iSBX 488™ GENERAL PURPOSE INTERFACE BUS (GPIB) MULTIMODULE™ BOARD HARDWARE REFERENCE MANUAL

Order Number: 143154-001

intel®

iSBX 488™ GENERAL PURPOSE INTERFACE BUS (GPIB) MULTIMODULE™ BOARD HARDWARE REFERENCE MANUAL

Order Number: 143154-001

Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051

REV.	REVISION HISTORY	PRINT DATE
-001	Original Issue	6/81

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

Intel Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in an Intel product. No other circuit patent licenses are implied.

The following are trademarks of Intel Corporation and its affiliates and may be used only to identify Intel products:

BXP
CREDIT
i
ICE
iCS
i_m
Insite
Intel
Int_el
Intelevision
Intellec
iRMX
iSBC
iSBX
Library Manager
MCS
Megachassis
Micromap
Multibus
Multimodule
PROMPT
Promware
RMX/80
System 2000
UPI
μScope

and the combination of ICE, iCS, iRMX, iSBC, iSBX, MCS, iMMX or RMX and a numerical suffix.

PREFACE

This manual provides general information, preparation for use instructions, programming information, principles of operation and service information for the iSBX 488 GPIB Multimodule Board. Additional information is available in the following publications:

- Intel *iSBX Bus Specification,* Order Number 142686.
- Intel *Multibus Specification,* Order Number 9800683.
- *Using the 8292 GPIB Controller,* Application Note AP-66.
- *IEEE Standard Digital Interface for Programmable Instrumentation,* Order Number IEEE Std 488-1978. Published by the Institute of Electrical & Electronics Engineers, Inc. Address: 345 East 47th Street, New York, N.Y. 10017.
- Intel *Component Data Catalog*

CONTENTS

TABLES

ILLUSTRATIONS

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The iSBX 488 GPIB Interface Multimodule Board implements the 1978 IEEE Standard 488 bus, using Intel Large Scale Integration (LSI) devices. The board is designed to interface a host iSBC Single Board Computer to one or more (up to 15) peripheral devices via the General Purpose Interface Bus (GPIB). The iSBX 488 Multimodule Board may reside directly on the component side of any iSBX Multimodule compatible iSBC board, and is interfaced to and powered by the host board through the iSBX connector (Figure 1-1).

1-2. DESCRIPTION

The iSBX 488 Multimodule Board utilizes several Intel support devices to perform most of the processing associated with the IEEE-488 bus. The 8291A GPIB Talker/Listener device is used to perform most of the interfacing between the host single board computer and the external IEEE-488 bus. Its capabilities include but are not limited to performing the functional subsets allowed by the IEEE-488 Standard. In general, these functions are Acceptor Handshake, Listener Handshake, Talker, Listener, Service Request, Remote-Local, Parallel Poll, Device Clear & Device Trigger.

The iSBX 488 Multimodule Board also utilizes an Intel 8292 GPIB Controller in conjunction with the 8291A. The 8292 acts as a slave processor to the host CPU thus performing the GPIB controller interface function. The actual electrical interface between the iSBX 488 Multimodule Board and the IEEE-488 bus is performed by two Intel 8293 GPIB Transceivers. These bidirectional drivers are specifically designed for GPIB applications.

The iSBX 488 Multimodule Board may be interfaced with the IEEE 488 bus by connecting to the iSBC 988, GPIB cable assembly, for connection to the IEEE 488 bus. This flat cable is approximately one-half meter long, and is terminated with a 26-pin edge connector at one end and a 24-pin GPIB plastic receptacle at the opposite end.

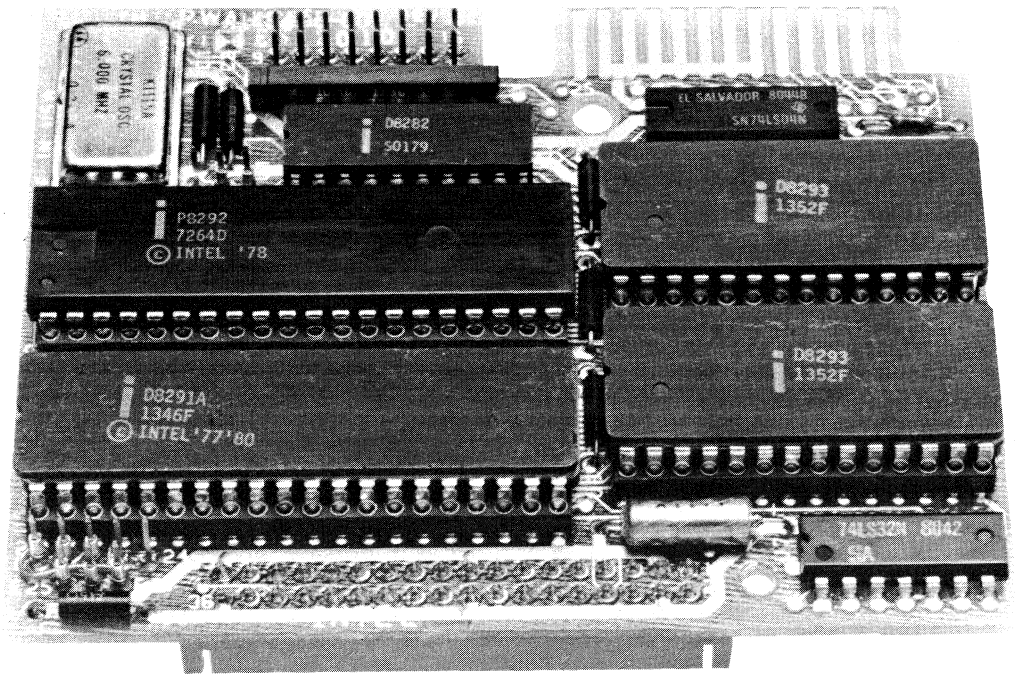


Figure 1-1. iSBX 488™ GPIB Board

1-3. DOCUMENTATION & EQUIPMENT SUPPLIED

The iSBX 488 Multimodule board is shipped with the following documentation & equipment:

a. Schematic Diagram
b. 1 Nylon Spacer (0.5 in. X 6/32 in.)
c. 2 Nylon Screws (0.25 in. X 6/32 in.)
d. 8 Jumper Receptacles

The iSBX 488 Multimodule Board is shipped from the factory with a corresponding set of schematic diagrams. These diagrams should be inserted into the back of this manual for future reference. See section 5-4 for related information.

1-4. SPECIFICATIONS

Specifications of the iSBX 488 Multimodule Board are provided in Table 1-1.

Table 1-1. iSBX 488 Multimodule™ Board Specifications

POWER REQUIREMENTS:	Vcc = +5 Vdc ± 5% Icc = 600 Milliamps maximum		
ENVIRONMENTAL REQUIREMENTS:			
Operating Temperature:	0° to 55° C		
Relative Humidity:	90% Maximum, non-condensing		
Heat Dissipation:	45.9 Gram-Calories/minute (0.18 Btu/min)		
PHYSICAL DIMENSIONS:			
Width:	9.40 cm (3.70 in.)		
Length:	7.24 cm (2.85 in.)		
Height:	2.04 cm (0.80 in.)		
Weight:	87.80 gm (3.10 oz.)		
GPIB FUNCTION	Symbol	Subsets	
Source Handshake	(SH)	SH0, SH1	
Acceptor Handshake	(AH)	AH0, AH1	
Talker	(T)	T0 through T8	
Extended Talker	(TE)	TE0 through TE8	
Listener	(L)	L0 through L8	
Extended Listener	(LE)	LE0 through LE8	
Service Request	(SR)	SR0, SR1	
Remote Local	(RL)	RL0, RL1	
Parallel Poll	(PP)	PP0, PP1*, PP2	
Device Clear	(DC)	DC0, DC1, DC2	
Device Trigger	(DT)	DT0, DT1	
Controller	(C)	C0 through C28	
INTERFACE CONNECTORS:			
Interface	No of Pins	Pin Centers in / mm	Mating Connectors
P1, iSBX	36	0.1 / 2.54	Intel 103059-001
J1, GPIB	26	0.1 / 2.54	3M 3462-0001 AMP 88373-5

*The host processor must interpret the remote commands PPU, PPC, PPE and PPD and send local message, Ipe, to the iSBX 488 board. Refer to Section 3-12, Parallel Poll Protocol.

