast line was false. During Data In sequences the IOP shall always empty the data buffer after ED has been set, before terminating the sequence, unless ED was set because an IOP Halt condition was detected.

When the IOP has driven the End Data line the IOP shall delay at least 185 nanoseconds before raising the RSA line.

The ES line shall be driven by the IOP only. The ES line shall signify, when true, that the Device Controller service shall terminate; that is, the Device Controller shall not raise RS again during the current service sequence. The Selector IOP shall control the ES line such that a Terminal Order shall always be sent to the Device Controller to terminate an Order or Data service sequence. The timing of the ES line respect to the RSA line shall be the same as the timing of the DA0-7 lines during the Terminal Order exchange.

7.3.2.9 Parity Check (PC) Line:

The PC line shall be driven by Device Controllers in which it use has been specified. The IOP; however, shall only respond to the PC line if the DX2 and DX4 lines are false. If the PC line is true under this condition then the IOP shall perform an odd parity check on the DA0-7 and DAP lines. The PC line, if driven, shall be raised with the initial RS of a Data In sequence and shall not be dropped until RSA and ES have been signalled by the IOP.

7.3.2.10 Interrupt Call (IC) Line:

The IC line shall be driven by the Device Controller to make an I/O Interrupt request to the CPU via the IOP. The IOP shall set an Interrupt Request memory element whenever the IC line is true and the CPU is not requesting an AIO service from the IOP. The IOP shall clear the Interrupt Request whenever the Reset I/O (RIO) line is driven or during the execution of an AIO request from the CPU. The IOP shall drive the Interrupt Request (IR) line at the CPU interface from the output of the Interrupt Request memory element.

7.3.2.11 Zero Byte Count Interrupt (ZBCI) and Fast Device Controller (FAST) Lines:

The FAST line shall be driven by those Device Controllers in which its use has been specified during Data Out/In service sequences. The IOP shall interpret this line as follows:

If, during Data Out/In service the FAST line is true and the Data Chain flag is true, the IOP shall not terminate a service sequence to send a Zero Byte Count Interrupt to the Device Controller via the Terminal Order.

The FAST line, if driven, shall be raised with the initial RS of a Data Out/In sequence and shall not be dropped until RSA and ES have been signalled by the IOP.

The ZBCI line shall be driven by the IOP and shall have the same state as the Zero Byte Count Interrupt status bit. If the Device Controller has been specified to drive the FAST line then it shall also take action when the ZBCI line goes true whenever the Device Controller is connected to the IOP for a Data Out/In service sequence. The device Controller shall take the same action when the ZBCI line is true under the condition described above as the Device Controller would take when the DAO (Interrupt) line is true during a Terminal Order exchange.

7.3.2.12 Enable two-byte Data path (EDX2), Enable Four-Byte data path (EDX4), two byte interface request (DX2), and four-byte Interface Request (DX4):

The SIOP shall always hold the EDX2 and EDX4 lines true. This signifies that the IOP can communicate with a DC (Device Controller) on a two-byte data path or four-byte data paths in addition to communication on a one-byte data path. The DX2 and DX4 lines shall be driven by those DC'S for which their use has been specified. The DX2 and DX4 lines shall specify to the IOP the width of the DC interface data path during Data Out/In exchange according to the following:

DX2	DX4	
0	0	One-Byte Data Path
1	0	Two-Byte Data Path
0	1	Four-Byte Data Path

The DX2 or DX4 lines, if driven, shall be raised with the initial RS of a Data Out/In sequence and shall not be dropped until RSA and ES have been driven by the IOP.

7.3.2.13 I/O Reset (RST), One Mhz clock (CL1), AVI from IOP (TRUE), and Reset I/O from JX58 (TESTRIO) lines: The I/O Reset (RST) line shall be driven by the IOP and shall have the same state as the RIO line driven by CPU or the TESTRIO line driven at the Device Controller interface by the JX58 tester when connected. The One Mhz Clock (CL1) line shall be driven by the IOP and shall have the same state as the CL1S line driven by the CPU.

2 lines shall be driven by the IOP when the IOP logic power supply is on to provide the AVI signal.
Device Controller connected to the IOP.

ne shall be driven at the Device Controller interface by the JX58 tester when connected to the IOP. This line
for initializing the IOP. The TESTRIO line shall cause the same IOP response as does the RIO line driven by the CPU.

