Publication Number 43-009R01

I/O INTERFACE

DESIGN SPECIFICATION

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, N.J. 07757

PAGE REVISION STATUS SHEET

PUBLICATION NUMBER 43-009

TITLE I/O Interface Design Specification

REVISION R01 DATE September 1978

PAGE	REV.	DATE	PAGE	REV.	DATE	PAGE	REV.	DATE
i/ii	R01	9/78						
iii	R01	9/78						
iv	R01	9/78						
v/vi	R01	9/78						
1-1 thru 1-7/ 1-8	R01	9/78						
2-1 thru 2-47/ 2-48	R01	9/78						
3-1 thru 3-19/ 3-20	R01	9/78						
4-1 thru 4-4	R01	9/78						
Printed Circuit Board General Description SK-653								
1	R02	2/74						
2	R02	2/74						
3	R02	2/74						

A1598

PREFACE

This manual defines the electrical and mechanical specifications of the Perkin-Elmer Input/Output (I/O) System.

Because there are several methods of communication between the Perkin-Elmer I/O bus, peripheral devices, and/or other system elements, a particular system interface may be tailored to the individual user's requirements. Each system interface may be easily expanded as the user's requirements grow.

Chapter 1 contains a general description of the Input/Output system, including the different types of systems interface compatibility. An introduction to the I/O systems modules is also discussed in Chapter 1.

Chapter 2 describes the types of I/O Bus operation. The signal lines on the I/O Bus are defined in this chapter as well as the signal characteristics. I/O bus timing considerations are thoroughly discussed in Chapter 2. A portion of this chapter explains the various module considerations in the Perkin-Elmer I/O System.

Chapter 3 discusses examples of standard I/O bus interface logic. The design of I/O interface programming characteristics and recommended status byte formats are described and discussed in this chapter. Chapter 3 describes the DC-DC Converter that is used in the Perkin-Elmer I/O System and the rules that must be adhered to in the design of this system.

Chapter 4 describes the printed circuit board modules on which all I/O device controllers are implemented. The chapter also identifies the back panel signal layout within the I/O chassis.

The readership level of this manual is geared to the design engineer.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

FIGURES

TABLES

DRAWINGS

CHAPTER 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

The term interface is used with digital systems to define a group of logic circuits connecting two different devices, elements, or pieces of equipment. The interface logic circuits may perform control and format sequences, timing, and translation of signal voltage levels.

Digital logic systems operate with one or more sources of input data and one or more output devices. Inputs may consist of digital data or analog signals (i.e., keyboard, card reader, data set, etc.). Outputs may be to a Visual Display Terminal (VDT) or a hard copy terminal (i.e., line printer or teletype) or digital/analog control signals. For successful interfacing, each signal processed by the interfacing hardware must be adequately specified and defined.

When planning an I/O system to which specific devices must interface, each line has a dedicated function such as:

- Transfer data to or from the processor
- Convey control and timing signals to the peripheral devices.
- Transfer status information from the peripheral devices to the processor.

This document defines the electrical and mechanical specifications of the Perkin-Elmer Input/Output (I/O) system. There are several methods of communication between the Perkin-Elmer I/O bus, peripheral devices, and/or other system elements. These methods vary in speed, sophistication, and the amount of supervision required from the processor. Thus, a particular system interface may be tailored to the individual user's requirements and it may be easily expanded as the user's requirements grow.

1.2 I/O CONFIGURATION

Block diagrams of Perkin-Elmer digital systems, Figures 1-1 and 1-2, show the different types of systems interface capabilities. There are several methods of interfacing peripheral devices or system elements:

- to the Multiplexor (I/O) bus
- to the Selector Channel (SELCH) I/O bus
- to the Enhanced Memory Access Multiplexor (EMAM) I/O bus

Throughout this specification, the term "I/O bus source" is used to describe the device that generates the specific I/O bus under discussion. In the case of the multiplexor bus, the I/O bus source is the processor. The SELCH, EMAM, I/O Switch (IOS), Subchannel Controller (SCC), and I/O Bus Buffer (BB) are also I/O bus source devices.

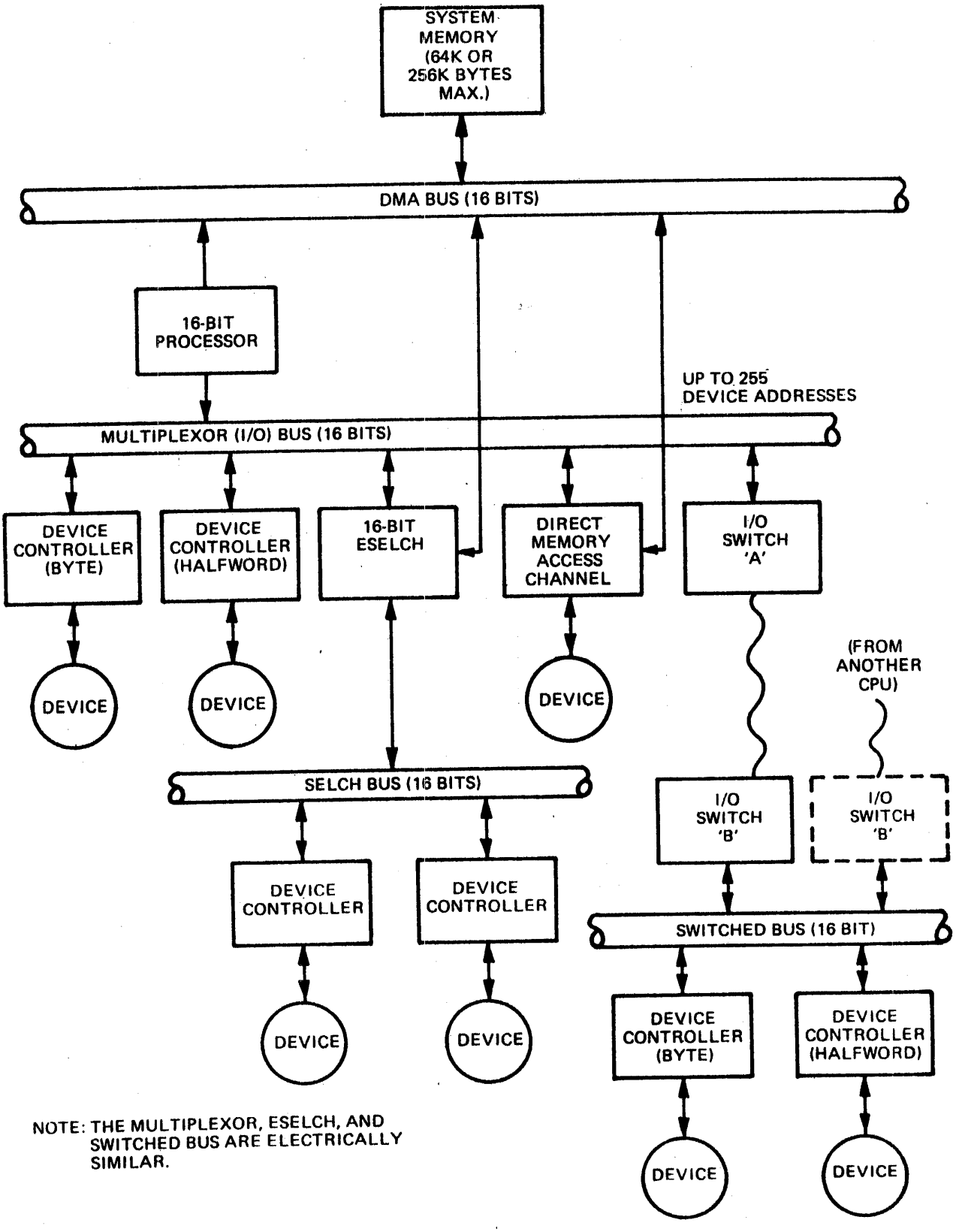


Figure 1-1 Block Diagram, Perkin-Elmer Digital Systems

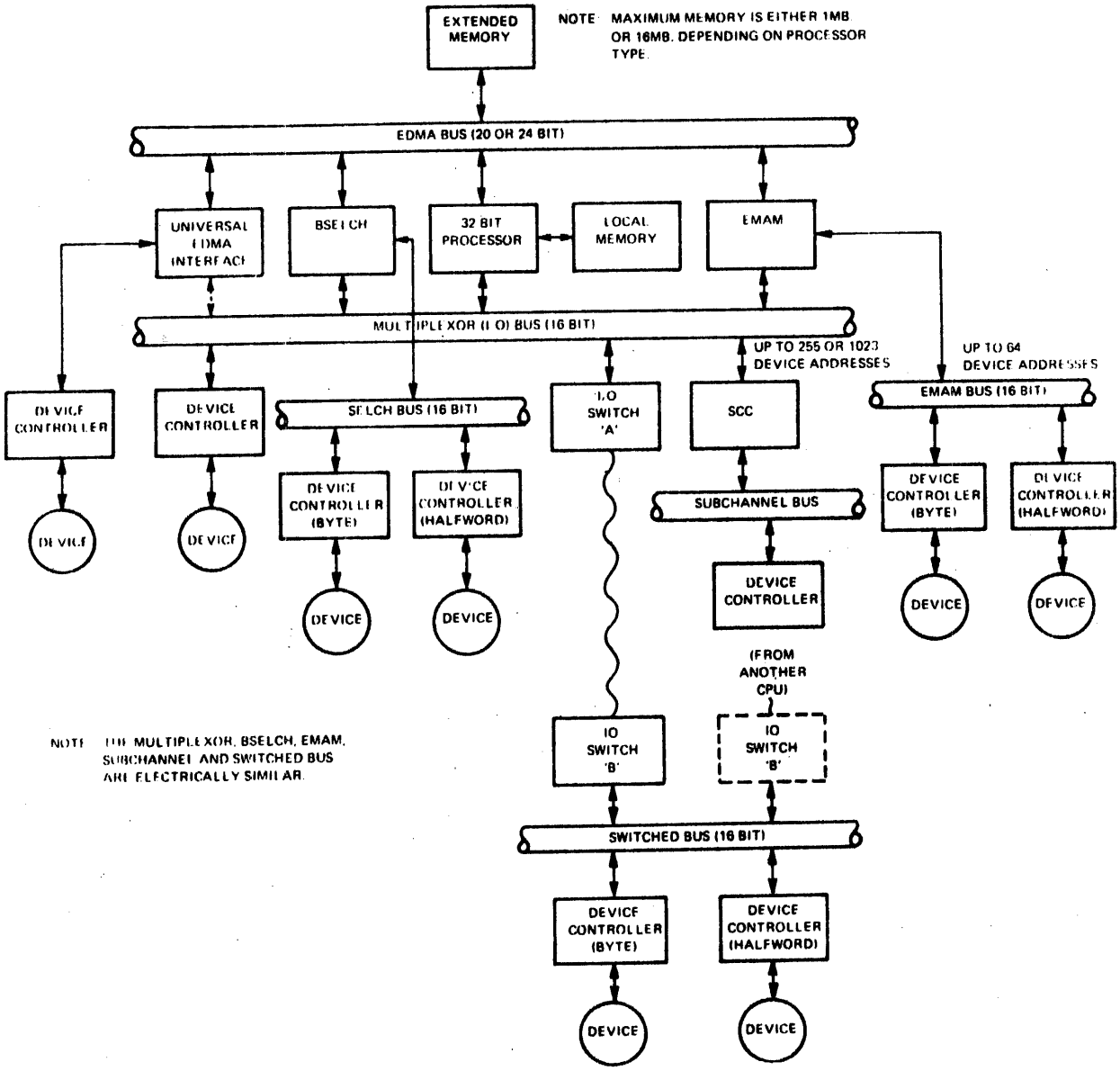


Figure 1-2 Block Diagram, Perkin-Elmer Digital Systems

1.3 INTRODUCTION TO I/O SYSTEMS MODULES

The Perkin-Elmer digital system incorporates several devices to increase I/O throughput, add configuration flexibility, and reduce I/O programming requirements. These devices are:

- Selector Channel (SELCH)
- I/O Switch (IOS)
- I/O Bus Buffer (BB)
- Sub-Channel Controller (SCC)
- Enhanced Memory Access Multiplexor (EMAM)

The above devices are described in the following paragraphs.

1.3.1 Selector Channel (SELCH)

Throughout this specification, the term SELCH is used to describe all types of Perkin-Elmer Selector Channels, unless specifically noted.

The Perkin-Elmer Selector Channel (SELCH) provides a high-speed direct memory access (DMA) port for block data transfer, bypassing the processor. For 16-bit processor systems, the 16-bit Extender Selector Channel (ESELCH) is used; for 32-bit processor systems, the Buffered Selector Channel (BSELCH) is used. The operation of these two devices is similar.

The SELCH generates a private I/O bus called the SELCH bus. When the SELCH is idle, the SELCH bus becomes an extension of the processor I/O bus. However, when the SELCH is active, the SELCH bus is disconnected from the processor I/O bus.

The SELCH operates in a status-polling mode with the selected I/O device on the SELCH bus. The SELCH uses the device controller's BUSY status bit (bit 12) to control the rate of data transfer; the transfer is terminated upon bad status from the device controller (status bit 13, 14, or 15 set) or by memory buffer transfer completion. The SELCH then raises an interrupt to the processor. For additional information on selector channel/device controller handshaking, refer to Section 2.5.1.1.

The SELCH transfers data to/from only one device controller during a block transfer. The SELCH does not perform any interrupt servicing; interrupts occurring on the SELCH bus during a block transfer are allowed to queue but are not gated to the processor until the termination of the current block transfer.

The SELCH is not capable of continuous or chained data transfers. After each block transfer is completed, the processor must issue new buffer parameters to the SELCH and restart the SELCH block transfer.

The Buffered Selector Channel (BSELCH) contains logic to support an optional high-speed handshake protocol. For further information, refer to Section 2.5.1.2.

1.3.2 Input/Output Switch (IOS)

The Perkin-Elmer I/O Switch (IOS) provides the capability of sharing I/O devices between two or more processors and/or I/O bus extension up to 100 feet. The I/O Switch may be installed on either 16- or 32-bit processors.

The IOS consists of two 7-inch by 15-inch printed circuit boards, IOS-A and IOS-B, with interconnecting cables. The IOS-A board may be installed on either the multiplexor bus, SELCH bus, subchannel bus, or EMAM bus. The IOS-B board generates a private I/O bus called the switched bus. Each IOS-A and IOS-B pair comprise a switched bus port. With the port selected, the switched bus becomes an extension of the I/O bus at the IOS-A board. Port selection can either be controlled manually or under program control. A maximum of six IOS-B boards may be configured together to form a multi-port switch bus.

For 32-bit systems, the IOS can support the optional high-speed handshake protocol between the BSELCH and device controller.

Throughput of the switched bus is affected by I/O switch cable length. For further information, refer to Section 2.5.2.

1.3.3 I/O Bus Buffer (BB)

The Perkin-Elmer I/O Bus Buffer (BB) provides increased fanout capability for the I/O bus. The BB may be installed on either 16- or 32-bit processors.

The Bus Buffer may be installed on either the multiplexor bus, SELCH bus, switched bus, or EMAM bus. The BB does not support the optional BSELCH/device controller optional high-speed handshake protocol.

1.3.4 Sub-Channel Controller (SCC)

The Perkin-Elmer Sub-Channel Controller (SCC) provides increased fanout capability for the I/O bus and provides increased I/O address capability (a maximum of 1023 device addresses). The SCC may be installed on either 16- or 32-bit processors. The SCC will be used in place of the BB for all new configurations.

The SCC has an address switch permitting the SCC to be assigned a block of 256 device addresses. Multiple SCCs can be configured to support device addresses in increments of 256 for a total of 1023 device addresses. The SCC generates a private I/O bus called the subchannel bus. All device controllers installed on the subchannel bus gain a 10-bit address identity as seen by the processor. The SCC handles detection of the high-order 2 bits of the 10-bit device address, plus appends the correct high-order bits as a device interrupt is acknowledged.

1.3.5 Enhanced Memory Access Multiplexor (EMAM)

The Perkin-Elmer Enhanced Memory Access Multiplexor provides the capability to interface up to 64 device controllers to the system memory. The EMAM can be installed only on a 32-bit processor.

The EMAM generates a private I/O bus called the EMAM bus. The EMAM services all the device controller interrupts on the EMAM bus, and transfers data between the device controller and system memory in a manner similar to the Auto Driver Channel. The EMAM bus is transparently-shared between the EMAM and the processor, allowing the processor to communicate with any device controller on the EMAM bus while one or more data transfers are in progress. The EMAM does not perform polling of the device controllers on the EMAM bus. The EMAM can detect sequences of data communications control characters (in BISYNC and ZBID protocols) for the purpose of controlling buffer switching. The EMAM is capable of continuous data transfers using a buffer-swap technique. Maximum throughput of the EMAM is approximately 200 Kbytes/second in byte mode and 400 Kbytes/second in halfword mode.

The EMAM contains SCC logic to support 10-bit I/O device addresses. The EMAM, plus all device controllers on its private bus, reside within a block of 256 device addresses assigned to the EMAM subchannel. All device controllers on the EMAM bus gain a 10-bit address identity as seen by the processor.

CHAPTER 2
I/O BUS CHARACTERISTICS

2.1 TYPES OF I/O BUS OPERATIONS

The multiplexor (I/O) bus is the primary data/control channel between the Perkin-Elmer processor and the system I/O devices. The processor initiates, monitors, and responds to all system I/O devices via multiplexor (I/O) bus operation sequences. Data transfers between the processor and the system I/O devices may either be performed directly by the processor over the multiplexor (I/O) bus, or controlled indirectly by the processor, using DMA data transfer devices such as the SELCH or the EMAM.

2.1.1 I/O Bus Operations

All sequences on the Perkin-Elmer multiplexor bus consist of at least two operations; as device address, and data/command/status are multiplexed onto one physical set of data lines. There are six types of operations that occur on the multiplexor bus:

OPERATION	PURPOSE
Address	Selects the desired I/O device.
Command	Transfers a command byte from the processor to the selected I/O device. The command byte is the primary element for control of I/O devices by the processor.
Status	Transfers a status byte from the selected I/O device to the processor. The status byte is the primary element for interrogation/monitoring of the I/O device by the processor.
Data Available (data byte/halfword)	Transfers an 8-bit data byte or 16-bit data halfword from the processor to the selected I/O device.
Data Request (data byte/halfword)	Transfers an 8-bit data byte or 16-bit data halfword from the selected I/O device to the processor.

OPERATION	PURPOSE
Interrupt Acknowledge	Transfers the address of an interrupting I/O device to the processor.

2.1.2 Programmed I/O Instructions

The machine-level instruction set of each Perkin-Elmer processor includes several instructions which perform sequences of I/O operations on the multiplexor bus. These instructions and the operations in each instruction sequence are summarized in Table 2-1.

TABLE 2-1 PROGRAMMED I/O INSTRUCTIONS

INSTRUCTION	16-BIT CPU	32-BIT CPU	I/O BUS OPERATIONS
OUTPUT COMMAND	X	X	1. ADDRESS 2. COMMAND
SENSE STATUS	X	X	1. ADDRESS 2. STATUS
WRITE DATA	X	X	1. ADDRESS 2. DATA AVAILABLE (byte)
READ DATA	X	X	1. ADDRESS 2. DATA REQUEST (byte)

TABLE 2-1 PROGRAMMED I/O INSTRUCTIONS (Continued)

INSTRUCTION	16-BIT CPU	32-BIT CPU	I/O BUS OPERATIONS
WRITE HALFWORD	X	X	1. ADDRESS if selected I/O device is a halfword device, then 2. DATA AVAILABLE (HALFWORD) otherwise: 2a. DATA AVAILABLE (most significant byte) 2b. DATA AVAILABLE (least significant byte)
READ HALFWORD	X	X	1. ADDRESS if selected I/O device is a halfword device, then 2. DATA REQUEST (HALFWORD) otherwise: 2a. DATA REQUEST (most significant byte) 2b. DATA REQUEST (least significant byte)
WRITE BLOCK	X	(some)	1. ADDRESS 2. STATUS if status bit 13, 14, or 15 set, terminate; otherwise if status bit 12 set, remain at step 2, otherwise: 3. DATA AVAILABLE (byte) go to step 2 if block transfer not complete, otherwise terminate.

TABLE 2-1 PROGRAMMED I/O INSTRUCTIONS (Continued)

INSTRUCTION	16-BIT CPU	32-BIT CPU	I/O BUS OPERATIONS
READ BLOCK	X	(some)	1. ADDRESS 2. STATUS if status bit 13, 14, or 15 set, terminate; otherwise if status bit 12 set, remain at step 2, otherwise: 3. DATA REQUEST (byte) go to step 2 if block transfer not complete, otherwise terminate.
ACKNOWLEDGE INTERRUPT	X	N/A	1. INTERRUPT ACKNOWLEDGE 2. ADDRESS (interrupting device) 3. STATUS

AUTOLOAD	X	X	1. ADDRESS (from memory location X'78')
			2. COMMAND (from memory location X'79')
			3. STATUS
			if status bit 13, 14, or 15 is set, terminate; otherwise if status bit 12 set, remain at step 3, otherwise:
			4. DATA REQUEST (byte)
			if byte is zero, go to step 3, otherwise:
			5. STATUS
			if status bit 13, 14, or 15 set, terminate; otherwise if status bit 12 set, remain at step 5, otherwise:
			6. DATA REQUEST (byte)
			go to step 5 if block transfer not complete, otherwise terminate.