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter provides instructions for installing the iSBX 488 Multimodule Board onto your host iSBC Single Board Computer. Instructions for configuring the Multimodule board jumpers are also given. Board DC and AC operating characteristics are specified in this chapter.

2-2. UNPACKING & INSPECTION

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or water-stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present when the carton is opened and the contents are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment contact the Intel Product Service HOTLINE to obtain a return authorization number and further instructions (see section 5-2). A purchase order will be required to complete the repair. A copy of the purchase order should be submitted to the carrier with your claim.

2-3. INSTALLATION CONSIDERATIONS

The iSBX 488 board can be installed on any 8-bit or 8/16-bit iSBC board equipped with an iSBX Multimodule connector. Power requirements and operating temperature ranges are provided in Chapter 1, Table 1-1.

2-4. POWER & COOLING REQUIREMENTS

The host iSBC board provides power to the iSBX 488 Multimodule board via the iSBX connector. The maximum power requirement for the iSBX 488 board is 600 mA @ 5V. (±0.25 V).

The iSBX 488 board dissipates a maximum of 45.9 gram-calories per minute of heat. Adequate air circulation must be provided to prevent a chassis temperature rise over 55°C (131°F).

2-5. PHYSICAL DIMENSIONS

Physical dimensions of the iSBX 488 board are provided in Figure 2-1. Mounting clearance detail is shown in Figure 2-2.

NOTE

In some cardcage models, two slots are used by the host iSBC & iSBX board combination.

2-6. INSTALLATION PROCEDURE

The iSBX 488 Multimodule Board can be easily installed without special equipment or tools. The following procedure outlines iSBX 488 Multimodule Board installation:

CAUTION

Host iSBC board must be removed from chassis or cardcage for proper installation. Turn off power before removal.

a. Some iSBC Single Board Computers have up to three iSBX Multimodule connectors. Choose the connector location which corresponds to the host I/O addressing you select. Refer to the host board hardware reference manual for the base address identification.

b. Install the supplied threaded spacer on the solder side of the Multimodule Board (at the hole). Secure the spacer by hand-tightening one of the supplied 1/4 inch screws through the component side of the iSBX 488 Multimodule Board (refer to Figure 2-3).

c. Locate pin 1 on the host iSBX connector. Similarly, locate pin 1 on the iSBX 488 Multimodule Board iSBX connector.

d. Carefully match the connectors at pin 1 and insert the iSBX 488 Multimodule Board into the host board iSBX connector until it is fully inserted and correctly seated. The iSBX 488 Multimodule Board J1 connector should be oriented in the same direction as the host board's I/O connectors.

e. Push the remaining 1/4 inch screw up through the bottom of the host board and thread it into the spacer.

f. Tighten down both screws as shown in Figure 2-3.

g. Refer to Section 2-7 for jumper connection information. If no jumper connections are required, install the host board back into its chassis.

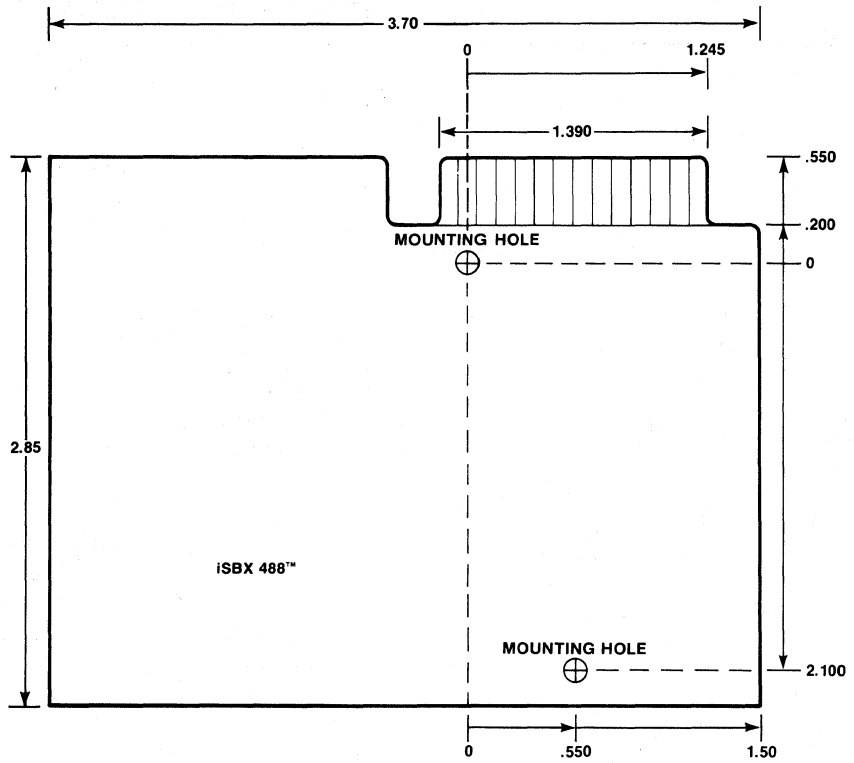


Figure 2-1. Physical Dimensions (Inches)

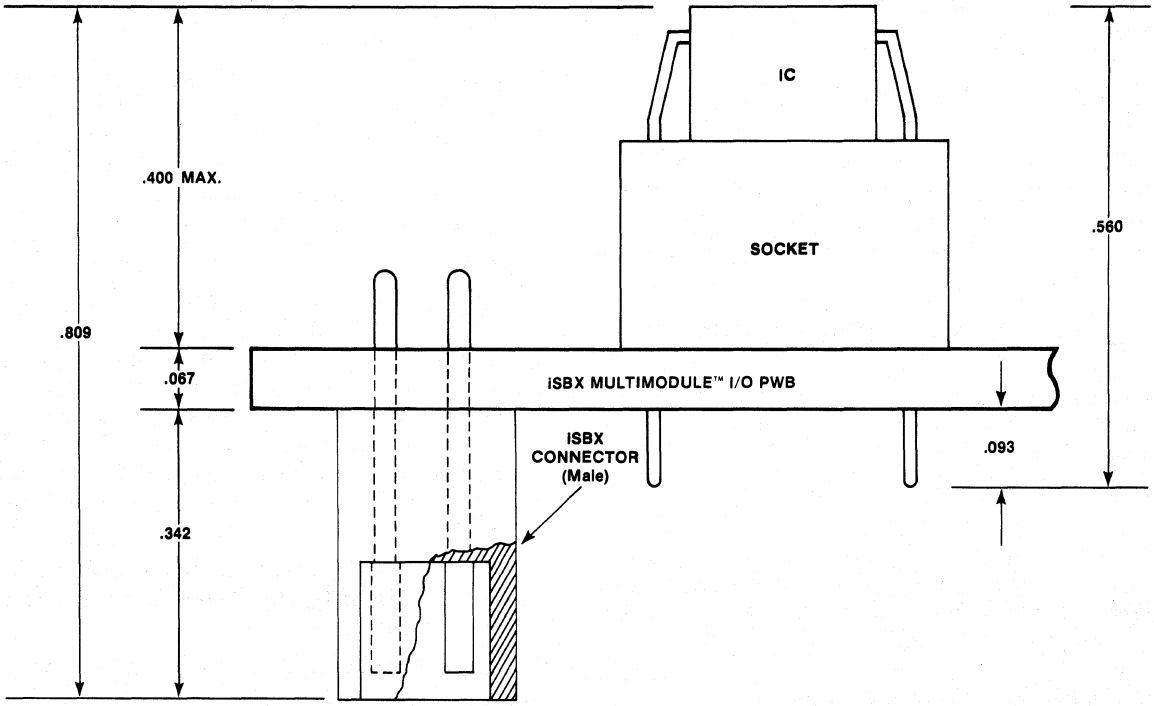


Figure 2-2. Mounting Clearances (Inches)

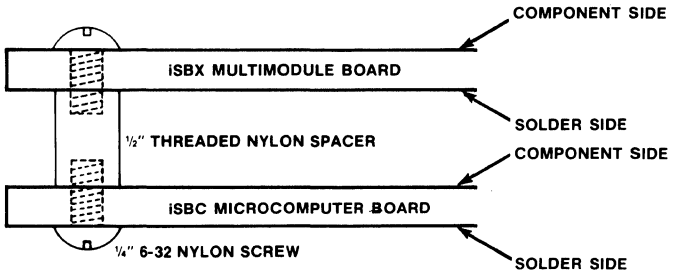


Figure 2-3. Mounting Technique

2-7. JUMPER CONFIGURATIONS

The iSBX 488 Multimodule Board has several optional jumper configurations which may be implemented to match your application.