OP OPERATIONS DURING DEVICE CONTROLLER SERVICE REQUESTS

of the IOP to Device Controller service requests will be defined in this section. Responses of the IOP to CPU service
have been defined in Sections 6.0 and 7.3.2.

viously defined in Section 7.3.2 the IOP shall respond to a Service Call from a Device Controller by raising the ASC Func-
Indicator line, delay, and then raise the Function Strobe line. A Device Controller indicates that it will connect for service
raising the Function Strobe Acknowledge Line; otherwise, the lowest priority Device Controller drives the AVO line and the
rvice request is aborted.

The Device Controller, if connected for service, then initially raises the Request Strobe line. The IOP detects the type of service (Order Out, Order In, Data Out, or Data In) by strobing and decoding the DOR and IOR lines. If an Order In operation is specified the IOP strobes the Operational Status Byte on the DA0-7 lines or strobes the data on the appropriate Data Lines if this is a Data In operation. If an output operation is specified the IOP shall access core memory and if this is an Order Out place the order on the DA0-7 lines or if this is a Data Out operation place data on the appropriate Data Lines. The state of the End Data and End Service lines shall then specify how the service shall terminate. The Selector IOP shall control the ES line such that a Terminal Order is exchanged to terminate every Device Controller service request.

In the paragraphs that follow the action of the IOP in response to the specific types of service requests (Order Out/In, Data Out/In) will be defined.

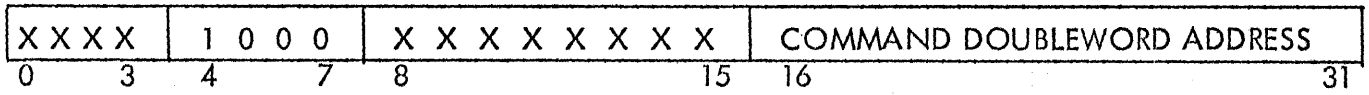
7.4.1 Order Out

If the Device Controller has requested that the IOP perform an Order Out operation by raising DOR and IOR with the initial RS then the IOP shall take the following action:

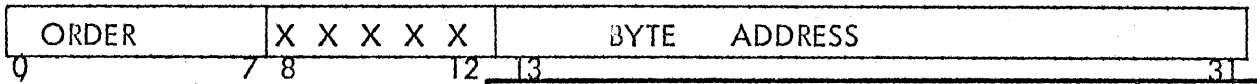
(a) Examine the state of the Chaining Modifier bit. If the Chaining Modifier is true the IOP shall increment the current Command Doubleword address by one and clear the Chaining Modifier bit.

(b) Request a core memory read operation from the the memory location specified by the Command Doubleword address. If a Memory Address Error was detected the IOP shall set the Memory Address Error and IOP Halt status bits. If a memory parity error was detected then the IOP shall set the IOP Memory Error and IOP Halt status bits. In either case the IOP shall respond immediately to the initial RS by raising RSA and shall report - at least - IOP Halt in the ensuing Terminal Order.

(c) If a Memory Address Error or Memory Parity Error was not detected then the IOP shall operate on the even Command word as follows:

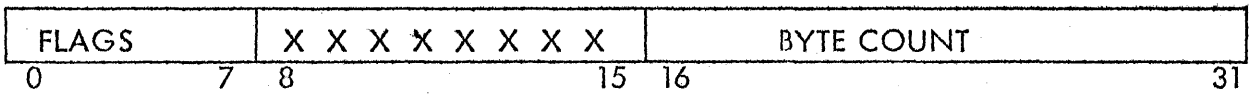


If bits 4-7 are coded as shown above then the IOP shall interpret this as a Transfer In Channel Command, ignore the state of bits 0-3 and bits 8-15 and store the contents of bits 16-31 into the Command Address register. If this is the second successive Transfer In Channel Command detected by the IOP then the IOP shall set the IOP Control Error and IOP Halt status bits and immediately respond to the initial RS by raising RSA and report - at least - IOP Halt in the Terminal Order. Otherwise, the IOP shall access the core memory location specified by the Command Address register and operate on the even Command word as follows (if no Memory Address Error or Memory Parity Error):



The IOP shall transfer bits 0-7 unchanged to the DA0-7 lines, delay, and then raise RSA. The IOP shall also examine bits 0-7 to determine whether the Read Backward order (XXXX 1100) has been coded. If this order has been coded then the IOP shall store this state as a control function during Data In operations. The IOP shall also examine the state of bit 7 of the Order field and initialize the data buffer accordingly. The subsequent service shall be interpreted as Data Out if bit 7 is true and as Data In if bit 7 is false. The IOP shall ignore the state of bits 8-12 and shall store bits 13-31 into the Byte Address register.