TABLE 2-1 PROGRAMMED I/O INSTRUCTIONS (Continued)

INSTRUCTION	16-BIT	32-BIT	I/O BUS OPERATIONS
SIMULATE INTERRUPT	No I/O operations performed	X	1. ADDRESS 2. STATUS

2.1.3 Automatic Interrupt Servicing

The Perkin-Elmer 32-bit processors include a microcoded routine for high speed I/O interrupt servicing which performs sequences of I/O operations on the multiplexor (I/O) bus, as shown in Table 2-2.

TABLE 2-2 AUTOMATIC INTERRUPT SERVICING I/O OPERATIONS

ROUTINE	I/O BUS OPERATIONS
AUTOMATIC INTERRUPT SERVICING	1. INTERRUPT ACKNOWLEDGE 2. ADDRESS 3. STATUS if I/O device is not enabled for auto-driver channel, exit to machine-level program, otherwise enter AUTO-DRIVER CHANNEL.
AUTO-DRIVER CHANNEL	if status is acceptable, either: 4. DATA REQUEST (halfword or byte) or 4. DATA AVAILABLE (halfword or byte)

2.1.4 Automatic I/O Channel

Certain Perkin-Elmer 16-bit processors include a microcoded routine for high speed I/O interrupt servicing and control, which performs sequences of I/O operations on the multiplexor (I/O) bus as shown in Table 2-3.

TABLE 2-3 AUTOMATIC I/O CHANNEL I/O OPERATIONS

ROUTINE	I/O BUS OPERATIONS
INITIALIZATION PHASE (ENTRY VIA SINT)	1. ADDRESS 2. COMMAND
I/O OPERATION-READ (ENTRY VIA I/O INTERRUPT)	1. INTERRUPT ACKNOWLEDGE 2. ADDRESS 3. STATUS if any of status bits 12, 13, 14, or 15 set, terminate; otherwise 4. DATA REQUEST (byte) repeat step 4 until desired number of bytes have been transferred (1 to 16), then exit.
I/O OPERATION-WRITE (ENTRY VIA I/O INTERRUPT)	1. INTERRUPT ACKNOWLEDGE 2. ADDRESS 3. STATUS if any of status bits 12, 13, 14, or 15 set, terminate; otherwise 4. DATA AVAILABLE (byte) repeat step 4 until desired number of bytes have been transferred (1 to 16), then exit.

2.1.5 Additional Multiplexor Bus Operations

In addition to the previously-listed multiplexor bus sequences, the multiplexor bus is also used in certain Perkin-Elmer processors for communications between processor modules; specifically, the following processor modules, if present, communicate with the processor via the multiplexor bus:

Multiply/divide unit (16-bit processors)
Floating-point processor (except 8/32)
Communication hardware assist (32-bit processor)
Display panel
Auto-load option

In most cases, only the multiplexor bus data lines are used with these devices; special control lines dedicated for these devices are implemented at the processor.

2.2 I/O BUS DEFINITION

The I/O bus is a byte or halfword oriented I/O system which communicates with a maximum of 1023 peripheral devices. The I/O bus consists of 27 signal lines: 16 bidirectional data lines, 7 control lines, 3 test lines, and an initialize line.

Table 2-4 shows the signal lines in the I/O bus.

NOTE

All the signal lines listed in Table 2-4 are applicable to private I/O buses generated by the I/O system modules listed in Section 1.3. Each signal mnemonic on the private bus has a prefix or suffix to identify the applicable private bus:

Prefix P - BSELCH, ESELCH, SCC
Prefix M - EMAM
Suffix A - IOS, BB

TABLE 2-4 MULTIPLEXOR BUS SIGNAL LINES

TYPE	MNEMONIC	DIRECTION PROCESSOR ⇄ DEVICE	NUMBER
Data lines	D000:150	⟷	16 lines
Control lines	ADRS0	→	1 line
	SR0	→	1 line
	DR0	→	1 line
	DA0	→	1 line
	CMD0	→	1 line
	RACK0	→ (daisy-chain)	1 line
	CLO70	→	1 line
Test lines	SYN0	←	1 line
	ATN0	←	1 line
	HW0	←	1 line
Initialize line	SCLR0	→	1 line

Signal Line Definitions:

Data lines:

D000:D150

The 16-low-active data lines (D000 through D150) are used to transfer, in parallel, an 8-bit byte or 16-bit halfword of data between the I/O bus source and the device controller. In the case of an 8-bit data byte, the data byte is transferred on the least significant 8 data lines (D080 through D150, back panel pins 111 through 218).

Control lines:

ADRSO (Address)

This low-active control line is activated by the I/O bus source to all the device controllers. It is accompanied by an 8- or 10-bit device address on the data lines, to select one device controller for subsequent I/O transfers. (Back panel pin 219).

SRO (Status Request)

This low-active control line is activated by the I/O bus source to the previously-selected device controller. The device controller gates its status byte onto the data lines. (Back panel pin 119).

DRO (Data Request)

This low-active control line is activated by the I/O bus source to the previously-selected device controller. The device controller gates a byte or halfword of data onto the data lines. (Back panel pin 120).

DAO (Data Available)

This low-active control line is activated by the I/O bus source to the previously-selected device controller, accompanied by a byte or halfword of data on the data lines. (Back panel pin 221).

CMDO (Command)

This low-active control line is activated by the I/O bus source to the previously-selected device controller, accompanied by a command byte on the data lines. (Back panel pin 220).

RACKO (Interrupt Acknowledge)

This low-active control line is activated by the I/O bus source to the device controllers in a serial "daisy chain" fashion. The first device controller in the priority chain having an interrupt pending (as RACKO is received) inhibits propagation of TACKO to lower-priority devices and gates its device address onto the data lines. (Back panel pins 122 and 222).

CLO7O (Early Power-Fail Warning)

This low-active control line is activated by the I/O bus source to all the device controllers when a power-fail condition is detected by the processor. This control line is held active until initialize (SCLRO) is activated. (Back panel pin 121).

Test lines:

SYNO (Synchronize)

This low-active test line is activated by the device controller to the I/O bus source to inform the I/O bus source that the device has properly recognized and responded to a control line signal. For an address operation SYNO is activated only by the device controller being addressed. SYNO is not activated by any device controller in response to the CLO7O control line. For status request, data request, command, and data available operations, SYNO is activated only by the previously-selected device controller. (Back panel pin 123).

ATNO (Attention)

This low-active test line is activated by any device controller to the I/O bus source to inform the I/O bus source that the device controller has an interrupt pending. The device controller holds this test line active until it has received an interrupt acknowledge (RACKO) control line signal. Several device controllers may activate ATNO concurrently. (Back panel pin 223).

HWO (Halfword)

This low-active test line is activated by a halfword-oriented device controller at all times while it is addressed. A byte-oriented device controller must not activate this test line. (Back panel pin 226).

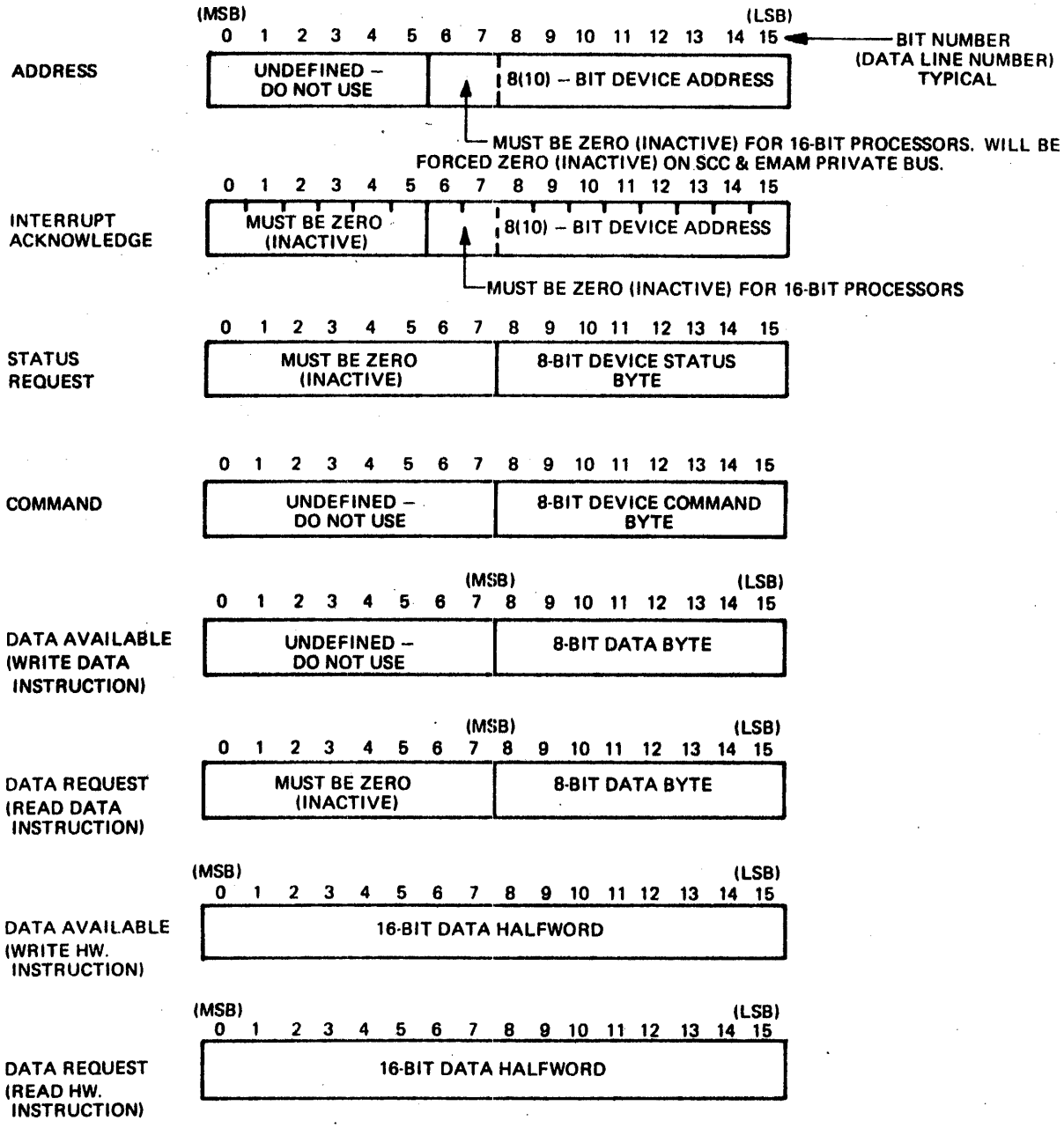
Initialize line:

SCLRO (Initialize)

This low-active line is activated during a system shutdown, power-up, or initialization operation. (Back panel pin 126).

With the exception of the serial RACKO/TACKO line, all I/O bus signal lines are connected in parallel to all device controllers on the I/C bus.

Figure 2-1 summarizes the I/O bus data line formats for each individual bus sequence.



NOTES:

1) ON INPUT OPERATIONS (INTERRUPT ACKNOWLEDGE, STATUS REQUEST, AND DATA REQUEST), THE DEVICE CONTROLLER MUST NOT ACTIVATE THE DATA BITS LABELED "MUST BE ZERO". ON OUTPUT OPERATIONS (ADDRESS, COMMAND, AND DATA AVAILABLE), THESE FIELDS ARE UNDEFINED.

2) ALL DATA LINES ARE LOW-ACTIVE (NEGATIVE-TRUE).

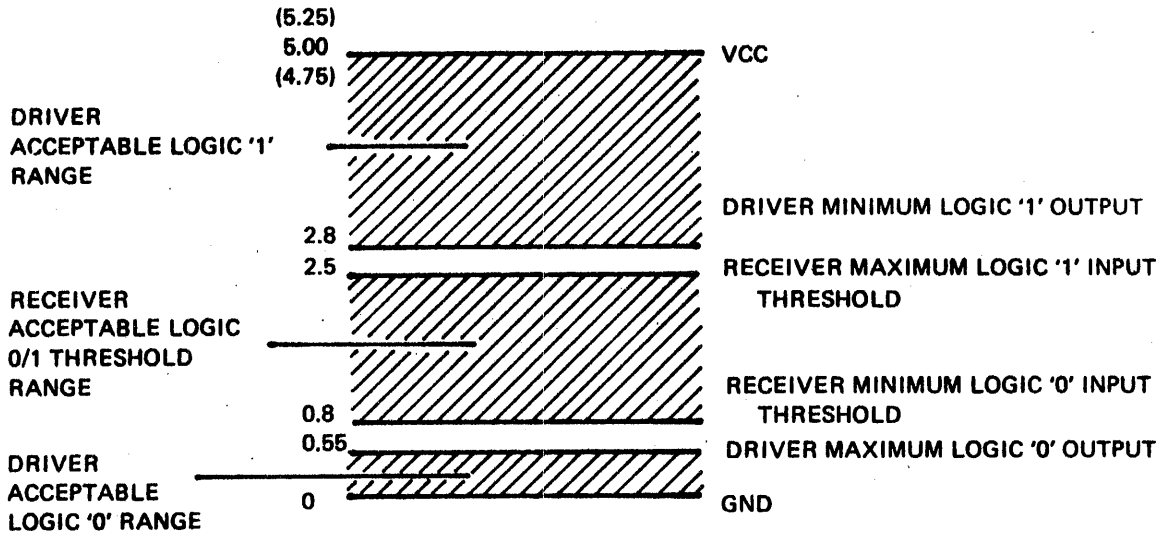
Figure 2-1 I/O Bus Data Formats (As Seen On Multiplexor Bus)

2.3 I/O BUS SIGNAL CHARACTERISTICS

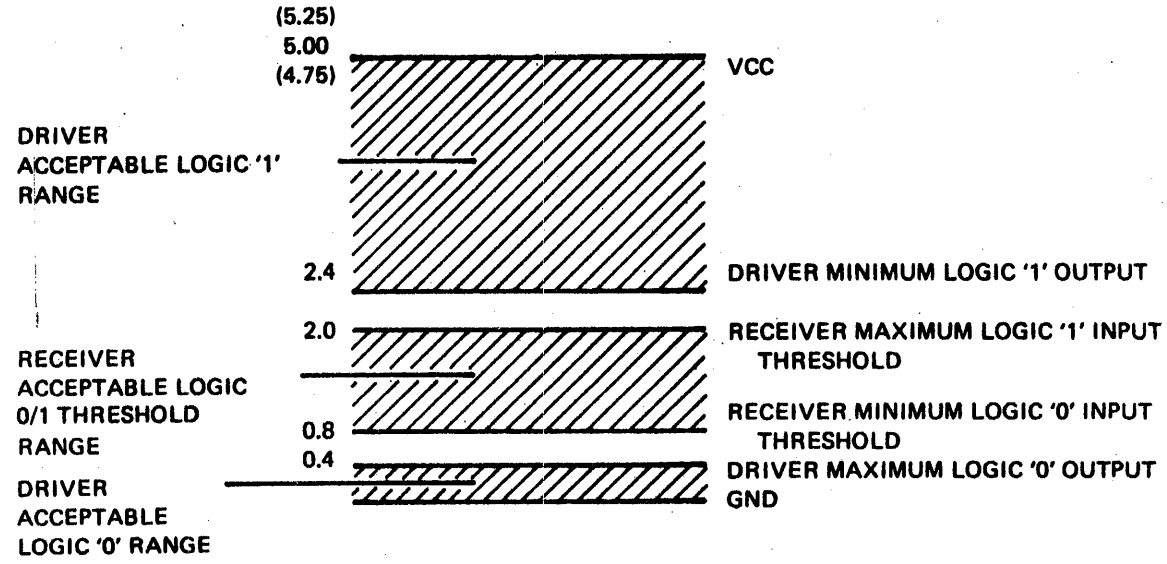
The Perkin-Elmer I/O bus data, control, initialize, and test lines may be implemented using standard 7400-series TTL (transistor- transistor logic) integrated circuit components provided that they comply with the requirements of Section 2.3.2. All signal lines are low-active and (with the exception of RACK0) must be open-collector signals tied in parallel to all I/O interfaces.

2.3.1 I/C Bus Logic Levels and Signal Loading

Figure 2-2 shows I/O bus logic levels. The logic levels shown are compatible with standard TTL levels. These logic levels guarantee a minimum noise margin of 0.25 volt at the logic low state, and 0.30 volt at the logic high state on the data lines, and 0.40 volt minimum noise margin at both the logic low and high states on all other I/O bus lines. Note that in most cases, the logic '1' voltage levels are guaranteed by the pullup resistor on each of the I/O bus signals at the bus source. Each I/O bus interface is permitted up to 2.0 milliamperes maximum low-level load and up to 300 microamperes maximum high-level load (including leakage current) on each of the I/O bus data, control, and initialize lines. Each I/O bus interface is permitted up to 750 microamperes maximum high-level leakage current on each of the I/O bus test lines.



DATA LINES D000:150



ALL OTHER SIGNAL LINES (CONTROL, TEST, INITIALIZE, INTERRUPT ACKNOWLEDGE)

Figure 2-2 I/O Bus Logic Levels

2.3.2 Standard I/O Interface Drivers and Receivers

Figure 2-3 shows the recommended IC types for each of the I/O bus data, control, initialize, and test lines. Use of the recommended IC types guarantee conformance to the I/O bus logic level and signal loading specification in Section 2.3.1, allowing configurations of up to 16 device loads on the I/O bus.

2.3.3 Use of Non-Standard I/O Interface Drivers and Receivers

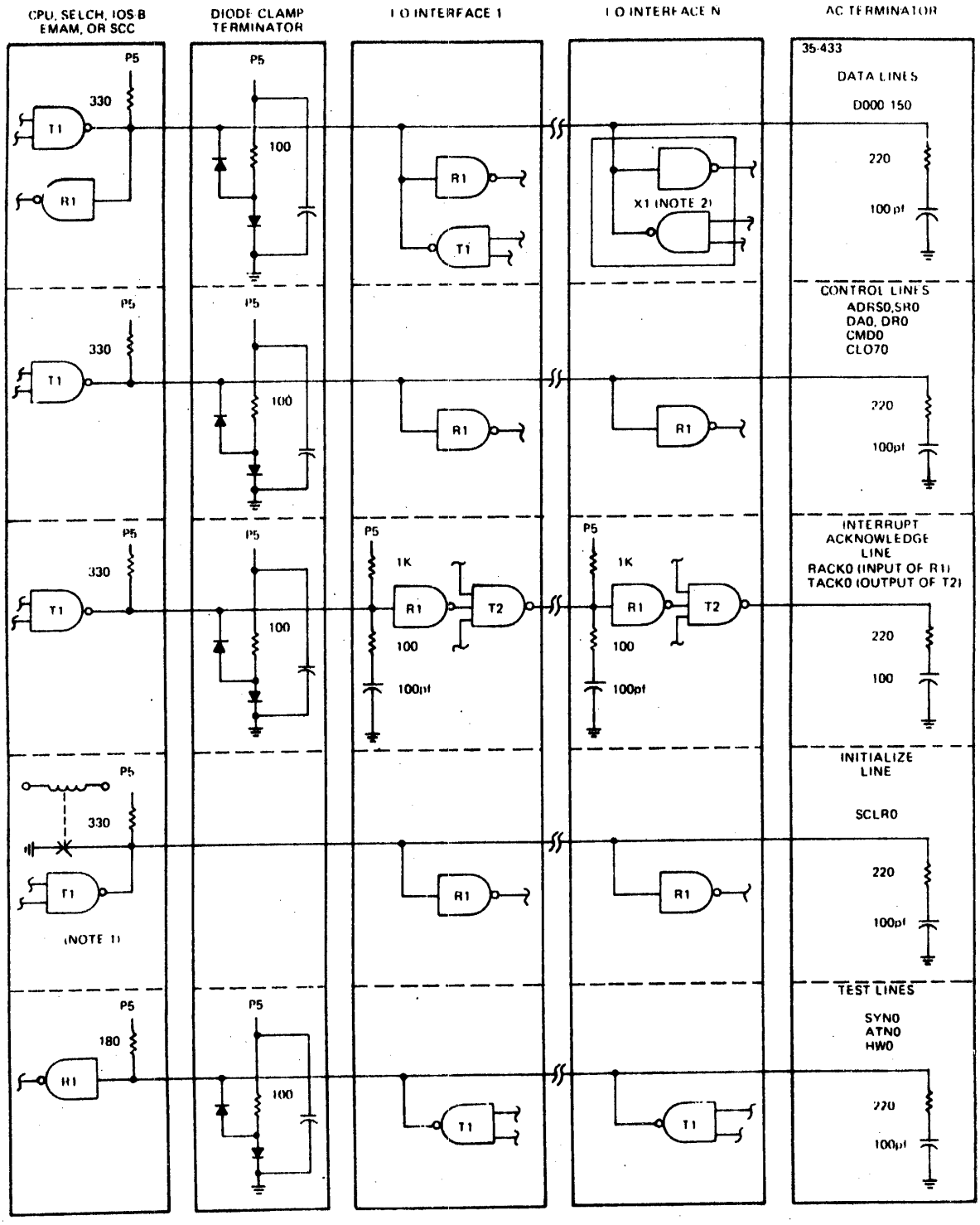
In general cases, use of IC types other than those listed in Figure 2-3 is to be avoided, unless the alternative IC types meet all the listed logic level, signal loading, drive capability, and propagation delay specifications. The use of Schottky type ICs is not allowed, with the exception of the data line transceiver IC (19-118). Tristate driver ICs are not allowed.