Address Jumpers

Table 2-1 provides the options available for the Address jumpers. These jumpers are not installed at the factory. Push-on jumper receptacles are provided for configuring the jumper options of your choice.

All jumper connections in Table 2-1 are general purpose, and must be used with the appropriate software programming to provide the desired function. The table indicates the function typically assigned to these particular jumpers.

The address jumper matrix allows you to set the 5-bit binary address, talk address bit, listen address bit, and jumper E8-E16 is used to indicate if the on-board 8292 circuit is or is not the GPIB system controller.

Interrupt Output Jumpers

Three jumper connections are used to route optional 8292 interrupts to the host iSBC board. The following table indicates the factory configuration:

Interrupt	Jumper Pair	Destination
OBFI	22 - 25	P1-30 (OPT0)
$\overline{\text{IBFI}}$	21 - 26	P1-28 (OPT1)
TCI	20 - 27	DI7 on 8282 Latch

The OBFI & $\overline{\text{IBFI}}$ interrupts are typically used when transferring data between the 8292 device and a host processor. The TCI interrupt flags the host processor that certain commands have been executed. In the factory configuration (E20-E27), the status of TCI may be read from the on-board 8282 latch (bit D7).

The 8291A device sources a jumper selected interrupt (DMA Request). This is connected to the iSBX board pin P1-34. Simultaneously, this signal may be jumpered from post E24 to the destination indicated in the table (i.e., E24-E25, E24-E26, or E24-E27).

NOTE

Only one interrupt souce may be connected to any individual destination.

Count Input Jumper

The count input jumper allows source selection of the Count Input pin on the 8292 device (pin 39). This jumper is default connected (E18 - E19) to count EOI transitions for sending or receiving multiple blocks of data. Alternatively, the COUNT input may be connected (E17 - E18) to count NDAC transitions for sending or receiving a single block of data.

TRIG Jumper

The TRIG signal originates from the 8291A device and may be jumpered to iSBX signals, OPT0 (E23 - E25) or to OPT1 (E23 - E26). This normally low signal, generates a 1 microsecond (minimum) high pulse in response to the Group Execute Trigger GPIB command.

Summary

Table 2-2 summarizes the factory installed and optional jumper configurations on the iSBX 488 board.

Table 2-1. Address Jumper Configurations

Jumper Pair	E7-E15	E6-E14	E5-E13	E4-E12	E3-E11	E2-E10	E1-E9
Recommended Assignment	ET/DT	EL/DL	AD5	AD4	AD3	AD2	AD1
Talker Address	ET	DL	Talk Address				
Listener Address	DT	EL	Listen Address				
Talker/Listener Address	ET	EL	Talk/Listen Address				
Don't Care	DT	DL	Don't Care				

Notes:
1. Address bit ADi = Logic 1 when Jumper is not installed.
2. Enable Talk Address (ET) Jumper IN
   Enable Listen Address (EL)
3. Disable Talk Address (DT) Jumper OUT
   Disable Listen Address (DL)
4. Jumper E18-E16 is not general purpose. A jumper installed disables the System Controller Function of the board.

NOTE: None of these jumpers are installed at the factory; only one System Controller (SYC) allowed per GPIB.

Table 2-2. iSBX 488™ Jumper Configurations Summary

Destination Location	Source Location						
	OBFI U3pin35	IBFI U3pin36	TCI U3pin32	DREQ** U5pin6	TRIG U5pin6	EOI U6pin3	NDAC U6pin10
OPT 0, P1-30	E22-E25*	E21-E25	E20-E25	E24-E25	E23-E25	—	—
OPT 1, P1-28	E22-E26	E21-E26*	E20-E26	E24-E26	E23-E26	—	—
DI7, U1pin8	E22-E27	E21-E27	E20-E27*	E24-E27	E23-E27	—	—
COUNT,U3pin39	—	—	—	—	E23-E18	E19-E18*	E17-E18

NOTES:
* = Factory Installed
** = DREQ should not be jumpered whenever the host baseboard is terminating DREQ at connector P1-34.
hyphen = Not applicable or recommended.

2-8. iSBX CONNECTOR PIN ASSIGNMENTS (P1)

Pin assignments for the iSBX connector (P1) on the iSBX 488 Multimodule Board are provided in Table 2-3. Signal descriptions are given in Table 2-4.

2-9. iSBX CONNECTOR AC & DC SIGNAL SPECIFICATIONS

Interface loading specifications for the iSBX connector signals are provided in Table 2-5. Timing specifications are shown in Figure 2-4 and Table 2-6.

2-10. GPIB CONNECTOR PIN ASSIGNMENTS (J1)

Pin assignments for the GPIB connector (J1) on the iSBX 488 Multimodule Board are provided in Table 2-7. Signal descriptions are given in Table 2-8.

2-11. GPIB AC and DC SIGNAL SPECIFICATIONS

The timing protocol for a typical GPIB transaction is shown in Figure 2-5. The AC specifications are given in Table 2-9. The DC loading specifications for the GPIB interface (J1 connector) are provided in Table 2-10.

Table 2-3. iSBX™ Connector Pin Assignments

PIN	MNEMONIC	DESCRIPTION	PIN	MNEMONIC	DESCRIPTION
35	GND	SIGNAL GROUND	36	+5V	+5 Volts
33	MDO	MDATA BIT 0	34	MDRQT	M DMA REQUEST
31	MD1	MDATA BIT 1	32	MDACK/	M DMA ACKNOWLEDGE
29	MD2	MDATA BIT 2	30	OPT0	OPTION 0
27	MD3	MDATA BIT 3	28	OPT1	OPTION 1
25	MD4	MDATA BIT 4	26	TDMA	TERMINATE DMA*
23	MD5	MDATA BIT 5	24		RESERVED*
21	MD6	MDATA BIT 6	22	MCS0/	M CHIP SELECT 0
19	MD7	MDATA BIT 7	20	MCS1/	M CHIP SELECT 1
17	GND	SIGNAL GROUND	18	+5V	+5 Volts
15	IORD	IO READ COMMAND	16	MWAIT/	M WAIT*
13	IOWRT/	IO WRITE COMMAND	14	MINTR0	M INTERRUPT 0
11	MA0	M ADDRESS 0	12	MINTR1	M INTERRUPT 1
9	MA1	M ADDRESS 1	10		RESERVED*
7	MA2	M ADDRESS 2	8	MPST/	M PRESENT
5	MRESET	RESET	6	MCLK/	M CLOCK*
3	GND	SIGNAL GROUND	4	+5V	+5 Volts
1	+12V	+12 Volts*	2	-12V	-12 Volts*

NOTE: * = Not used on iSBX 488 board.