(d) The IOP shall then increment the even Command word address by one and access core memory for the odd Command word and shall also increment the Command Doubleword Address register by one. The IOP shall operate on the odd Command word as follows (if no Memory Parity Error):



Bits 0-7 contain the flags. The IOP shall store these into the flags register. The response of the IOP to the flags will be defined below. The IOP shall ignore the state of bits 8-15 and shall store bits 16-31 into the Byte Count register.

(e) The response of the IOP to the flags shall be as follows:

Bit 0: Data Chain (DC) flag - During a Data Out/In operation if the Byte Count has gone to zero and the Data Chain flag was true then the IOP shall access the next Command Doubleword. During the Data Chaining operation the IOP shall perform all those operations previously defined for the Order Out operation except that the IOP shall not send the order in bits 0-7 of the Command even word to the Device Controller. If a Zero Byte Count Interrupt has been requested then the IOP shall, during a Data In operation if the FAST line was false, cause End Data to be set, empty the data buffer, and in the ensuing Terminal Order report - at least - Interrupt. If a Zero Byte Count Interrupt has been requested during a Data Out operation and the FAST line was false then the IOP shall cause End Data to be set and report - at least - Interrupt in the ensuing Terminal Order.

Bit 1: Interrupt on Zero Count (IZC) flag - If, during a Data Out/In operation the Byte Count has gone to zero and this flag was true the IOP shall set the Zero Byte Count Interrupt status bit and shall drive the ZBCI line at the Device Controller interface with the same state as the status bit. If the Data Chain flag was true and the FAST line was false the IOP shall also terminate the service sequence and report - at least - Interrupt in the Terminal Order. Otherwise, the IOP shall terminate normally and report Interrupt in the Terminal Order.

Bit 2: Command Chain (CC) flag - During each Terminal Order in which the CC flag is true, the IOP shall report Command Chain.

Bit 3: Interrupt at Channel End (ICE) flag - If, during an Order In, the Device Controller reported Channel End and this flag was true then the IOP shall set the Channel End Interrupt status bit and shall report Interrupt in the ensuing Terminal Order.

Bit 4: Halt on Transmission Error (HTE) flag - This flag may only be changed during an Order Out and may not be changed during Data Chaining. If, during an Order In, the HTE flag was true and the Device Controller reported Transmission Error or reported Incorrect Length and the Suppress Incorrect Length flag was false then the IOP shall set the IOP Halt status bit and shall report IOP Halt in the ensuing Terminal Order. During a Data In operation if the HTE flag was true and the IOP detected a Transmission Data Error (byte parity error at Device Controller interface) then the IOP shall, at the next termination, set the IOP Halt status bit and report IOP Halt in the Terminal Order. If, during Data Out/In operations, the HTE flag was true and the IOP detected a Transmission Memory Error (memory parity error during read or partial write for data exchange) then the IOP shall, at the next termination, set the IOP Halt status bit and report IOP Halt in the Terminal Order.

Bit 5: Interrupt on Unusual End (IUE) flag - If, during an Order In, the Device Controller reported Unusual End or if the IOP Halt state existed at that time and the IUE bit was true then the IOP shall set the Unusual End Interrupt status bit and report - at least - Interrupt in the ensuing Terminal Order.

Bit 6: Suppress Incorrect Length (SIL) flag - If, during an Order In operation, the SIL flag was false and the HTE flag was true and the Incorrect Length status bit has been set then the IOP shall set the IOP Halt status bit and report - at least - IOP Halt in the ensuing Terminal Order. Note that the Incorrect Length status bit may be set during an Order In if the Device Controller reported Incorrect Length or during an Data In operation if the Data Chain flag was false and Zero Byte Count was detected and the IOP data buffer was not empty.

Bit 7: Skip (S) flag - If this bit was true the IOP shall during Data Out/In operations perform all functions associated with that operation except core memory accesses.