2.3.4 I/O Bus Loading Rules

The multiplexor bus generated by the processor is capable of driving a total of 16 I/O controllers. Certain Perkin-Elmer processors contain specialized built-in I/O controllers or processor modules which interface to the processor via the multiplexor bus; these devices are considered (if present) as one I/O load a piece. If any 10-inch device controller is present on the bus, the maximum number of I/O controllers is reduced to 9.

The BB, SCC, ESELCH, BSELCH, EMAM, or IOS extend the bus driver capability by regenerating the multiplexor bus. These devices each represent one load to the bus they are driven by. Each of these devices is capable of driving up to 16 I/O controllers. If any 10-inch device controller is present on the regenerated bus, the maximum number of I/O controllers on the regenerated bus is reduced to 9.

Table 2-5 summarizes the drive capabilities of the various Perkin-Elmer processors and system modules.



NOTE 1 RELAY IS NOT USED ON SELCH, IOS-B, SCC, OR EMAM. T1 IS NOT USED ON CPU OR BUS BUFFER

NOTE 2 X1 TRANSCEIVER MAY BE SUBSTITUTED FOR R1 AND T1 ON DATA LINES ONLY. X1 (OR PORTIONS THEREOF) MUST NOT BE USED ON CONTROL, INTERRUPT ACKNOWLEDGE, INITIALIZE, OR TEST LINES.

Figure 2-3 Recommended IC Types, Multiplexor Bus

ELECTRICAL CHARACTERISTICS
TRANSMITTER

PARAMETER	T1	T2
VOL, Low-Level Output Voltage	0.4 V max. @ 48 ma.	0.4 V max. @ 20 ma.
VOH, High-Level Output Voltage	≤VCC	2.4 V max. @ 500 ua.
IOH, High-Level Leakage Current	250 ua max. @ 5.5 V	not applicable
tPLH, Delay, Low-to-High Output	22 ns max.	10 ns max.
tPHL, Delay, High-to-Low Output	18 ns max.	10 ns max.
Recommended IC Type	7438 (19-036)	74H10 (19-017)

RECEIVER

PARAMETER	R1
VIH, Input Threshold Voltage, High	2.0 V max.
VIL, Input Threshold Voltage, Low	0.8 V min.
IIH, Input Leakage Current, High	1.0 ma max. @ 5.5 V 50 ua max. @ 2.4 V
IIL, Input Current, Low	2.0 ma max. @ 0.4 V
tPLH, Delay, Low-to-High Output	15 ns max.
tPHL, Delay, High-to-Low Output	12 ns max.
Recommended IC Type	74H04 (19-015) or 74H00 (19-016)

Figure 2-3 Recommended IC Types, Multiplexor Bus (Continued)

DATA LINE TRANSCEIVER

BUS DRIVER

PARAMETER	X1
VOL, Low-Level Output Voltage	0.5 V max. @ 43 ma 0.55 V max. @ 48 ma
VOH, High-Level Output Voltage	5.5 V max.
tpLH, Delay, Low-to-High Output (from enable input)	30 ns max.
tpHL, Delay, High-to-Low Output (from enable input)	25 ns max.

BUS RECEIVER

PARAMETER	X1
VIH, Input Threshold Voltage, High (low going input)	2.5 V max.
VIL, Input Threshold voltage, Low (high going input)	1.0 V min.
IIH, Input Leakage Current, High	1.0 ma @ 5.5 V 40 ua @ 2.4 V
IIL, Input Current, Low	1.6 ma @ 0.4 V
tpLH, Delay, Low-to-High Output	30 ns max.
tpHL, Delay, High-to-Low Output	30 ns max.
RECOMMENDED IC TYPE	Selected Am26S12A (19-118)

Figure 2-3 Recommended IC Types, Multiplexor Bus (Continued)

TABLE 2-5 I/O BUS LENGTH RESTRICTIONS

BUS SOURCE	MAXIMUM ADDITIONAL BUS LOADS	BUS LENGTH
6/16	16 (Note 1)	Note 5
8/16, 8/16E, 7/32C II	15 (Note 1)	Note 5
1610, 1620, 1630	15 (Note 3)	Note 5
8/32C/D	14 (Note 2)	Note 6
BUS BUFFER/ SUBCHANNEL CONTROLLER	16	Note 5
BSELCH, 16-BIT ESELCH	16	Note 5
I/O SWITCH	16 (Note 4)	Note 5
EMAM	15	Note 5

NOTES

1. This number reflects loading over and above load from display controller/ALO option.

2. This number reflects loading over and above loads from TTY interface and display controller.

3. This number reflects loading over and above load from STC (System Terminal Controller).

4. Each additional port (up to five additional) uses one unit load.

5. Three adjacent 7-inch chassis units. All chassis must be less than two inches apart. Maximum two interconnecting cables four inches each; or: one cable, maximum 36 inches in length, connecting to two adjacent 7-inch chassis units. Both chassis must be less than two inches apart, with interconnecting cables maximum four inches in length.

6. Maximum 36-inch cable can be driven directly by processor multiplexor bus.

2.3.5 I/O Bus Length Restrictions

The processor's multiplexor bus must be complete within the 7-inch chassis and two adjacent 7-inch expansion chassis. A multiplexor bus originating at a 14-inch twin-chassis processor may extend to one adjacent 7-inch expansion chassis only. The multiplexor bus may not be extended to any expansion chassis by use of a cable longer than 4 inches. If longer cables (up to 36 inches) are required between chassis, a BB or SCC must be used to drive the cable and extended bus.

Private I/O busses generated by a BB, SCC, ESELCH, BSELCH, EMAM, or IOC-B, must be complete within the 7-inch chassis the bus is generated in, plus a maximum of two 7-inch expansion chassis. Any private I/O bus may be extended by no more than one 36-inch (maximum) cable plus an additional cable not greater than 4 inches in length.

Configurations requiring cable lengths greater than 36 inches must use an I/O switch. The I/O switch is capable of generating a private I/O bus up to 100 feet from the I/O bus source.

2.3.6 I/O Bus Termination

Two types of I/O bus terminators are used in Perkin-Elmer systems: the 35-433 A.C. terminator and the 35-722 diode clamp terminator. An A.C. terminator must be installed at the end of each multiplexor bus or private I/O bus in the system. If the multiplexor bus or private I/O bus appears on both connector 0 and connector 1 of a chassis, A.C. terminators must be installed on both sides.

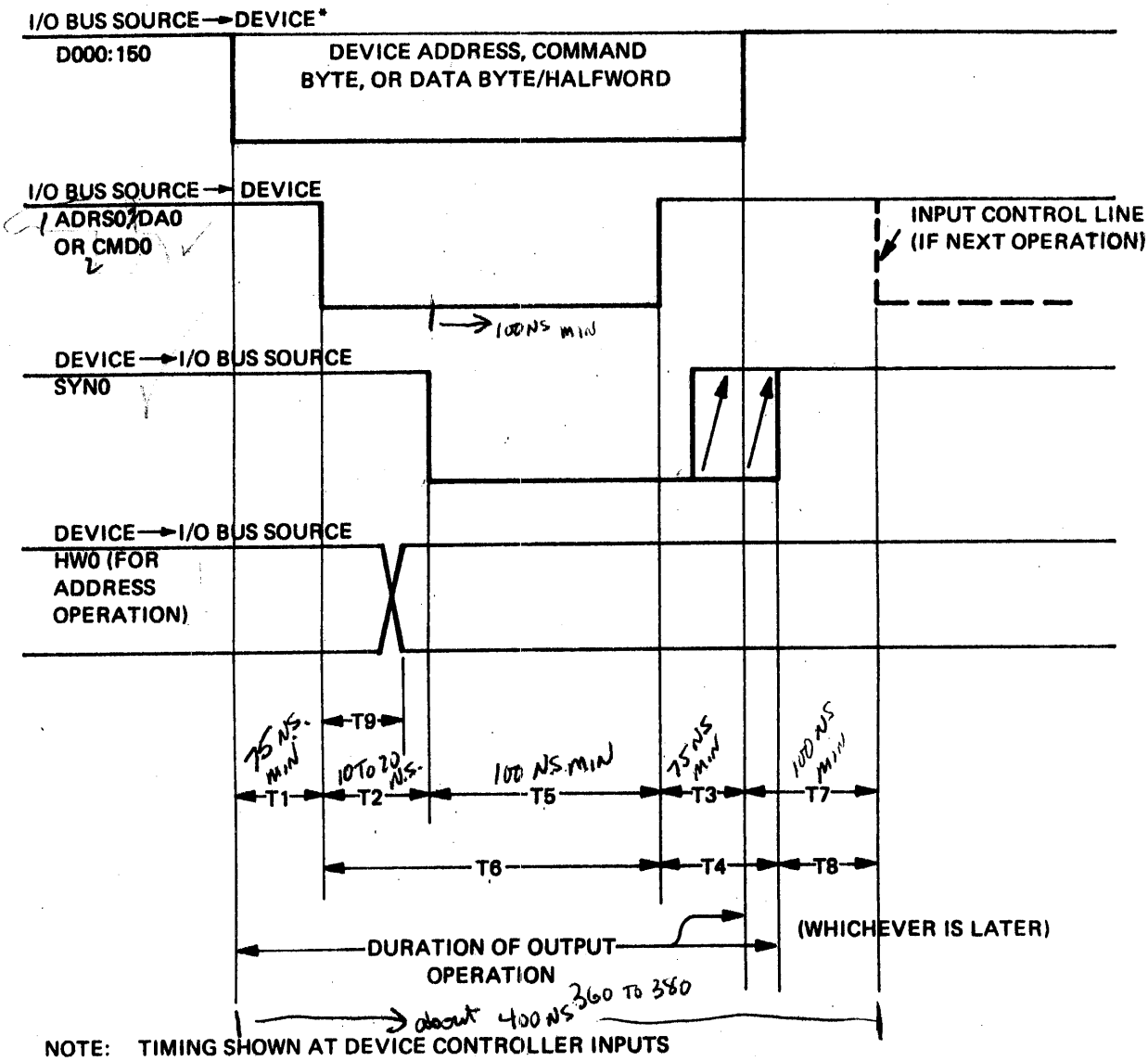
The diode clamp terminator must be installed at the source of each multiplexor bus or private I/O bus in the system. The source is considered to be the point where the bus originates on connector 0 and 1 of the chassis. If the bus originates on both connector 0 and 1 of a chassis, diode clamp terminators are required on both sides.

2.4 I/O BUS TIMING

Request/response signaling is used for both input and output I/O bus operations. This allows the I/O bus to run at its maximum speed whenever possible, but permits a graceful slowdown if the characteristics of a particular device controller require signals of longer duration. Device controller designs should keep I/O bus timing as fast as possible, consistent with practical circuit margins and the required minimum timing as shown in the following sections. Any delays in the I/O bus timing over the specified minimum timing results in increased I/O instruction execution time; hence, system throughput is degraded.

2.4.1 Output I/O Operation Timing

Figure 2-4 shows the timing for the three types of output I/O operations (Address, Command, and Data Available). To begin the output operation, the I/O bus source activates the I/O bus data lines with the desired data, followed by the activation of the appropriate I/O bus control line (ADRSO, CMDO, or DAO). The time between the activation of the data lines and the activation of the control line (T1) is variable, but is guaranteed to be at least 75 nanoseconds (minimum) at the device controller inputs. This allows the device controller to recognize an address match or decode data or command information prior to the leading edge of the control line signal. The leading edge of the control line signal should be used where edge-triggered latches or flip-flops are used to store command, data, or address match. When the device controller has accepted the command or data, or recognized an address match, it should activate the SYNO test line to the I/O bus source.



NOTE: TIMING SHOWN AT DEVICE CONTROLLER INPUTS

T1, T3 } 75 NS. GUARANTEED MINIMUM AT CONTROLLER INPUTS

T2 – AS SHORT AS POSSIBLE AFTER DEVICE RECOGNIZES ADDRESS OR ACCEPTS DATA OR COMMAND.

T4 – AS SHORT AS POSSIBLE

T5 – 100 NS. MINIMUM

T6 – 350 NS. MINIMUM FOR ADRS0. DA0 AND CMD0 HAVE NO MINIMUM BUT DEACTIVATE AFTER SYN0 ACTIVATES

T7 – 100 NS. MINIMUM

T8 -- MUST BE GREATER THAN ZERO

T9 – MUST BE LESS THAN T2

* THE PROCESSOR MUST DEACTIVATE THE DATA LINES MINIMUM 100 NANOSECONDS BEFORE ACTIVATING ANY CONTROL LINE FOR THE NEXT I/O OPERATION.

Figure 2-4 Timing For Types of Output I/O Operation

For Command and Data Available operations, the I/O bus source deactivates the control line signal (after a delay of at least 100 nanoseconds) after the device controller activated SYNO. In the case of the address operation, the I/O bus source holds the address control line active, despite the presence of SYNO, for at least 350 nanoseconds. After the control line is deactivated, the I/O bus source holds the data lines active for an additional 75 nanoseconds minimum (T3). The device controller must minimize the SYNO deactivation time (T4) after the control line signal is deactivated, as the I/O bus source cannot begin another input or output operation until SYNO deactivates. After the I/O bus source deactivates the data lines, it must wait at least 100 nanoseconds before activating any control line for an input I/O operation.

It should be emphasized that the timing shown in Figure 2-4 is defined at the device controller inputs on the I/O bus. Within a given device controller, one data line may propagate through more gates than another data line, generating a skew between data lines which must be considered for practical designs.

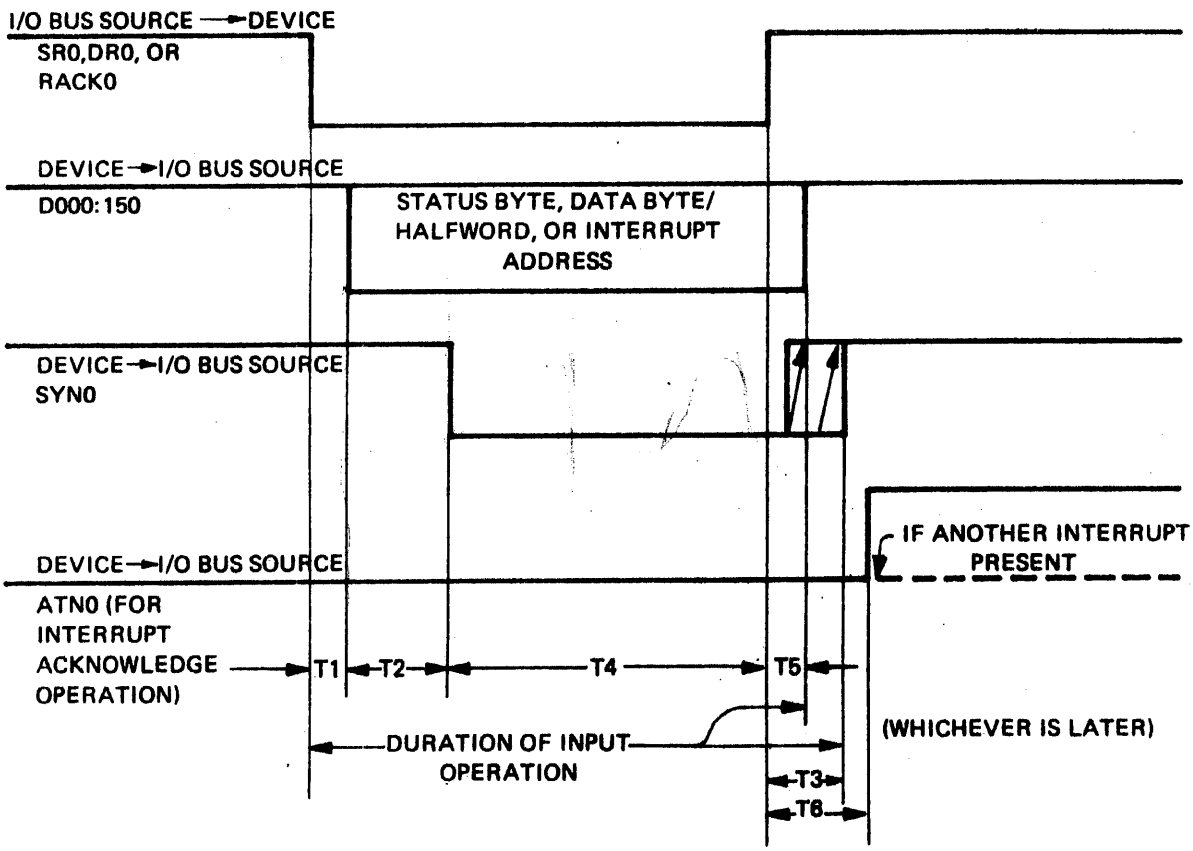
NOTE

The I/O bus source must guarantee that the data overlaps the associated control line by at least 100 nanoseconds. All device controllers must be designed assuming no more than 75 nanoseconds overlap is present, as data line skew is introduced if the controller is configured on a SELCH, IOS, or SCC bus.

2.4.2 Input I/O Operation Timing

Figure 2-5 shows the timing for the three types of input I/O operations (Status, Data Request and Interrupt Acknowledge). The input operation begins as the I/O bus source activates one of the applicable control lines (SRO, DRO, or RACKO). The device controller then gates the appropriate data onto the I/O bus data lines, with as little delay as possible, and then activates the SYNO test line. Data must be stable on the bus before SYNO is activated. The I/O bus source must take into account the worst-case data slew of 25 nanoseconds before latching the data after SYNO goes active. In response to SYNO, the I/O bus source accepts the contents of the data lines, then deactivates the control line with a minimum delay of 100 nanoseconds after SYNO is activated. As the control line is deactivated, the device controller must deactivate the I/O bus data lines and deactivate SYNO with as little delay as possible. The I/O bus source considers the operation complete when SYNO deactivates.

In the case of the Interrupt Acknowledge operation, the RACKO control line signal is propagated in daisy-chain fashion, to the highest priority (by position) device controller with an interrupt active. Note that this may or may not be the presently-selected device controller. The T1 delay in this case includes all the cumulative gate delays for all the device controllers between the I/O bus source and the responding device controller.



NOTE: TIMING SHOWN AT DEVICE CONTROLLER INPUTS

T1 - AS SHORT AS POSSIBLE
T2 - MUST BE GREATER THAN ZERO
T3 - AS SHORT AS POSSIBLE
T4 - 100 NS. MINIMUM
T5 - AS SHORT AS POSSIBLE
T6 - MUST BE GREATER THAN ZERO -- I.E. DEVICE MUST NOT
DEACTIVATE ATN0 BEFORE RACK0 (AT DEVICE INPUT) DEACTIVATES.

Figure 2-5 Timing for Types of Input I/O Operation

2.4.3 I/O Timeouts

All Perkin-Elmer processors include a mechanism to prevent a system 'hang' if the I/O device controller fails to respond (by activating SYNO) to a processor I/O operation. As the multiplexor bus control line is activated, a false SYNC timer is started. If SYNO does not activate before the timer completes its cycle, then as the timeout occurs, the processor deactivates the control line and proceeds to execute the next user instruction. The flag bits of the Program Status Word (PSW) indicate that a false SYNC timeout has taken place. Typically, a false SYNC timeout occurs due to a hardware fault or software error (i.e., attempting an I/O operation to a non-existent device controller). The time value of the false SYNC timeout varies from processor to processor; typical timeout values are summarized in Table 2-6.

TABLE 2-6 TYPICAL I/O TIMEOUT VALUES

PROCESSOR TYPE	I/O TIMEOUT VALUE (TYPICAL)
5/16	14.0 usec.
6/16	29.0 usec.
8/16, 8/16E	29.0 usec.
7/32C II	29.0 usec.
8/32C/D	25.0 usec.
EMAM	5.875 usec.

The following rules and constraints apply to I/O timeout situations:

1. The duration of the I/O bus control line pulse is undefined.

2. For output operations the I/O bus source must generate the normal data line/control line timing relationship.

3. For input operations, the I/O bus source must ignore the content of the data lines.

4. If the I/O bus source is executing an I/O instruction or microprogrammed sequence involving multiple input or output (or both) operations to a particular device controller, the I/O bus source must not attempt to perform any operation after detecting an I/O timeout from the device.

5. All the above rules apply to any I/O system module (such as a SELCH or EMAM) which performs I/O operations to devices on its private bus. In this case, the I/O system module must detect an I/O timeout and report it to the processor.

NOTE

All existing Perkin-Elmer selector channels do not detect I/O timeouts.

2.4.4 Additional I/O Timing Considerations

In addition to the timing relationships as shown in the above paragraphs, several other general rules apply:

1. No device controller or I/O system module may activate any of the I/O bus data lines except when requested by the I/O bus source; i.e., unless the device controller has previously been selected and the I/O bus source activates either the SRO or DRO control lines, or if the device controller has activated the ATNO test line and the RACKO control line goes active at the input to the device controller. No device controller may introduce any condition causing increased settling time of the multiplexor bus data lines.

2. Any I/O system module which propagates the multiplexor bus onto the module's private I/O bus has the responsibility to restore any I/O timing relationships destroyed as the I/O signals propagate through the I/O system module. A maximum delay of 20 nanoseconds (with the exception of the I/O switch) is allowed in regenerating any multiplexor bus data line to or from the private I/O bus data line.

2.5 I/O SYSTEM MODULES CONSIDERATIONS

Unique sequences of I/O operations and/or dramatic variations of I/O timing occur when operating a device controller with certain I/O systems modules listed in Section 1.3. This section presents these differences in detail.