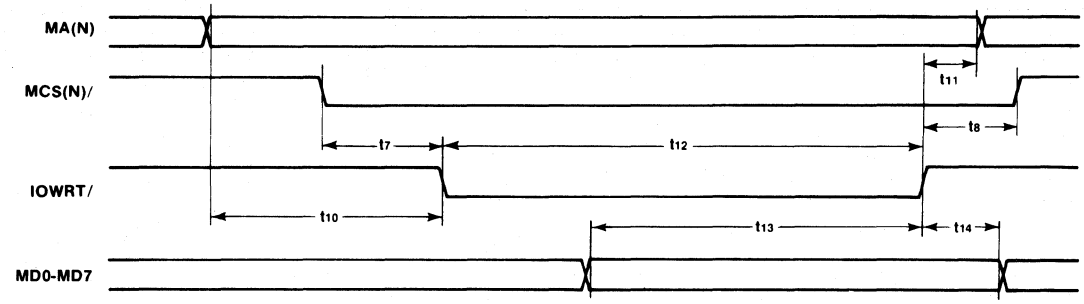
Table 2-4. iSBX™ Connector Signal Descriptions

IORD/	Commands the Multimodule board to perform the read operation.
IOWRT/	Commands the Multimodule board to perform the write operation.
MRESET/	Initializes the Multimodule board to a known internal state.
MCS0/	Chip select 0.
MCS1/	Chip select 1.
MA0-2	Least significant three bits of the I/O address. Used in conjunction with the chip select and command lines.
MPST/	Multimodule present indicator. Informs host board that a Multimodule board(s) is installed.
MINTR0-1	Interrupt request lines from the Multimodule board to the host board interrupt matrix.
OPT0-1	Optional use lines. May be used for additional interrupt request lines.
MD0-7	Bidirectional data lines.
MDRQT	Multimodule DMA Request issued by iSBX 488 Board.
MDACK/	DMA Acknowledge response from the host board DMA controller.

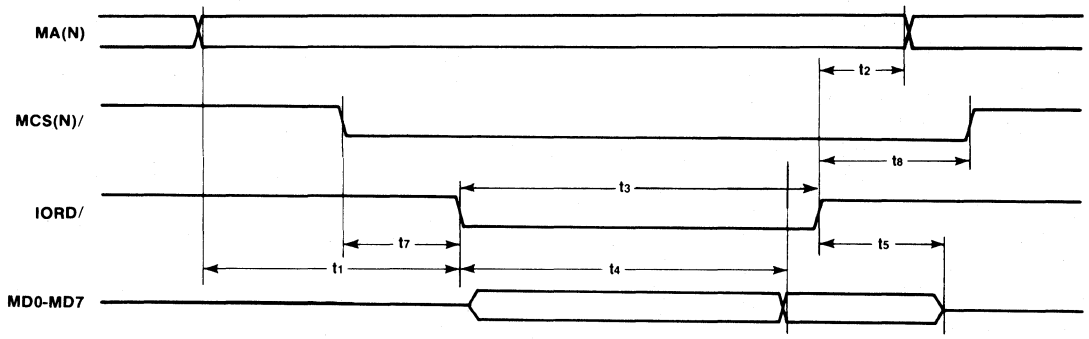
Table 2-5. iSBX™ Multimodule Board I/O DC Specifications (P1)

Output						
Bus Signal Name	**Type Drive**	**IOL MAX -Min (mA)**	**@ Volts (Vol Max)**	**IOH MAX -MIN (μA)**	**@ Volts (VOH Min)**	**C0 (Min) (pf)**
MD0-MD7	TRI	1.6	0.5	-200	2.4	130
MINTR0-1	TTL	2.0	0.5	-100	2.4	40
MDRQT	TTL	1.6	0.5	- 50	2.4	40
OPT1-2	TTL	1.6	0.5	- 50	2.4	40
MPST/	**					
Input						
Bus Signal Name	**Type Receiver**	**IIL MAX (mA)**	**@ VIN Max (volts)**	**IIH MAX (μA)**	**@ Vin Max (volts)**	**C0 (Min) (pf)**
MD0-MD7	TRI	-0.5	0.8	70	2.2	40
MA0-MA2	TTL	-0.5	0.8	70	2.0 (2.2)*	40
MCS0/-MCS1/	TTL	-4.0	0.8	100	2.0	40
MRESET	TTL	-2.1	0.8	100	2.0	40
MDACK/	TTL	-1.0	0.8	100	2.0	40
IORD/ IOWRT/	TTL	-1.0	0.8	100	2.2	40
OPT1-OPT2	TTL	-2.0	0.8	100	2.0	40

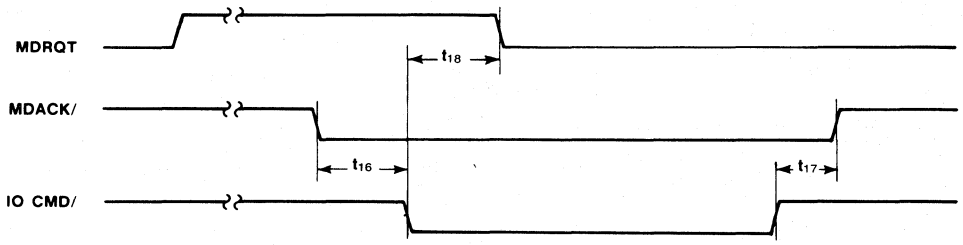
NOTES:
TTL = standard totem pole output. TR1=Three-state
* = VIN $\geqslant$ 2.2 volts required for MA0 only
** = MPST/ is connected to signal ground



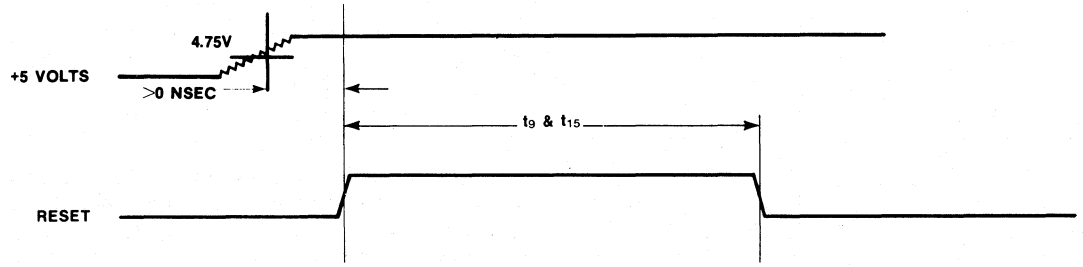
iSBX Multimodule™ Board I/O Write Timing



iSBX Multimodule™ Board I/O Read Timing



iSBX Multimodule™ Board I/O DMA Timing



iSBX Multimodule™ Board I/O Reset Timing

Figure 2-4. P1 Interface Timing Specifications

Table 2-6. iSBX Multimodule™ Board I/O AC Specifications

Symbol	Parameter	Min (ns)	Max (ns)	Figure Reference
t_1	Address stable before read	50	—	2-4
t_2	Address stable after read	30	—	2-4
t_3	Read pulse width	300	—	2-4
t_4	Data valid from read	0	250	2-4
t_5	Data float after read	0	150	2-4
t_6	Time between RD and/or WRT	—	Note 3	2-4
t_7	CS stable before CMD	25	—	2-4
t_8	CS stable after CMD	30	—	2-4
t_9	Power up reset pulse width	50 Msec	—	2-4
t_{10}	Address stable before WRT	50	—	2-4
t_{11}	Address stable after WRT	30	—	2-4
t_{12}	Write pulse width	300	—	2-4
t_{13}	Data valid to write	250	—	2-4
t_{14}	Data valid after write	30	—	2-4
t_{15}	Reset pulse width	10 Msec	—	2-4
t_{16}	DACK set up to I/O CMD	100	—	2-4
t_{17}	DACK hold	30	—	2-4
t_{18}	CMD to DMA RQT removed to end of data cycle	—	200	2-4

NOTE:
Time dependent on the host iSBC board to which the Multimodule board is connected.