(f) After operating on the Command odd word as defined in (d) above the IOP shall wait for the Terminal Order Request Strobe from the Device Controller. The IOP shall then set End Service, place the Terminal Order on the DA0-7 lines, delay, and raise RSA to terminate the Order Out sequence. The information exchanged during the Terminal Order will be defined in a later section.

7.4.2 Order In

If the Device Controller has requested that the IOP perform an Order In by raising only the DOR line with the initial RS then the IOP shall take the following action:

(a) The IOP shall strobe the End Data line and the DA0-7 lines. Sensing that the Device Controller has raised the ED line, the IOP shall also drive the ED line. The IOP shall respond to the Operational Status Byte on the DA0-7 lines as follows:

DA0 - Transmission Error: If this line was true the IOP shall set the Transmission Data Error status bit. If the HTE flag was true then the IOP shall also set the IOP Halt status bit and report IOP Halt in the ensuing Terminal Order.

DA1 - Incorrect Length: If this line was true the IOP shall set the Incorrect Length status bit. If the SIL flag was false and the HTE flag was true the IOP shall also set the IOP Halt status bit and report IOP Halt in the ensuing Terminal Order.

DA2 - Chaining Modifier: If this line was true then the IOP shall store this condition. During the next Order Out the IOP shall increment the Command Doubleword Address in the Command Address register by one, as described in Section 7.4.1, and shall clear the Chaining Modifier control bit. IOP shall also during a successful SIO operation clear the Chaining Modifier control bit.

DA3 - Channel End: If this line was true the IOP shall, if the Interrupt at Channel End flag was true, set the Channel End Interrupt status bit and shall also signal Interrupt in the ensuing Terminal Order. If the Command Chain flag was false or if the IOP Halt status bit was set then the IOP shall also clear the IOP Busy status bit.

DA4 - Unusual End: If this line was true the IOP shall, if the Interrupt at Unusual End flag was true, set the Unusual End Interrupt status bit and also signal Interrupt in the ensuing Terminal Order. The IOP shall also clear the IOP Busy status bit.

(b) The IOP shall delay after raising the ED line and then raise the RSA line. The IOP shall then wait for the Terminal Order Request Strobe, place the Terminal Order on the DA0-7 lines, raise the ES line, delay, and raise RSA to terminate the Order In sequence.

7.4.3 Data Out

If the Device Controller requests that the IOP perform a Data Out operation by raising only the IOR line with the initial RS of a service sequence then the IOP shall respond as follows:

(a) The IOP shall access the core memory address designated by the Byte Address register and shall load that data into the data buffer. The IOP shall increment the Byte Address register and decrement the Byte Count register. This cycle shall be repeated until either the Byte Count goes to zero or End Data is set. If the Byte Count goes to zero then the IOP shall perform Data Chaining if the Data Chain flag is true or terminate the service sequence (Count Done).

(b) At the Device Controller interface the IOP shall access the data buffer, align the data as required by the Interface data path width and the state of the Byte Address and Byte Count register, place the data on Data Lines, generate the DAP parity bit, delay, and raise the RSA line.

(c) The preceding operations shall continue until the End Data line has been raised. The IOP shall then wait for the Terminal Order Request Strobe, place the Terminal Order on the DA0-7 lines, raise the ES line, delay, and raise the RSA line to terminate the service sequence.

7.4.4 Data In

If the Device Controller requests that the IOP perform a Data In operation by holding DOR and IOR false with the Initial RS then the IOP shall respond as follows:

(a) The IOP shall delay after receiving each RS, strobe the Data lines (except during the Terminal Order), check parity if required, load the data into the data buffer, and raise RSA. The IOP shall repeat these operations at the Device Controller Interface until the End Data line has been raised.

(b) At the core memory interface the IOP shall access the data buffer, align the data as required by the current state of the Byte Address and Byte Count registers, and request core memory write or partial write operations. During each core memory interface cycle the IOP shall increment the Byte Address register (decrement if Read Backward was indicated) and decrement the Byte Count register. The IOP shall repeat these operations until a Memory Address Error occurs, or the Byte Count goes to zero, or End Data has been signalled. If a Memory Address Error occurs the IOP shall terminate the Data In sequence by causing End Data to be raised at the Device Controller interface. If the Byte Count goes to zero and the Data Chain flag is true the IOP shall perform the chaining operation and then return to the exchange of data between the data buffer and core memory. If the Byte Count goes to zero and the Data Chain flag is false then the IOP shall cause End Data to be raised and terminate. If End Data has been raised then the IOP shall empty the data buffer (except for Memory Address Error or Zero Byte Count and Data Chain flag false) and terminate the Data In sequence.