2.5.1 Selector Channel (ESELCH or BSELCH)

Whenever the SELCH is idle, the private SELCH I/O bus becomes a regenerated multiplexor bus. When the SELCH is started (via an output command with the SELCH GO bit set to the SELCH), the SELCH bus is disconnected from the multiplexor bus and a special I/O handshake sequence begins, with the selected device controller on the private SELCH bus. There are two types of SELCH I/O handshake procedures, normal and high speed, which are described in the following paragraphs. In both cases, the desired device controller is selected before the data transfer begins by the processor performing an I/O operation (typically an output command) to the desired device controller on the private SELCH bus, then issuing an output command GO to the SELCH. As the address portion of the output command sequence to the SELCH is being performed, address match at the SELCH is used to inhibit the private SELCH bus address control line (PADRSO), causing the previously selected device controller to remain selected.

2.5.1.1 Normal Selector Channel Handshake Procedure

The normal SELCH handshake I/O timing is shown in Figure 2-6. As the SELCH transfer begins, the SELCH activates Private Status Request (PSRO) to the selected device controller. The controller responds by gating its status byte onto the private SELCH bus data lines, and then activating PSYNO (private SYNC return). BUSY (bit 12 of the status byte) controls the period between individual data transfers. BUSY is active at all times when the controller is not ready to transfer data. The SELCH holds PSRO active until PD120 deactivates (indicating the controller is now ready to transfer data), or either PD130, PD140, or PD150 activate (indicating bad status), or the processor halts the SELCH.

After PD120 deactivates, a data transfer takes place; in read mode, PDRO (private data request) is activated and the controller gates a byte or halfword of data onto the private SELCH bus, then activates PSYNO. In write mode, the byte or halfword of data is gated onto the private data lines by the SELCH, then PDAO (private data available) is activated. The controller activates PSYNO after it accepts the data. In both read and write modes, the state of PHWO controls whether a byte or halfword data transfer takes place. The selected device controller activates PHWO if it is a halfword-oriented device; with PHWO active the SELCH transfers successive halfwords of data to or from the controller. With PHWO inactive, the SELCH transfers successive bytes of data to or from the controller.

After the data transfer takes place, the SELCH continues by repeating the status check/data transfer sequence until the memory data buffer is completed or the transfer is prematurely terminated due to bad controller status.

Maximum data throughput rate for the normal handshake procedure is dependent primarily on PSYNO activation and removal times by the device controller, as seen at the SELCH. The SYNC return turn-on delay should be minimized, consistent with practical circuit margins.

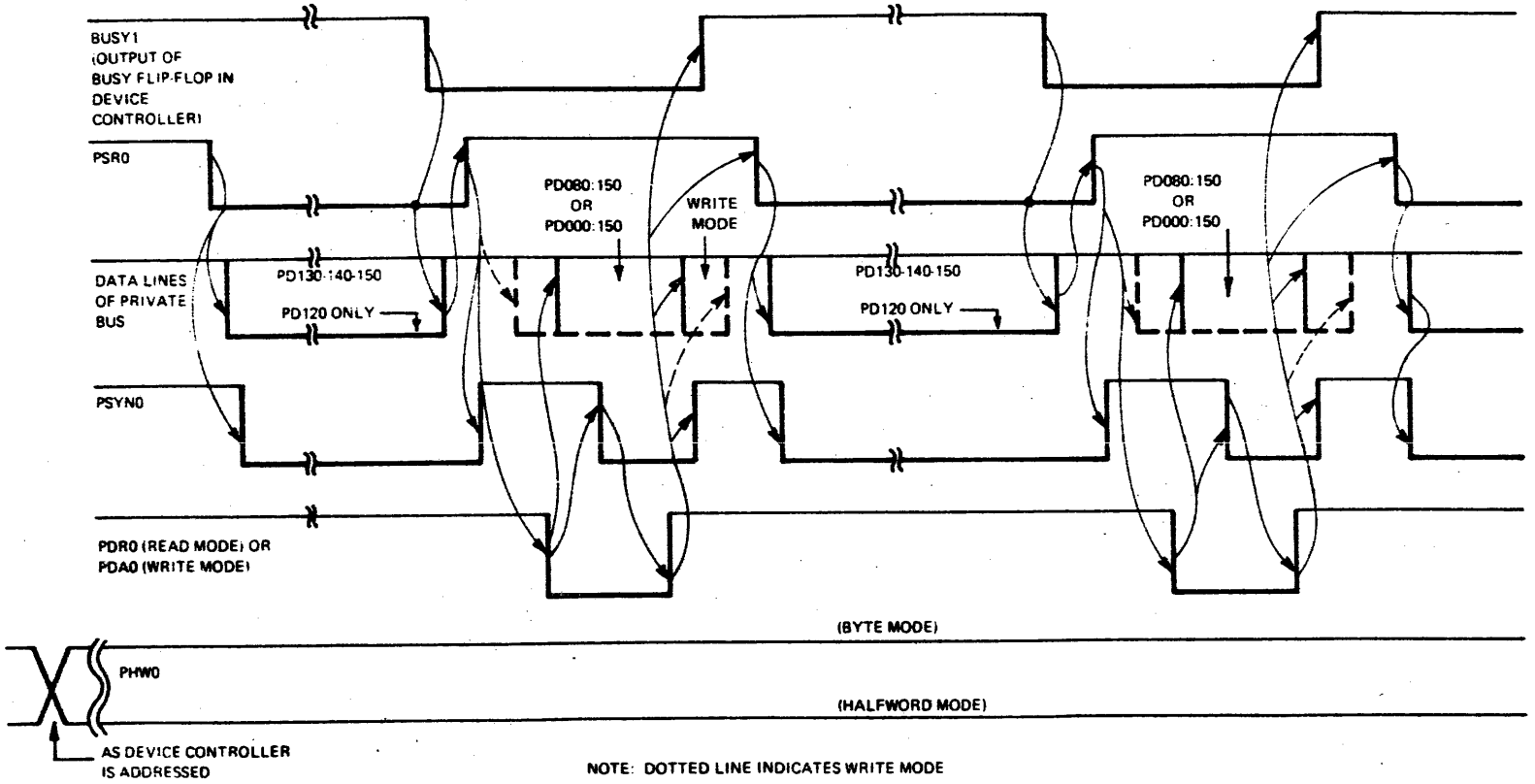


Figure 2-6 Normal SELCH Handshake Timing

Maximum data throughput may be calculated using the formula:

$$X = \frac{B}{Y + 2\,(Sn + Sm)}$$

Where X = transfer rate in Megabytes/second
B = bytes per DA/DR pulse
(B = 1 for byte devices)
(B = 2 for halfword devices)
Sn = SYNC return turnon delay time in microseconds
Sm = SYNC removal delay time in microseconds
(Sn and Sm are measured at the selector channel PSYNO input)
Y = 0.45 for 16-bit ESELCH
0.73 for BSELCH

NOTE

This formula assumes no I/O switch or SCC between the SELCH and the device controller.

2.5.1.2 High Speed Selector Channel Handshake Protocol

The high speed (new) handshake protocol has recently been introduced to support device controllers which must sustain higher throughputs than are achievable using the normal selector channel procedure. Higher throughput is achieved by eliminating the status check operation and streamlining the data transfer procedure. The new protocol is now available only on the M48-050 M01 BSELCH (32-bit machines only).

Table 2-7 shows the three additional control/test signals required for the high speed protocol (1 control, 2 test). The logic levels are the same as the control line logic levels shown in Figure 2-2. Each device controller is permitted a maximum of 2.0 milliamperes low-level load and a maximum of 100 microamperes high level (including leakage current) on SELCH busy (SBSY0). Each device controller is allowed a maximum of 250 microamperes high-level leakage current on SCHK0 and SNS0.

TABLE 2-7 HIGH SPEED PROTOCOL SIGNAL LINES

TYPE	MNEMONIC	DIRECTION BSELCH ⇄ DEVICE	NUMBER
Control Line	SBSYO	→	1 line
Test Lines	SNSO	←	1 line
	SCHKO	←	1 line

(In addition to the 27 lines listed in Table 2-4)

Signal Line Definitions:

Control Line:

SBSYO (SELCH Busy)

This low-active control line is activated by the BSELCH to the previously-selected device controller, indicating that a SELCH block data transfer is in progress. This line uses back panel pin 224-1.

Test Lines:

SNSO (Switch to New Sequence)

This low-active test line is activated by the previously-selected device controller to the BSELCH to specify that the controller supports the new protocol. It remains active at all times while the device controller is selected. This line uses backpanel pin 124-1.

SCHKO (Status Check)

This low-active test line is activated by the previously-selected device controller to the BSELCH to indicate the occurrence of a 'bad status' condition (bad status = S13 + S14 + S15). This line may be activated only while either PDAO or PDRO is active, and at least 50 nanoseconds before the device controller activates PSYNO. Once activated, it must remain active until after SBSYO deactivates. This line uses back panel pin 225-1.

All the above-listed control and test lines are connected in parallel to all devices on the SELCH bus.

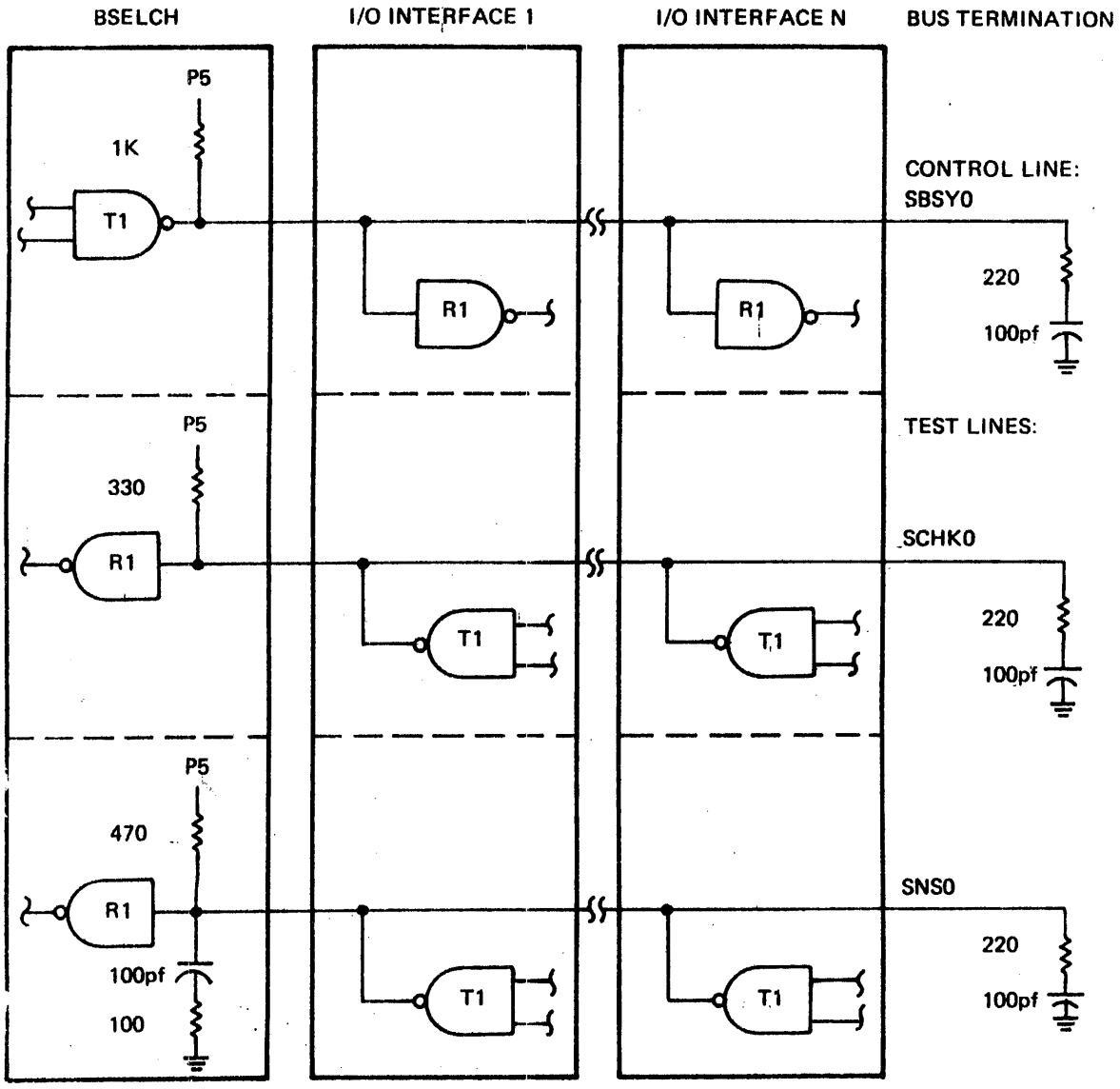
Figure 2-7 shows the recommended IC types for each of the new control and test lines. Use of the recommended IC types guarantees conformance to the logic level and signal loading specifications above, allowing configuration of a maximum of 16 device loads on the SELCH bus.

Figure 2-8 shows the high speed protocol data transfer timing. As a read mode BSELCH transfer begins, the BSELCH activates PDR0 (private data request). When the device controller is ready to transfer a byte or halfword of data (as BUSY1 deactivates), the controller gates the data onto the private BSELCH bus, then activates PSYN0. The BSELCH responds by accepting the data and deactivating PDR0, causing the controller to deactivate the private data lines and PSYN0. This handshake procedure is repeated until the data transfer terminates.

As a write mode BSELCH transfer begins, the BSELCH gates the data byte or halfword onto the private data lines, then activates PDA0 (private data available). When the controller is ready to accept the data (as BUSY1 deactivates), the controller latches the data and then activates PSYN0. The BSELCH responds by deactivating PDA0, causing the controller to deactivate PSYN0. This handshake procedure is repeated until the data transfer terminates.

In both read and write modes, the device controller may terminate the data transfer by activating SCHK0 (status check) after PDR0 or PDA0 is activated but before PSYN0 is activated. (See Figure 2-9). SCHK0 must remain active until the BSELCH deactivates SBSY0 (SELCH busy). As in normal protocol, the halfword line (PHW0) controls whether the data transfer is performed in byte or halfword mode.

As with normal protocol, maximum throughput rate is dependent primarily on PSYN0 activation and removal times by the device controller, as seen at the BSELCH. The SYNC return turn-on delay should be minimized, consistent with practical circuit margins.



NOTE 1: SEE FIGURE 2.3 FOR ELECTRICAL CHARACTERISTICS OF R1 AND T1
NOTE 2: XI (TRANCEIVER) MUST NOT BE SUBSTITUTED FOR T1 OR R1

Figure 2-7 Recommended IC Types, BSELCH High Speed Protocol Lines

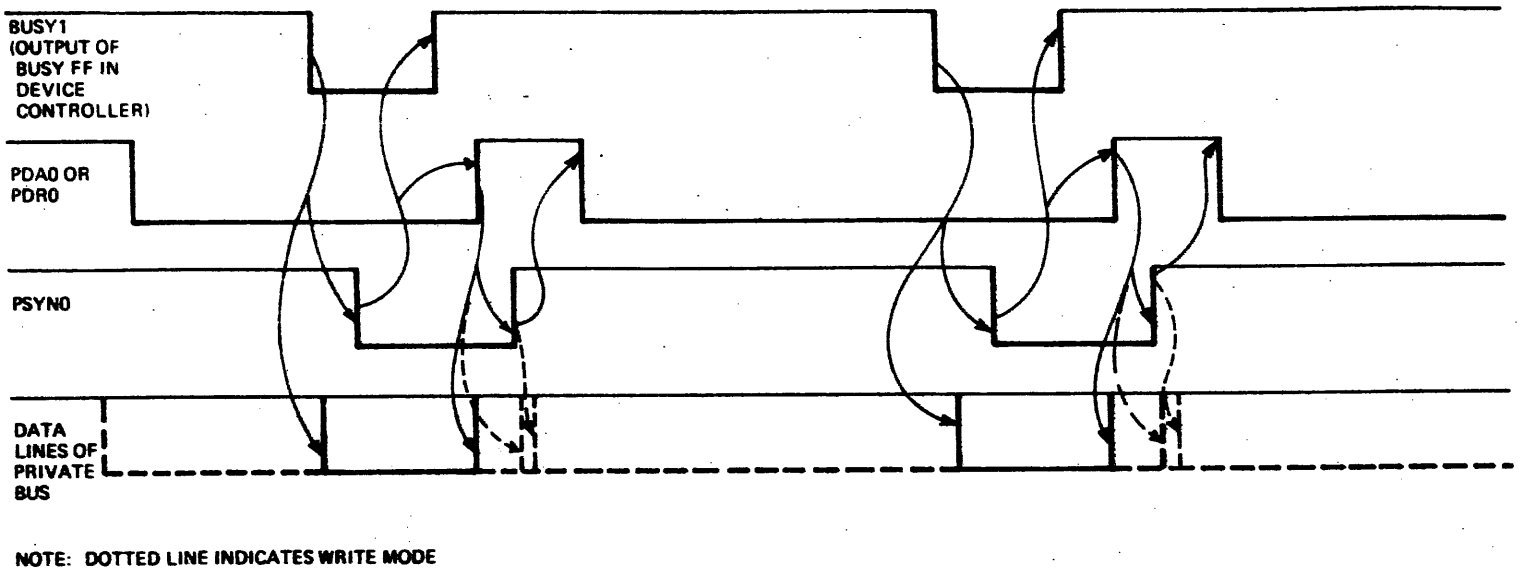


Figure 2-8 High Speed Protocol Data Transfer Timing

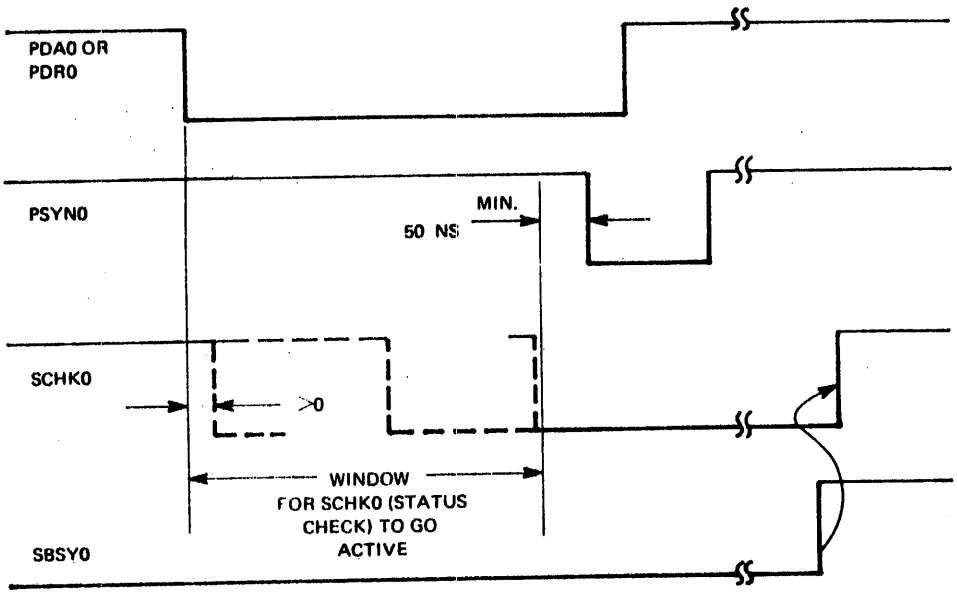


Figure 2-9 Termination of BSELCH Data Transfer

Maximum data throughput may be calculated using the formula:

$$X = \frac{B}{.390 + (Sn + Sm) + DA}$$

Where X = transfer rate in Megabytes/second
B = bytes per DA/DR pulse
(B = 1 for byte devices)
(B = 2 for halfword devices)
Sn = SYNC return turn-on delay time in microseconds
Sm = SYNC removal delay time in microseconds
(Sn and Sm are measured at the BSELCH PSYNO input)
DA = .075 for write to device,
0 for read from device.

NOTE

This formula assumes no I/O switch between the BSELCH and the device controller.

An example of the handshake logic required at the device controller is shown in Figure 2-10.