Table 2-7. GPIB Connector (J1) Pin Assignments

Pin No	Signal	Pin No.	Signal
1	DIO5	14	NRFD
2	DIO1	15	GND
3	DIO6	16	NDAC
4	DIO2	17	GND
5	DIO7	18	IFC
6	DIO3	19	GND
7	DIO8	20	SRQ
8	DIO4	21	GND
9	REN	22	ATN
10	EOI	23	GND
11	GND	24	SHIELD
12	DAV	25	SHIELD
13	GND	26	SHIELD

2-12. CONNECTOR & CABLE INFORMATION

The iSBX 488 Multimodule Board is compatible with the iSBC 988 GPIB interface cable/connector assembly. The pin numbering conventions used on the board and the two cable connectors are not identical. Table 2-11 provides pin correspondence among the three connectors.

The 26-pin connector should be inserted onto the iSBX 488 Multimodule Board J1 edge connector. The 24-pin connector should be connected to the GPIB. The two extra ground lines are shield lines which can be used for earth termination purposes.

Since the J1 connector is actually a two-row 13-pin connector, care must be exercised when installing the cable assembly. Odd numbered pins are on the component side of the iSBX 488 Multimodule Board, with pin 1 located at the corner edge (board is marked accordingly). Both board and cable connector have triangle reference marks which should be aligned as shown in Figure 2-6 to ensure proper board to cable interface.

Table 2-8. GPIB Connector Signal Descriptions

Signal	Description
Data Bus	Lines DI01 through DI08 are used to transfer addresses, control information and data. The formats for addresses and control bytes are defined by the IEEE 488 standard. Data formats may be ASCII (with or without parity) or binary. DI01 is the Least Significant Bit (bit 0).
Management Bus	
ATN	*Attention.* This signal is asserted by the Controller to indicate that it is placing an address or control byte on the Data Bus. ATN is de-asserted to allow the assigned Talker to place status or data on the Data Bus. The Controller regains control by reasserting ATN.
EOI	*End or Identify.* This signal has two uses as its name implies. A talker may assert EOI simultaneously with the last byte of data to indicate end of data. The Controller may assert EOI along with ATN to initiate a Parallel Poll.
SRQ	*Service Request.* This line is like an interrupt: it may be asserted by any device to request the Controller to take some action. The Controller must determine which device is asserting SRQ by conducting a Serial Poll at its earliest opportunity.
IFC	*Interface Clear.* This signal is asserted only by the System Controller in order to initialize all device interfaces to a known state.
REN	*Remote Enable.* This signal is asserted only by the System Controller. Its assertion does not place devices into Remote Control mode; REN only *enables* a device to go remote when addressed to listen.
Transfer Bus	
NRFD	*Not Ready For Data.* This handshake line is asserted by a listener to indicate it is not yet ready for the next data or control byte.
NDAC	*Not Data Accepted.* This handshake line is asserted by a Listener to indicate it has not yet accepted the data or control byte on the DIO lines.
DAV	*Data Valid.* This handshake line is asserted by the Talker to indicate that a data or control byte has been placed on the DIO lines and has had the minimum specified settling time.

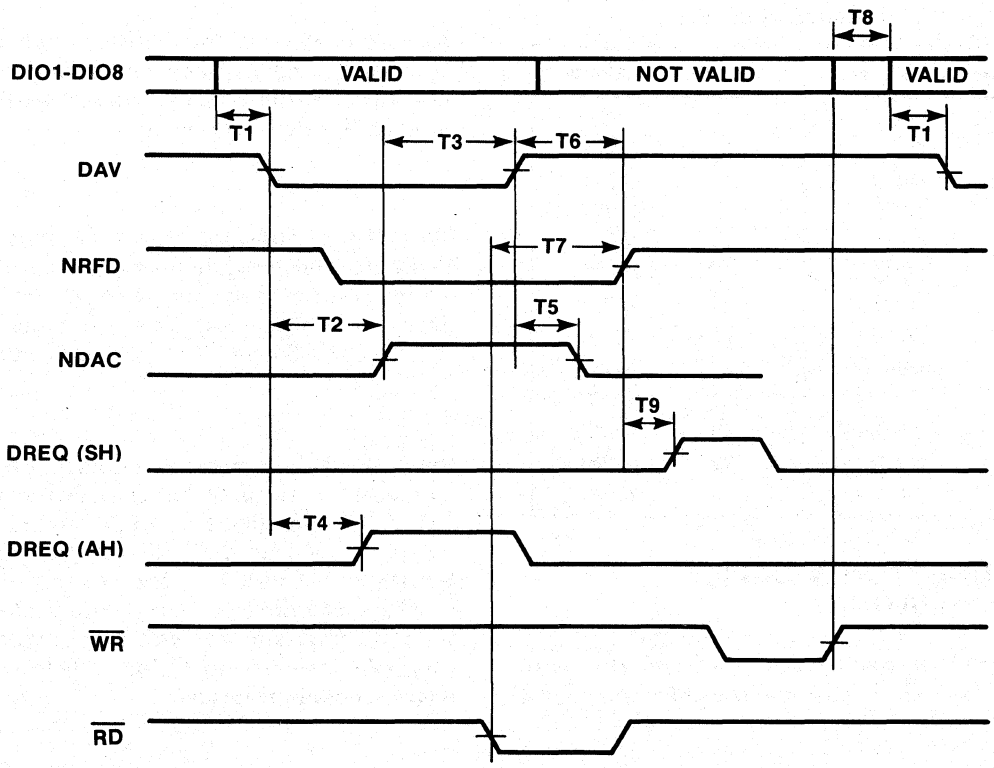


Figure 2-5. GPIB AC Timing Waveforms

Table 2-9. GPIB AC Timing Specifications

Symbol	Parameter	GPIB State	Min (ns)	Max (ns)	Reference
T1	DIO Valid to DAV	SDYS	167	2750	Section 3-11
T2	DAV to DAC	ACDS		730	Figure 2-5
T3	DAC to DAV false	SWNS		430	Figure 2-5
T4	DAV to DREQ	AH & LACS		650	Figure 2-5
T5	DAV false to NDAC	ANRS		440	Figure 2-5
T6	DAV false to RFD	ACRS		430	Figure 2-5
T7	IORD/ to RFD	ACRS		530	Figure 2-5
T8	IOWR/ false to DIO	TACS		310	Figure 2-5
T9	RFD to DREQ	SH & TACS		450	Figure 2-5

Table 2-10. GPIB Interface, J1, DC Specifications

Symbol	Parameter	Limits Min	Limits Max	Unit	Test Condition
V_{IL}	Input Low Voltage		0.8	V	I_{OL}=48mA
V_{OL}	Output Low Voltage		0.5	V	I_{OH}=-5.2mA
V_{OH}	Output High Voltage	2.4		V	
$V_{T+}-V_{T}$	Receiver Input Hysteresis	400		mV	
V_{IT}	Receiver Threshold H to L	0.8		V	
	L to H		2.0	V	V_{CC}=OV
I_{PD}	Bus Power Down Leakage Current	-10	10	μA	V_{IL}=0.4V
I_{IL}	Low Input Load Current	- 3.2	- 1.3	mA	V_{IH}=3.7V
I_{IH}	High Input Load Current	0.0	2.5	mA	

Table 2-11. Pin Correspondance

GPIB Signal Name	PWA, J1 Connector	26-Contact Edge Connector	24-Contact Receptacle
DIO 1	2	1	1
DIO 2	4	3	2
DIO 3	6	5	3
DIO 4	8	7	4
DIO 5	1	2	13
DIO 6	3	4	14
DIO 7	5	6	15
DIO 8	7	8	16
DAV	12	11	6
NRFD	14	13	7
NDAC	16	15	8
EOI	10	9	5
REN	9	10	17
IFC	18	17	9
SRQ	20	19	10
ATN	22	21	11
GND	11	12	18
GND	13	14	19
GND	15	16	20
GND	17	18	21
GND	19	20	22
GND	21	22	23
GND	23	24	24
SHIELD	24	23	12
SHIELD	25	26	Void
SHIELD	26	25	Void

2-13. INSTALLATION SUMMARY

The following list summarizes the complete iSBX 488 board installation procedure:

a. Perform any required jumper modifications on the iSBX 488 board (refer to Section 2-7).

b. Install the iSBX 488 board onto the host iSBC board (refer to Section 2-6). Ensure that host iSBC board is removed from its cardcage and power is not applied.

c. Install the host iSBC board and iSBX 488 board combination back into the cardcage. Ensure that power is not present and that physical clearance is provided for combined board height.

d. Install iSBC 988 Cable Assembly (or equivalent) between iSBX 488 board and GPIB (refer to section 2-12). Ensure that power is not applied to the iSBC/iSBX system or the GPIB system.

e. Check other I/O cables on host iSBC board system for correct seating.