(c) The preceeding operations shall continue until ED and RSA have been signalled. The IOP shall then wait for the Terminal Order Request Strobe, place the Terminal Order on the DA0-7 lines, raise the ES Line, delay, and raise RSA.

7.4.5 Terminal Order

As previously defined the IOP shall terminate each Device Controller service sequence with a Terminal Order exchange. The definition of the DA0-3 Data lines during the Terminal Order exchange shall be as follows:

DA0 Interrupt: This line shall be raised by the IOP during the Terminal Order for the following reasons:

(a) During an Order In the Interrupt at Channel End flag was true and the Device Controller signalled Channel End.

(b) During an Order In the Interrupt at Unusual End flag was true and the Device Controller signalled Unusual End or the IOP Halt status bit was true.

(c) During Data Out if the Byte Count goes to zero and the Data Chain and Interrupt at Zero Byte Count flags are true or if Count Done is signalled in the Terminal Order and the Interrupt at Zero Byte Count flag is true.

(d) During Data In if the Byte Count goes to zero and the Interrupt at Zero Byte Count flag is true.

DA1 Count Done: The IOP shall raise this line during the Terminal Order for the following reasons:

(a) During Data Out operations if the Byte Count has gone to zero, and the Data Chain flag was false, and the IOP had transmitted all data in the data buffer to the Device Controller.

(b) During Data In operations the Byte Count has gone to zero and the Data Chain flag was false.

DA2 Command Chain: The IOP shall raise this line during every Terminal Order exchange in which the Command Chain flag is true.

DA3 IOP Halt: The IOP shall raise this line during every Terminal Order exchange in which the IOP Halt status bit is true.

7.5 CABLE AND CONNECTOR PIN ASSIGNMENT

Six cables shall interconnect the IOP and Device Controllers. As specified in section 4.0, cables shall be connected by cable connectors to Cable Driver/Receiver Modules. The maximum length of cable interconnecting the IOP and the furthest Device Controller shall not exceed 100 feet. (priority cable may be 200 feet). The pin assignment of signals in those cables at the Cable Driver/Receiver and Cable Connector Interface shall be as specified below:

Cable 1

AT11 Cable Driver/Receiver Module

Connector Pin	Signal Desig. at Connector	Signal Function Description
01	DA7	Data Input/Output, Byte 1 (2^0)
02	DA6	Data Input/Output, Byte 1 (2^1)
03	DA5	Data Input/Output, Byte 1 (2^2)
04	DA4	Data Input/Output, Byte 1 (2^3)
05	DA3	Data Input/Output, Byte 1 (2^4)
06	DA2	Data Input/Output, Byte 1 (2^5)
07	DA1	Data Input/Output, Byte 1 (2^6)
08	DA0	Data Input/Output, Byte 1 (2^7)
09	DAP	Data Parity Input/Output, Byte 1
10	ED	End Data
11	PC	Parity Check
12	DOR	Data/Order Request
13	SC	Service Call
14	Not Assigned	

Cable 2

AT11 Cable Driver/Receiver Module

Connector Pin	Signal Desig. at Connector	Signal Function Description
01	DB0	Data Input/Output, Byte 2 (2^7)
02	DB1	Data Input/Output, Byte 2 (2^6)
03	DB2	Data Input/Output, Byte 2 (2^5)
04	DB3	Data Input/Output, Byte 2 (2^4)
05	DB4	Data Input/Output, Byte 2 (2^3)
06	DB5	Data Input/Output, Byte 2 (2^2)
07	DB6	Data Input/Output, Byte 2 (2^1)
08	DB7	Data Input/Output, Byte 2 (2^0)
09	EDX2	Enable Two-Byte Interface
10	DC0	Data Input/Output, Byte 3 (2^7)
11	DC1	Data Input/Output, Byte 3 (2^6)
12	DC2	Data Input/Output, Byte 3 (2^5)
13	DC3	Data Input/Output, Byte 3 (2^4)
14	EDX4	Enable Four-Byte Interface