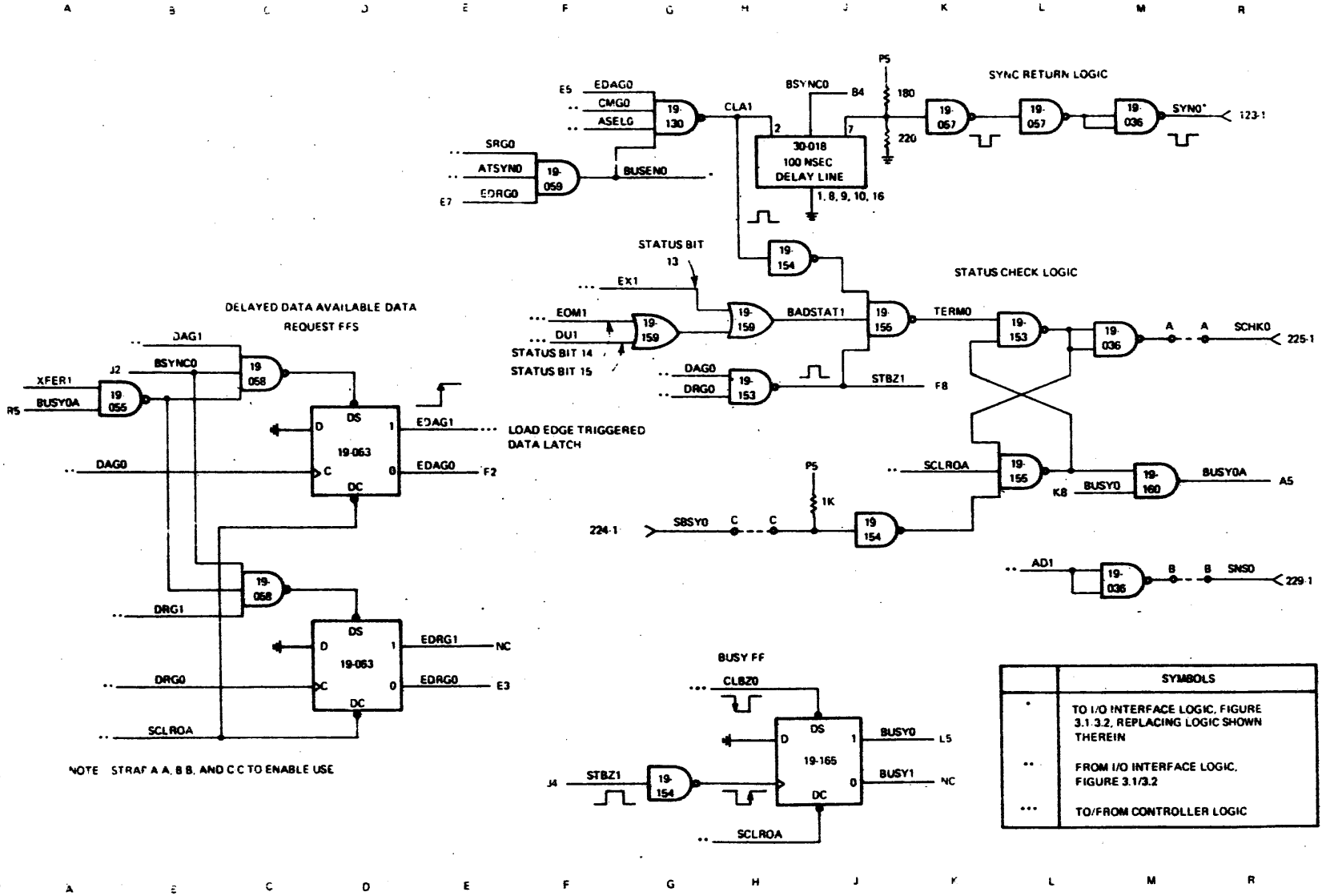


Figure 2-10 Example of Handshake Logic at Device Controller

2.5.2 I/C Switch

All I/O operations through an I/O switch (IOS) incur additional propagation delays which in high throughput applications may become critical. The amount of propagation delay added to each I/O operation may be calculated using the formula:

$$D = K + .004F$$

Where D = Worst-case propagation delay added by IOS in microseconds
F = Length of cable between the IOS-A and IOS-B boards in feet.
K = .416 for SR, DR, or RACK I/O operation (input operations)
.548 for DA, CMD, or ADRS I/O operation (output operations)

When the processor executes an I/O instruction to a device controller through an IOS, the instruction execution time may be approximated using the formula:

$$Ia = In + (Din)(Nin) + (Dout)(Nout)$$

Where Ia = Approximate instruction execution time in microseconds
In = Nominal instruction execution time in microseconds (as listed in the appropriate processor user's manual)
Nin,Nout = Number of input or output I/O operations per instruction (see Table 2-1)
Din,Dout = Worst-case propagation delay added by IOS in microseconds for each input or output I/O operation

The most critical case is usually when an IOS is used to extend a SELCH bus. In this case, throughput may be calculated using the following formula:

$$X = \frac{B}{D + P\,(Sn \,\&\, Sm) + P\,(.008F + (P)(C) + DA}$$

Where X = Transfer rate in Megabytes/second

B = Number of bytes per DA/DR pulse
(B = 1 for byte mode)
(B = 2 for halfword mode)

D = SELCH delays (worst case)
(D = .390 for BSELCH w/high speed protocol)
(D = .730 for BSELCH w/normal protocol)
(D = .450 for 16-bit ESELCH normal protocol only)

P = Protocol
(P = 1 for high speed protocol on BSELCH)
(P = 2 for normal protocol)

Sn = Sync return turn on time in microseconds

Sm = Sync removal time in microseconds
(Sn and Sm are measured at the SELCH PSYNO input)

F = Length of cable between the IOS-A and IOS-B boards in feet

C = Worst-case IOS-A and B logic delay in microseconds = .416

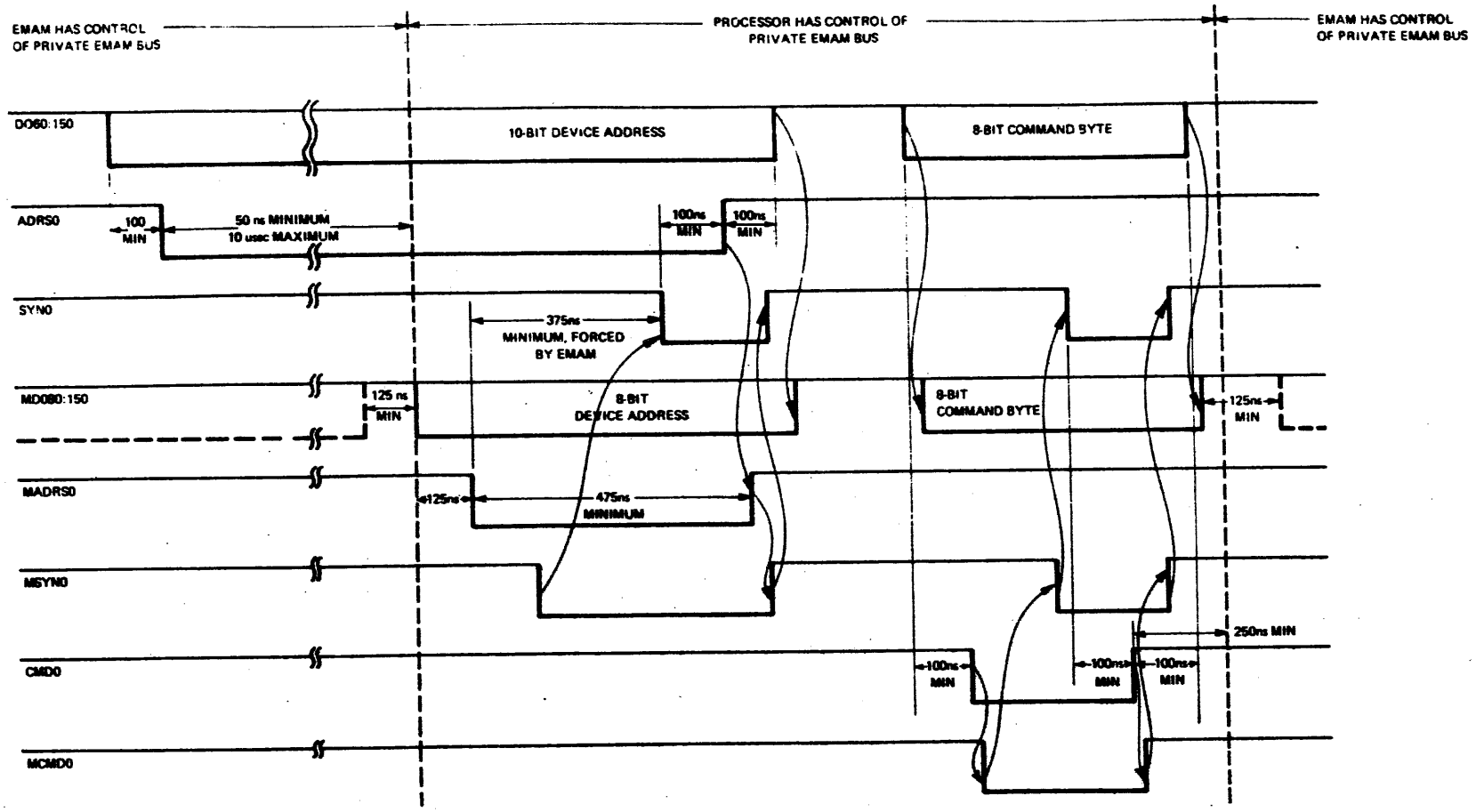
DA = .132 for write to device
0 for read from device

Adding additional ports to the switched bus (i.e., additional IOS-B boards) causes no additional propagation delay. Also, no additional propagation delay occurs when performing I/O operations to device controllers on the multiplexor bus with an IOS-A board present on the multiplexor bus.

2.5.3 EMAM

While the EMAM is in the active mode, control of the EMAM private bus is transparently switched between the EMAM and the processor, on a round-robin priority basis. Figure 2-11 shows the timing for the acquisition of the EMAM bus by the processor to perform one I/O instruction to a controller on the EMAM bus. The processor requests the EMAM private bus by initiating an address I/O operation to the desired device controller on the EMAM's subchannel. The EMAM, as it recognizes a match on its upper two address bits (subchannel number) and a non-match on its lower eight address bits, queues the bus request from the processor. At the conclusion of the present EMAM interrupt service cycle in progress, the EMAM bus is turned over to the processor. The delay in acquisition of the bus can be 50 nanoseconds to 10 microseconds after ADRS0 was activated by the processor. Since the timing relationship of ADRS0 to the device address on the multiplexor bus data lines has been destroyed, the EMAM, in propagating the device address onto the private EMAM bus, reconstructs the timing by delaying MADRS0 (private ADRS0) 125 nanoseconds after the device address is gated onto the EMAM bus data lines. In addition, to ensure that MADRS0 is at least 350 nanoseconds wide, MSYN0 (private SYN0) is prevented from propagating to the multiplexor bus as SYN0 until 375 nanoseconds after MADRS0 goes active.

The EMAM monitors the presence of MSYN0 during the processor's address operation on the private EMAM bus; and if MSYN0 occurs, allows the processor control of the EMAM bus to continue. Should MADRS0 deactivate without MSYN0 active, the EMAM assumes that the processor's request for the EMAM bus has been aborted. This situation can occur if the desired device controller is not on the EMAM private bus, or if the processor attempted an I/O operation to a non-existent device on the EMAM's subchannel. Assuming that MSYN0 is active during the address I/O operation, the processor is allowed to perform one I/O operation to the selected controller on the EMAM bus. At the conclusion of the following I/O operation, control of the EMAM private bus by the processor is terminated. If an interrupt is queued on the EMAM private bus during the processor access, the EMAM performs an interrupt service cycle before honoring the next processor request for the EMAM bus. Note that all I/O instructions listed in Table 2-1 which issue multiple I/O operations after the address I/O operation must not be issued to a device controller on the EMAM bus while the EMAM is active.



NOTE: PROCESSOR EXECUTES AN 'OUTPUT COMMAND' INSTRUCTION TO A DEVICE CONTROLLER ON THE EMAM PRIVATE I/O BUS, PRECEDED AND FOLLOWED BY AN EMAM INTERRUPT SERVICE CYCLE (i.e., EMAM IN ACTIVE MODE).

Figure 2-11 EMAM Bus Acquisition Timing

Figure 2-12 shows the timing for an EMAM interrupt service cycle to a device controller on the EMAM bus. The sequence of I/O operations is somewhat similar to the 32-bit processor automatic interrupt servicing (Table 2-2) except that a command may also be issued during the cycle. As soon as the EMAM bus is available (i.e., the processor completes any I/O access in progress), after MATNO goes active, the EMAM acknowledges the device interrupt by activating MRACK0. The highest-priority (by position) interrupting device controller captures the MRACK0 pulse, gates its address onto the EMAM bus, then activates MSYN0 (private SYNC return). The EMAM then deactivates MRACK0, gates the interrupting device's address onto the EMAM bus, then activates MADRS0 (private address). The interrupting device is now selected, and responds by activating MSYN0. The EMAM then deactivates MADRS0, and 250 nanoseconds later activates MSR0 (private status request). The selected device responds by gating its status byte onto the EMAM bus, and then activating MSYN0. The EMAM then deactivates MSR0.

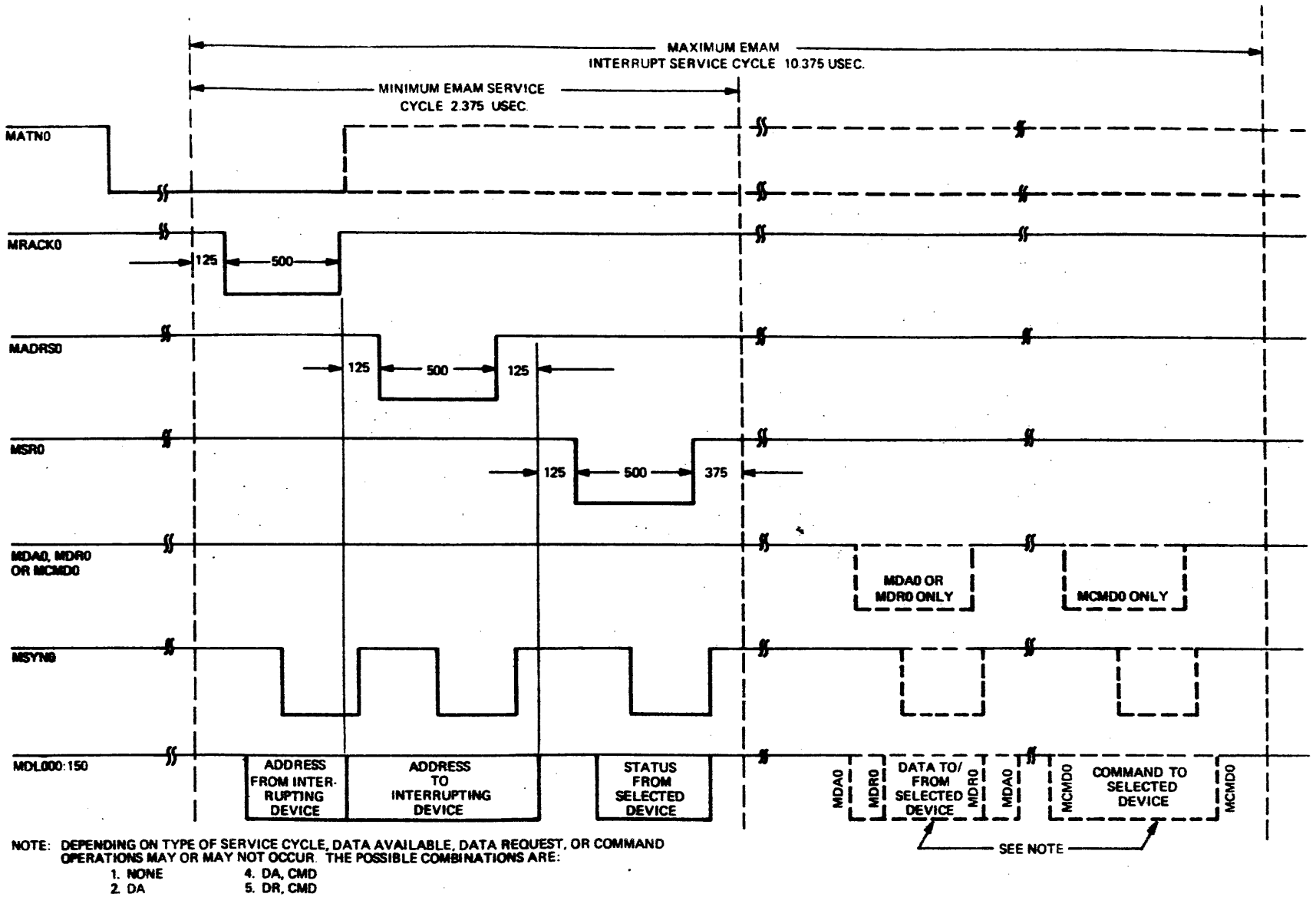
Following the status request I/O operation, one or two additional I/O operations may occur, depending on the type of interrupt service cycle.

With the EMAM inactive (kill mode), all device interrupts on the private EMAM bus are passed to the multiplexor bus, and serviced by the processor. In this mode, the EMAM operates identically to a SCC.

EMAM throughput varies according to type of interrupt service, protocol support, SYNC return delays, and amount of processor I/O to devices on the EMAM's subchannel. Maximum EMAM throughputs are as follows:

MODE	EMAM THROUGHPUT (KBYTES/SECOND)
Halfword	400
Byte, no Protocol	200
Byte, BISYNC Protocol	195
Byte, ZBID Protocol	130

Refer to the EMAM Maintenance Manual, Publication Number 29-609, for additional details.



NOTE: DEPENDING ON TYPE OF SERVICE CYCLE, DATA AVAILABLE, DATA REQUEST, OR COMMAND OPERATIONS MAY OR MAY NOT OCCUR. THE POSSIBLE COMBINATIONS ARE:

1. NONE
2. DA
3. DR
4. DA, CMD
5. DR, CMD

Figure 2-12 EMAM Interrupt Service Cycle Timing

2.5.4 Subchannel Controller

Figure 2-13 shows I/O timing for an address operation when performed with a SCC. As the processor begins an address operation to a device controller on the subchannel bus, it gates the 10-bit address onto the multiplexor bus data lines, then activates ADRS0. At the SCC, all 10-bits are gated onto the private bus until ADRS0 activates; at this time the upper two address bits are forced inactive at the private bus data lines PD060 and PD070. If the upper two address bits match the subchannel controller's channel address bits, PADRS0 (private address) is activated after a 100 nanosecond delay. In response to PADRS0, the selected device controller activates PSYN0, which is propagated to the multiplexor bus by the SCC as SYN0. The processor then deactivates ADRS0. At the SCC, the subchannel address flip-flop is set and SCAD1 (subchannel addressed) is activated, if PSYN0 is active as ADRS0 deactivates. SCAD1 active enables the multiplexor bus control lines DA0, DR0, CMD0, and SR0, allowing control line signals to propagate to the private bus.

Figure 2-14 shows I/O timing for an interrupt acknowledge operation when performed with a SCC. As an interrupt occurs on the private bus, PATN0 (private attention) generates ATN0 at the multiplexor bus. When the processor responds to the interrupt with an acknowledge, RACK0 activates PTACK0 (private acknowledge) onto the private bus, and the subchannel number is gated onto D060 and D070 by the SCC. The highest priority interrupting device controller on the private bus captures the private acknowledge, gates its 8-bit device address onto the private bus data lines PD080:150, then activates PSYN0. At the SCC, PD080:150 are gated onto D080:150 (forming the full 10-bit interrupting device address on D060:150) and PSYN0 activates SYN0.

Subchannel 0 is normally dedicated to the processor multiplexor bus. A SCC is not necessary for device controllers assigned to subchannel 0, except in the case of device controllers not having logic to prevent an address match if either D060 or D070 is active during an address operation. Multiple SCCs may be assigned the same subchannel number for fanout purposes, provided that each device controller under the SCCs has a unique 8-bit device address.