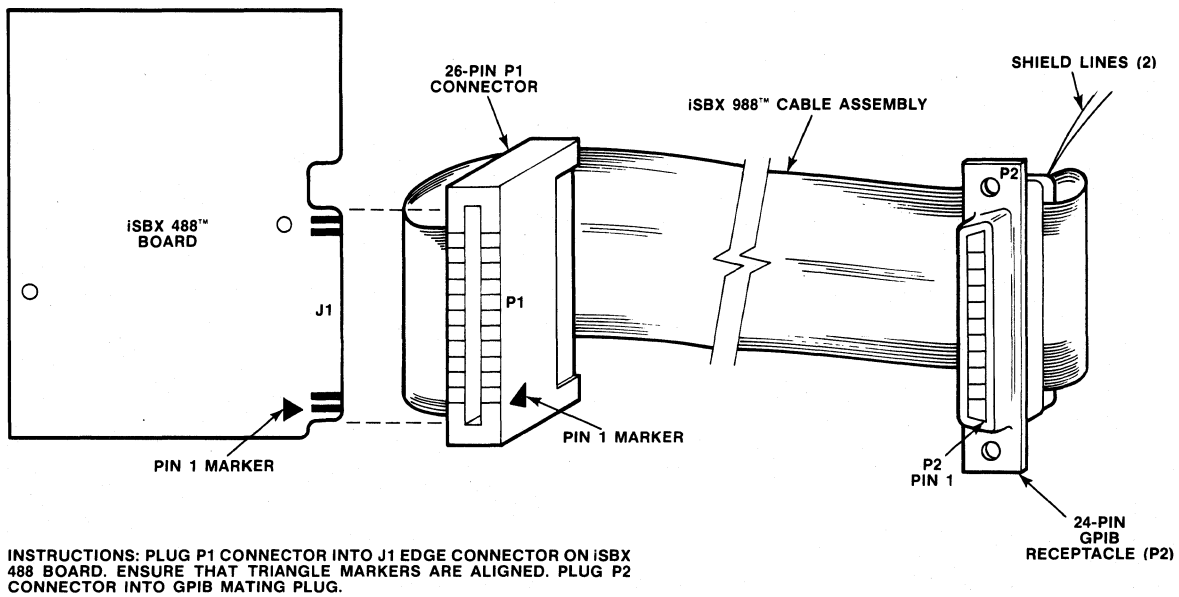


Figure 2-6. iSBX 988™ Cable Installation

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

This chapter provides programming instructions and protocol information for the iSBX 488 Multimodule Board. This information includes I/O addressing, register descriptions, system initialization, and programming examples.

3-2. iSBX 488 MULTIMODULE BOARD PROTOCOL

All communication between the host iSBC board and the iSBX 488 Multimodule Board is executed via the iSBX connector. The 8291A device and the 8292 device can each communicate independently with the host processor.

The 8291A device handles all GPIB non-controller functions in which data or command/status information may be read from or written to the host board system program. Direct Memory Access (DMA) operation is available for bus data transfer operations. DMA is discussed in section 4-6. Indicating 8291A status may be interrupt driven or polled.

The 8292 device handles all GPIB controller functions. Communication to the host CPU may be interrupt driven or status polled.

The 8282 device is used as a general purpose read-only register, except bit D7. Bit D7 may be used to indicate TCI (Task Completion Interrupt) status in polling 8292 related routines.

3-3. iSBX I/O PORT ADDRESSING

I/O Port addressing for the iSBX 488 Multimodule Board can be divided into three groups: 8291A registers, 8292 registers, and the 8282 register. The host board must assert the proper chip select signal in conjunction with the desired address to perform I/O read or write operations. Table 3-1 summarizes the I/O addresses and chip select requirements.

Table 3-1. I/O Port Addresses & Chip Select Assignments

		iSBX Address Lines			iSBX I/O Port
8291A Registers Multimodule Chip Select 0 (MCSO/=0)		MA2	MA1	MA0	Hex Address ***
Read	**Write**				
Data In	Data Out	0	0	0	X0
Interrupt Status 1	Interrupt Mask 1	0	0	1	X1
Interrupt Status 2	Interrupt Mask 2	0	1	0	X2
Serial Poll Status	Serial Poll Mode	0	1	1	X3
Address Status	Address Mode	1	0	0	X4
Command Pass Through	Aux Mode	1	0	1	X5
Address 0	Address 0/1	1	1	0	X6
Address 1	EOS	1	1	1	X7
8292 Registers Multimodule Chip Select 1 (MCS1/=0)					
Read	**Write**				
*Interrupt Mask	**Interrupt Mask	N	1	0	XA or XE
*Error Flag		N	1	0	XA or XE
*Error Mask	**Error Mask	N	1	0	XA or XE
*Event Counter Status	*Event Counter	N	1	0	XA or XE
*Time Out Status	*Time Out	N	1	0	XA or XE
*Controller Status		N	1	0	XA or XE
*GPIB Status		N	1	0	XA or XE
Interrupt Status	Command Field	N	1	1	XB or XF
8282 Register					
Read (MCS1/=0)		N	0	N	X8 or X9 or, XC or XD

NOTES:

* This register is accessed for an appropriate read or write by first writing a specific byte to the Command Field Register. See section 3-14 for more details.

** The Interrupt Mask and Error Mask register are distinguished from each other by the value of the most significant bit, D7, in the byte written to the 8292. See Table 3-6 for more details.

N indicates an irrelevant condition.
0 indicates a Low Voltage state.
1 indicates a High Voltage state.

*** The hex addresses correspond to an 8-bit iSBC board only. The first digit of each I/O address is represented as X since it will change depending on the type of host iSBC board used. Refer to the HRM for your host iSBC board to determine the first digit required for the I/O address.

3-4. 8291A REGISTERS

The 8291A circuit utilizes 16 internal registers to communicate with the host board. Register addressing was described in section 3-3. Table 3-2 provides a summary of register bit identification, and the associated iSBX I/O port address.

Sections 3-5 through 3-14 describe the 8291A registers.

3-5. DATA REGISTERS

D17	D16	D15	D14	D13	D12	D11	D10

DATA IN

D07	D06	D05	D04	D03	D02	D01	D00

DATA OUT

The 8291A utilizes two data registers: the data-in register and the data-out register. The data-in register is used to move data from the GPIB to the host iSBC board when the 8291A is an active listener. The data-out register is used to move data from the host iSBC board to the GPIB when the 8291A is an active talker. The data from the GPIB is latched into the data-in register, and its contents are not destroyed by writing to the data-out register. Likewise, a read of the data-in register does not destroy information in the data-out register.

3-6. INTERRUPT REGISTERS

CPT	APT	GET	END	DEC	ERR	BO	BI

INTERRUPT STATUS 1

INT	SPAS	LLO	REM	SPC	LLOC	REMC	ADSC

INTERRUPT STATUS 2

CPT	APT	GET	END	DEC	ERR	BO	BI

INTERRUPT MASK 1

0	0	DMAO	DMAI	SPC	LLOC	REMC	ADSC

INTERRUPT MASK 2

The 8291A is configured to generate an interrupt (to the host iSBC board) on occurrence of any of 12 GPIB conditions or events. The host board reads the 8291A interrupt status register to determine which event has occurred, and then executes the appropriate service routine. The bit mnemonics are summarized in Table 3-3. Most bits in the Interrupt Status 1 & 2 Registers have a corresponding bit mask. The bit is enabled by writing a logic one to the mask. Notice that four of the bits in the Interrupt Status 2 Register are status-only bits and do not generate interrupts.