Cable 3

AT11 Cable Driver/Receiver Module

Connector Pin	Signal Desig. at Connector	Signal Functional Description
01	DC4	Data Input/Output, Byte 3 (2^3)
02	DC5	Data Input/Output, Byte 3 (2^2)
03	DC6	Data Input/Output, Byte 3 (2^1)
04	DC7	Data Input/Output, Byte 3 (2^0)
05	Not Assigned	
06	DD0	Data Input/Output, Byte 4 (2^7)
07	DD1	Data Input/Output, Byte 4 (2^6)
08	DD2	Data Input/Output, Byte 4 (2^5)
09	DD3	Data Input/Output, Byte 4 (2^4)
10	DD4	Data Input/Output, Byte 4 (2^3)
11	DD5	Data Input/Output, Byte 4 (2^2)
12	DD6	Data Input/Output, Byte 4 (2^1)
13	DD7	Data Input/Output, Byte 4 (2^0)
14	Not Assigned	

Cable 4

AT10 Cable Receiver Module

Connector Pin	Signal Desig. at Connector	Signal Functional Description
01	FR7	Function Response (2^0)
02	FR6	Function Response (2^1)
03	FR5	Function Response (2^2)
04	FR4	Function Response (2^3)
05	FR3	Function Response (2^4)
06	FR2	Function Response (2^5)
07	FR1	Function Response (2^6)
08	FR0	Function Response (2^7)
09	RS	Request Strobe
10	IOR	Input/Output Request
11	FSL	Function Acknowledge Strobe
12	DX2	Two Byte Interface Signal
13	IC	Interrupt Call
14	DX4	Four Byte Interface Signal

Connector Pin	Signal Desig. at Connector	Signal Functional Description
01	RST	I/O Reset
02	CL1	Clock, 1 megahertz
03	ES	End Service
04	RSA	Request Strobe Acknowledge
05	SIO	Start I/O Function Indicator
06	HIO	Halt I/O Function Indicator
07	TIO	Test I/O Function Indicator
08	TDV	Test Device Function Indicator
09	AIO	Acknowledge Interrupt Function
10	ASC	Acknowledge Service Call Indicator
11	FS	Function Strobe
12	ZBCI	Zero Byte Count Interrupt
13	WADR	Word-Aligned Data Required
14	Not Assigned	

Cable 6

AT11 Cable Driver/Receiver Module

Connector Pin	Signal Desig. at Connector	Signal Functional Description
01	HPI*	High Priority Interrupt
02	HPS*	High Priority Service
03	FAST	Fast Device Controller
04	AVO	Available Output
05	Not Assigned#	
06	Not Assigned#	
07	TRUE-2#	AVI from IOP
08	TRUE-1#	AVI from IOP
09	Not Assigned#	
10	Not Assigned#	
11	Not Assigned#	
12	TEST RIO#	I/O Reset from JX58
13	Not Assigned#	
14	Not Assigned#	

*Not used by IOP.

#Device Controllers may not use these lines.

8.0 BUS-SHARING OPTION

In this section are described the actions and responses of a Selector IOP when this option is installed. Also defined is the Bus-sharing interface through which two Selector IOP's exchange various control signals related to this option.

8.1 Cable Connections

Selector IOP's that share a core memory bus shall connect the five IOP/Core Memory cables to a common memory bus as shown in Figure 3.3. In addition, the Bus-sharing interface of the two Selector IOP's shall be interconnected with a single cable. Signals exchanged at the Bus-sharing interface shall be via AT11 Cable Driver/Receiver modules.

8.2 Switch Settings

Two switches on the LT26 module located in each of the Bus-sharing Selector IOP's shall be set when this option is installed. The first switch designates that the Bus-share option is installed. This switch shall be set to the ONE state in each IOP. The second switch designates the relative priority of the two IOP's in gaining the use of the core memory bus when requests by each are made at the same time. The higher priority or AIOP shall have this switch set to the ONE state while the lower priority or B IOP shall have this switch set to the ZERO state.