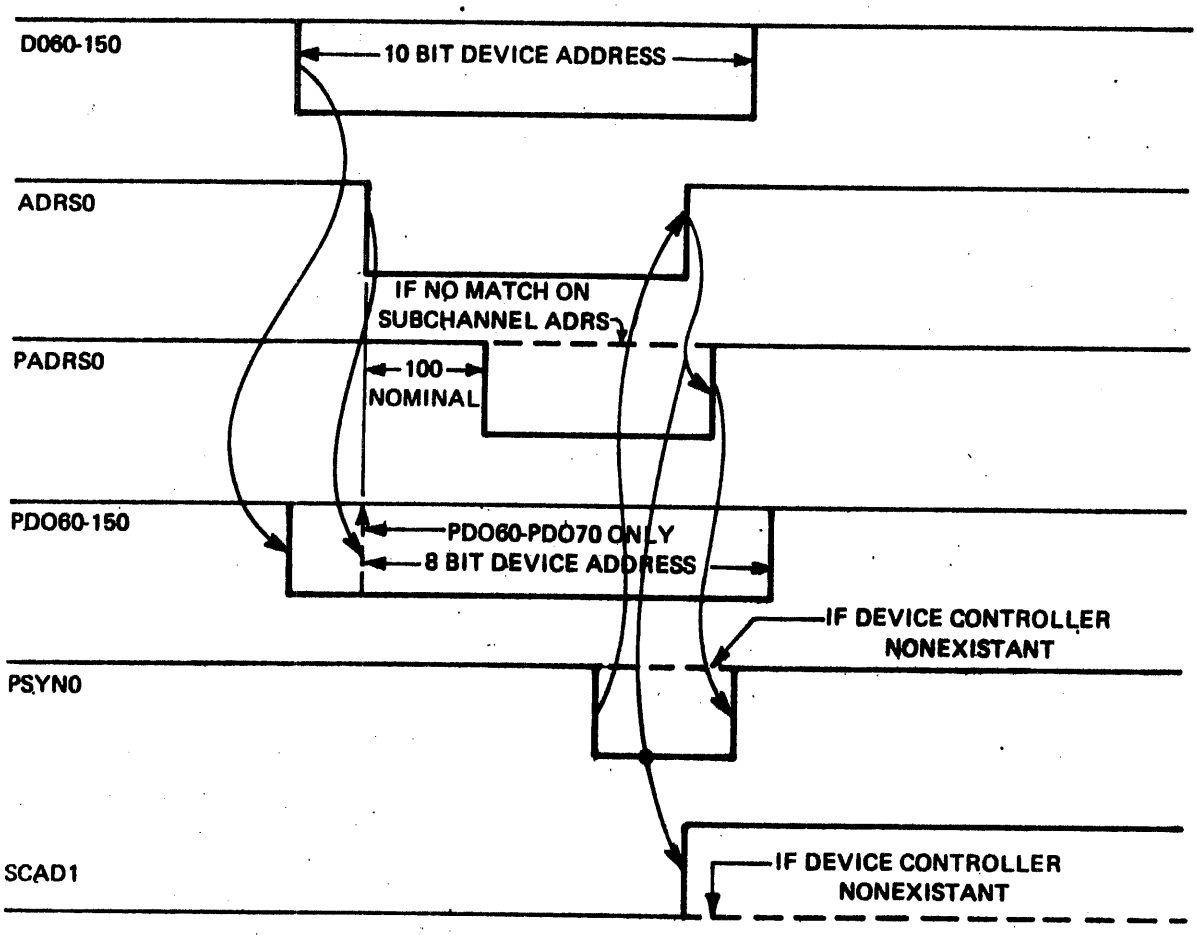


Figure 2-13 I/O Address Operation Timing With SCC

INTERRUPT ACKNOWLEDGE I/O OPERATION

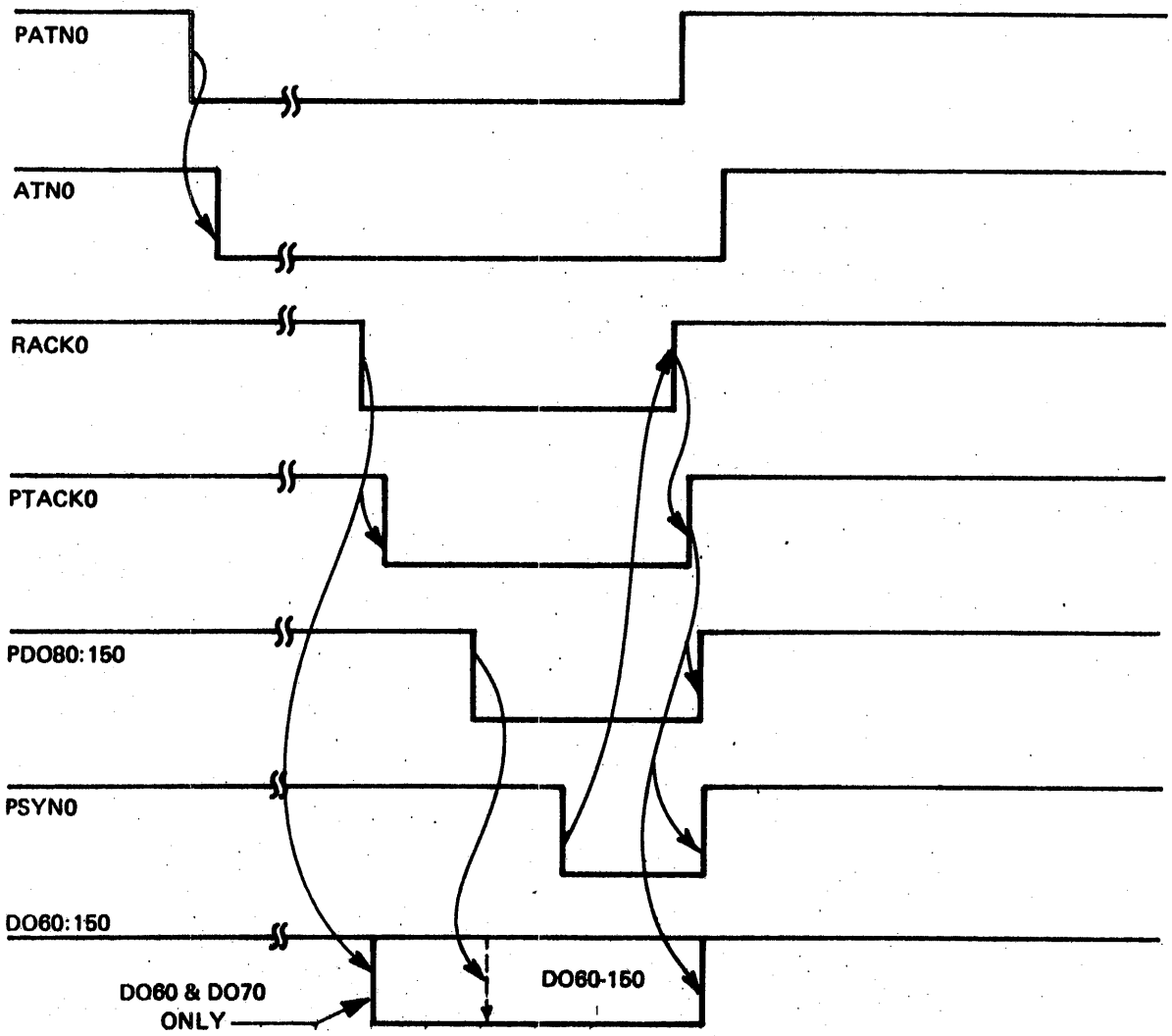


Figure 2-14 I/O Interrupt Acknowledge Timing with SCC

CHAPTER 3
I/O BUS CONTROLLER DESIGN

3.1 STANDARD I/O BUS INTERFACE LOGIC

Figures 3-1 and 3-2 present two examples of generalized I/O bus interfaces suitable for most applications. Each of these interfaces consists of a single device address, 8- or 16-bit I/O interface with interrupt logic. The circuit in Figure 3-1 uses the R1/T1 I/O bus drivers and receivers (19-016/19-036), while the circuit in Figure 3-2 uses the X1 I/O bus transceivers (19-118). Both interfaces function identically.

3.1.1 Address Match and Address Flip-Flop

Referring to Figure 3-1, as the address I/O operation begins, the processor gates the desired device address onto the multiplexor bus, then activates ADRS0. The low-order 8 bits of the multiplexor bus (D080:150) are received at E1-L1 and appear, in true and complement form, at the inputs of the two hexadecimal device address select switches (E3-L3). At the desired device controller, the 8 outputs of the hexadecimal switches are all high, causing the output of the 8-input NAND gate (E5) to go low. This output signal is inverted and ANDed with D060A and D070A to activate ADMCH1 (address match) (D6). Note that D060A and D070A must both be high (inactive) for an address match to occur. This allows configuration of the controller directly on subchannel 0 (multiplexor bus) without a SCC. For device controllers configured under a SCC (or EMAM) on subchannels 1, 2, or 3, the SCC forces D060 and D070 inactive on its private bus as the device address is propagated.

With ADMCH1 high, on the low-to-high transition of ADRS1, the address flip-flop (D7) is set and AD1 (device addressed) (D9) is activated. Simultaneously, all other device controllers on the multiplexor bus have their address flip-flop reset (ADMCH1 low as the ADRS1 low-to-high transition occurs). With AD1 high, if the controller is designed to transfer parallel 16-bit halfwords, HW0 (C9) is activated.

The address match logic shown in Figure 3-2 functions in a similar manner. Two 4-bit comparator ICs are used to compare the 8-bit device address (SW081:151) (F1-J1) from the two hexadecimal device address select switches (E4, H4) with the high-active buffered low-order 8 multiplexor bus data lines D081:151 (G5, J5). The two high-active A=B outputs of the comparators are ANDed with D060(A) and D070(A) to activate ADMCH1. The address flip-flop (D6) is similar to Figure 3.1.

3.1.2 Control Lines and SYNC Return

The control line inputs are shown at A1-B1 of Figures 3-1 and 3-2. The buffered control line input is gated with AD1 causing only the selected device controller to respond to control line pulses. The gated control line signals DAGO (A4), DRGO (B4), CMGO (B4), and SRGO (C4) are available to load or unload data registers, load command registers, or unload status registers within the device controller. With either of these four gated control lines active, or ASELO (address selected) (C6) or ATSYNO (interrupt acknowledge captured) (N6) active, CLA1 (control line active) (A7) goes high. The low-to-high transition removes the direct-clear from the SYNC return flip-flop (B9) and triggers the SYNC return delay one-shot (B7). When the one-shot times out, the low-to-high transition of its 0 output sets the SYNC return flip-flop, activating SYNO (C9). In response to SYNO, the processor deactivates the control line, causing CLA1 to go low. This immediately resets the SYNC return flip-flop, deactivating SYNO.

The initialize signal (SCLRO) input is shown at D1. The buffered initialize signals are available to the device controller logic to reset all controller functions on a power- up or system initialization. The power-fail early warning signal (CL070) (A1) is available, if needed, for lookahead reset of control functions on a system power down.

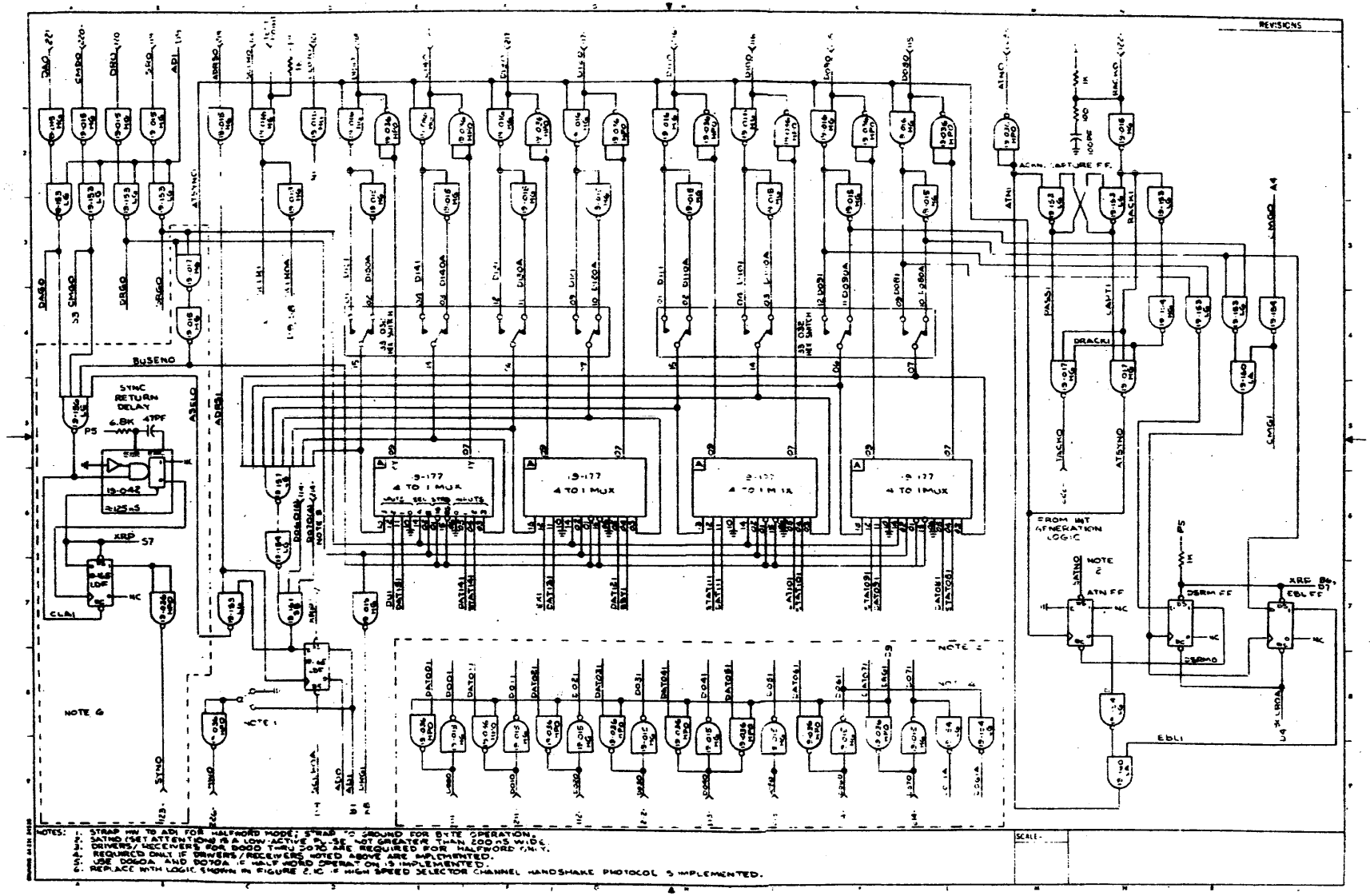


Figure 3-1 I/O Bus Interface Logic

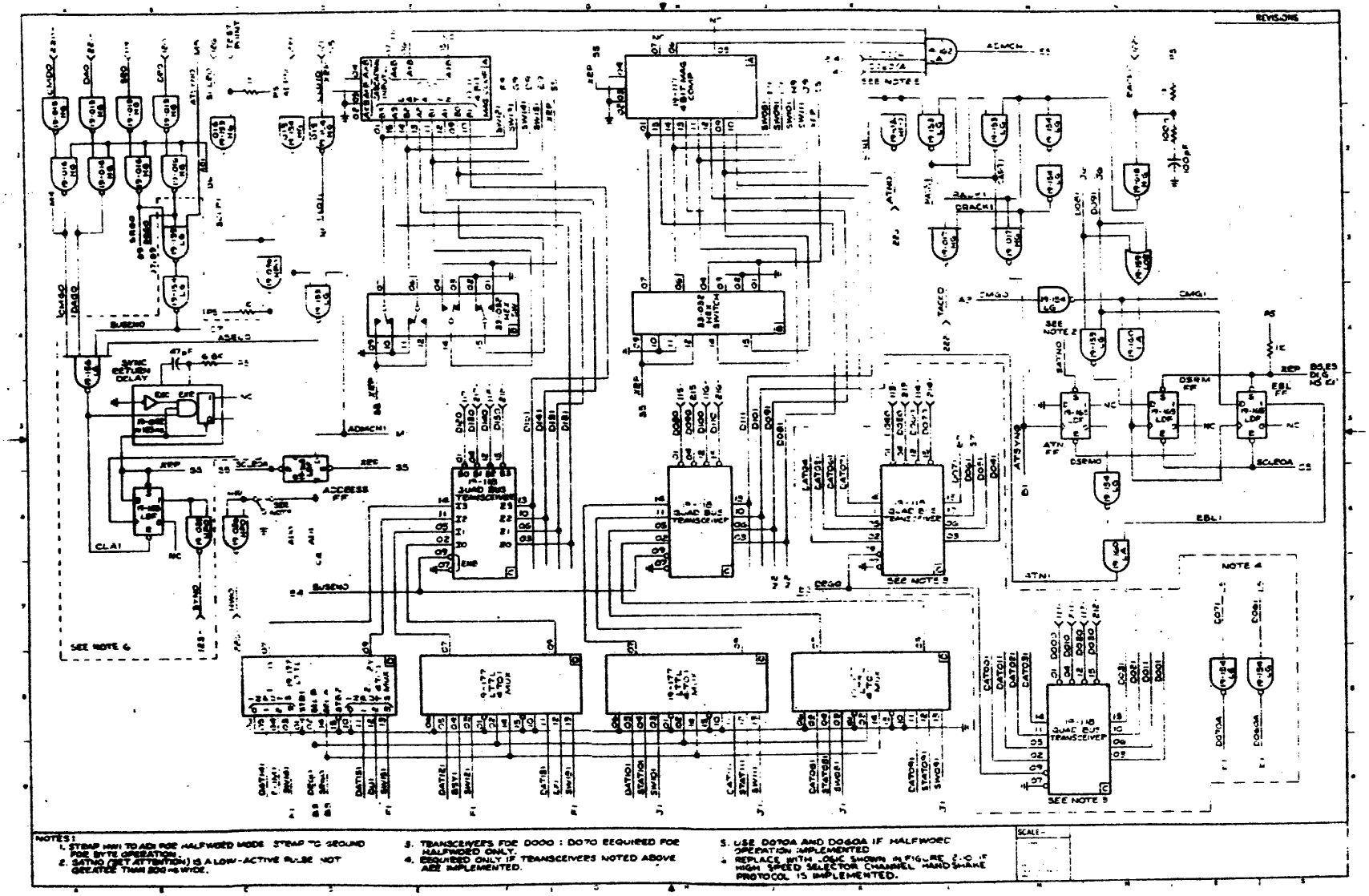


Figure 3-2 I/O Bus Interface Logic

3.1.3 Data/Status/Command

The four 4-to-1 multiplexor ICs shown at P5-L5 of Figure 3-1 select one of three sources for information gated from the controller to the multiplexor bus. With DRG0 active (low) and SRG0 inactive (high), a binary code of '01' on the multiplexor's select input selects the data byte (DAT081-151) to be gated back to the processor. With SRG0 low and DRG0 high, a binary code of '10' on the select inputs selects the status byte (STAT008:111, BSY1, EX1, EOM1, and DU1). With both DRG0 and SRG0 high, a binary code of '11' selects the interrupt device address. The multiplexor ICs are enabled by BUSEN0 (enable gating to multiplexor bus) low. BUSEN0 goes active if either SRG0, DRG0, or ATSYN0 go active. Note that the interrupt address inputs to the multiplexor ICs assume their correct state only when ATSYN0 goes low, forcing all the lower eight buffered data inputs D081:151 active.

For devices designed to transfer parallel 16-bit halfwords of data, the eight driver gates at F8-L8 gate back the upper 8 data bits (DAT001-071) while DRG1 is active.

For data available and command operations, the device controller requires data and command registers, strobed by DAG1/0 or CMG1/0 as appropriate, to latch the contents of the buffered data lines D081:151. For halfword devices, a 16-bit data register is required.

The data/status/command logic of Figure 3-2 is identical to the logic of Figure 3-1 except that the interrupt address is taken from the hexadecimal switch outputs SW081:151.

3.1.4 Interrupt Flip-Flops and Acknowledge Logic

Control of device controller interrupts is developed at the interrupt flip-flops at N7-S7 of Figure 3-1. Command bits 8 and 9 perform the interrupt enable/disable/disarm function as further explained in Section 3.2.2. A command with bits 8, 9 = '11' (high-active data assumed) or an initialize, sets the controller to the disarm state, inhibiting interrupts from the controller. With D081 and D091 high, the D input of the DSRM flip-flop goes low, and on the leading edge of CMG1, the '1' output of the DSRM flip-flop goes low. This holds the attention flip-flop (N7) reset. A command with bits 8, 9 = '10' sets the controller to the disable state, allowing an interrupt to queue but preventing its gating to the processor. In this case, the '1' output of the DSRM flip-flop is high, but D080A low at the leading edge of CMG1 causes the '1' output of the EBL flip-flop (S7) to go low. This low inhibits the output of the attention flip-flop from reaching the multiplexor bus as ATN0. A command with bits 8, 9 = '01' sets the controller to the enable state, allowing interrupts to queue and to be passed to the processor. Here, the '1' output of the EBL flip-flop is high, enabling the output of the attention flip-flop. A command with bits 8, 9 = '00' causes no change to

the DSRM and EBL flip-flops; D080A and D090A inhibit CMG1 from clocking these two flip-flops.

When the device controller detects an interrupt condition, it generates a short low pulse on SATN0 (set attention) (N6). With the controller's interrupts enabled, this sets the attention flip-flop and activates ATN0 (M1). Later, in response to ATN0, the processor activates the Interrupt Acknowledge line. Assuming that this controller is the highest priority device presently interrupting, RACK0 (N1) goes active. This input is inverted to form RACK1 at the input to the acknowledge capture flip-flop (N2). With ATN1 high before RACK1 is activated, PASS1 (pass acknowledge to next device N4) is inactive and CAPT1 (acknowledge captured N4) is active. CAPT1 is ANDed with RACK1 and DRACK1 (delayed acknowledge N5) to create ATSYN0 (interrupt acknowledge captured N6). On the non-interrupting higher-priority controllers, ATN1 low causes PASS1 to be high. As RACK1 is activated CAPT1 deactivates. PASS1 is ANDed with RACK1 and DRACK1 to form RACK0 (N5) which passes the acknowledge signal to the next device controller.

With ATSYN0 active, the interrupting device address is gated onto the multiplexor bus (see Section 3.1.3) and then SYN0 is activated (see Section 3.1.2). In response, the processor latches the address on the data lines and then deactivates RACK0. When RACK0 deactivates, ATSYN0 is deactivated at the interrupting controller. The low-to-high transition of ATSYN0 resets the attention flip-flop (N7).

The interrupt logic of Figure 3-2 is identical to that of Figure 3-1.

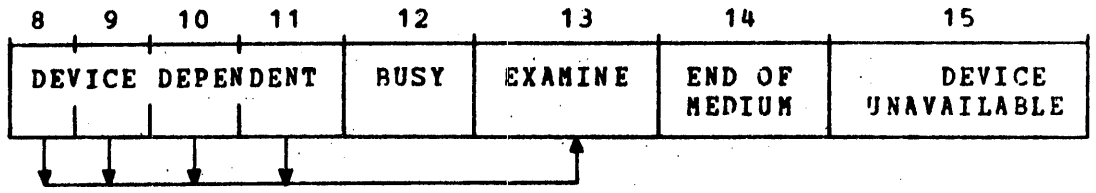
3.2 DESIGN OF I/O INTERFACE PROGRAMMING CHARACTERISTICS

3.2.1 Device Controller Status Byte

The recommended format for the status byte of an I/O device controller is shown in Table 3-1. There may be some exceptions to the status format, depending on the function of the I/O device controller; however, any exceptions may preclude use of the device controller with a SELCH, EMAM, or the processor auto driver channel.