The interrupt status registers are cleared when read or when a local PON message is executed. The bits in the interrupt status registers are enabled regardless of the corresponding enable bit in the Interrupt Mask Registers. The INT bit is cleared whenever the appro priate interrupt bit(s) is read and cleared from the Interrupt Status 1 and/or 2 Registers.

Table 3-2. 8291A Registers

READ REGISTERS								PORT # (HEX)	WRITE REGISTERS							
D17	D16	D15	D14	D13	D12	D11	D10	'X' 0	D07	D06	D05	D04	D03	D02	D01	D00
DATA IN									DATA OUT							
CPT	APT	GET	END	DEC	ERR	BO	BI	'X' 1	CPT	APT	GET	END	DEC	ERR	BO	BI
INTERRUPT STATUS 1									INTERRUPT MASK 1							
INT	SPAS	LLO	REM	SPC	LLOC	REMC	ADSC	'X' 2	0	0	DMAO	DMAI	SPC	LLOC	REMC	ADSC
INTERRUPT STATUS 2									INTERRUPT MASK 2							
S8	SRQS	S6	S5	S4	S3	S2	S1	'X' 3	S8	rsv	S6	S5	S4	S3	S2	S1
SERIAL POLL STATUS									SERIAL POLL MODE							
ton	lon	EOI	LPAS	TPAS	LA	TA	MJMN	'X' 4	TO	LO	0	0	0	0	ADM1	ADM0
ADDRESS STATUS									ADDRESS MODE							
CPT7	CPT6	CPT5	CPT4	CPT3	CPT2	CPT1	CPT0	'X' 5	CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0
COMMAND PASS THROUGH									AUX MODE							
INT	DT0	DL0	AD5-0	AD4-0	AD3-0	AD2-0	AD1-0	'X' 6	ARS	DT	DL	AD5	AD4	AD3	AD2	AD1
ADDRESS 0									ADDRESS 0/1							
X	DT1	DL1	AD5-1	AD4-1	AD3-1	AD2-1	AD1-1	'X' 7	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
ADDRESS 1									EOS							

BO & BI Interrupts

The BO and BI interrupts are used to perform data transfer cycles. BO indicates that a data byte should be written to the Data Out Register. Similarly, BI indicates that a data byte may be read from the Data In Register. BO is set whenever the 8291A is in TACS • (SWBS + SGNS) state and the RFD signal is true (passive high). When an active GPIB controller (other than the 8292 component) takes control (synchronously), the 8291A will source handshake the last byte out successfully. If the controller takes control asynchronously the 8291A will clear the output data and enter TADS • SIDS. The BO interrupt will reset. If the controller enters the standby mode, releasing ATN, the 8291A will enter TACS • SGNS and BO will be set.

If an IFC is issued by the System Controller the 8291A will exit TACS and enter TIDS, clearing BO. After IFC returns false and the 8291A is in the talk-only mode (refer to section 3-9 for information on addressing modes) the 8291A will enter TACS and set BO after ATN is released. BI is set whenever the 8291A is in the LACS • ACDS state.

The BI (BO) interrupt is reset after a byte has been read from (written to) the 8291A. BO and BI are also reset by issuing a "pon" command (refer to Auxiliary Commands in section 3-11), or by reading the Interrupt Status 1 Register. Data cycles may be performed without reading the Interrupt Status 1 register if all interrupts except BO & BI are disabled. BO & BI will reset automatically after each byte is transferred.

Serial Poll Complete Interrupt (SPC)

The serial poll complete interrupt is set when the Controller-In-Charge has accepted (DAC true) the 8291A status byte after the 8291A has requested service.

Command Pass Through (CPT) Interrupt

This interrupt indicates (to the host iSBC board) that an undefined command or a secondary command following an undefined command, has been received from the GPIB. Any message not decoded by the 8291A becomes an undefined command. The command is stored in the CPT register for use by the host board. See section 3-10 for further details of the CPT register and defined/undefined commands.

APT Interrupt

This interrupt indicates (to the host iSBC board) that a secondary address has been received and is ready to be validated by reading the command pass through register. This interrupt will only occur in Mode 3 addressing. See section 3-9.

Group Execute Trigger (GET)

This interrupt is set by the 8291A when the GET message is received. The 8291A must be addressed to listen. The TRIG output is asserted for at least 1 microsecond when the GET message is received.

END Int

The END interrupt bit is used to detect the end of a multibyte transfer. The bit is set when the 8291A is an active Listener and EOS (if enabled by Aux Reg A) or EOI is received. See Aux Mode Register, section 3-11.

DEC Int

The DEC Bit is set whenever a DCL message is received or the 8291A is addressed to Listen and a SDC message is received.

ERR Int

The ERR bit is set when the 8291A is an active talker and tries to send a byte to the GPIB and no Listeners are active.

3-7. STATUS BITS

Bits 4 through 7 of the Interrupt Status Register 2 are available to the Host board as status bits. These bits are status only. They will not generate Interrupts nor do they have corresponding mask bits. For example, if the Host board receives a REMC interrupt the nature of the interrupt can be determined by reading the REM Status Bit.

Bits 4 and 5 (DMAI, DMAO) of the Interrupt Mask 2 Register are available to enable direct data transfers between the iSBX data bus and the GPIB: DMAI (DMA in) enables the DREQ (DMA request) pin of the 8291A to be asserted upon the occurrence of BI. Similarly, DMAO (DMA out) enables the DREQ pin to be asserted upon the occurrence of BO. One might note that the DREQ pin may be used as a second interrupt output pin, monitoring BI and/or BO and

enabled by DMAI and DMAO. One should note that the DREQ pin is not affected by a read of the Interrupt Status 1 Register. It is reset whenever a byte is written to the Data Out Register or read from the Data In Register.

Bit 3 (SPC) of the Interrupt Status 2 Register indicates when a serial poll is complete, (assuming the 8291A had previously issued a service request, [SRQ], from the Controller-In-Charge). The SPC flag and interrupt (if unmasked) is set when the 8291A is exiting the SPAS • APRS state and entering the TADS • NPRS state. (I.e., when ATN is reasserted by the Controller-In-Charge.)

Bits 0, 1 and 2 (ADSC, REMC and LLOC) in the Interrupt Status 2 Register are used to indicate state changes. Bit 0, ADSC, indicates a transition in LIDS ↔ LADS or TIDS ↔ TADS or Major/Minor addressing (refer to section 3-9 for information on addressing modes). Bit 1, REMC, indicates a transition in LOCS ↔ REMS. Bit 2, LLOC, indicates a change in LWLS ↔ RWLS. The status bit and interrupt (if unmasked) will remain set even if more than one transition in that particular state has occurred (e.g., LIDS → LADS → LIDS). The nature of the state may be interrogated by reading bits 4, 5 or 6 of the Interrupt Status 2 Register.

3-8. SERIAL POLL REGISTERS

S8	SRQS	S6	S5	S4	S3	S2	S1

SERIAL POLL STATUS

S8	rsv	S6	S5	S4	S3	S2	S1

SERIAL POLL MODE

The serial poll mode register is used to establish the status byte that the iSBX 488 Multimodule Board will issue to the GPIB in response to the serial poll enable (SPE) message. Setting bit 7 (rsv) causes the 8291A to assert the SRQ line (J1-20), indicating its need for attention from the Controller-In-Charge of the GPIB. When service has been granted, the rsv bit is automatically cleared by the 8291A. The SPC interrupt is generated (if unmasked) after the Controller-In-Charge has reasserted ATN, ending the serial poll. The other bits of the register are available for sending status information over the GPIB.

The CPU may request service by writing another byte to the Serial Poll Mode Register with the rsv bit set. If the controller performs a serial poll when the rsv bit is clear, the last status byte written will be read, but the SRQ line will not be driven by the 8291A and the SRQS bit will be cleared in the status byte.