8.3 Signal Lines Interconnecting IOP's at the Bus-sharing Interface.

8.3.1 The lines tabulated below shall interconnect two Bus-sharing IOP's at the Bus-share interface:

DESCRIPTION	DESIGNATION	NO. OF LINES	SIGNAL FLOW DIRECTION	
			A to B IOP	B to A IOP
Memory Bus Request	MEMRQ	1		X
Memory Interface Clear	MEMRESET	1	X	
Memory Cycle Request Enable	MEMCYCB	1	X	
Memory Cycle Request Disable	NMEMCYCX0	1		X
Memory Bus Request Disable	MEMCYCB/1	1	X	
Memory Bus Request Enable	NMEMCYC/1X0	1		X

8.3.2 The definition of the lines interconnecting the Bus-sharing IOP's at the Bus-share interface shall be as follows:

Memory Bus Request (MEMRQ): This line shall be driven by the low-priority (B) IOP only. The B IOP shall drive this line to request the use of the memory bus during those IOP operations in which IOP/Core Memory data exchanges are required. These operations include all CPU requests for service that are normally executed by the IOP and all Device Controller Service sequences except the Order In sequence. This line shall be dropped by the B IOP after the high priority (A) IOP has driven the MEMCYCB line. The A IOP shall take action on this request as follows:

If the A IOP is not currently using the memory bus and also has not requested use of the memory bus then the A IOP shall grant use of the core memory bus to the B IOP by raising the MEMCYCB and MEMCYCB/1 lines.

emory Interface Clear (MEMRESET)

his line shall be driven by the A IOP as the initial response to a memory bus request by either the A IOP or the B IOP. .n response to this signal the A IOP and B IOP shall clear the following IOP/Core Memory Interface lines:

Data lines	MX0 through MX31
Address lines	LX15 through LX31
Write Byte lines	MW0X through MW3X

The signal on this line shall be a positive-going pulse not less than 60 nanoseconds nor longer than 150 nanoseconds in duration.

Memory Cycle Request Enable (MEMCYCB):

The A IOP shall drive this line in response to a memory bus request by the B IOP via the MEMRQ line. The A IOP shall drop this line in response to a false signal received from the B IOP on the NMEMCYCX0 line. When the MEMCYCB line goes true the B IOP shall be enabled to make core memory accesses.

Memory Cycle Request Disable (NMEMCYCX0):

The B IOP shall normally hold this line true. After the A IOP has granted use of the core memory bus to the B IOP by raising the MEMCYCB line the B IOP shall cause this line to go false after the last memory request of the permitted sequence of requests has been made. The B IOP shall hold the line false for an interval of time not less than 60 nanoseconds nor longer than 100 nanoseconds. Permitted sequences of requests shall be as shown on the following page.

(a) During normally executed CPU service requests the B IOP shall read cell 20* if not an AIO and shall write in cells 20* and 21* before relinquishing the core memory bus.

(b) During Order Out or Data Chaining operations the B IOP shall respond to the Transfer in channel order if requested and shall access the command doubleword before relinquishing the core memory bus.

(c) During Data Out or Data In operations the B IOP shall access one core memory location and then relinquish the core memory bus.

Memory Bus Request Disable (MEMCYCB/1):

The A IOP shall drive this line true at the same time that the MEMCYCB line is driven true. The A IOP shall drop this line in response to a false signal received from the B IOP on the NMEMCYC/1X0 line. While this line is true the A IOP shall not respond to requests for the memory bus from either the A IOP or the B IOP.

Memory Bus Request Enable (NMEMCYC/1X0):

The B IOP shall normally hold this line true. After the A IOP has granted use of the core memory bus to the B IOP by raising the MEMCYCB line the B IOP shall cause this line to go false after all core memory interface operations of the last request of a permitted sequence have been completed. The B IOP shall hold this line false for an interval of time not less than 60 nanoseconds nor longer than 150 nanoseconds.

*Base 16

8.4 Cables

As specified in section 8.1 a single cable shall interconnect Selector IOP's at the Bus-share interface. The maximum length of this cable shall be 5.5 feet. The pin assignment for signals in that cable at the Cable Driver/Receiver and Cable Connector interface shall be as specified below:

AT11 Cable/Driver/Receiver Module

Connector Pin	Signal Desig. at Connector	Signal Functional Description
01	MEMCYCB	Memory Cycle Request Enable
02	MEMRQ	Memory Bus Request
03	MEMRESET	Memory Interface Clear
04	NMEMCYCX0	Memory Cycle Request Disable
05	MEMCYCB/1	Memory Bus Request Disable
06	NMEMCYC/1X0	Memory Bus Request Enable
07	Not Assigned	
08	Not Assigned	
09	Not Assigned	
10	Not Assigned	
11	Not Assigned	
12	Not Assigned	
13	Not Assigned	
14	Not Assigned	