TABLE 3-1 RECOMMENDED STATUS BYTE FORMAT

Multiplexor Bus Bits:



BIT NUMBER	MEANING
8,9,10,11	DEVICE DEPENDENT - These status bits may be used, if required, to report auxiliary conditions within the device controller or I/O device, of interest to the processor. In many cases, (but not all) the assigned condition reported by this status bit indicates an error condition precluding further correct data transfer between the device controller and the processor. In this case, the EXAMINE status bit must also set. These status bits must be implemented such that each bit, when reset, indicates 'acceptable' device status, and when set, indicates 'unacceptable' device status. Examples of status bit assignments are: Overflow, Parity Error.
12	BUSY - This status bit indicates whether the device controller is ready to transfer data to/from the processor. When set, the device controller is not ready for a data transfer. When reset, the device controller is ready for a data transfer.

TABLE 3-1 RECOMMENDED STATUS BYTE FORMAT (Continued)

BIT NUMBER	MEANING
13	EXAMINE - This status bit, when set, indicates that an error condition or other significant auxiliary condition has been detected by the device controller. One or more of status bits 8 through 11 must be set simultaneously with examine. In no case may a device controller present a status byte of X'04' (EXAMINE only) to the processor, as this status condition is reserved to indicate an I/O timeout (false SYNC) at the processor.
14	END OF MEDIUM - This status bit typically indicates, when set, that a media fault has occurred. For example, this bit could indicate 'out of paper' for a line printer, 'end of tape' for a mag tape drive, or 'loss of carrier' (or CL2S) for a data communications interface. This status bit must be implemented such that the reset condition indicates 'acceptable' status, and the set condition indicates 'unacceptable' status.
15	DEVICE UNAVAILABLE - This status bit, when set, typically indicates that the peripheral device has lost AC power, has been manually taken 'off-line', or other major peripheral-oriented fault. This status bit must be implemented such that the reset condition indicates 'acceptable' status, and the set condition indicates 'unacceptable' status.

NOTE

Any status bit not implemented should be forced to the '0' state by the device controller. Programming documentation for the device controller must state that the state of the unimplemented status bit is undefined (may be either '1' or '0'). This allows enhancement of the device controller status byte at a future date.

When the processor executes a Sense Status instruction, the status byte of the selected device controller is copied into the low byte of a processor general register, or a byte memory location. In addition, the low 4-bits of the device status byte are copied into the condition code (CC) portion of the processor program status word (PSW); thus, the next instruction, if a branch instruction, may cause a branch in instruction execution based on the state of any one or more of the low 4 device status bits.

When the processor executes a Read Block or Write Block instruction, the lower 4-bits of the device status byte are tested before the data transfer is performed. If any of status bits 13, 14, or 15 (Examine, End of Medium, or Device Unavailable) become set, the block transfer is terminated. If status bit 12 (BUSY) is reset, along with status bits 13, 14, and 15 reset, one byte of data is transferred. If status bit 12 is set, with status bits 13, 14, and 15 reset, the device status byte is repeatedly examined until either status bit 12 resets or one or more of status bits 13, 14, or 15 become set.

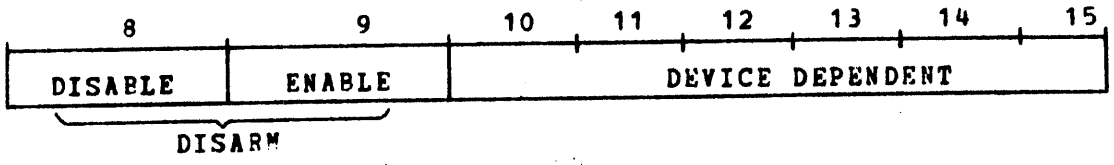
For data transfers using a SELCH in normal protocol mode, the device status byte is used to control the data transfer in an identical fashion to the processor Read Block and Write Block instructions. For data transfers using the processor Auto-Driver Channel or the EMAM, in response to an interrupt from the device controller, the status byte is obtained and all 8 status bits are ANDed with an 8-bit mask byte. If the result of the AND operation is true for any bit, unacceptable status is indicated and the buffer transfer is terminated. The data transfer may continue using the alternate data buffer if, on the next device interrupt, the status is acceptable. If the result of the AND operation is not true for any bit, acceptable status is indicated and a data transfer occurs.

3.2.2 Device Controller Command Byte

The recommended format for the command byte of an I/O device controller is shown in Table 3-2 The format shown must be adhered to for all devices operating in an operating system environment.

TABLE 3-2 RECOMMENDED COMMAND BYTE FORMAT

Multiplexor Bus Bits:



BIT NUMBER	MEANING
8,9	Interrupt Enable/Disable/Disarm control - specific combinations of these two command bits perform the following functions:

BITS 8	BITS 9	FUNCTION
0	1	Enable device interrupts - allow interrupts to queue and pass them to the processor.
1	0	Disable device interrupts - allow interrupts to queue but do not pass them to the processor.
1	1	Disarm device interrupts - clear any queued interrupts and do not allow further interrupts to queue.
0	0	No Change - do not modify the previous enable/disable/disarm interrupt control state.

System initialization must set the device controller to the disarm state.

TABLE 3-2 RECOMMENDED COMMAND BYTE FORMAT (Continued)

BIT NUMBER	MEANING
10,11,12,13,14,15	DEVICE DEPENDENT - These command bits may be used, if required, to control various device functions. They should be implemented such that the set state of the command bit turns on the desired device function. System Initialization must deactivate all command bit device functions.

NOTE

Any command bit not implemented must be defined as 'undefined-must be zero' in the programming documentation of the device controller. This allows enhancement of the device controller command byte at a future date.

I/O device controllers requiring additional control bits over and above those available in the recommended command byte format may have multiple command bytes implemented. In this case, one or more of the undefined bits in the recommended command byte format must be used as steering bits. Additional command bytes have no format restriction. Table 3-3 shows an example of multiple command byte format.

TABLE 3-3 MULTIPLE COMMAND BYTE FORMAT EXAMPLE

	8 9	10 11 12 13	14	15
Recommended Command Byte	INTERRUPT ENABLE/ DISABLE/DISARM	DEVICE DEPENDENT	0	0
Additional Command Bytes ↓	DEVICE DEPENDENT		0	1
	DEVICE DEPENDENT		1	0
	DEVICE DEPENDENT		1	1
			↑ STEERING	↑ BITS

NOTE

The all-zero state of the steering bit(s) must be reserved for the command byte containing the interrupt ENABLE/ DISABLE/DISARM bits.

3.2.3 Device Controller Interrupt Conditions

When enabled, the device controller may raise an interrupt at any time to indicate that an extraordinary condition has occurred of interest to the processor. Typically, interrupts occur resulting from status bit transitions. For example, using the recommended status byte format shown in Table 3-3, any one of the following status transitions could be defined to cause an interrupt:

STATUS BIT	TRANSITION	MEANING
BUSY	1 to 0	Device controller is now ready to transfer a byte/halfword of data.
EXAMINE	0 to 1	An error condition has been detected by the device controller precluding further correct data transfer.
END OF MEDIUM	0 to 1	The data medium has been exhausted; no more data can be transferred.
DEVICE UNAVAILABLE	0 to 1	The peripheral device or controller has lost power or has been manually taken 'off-line'.
DEVICE UNAVAILABLE	1 to 0	Power has been restored at the peripheral device or the peripheral device has been manually placed in the 'on-line' mode.

The specific interrupt conditions are a function of the status bit implementation and requirements of the peripheral device. Device controller interrupt conditions should be designed such that any error condition detected at the peripheral or device controller, which precludes completion of an operation in progress, be reported to the processor via an interrupt; otherwise, when used in an interrupt-driven I/O environment, the device controller appears to 'hang' (cease interrupting) if the error condition occurs.

When designing device controllers intended for use on the processor Auto-Driver Channel or the EMAM, consideration should be made of the device startup procedure to insure that data transfer interrupts will begin. For example, if a device controller already has the BUSY status bit reset (indicating that it is ready to transfer data) as it is enabled by a Processor command, no interrupt can occur from the BUSY 1 to 0 transition.

As a result, the device controller hangs and the interrupt-driven data transfer does not commence.

3.2.4 Device Controller Address Assignment

The device controller preferred device address must be assigned to avoid conflict with all other device controllers within the system. Figure 3-3 shows the preferred device addresses for all presently-supported Perkin-Elmer device controllers. The device controller must be designed with wirewrap stakes or hexadecimal switches allowing address assignment anywhere within the range of X'01' - X'FF'. In cases where multiple device controllers are designed onto one printed circuit board, it is acceptable to design one set of wirewrap stakes or hexadecimal switches to assign a contiguous block of device addresses for that controller, instead of stakes or switches on a per-address basis. In this case, the block of device addresses must be contiguous (example: X'A0', X'A1', X'A2', X'A3', etc.) The device controller address match logic must match across all 10 address bits; in the case of device controllers not intended for use on subchannels 1, 2, and 3 of the multiplexor bus without a SCC or EMAM, it is acceptable to match only on zeroes for address bits 6 and 7. Any device controller assigned a device address on subchannels 1, 2, and 3 of the multiplexor bus without a SCC or EMAM present must return a 10-bit address when responding to an Interrupt Acknowledge.

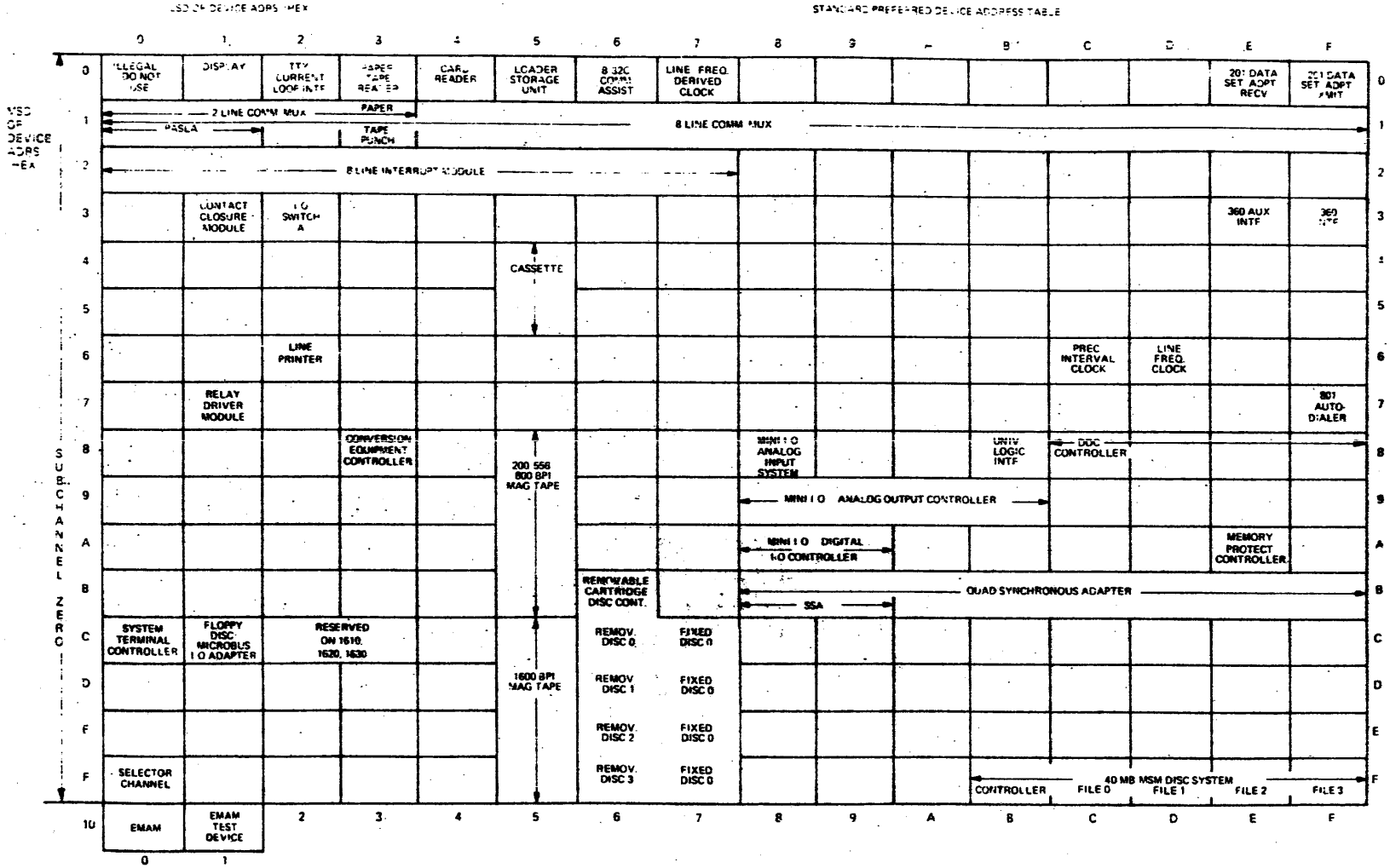


Figure 3-3 Preferred Device Controller Address

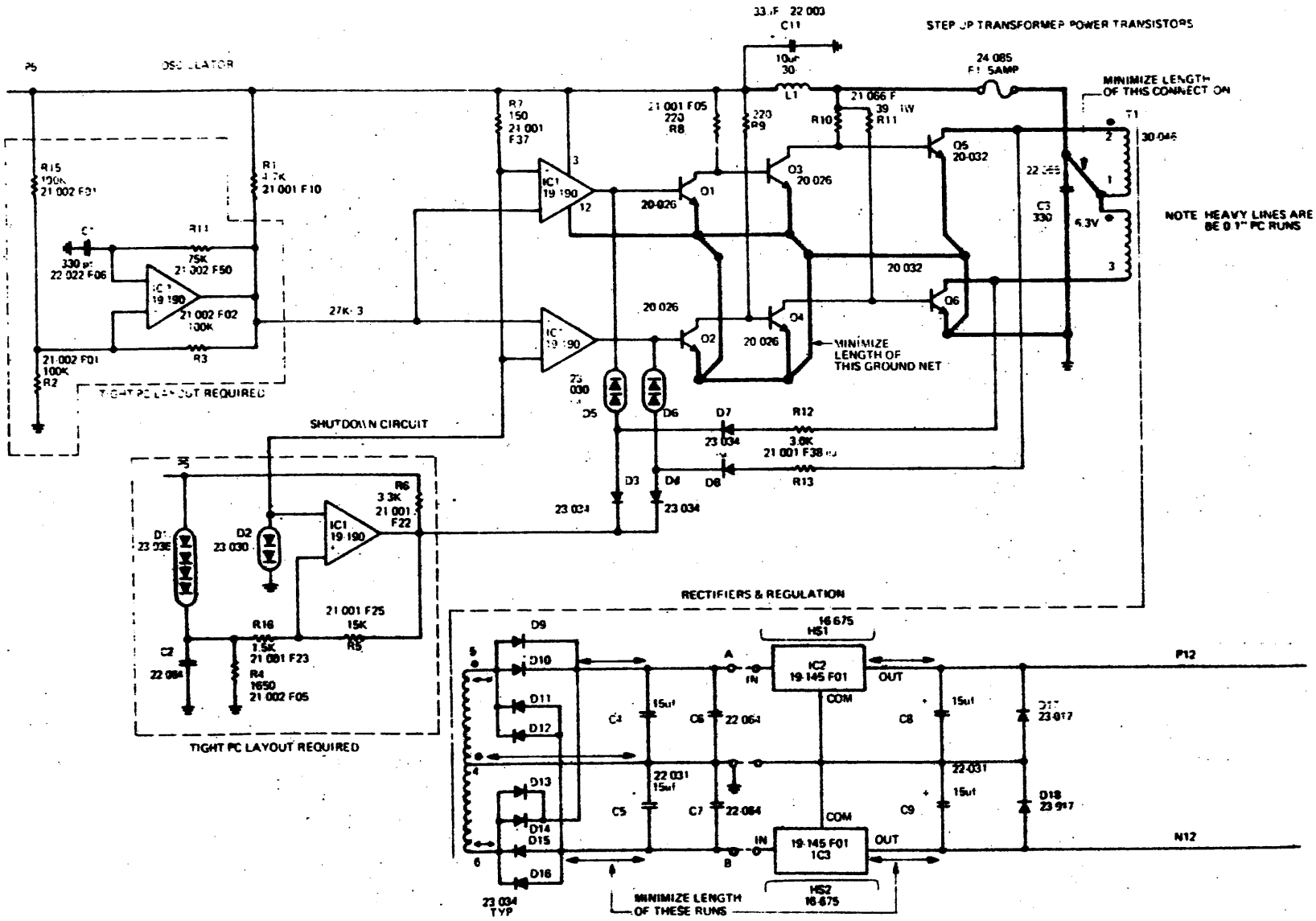


Figure 3-4 General Purpose DC-DC Converter

3.3 DC-DC CONVERTER

All I/O interface designs must use only the +5 volt ± 5% logic power available from the processor or expansion chassis power supply. Other voltages from the system power supply should not be used in the design of the interface, as no other voltages are guaranteed to be available in all systems configurations. If voltages other than +5 volts are required in the design of the interface, an on-board DC-to-DC converter must be used. Figure 3-4 shows a general-purpose DC-to-DC converter capable of developing low current voltage levels suitable for operational amplifiers, RS-232 or Teletypewriter (TTY) interface circuitry, etc. from the +5 volt logic power supply.

3.3.1 DC-DC Converter Description

The circuit is comprised of a 27 Khz. free-running oscillator (1/4 IC1, C1, R1-R3, R14-R15) whose complementary outputs drive a transistor array (Q1-4) which provides current gain to drive the primary of step-up transformer T1, whose secondary outputs are rectified by D9-D16 and filtered by C4-C7. IC2 and IC3 provide output regulation if required. The circuit includes a shutdown circuit (1/4 IC1, C2, R4-R6, R16, D1-D2) which turns off Q5 and Q6 whenever the 5 volt DC supply is less than 4.0 volts. A feedback path via R12-D7 and R13-D8 ensures that Q5 and Q6 are never turned on simultaneously.

3.3.2 DC-DC Converter Specifications

Input:	+5 volts DC ± 5% at 100 to 1350 at rated output.
Output:	The DC-DC converter provides + and - 12.0 volts DC at no load (with +5.0 volt input). line and load regulation is 11.4 minimum to 12.6 maximum. Maximum output is 120 ma DC from both outputs simultaneously with a peak output of 350 ma.
Output Ripple:	The DC-DC converter provides 100 millivolts peak-to-peak (p/p) maximum at full load.
Operating Frequency:	27 Khz nominal.

3.4 ADDITIONAL DESIGN RULES

In addition to the design criteria presented in Chapters 2 and 3, the following rules must be adhered to:

1. All new designs for halfword-oriented device controllers must use 16-bit wide data paths to/from the multiplexor bus.

2. In general, all unused inputs of logic packages should be terminated to +5 volts through a pull-up resistor.

3. Avoid the use of capacitors for delaying the edge of a logic signal. Instead, use a one-shot (19-031 or 19-042) or a delay line (30-018, 30-019, etc.) to generate the required delay.

4. Avoid the use of RC networks for differentiation of logic signal edges. Instead, use a one-shot (19-031 or 19-042) for differentiation.

5. When one-shots such as the 19-031 or 19-042 are used, the timing components (resistor and capacitor) must be located no farther than one inch from the one-shot IC. The +5 volt connection to the timing resistor must be made no farther than one inch from the Vcc pin of the one-shot IC.

6. High frequency decoupling capacitors should be located adjacent to the ICs as required. The number of decouplers required is a function of the logic family being used - TTL, STTL, LSTTL, etc. As a minimum, there should be one decoupler for each two ICs. Provide extra decoupling and/or isolated +5 volts (P5) and GND connections for high-current driver ICs. The 0.1 uF tubular ceramic capacitor (22-033) is recommended for decoupling purposes. In addition, each controller should have low-frequency mass decoupling capacitors distributed along the board side edges. Eight each 15 uF electrolytic capacitors (22-002) are recommended.

7. It is good design practice, in cases where address, command, or data is loaded from the multiplexor bus into an edge-triggered latch, to perform the load operation on the leading edge of the appropriate control line signal (i.e., on the high-to-low transition of ADRS0, CMG0, or DAG0).

8. The maximum P5 current available for a device controller is limited by the back panel connector to 13 amperes for a 15-inch controller and 7 amperes for a 7-inch controller.

9. All applications where a counter is clocked by a gated I/O control line should be designed to reject noise on the control line while it is active.

CHAPTER 4
I/O INTERFACE PHYSICAL PACKAGING

4.1 GENERAL DESCRIPTION

All I/O device controllers are implemented on 15-inch by 15-inch printed circuit boards or 7-inch by 15-inch printed circuit half-boards, as required. Hereafter, these boards are referred to as 15-inch boards or 7-inch half-boards, respectively. The size of the printed circuit board used for a device controller depends upon the amount of logic circuitry in the design.

4.2 15-INCH BOARDS

Typically, a 15-inch double-sided printed circuit board may hold a maximum of 180 14- and 16-pin integrated circuits; a 15-inch multiwire board may hold a maximum of 221 14- and 16-pin integrated circuits. Multiple 15 inch boards may be used to implement device controllers requiring more integrated circuits than will conveniently fit on one 15-inch board. In this case, front-edge cables must be used for all board-to-board interconnect signals.

It is recommended that all 15-inch device controllers pick up the I/O signals from the back panel via connector 1.

NOTE

Do not use some I/O signals from connector 0 and some from connector 1 for convenience of layout. Always use one connector only.

4.3 7-INCH HALFBOARDS

Typically, a 7-inch double-sided printed circuit board may hold a maximum of 70 14- and 16-pin integrated circuits.

Seven inch halfboards are mounted in pairs in the I/O chassis slot using the 16-398 half-board adapter kit. This kit contains hardware to install either two active 7-inch device controllers, or one active 7-inch controller and a 7-inch blank card (included in kit) into the I/O chassis slot.

4.4 I/O SYSTEM INTERFACE MODULES

Perkin-Elmer offers two general purpose interface modules which may be used by the I/O Controller designer to simplify the design effort. These modules on a 7-inch or 15-inch double-sided copper printed circuit board and contain the standard logic connections to the I/C system.

4.4.1 General Purpose Interface Board (15-Inch)

This product (M48-002) is a 15-inch PC which is used to house custom design interface circuits. This board contains mounting positions for up to 117 ICs of the 14- or 16-pin dual in-line variety and 416 axial lead components. IC sockets are not provided. Designers may elect to solder ICs or discrete components on the board, or use 300 mil mounting IC sockets. Connections to components and connectors are made via 25-mil square wire-wrap stakes. Associated manual is 29-271 GENERAL PURPOSE WIRE WRAP BOARD MANUAL.

4.4.2 Universal Interface Module (7-inch)

The Universal Interface Module (UIM) contains the complete interface to the I/O system. Also included is buffering to permit driving external logic which may be located up to ten feet away.

The UIM requires a 16-398 halfboard mounting kit. Included is a 25-pin cinch (25S) and a mating cinch (25P) connector for user's customer cable fabrication. The associated manual is 29-273 UNIVERSAL INTERFACE MODULE INSTRUCTION MANUAL.

4.5 I/O BACK PANEL LAYOUT

Figure 4-1 shows the back panel signal layout for a typical slot within the I/O chassis. Note that the I/O signals are duplicated on both the connector 0 and 1 sides of the slot. Depending on the system configuration, these signals may be identical, or they may be independent (i.e., multiplexor bus on one side and private I/O bus on the other).

NOTE

All unlabeled back panel pins shown in Figure 4-1 are reserved for other system buses and control signals. The I/O interface must not connect to any other signals other than the standard signals shown.

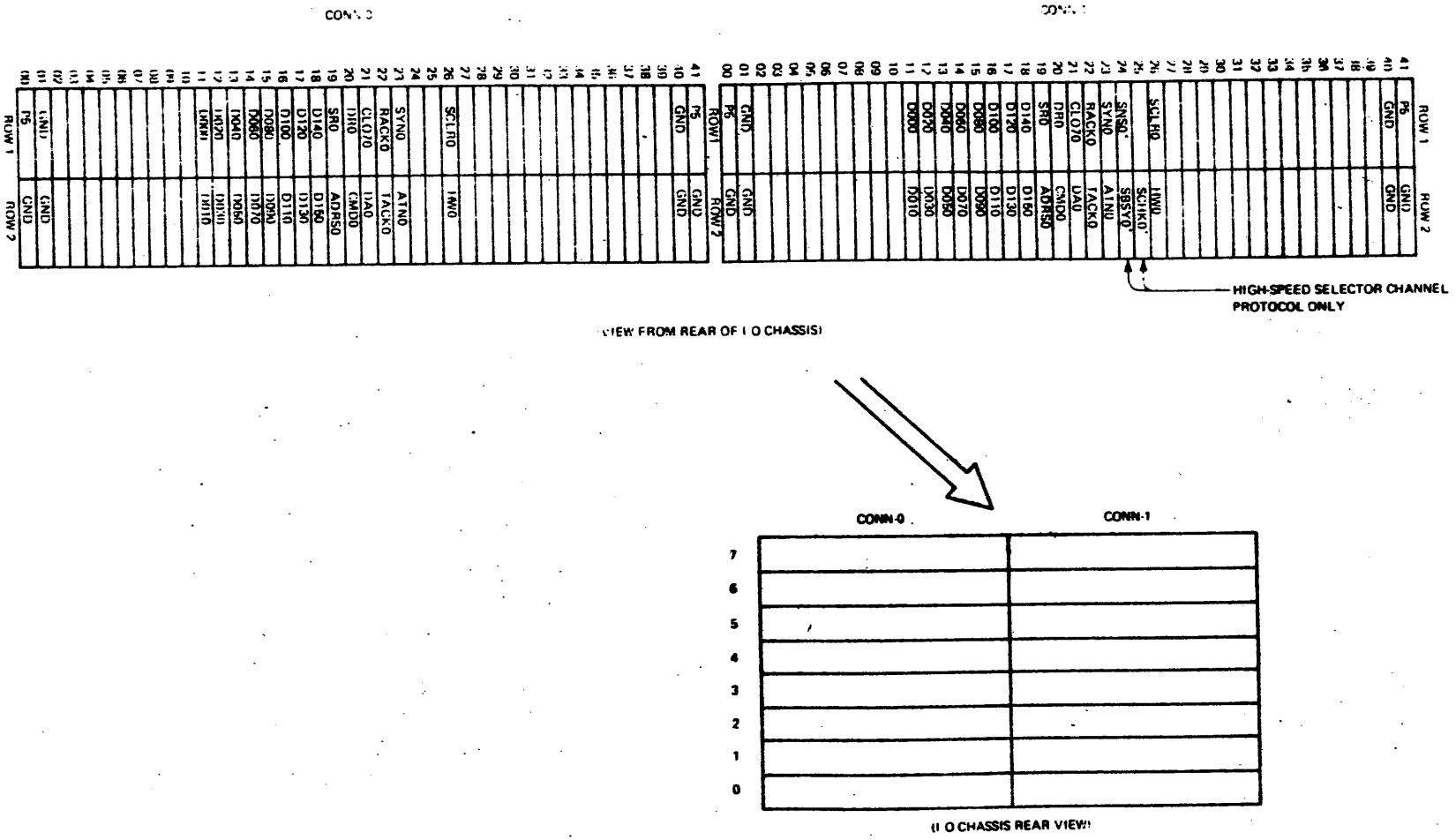
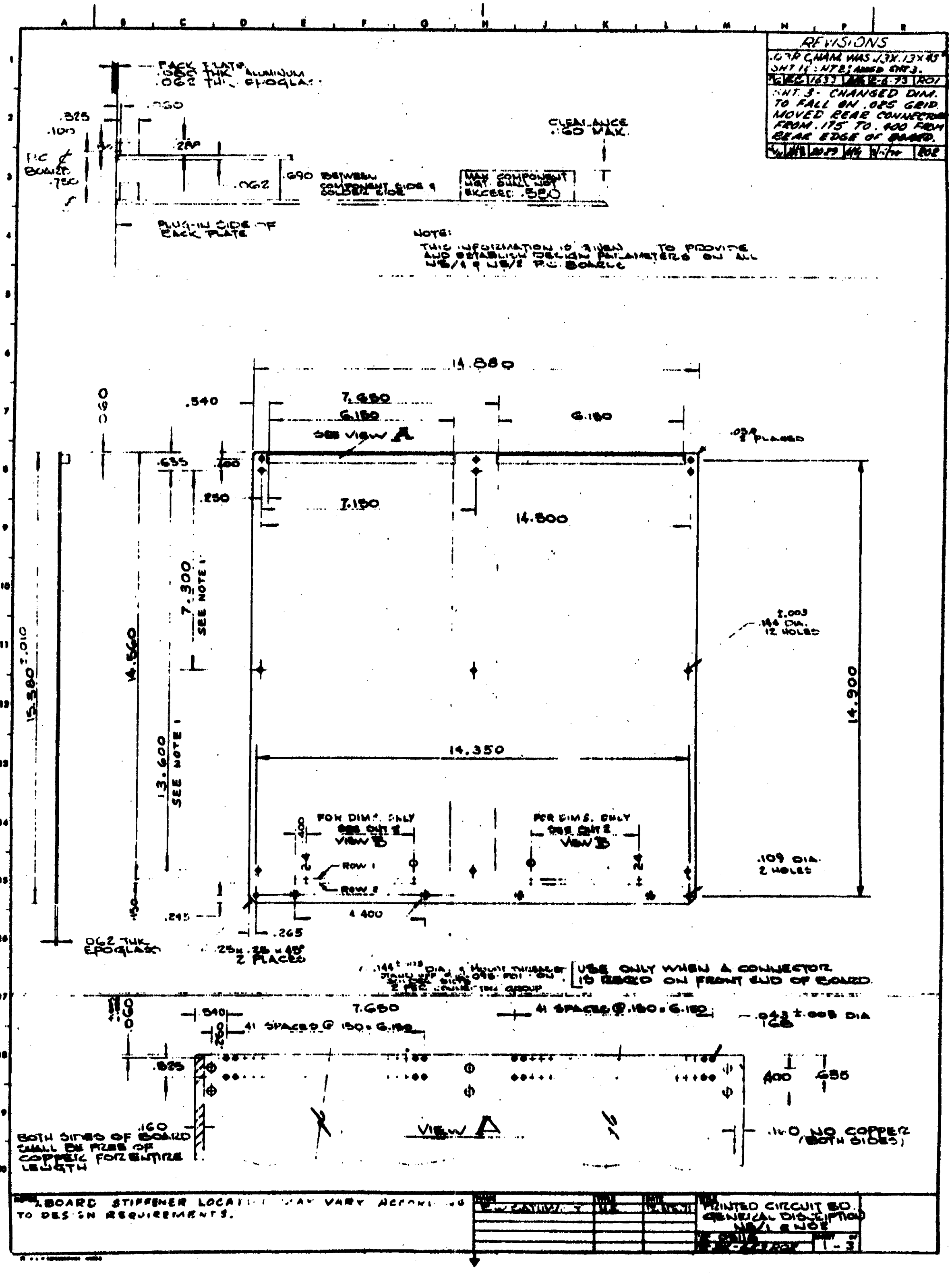
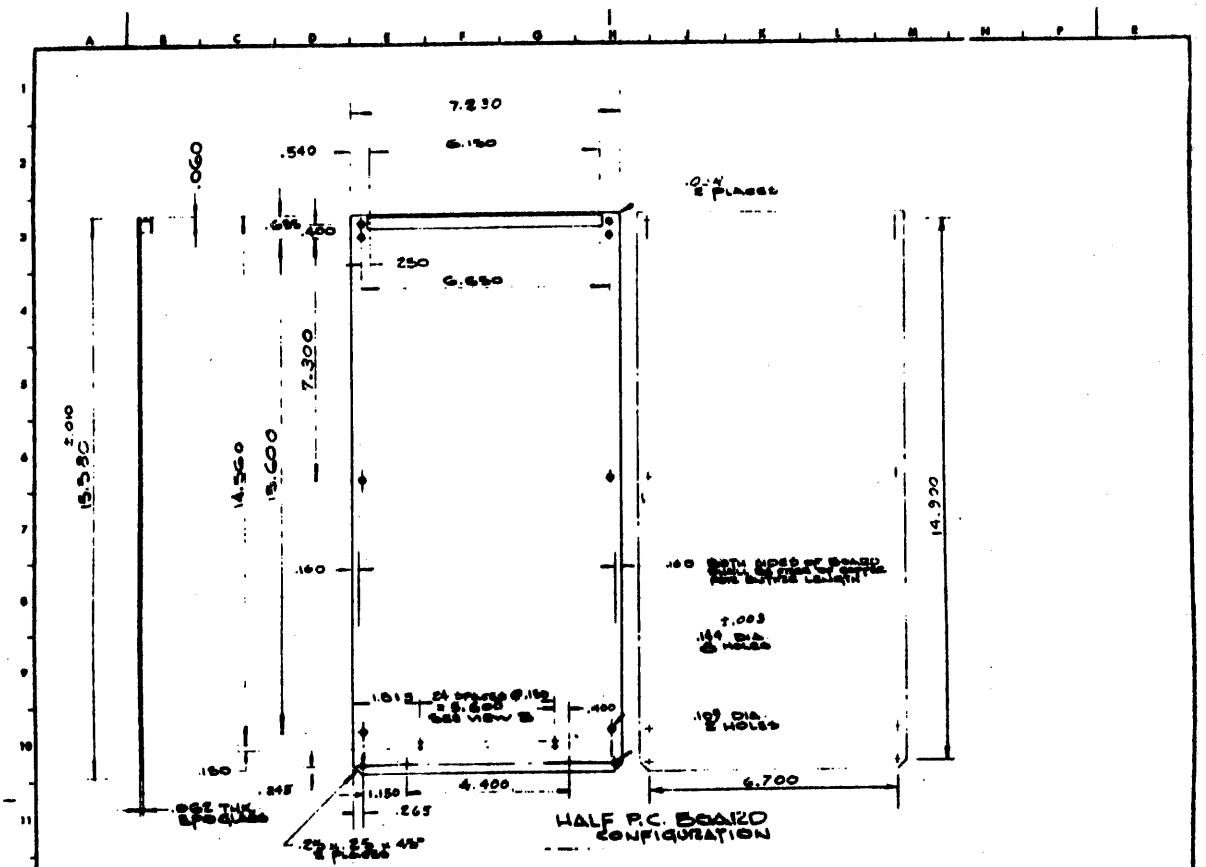
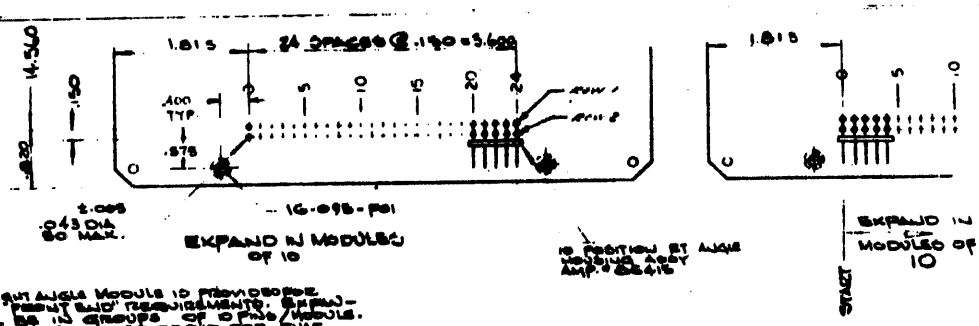


Figure 4-1 Typical I/O Slot Back Panel Signal Layout


REVISIONS
.07R CHAM WAS .13X.13X45°
SHT 1: NT2; ADDED SHT 3.
SHT. 3 - CHANGED DIM. TO FALL ON .025 GRID. MOVED REAR CONNECTOR FROM .175 TO .400 FROM REAR EDGE OF BOARD.
BACK PLATE .062 THK. ALUMINUM .062 THK. EPOGLAS
CLEARANCE .160 MAX.
.690 BETWEEN COMPONENT SIDE & SOLDER SIDE
MAX COMPONENT HGT. SHALL NOT EXCEED .550
PLUG-IN SIDE OF BACK PLATE
NOTE:
THIS INFORMATION IS GIVEN TO PROVIDE AND ESTABLISH DESIGN PARAMETERS ON ALL NS/1 & NS/2 P.C. BOARDS
14.880
7.650
6.150
6.150
SEE VIEW A
7.150
14.800
7.300 SEE NOTE 1
13.600 SEE NOTE 1
14.560
15.380 ±.010
14.350
14.900
.109 DIA. 2 HOLES
FOR DIMS. ONLY SEE SHT 2 VIEW B
ROW 1
ROW 2
4.400
.062 THK. EPOGLAS
.25 x .25 x 45° 2 PLACES
USE ONLY WHEN A CONNECTOR IS REQ'D ON FRONT END OF BOARD.
41 SPACES @ .150 = 6.150
.043 ±.003 DIA
VIEW A
BOTH SIDES OF BOARD SHALL BE FREE OF COPPER FOR ENTIRE LENGTH
.160 NO COPPER (BOTH SIDES)
1. BOARD STIFFENER LOCATION MAY VARY ACCORDING TO DESIGN REQUIREMENTS.
PRINTED CIRCUIT BD. GENERAL DESCRIPTION NS/1 & NS/2
1 - 3



NOTE:
1. WHEN HALF-BOARD CONFIGURATIONS ARE REQ'D AND ONLY 1 SIDE IS NEEDED, A BLANK MUST BE PROVIDED FOR THE UNUSED SIDE. THE BLANK BOARD WILL TIE TOGETHER, STIFFENING & HEADER GUIDANCE SYSTEM.
2. ON ALL HALF BOARD ASSY DRAWINGS PROVIDE ONLY 1 PLASTIC SHELL SYMBOL FOR HALF BOARD IN THE APPROPRIATE CORNER.

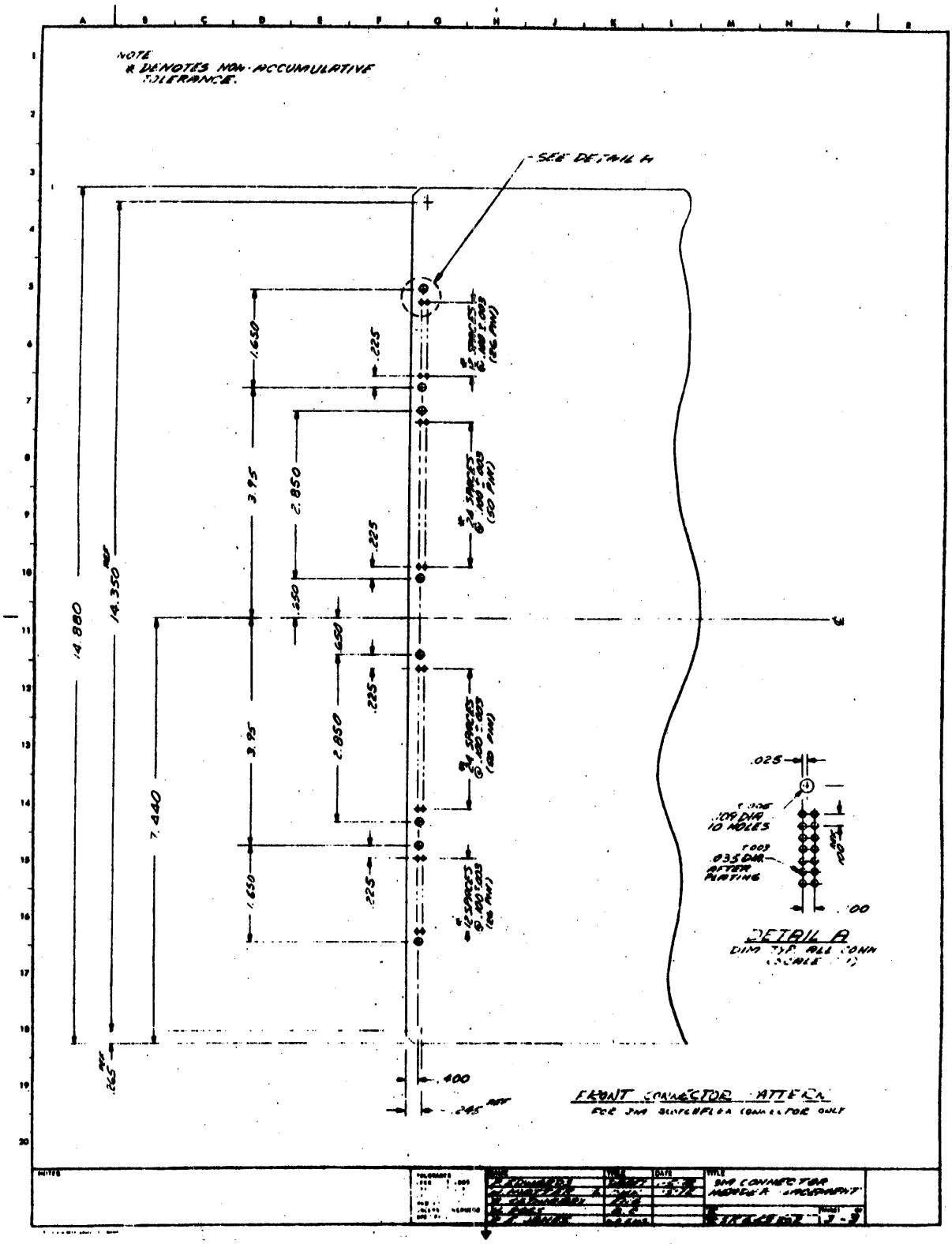


NOTE:
1. THE 10 PIN RIGHT ANGLE MODULE IS PROVIDED FOR P.C. BOARD "EDGE" REQUIREMENTS. SYSTEM DESIGN SHALL BE IN GROUPS OF 10 PINS/MODULE. CALL ONLY THOSE HOLES REQ'D FOR THAT SPECIFIC APPLICATION.

EXAMPLE:
10 PINS REQ'D - 1 MODULE - 10 HOLES
20 PINS REQ'D - 2 MODULES - 20 HOLES
24 PINS REQ'D - 3 MODULES - 30 HOLES

THE STARTING LOCATIONS AND EXPANSION CRITERIA AS SHOWN ABOVE APPLIES TO BOTH FULL & HALF BOARDS

VIEW B


NOTE
R DENOTES NON-ACCUMULATIVE TOLERANCE.
SEE DETAIL A
14.880
14.350 REF
7.440
1.650
3.95
2.850
.450
.650
.225
.265 REF
.400
.245 REF
.025
.109 DIA 10 HOLES
.035 DIA AFTER PLATING
.100
DETAIL A
FRONT CONNECTOR ATTACH
FOR SMA SURFACE FLEX CONNECTOR ONLY

INDEX

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

_____

_____

FOLD FOLD

Check the appropriate item.

☐ Error Page No. _____ Drawing No. _____

☐ Addition Page No. _____ Drawing No. _____

☐ Other Page No. _____ Drawing No. _____

Explanation:

FOLD FOLD

CUT ALONG LINE

Fold and Staple

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

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER
Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD

FOLD

STAPLE

STAPLE