Table 3-3. Interrupt Register Bit Identification

BIT 7		CPT	An undefined command has been received.	INTERRUPT STATUS 1 REGISTER
6		APT	A secondary address must be passed through to the microprocessor for recognition.	
5		GET	A group execute trigger has occured.	
4		END	An EOS or EOI message has been received.	
3		DEC	Device Clear Active State has occurred.	
2		ERR	Interface error has occurred; no listeners are active.	
1		BO	A byte has been output.	
0		BI	A byte has been input.	
7	Shows status of the INT pin	INT	These are status only. They will not generate interrupts, not do they have corresponding mask bits.	INTERRUPT STATUS 2 REGISTER
6	The 8291A has been enabled for a serial poll	SPAS		
5	The 8291A is in local lock out state.	LLO		
4	The 8291A is in a remote state.	REM		
3		SPC	Serial Poll Complete Interrupt	
2		LLOC	Local lock out change interrupt LLO ↔ NO LLO	
1		REMC	Remote/Local change interrupt. Remote ↔ Local	
BIT 0		ADSC	Address status change interrupt.* Addressed ↔ Unaddressed	

*In ton (talk-only) and lon (listen-only) modes, no ADSC interrupt is generated.

The Serial Poll Status Register is available for reading the status byte in the Serial Poll Mode Register. The processor may check the status of a request for service by polling bit 7 of this register, which corresponds to the SRQS (Service Request State). When a Serial Poll is conducted and the Controller-In-Charge reads the status byte, the SRQS bit is cleared. The SRQ line is tied to this bit, so that a request for service is terminated when the 8291A status byte is read by the Controller-In-Charge. The Controller-In-Charge may, however, read the status more than once before ending the serial poll. The 8291A will continue to source handshake the status stored in the status register until the controller ends the poll by asserting ATN.

3-9. ADDRESS REGISTERS

There are five separate registers associated with GPIB addressing:

a. Address Mode Register
b. Address Status Register
c. Address 0/1 Register
d. Address 0 Register
e. Address 1 Register

Address Mode Register

TO	LO	0	0	0	0	ADM1	ADM0

ADDRESS MODE

The address mode register is used to select one of the five addressing modes available for the 8291A device. Setting the "TO" bit puts the 8291A in a Talk only Mode while setting the "LO" bit puts the 8291A in a Listen only Mode. These bits may be used by the Controller-In-Charge to set itself up for remote commands or data communication. These may also be used to allow a device to operate as a Talker or a Listener in an interface system without a controller.

In Mode 1, the content of the Address 0 Register constitutes the "Major" talker/listener address while the Address 1 Register represents the "Minor" talker/listener address. In applications where only one address is needed, the major talker/listener is used, and the minor talker/listener should be disabled. Loading an address via the Address 0/1 Register into Address Registers 0 and 1 enables the major and minor talker/listener functions respectively.

In Mode 2, the 8291A recognizes two sequential address bytes: a primary followed by a secondary. Both address bytes must be received to enable the device to talk or listen. In this manner, Mode 2 addressing implements the extended talker and listener functions as defined in the IEEE-488 standard.

To use Mode 2 addressing, the primary address must be loaded into the Address 0 Register, and the Secondary Address is placed in the Address 1 Register. With both primary and secondary addresses residing in the 8291A, all addressing sequences are handled without processor intervention.

In Mode 3, the 8291A handles addressing as in Mode 1, except each Major or Minor primary address must be followed by a secondary address. All secondary addresses must be verified by the microprocessor when Mode 3 is used. When the 8291A is in TPAS or LPAS (talker/listener primary addressed state), and it does not recognize the byte in the DIO lines, an APT interrupt is generated. The GPIB handshake is held off with NRFD true and the byte becomes available in the CPT (Command Pass-Through) Register. As part of its interrupt service routine, the microprocessor must read the CPT Register and write one of the following responses to the Auxiliary Mode Register:

a. 07H implies a non-valid secondary address
b. 0FH implies a valid secondary address

See section 3-11 for details on the Auxiliary Mode Register.

Address mode is selected by writing one of the following bytes to the address mode register:

Command Byte	Mode
10000000	Enable Talk-Only Mode (TON) Bit TO=1
01000000	Enable Listen-Only Mode (LON) Bit LO=1
11000000	8291A Self Talk, Bits TO=1, LO=1
00000001	Mode 1: Primary - Primary Bit ADM0 =1
00000010	Mode 2: Primary - Secondary Bit ADM1=1
00000011	Mode 3: Primary / APT - Primary / APT Bits ADMO=1, ADM1=1

Address Status Register

ton	lon	EOI	LPAS	TPAS	LA	TA	MJMN

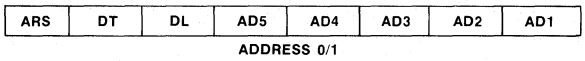
ADDRESS STATUS

The address status register is used by the host board to handle its own addressing. This includes status bits which monitor the address state of each talker/listener.

The Address Status Register contains information used by the microprocessor to handle its own addressing. This information includes status bits that monitor the address state of each talker/listener, "ton" and "lon" flags which indicate the talk and listen only states, and an EOI bit which, when set, signifies that the END message arrived with the last data byte. LPAS and TPAS indicate that the listener or talker primary address has been received. The microprocessor can use these bits when the secondary address is passed through to determine whether the 8291A is addressed to talk or listen. The LA (listener addressed) bit will be set when the 8291A is in LACS (Listener Active State) or in LADS (Listener Addressed State). Similarly, the TA (Talker Addressed bit) will be set to indicate TACS or TADS, but also to indicate SPAS (Serial Poll Active State).

The MJMN bit is used to determine whether the information in the other bits applies to the Major or Minor talker/listener. It is set to "1" when the Minor talker/listener is addressed. Note that only one talker/listener may be active at any one time. Thus, the MJMN bit will indicate which, if either, of the talker/listeners is addressed or active.

Address 0/1 Register



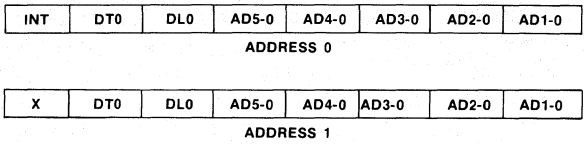
The Address 0/1 Register is used for specifying the device addresses according to the format selected in the Address Mode Register. Five bit addresses may be loaded into the Address 0 and Address 1 Registers by writing into the Address 0/1 Register. The Address Register Select (ARS) bit is used to select which of these registers the other seven bits will be loaded into. The DT and DL bits may be used to disable the talker or listener function at the address signified by the other five bits. When Mode 1 addressing is used and only one primary address is desired, both the talker and the listener should be disabled at the Minor address.

As an example of how the Address 0/1 Register might be used, consider an example where two primary addresses are needed in the device. The Major primary address will be selectable only as a talker and the Minor primary address will be selectable only as a listener. This configuration of the 8291A is formed by the following sequence of write operations by the microprocessor.

Operation	Write Data	Port Address Hex
1. Select addressing Mode 1	00000001	X4
2. Load major address into Address 0 Register with listener function disabled.	001AAAAA	X6
3. Load minor address into Address 1 Register with talker function disabled.	110BBBBB	X6

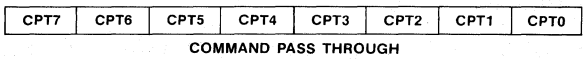
At this point, the addresses AAAAA and BBBBB are stored in the Address 0 and Address 1 Registers respectively, and are available to the host microprocessor. Thus, it is not necessary to store any address information elsewhere. Also, with the information stored in the Address 0 and Address 1 Registers, processor intervention is not required to recognize addressing by the controller. Only in Mode 3, where secondary addresses are passed through, must the processor intervene in the addressing sequence.

Address 0 and Address 1 Register



The Address 0 Register contains a copy of the Interrupt Status 2 Register (INT) bit 7. This is to be used when polling for interrupts. Software should poll register 6 checking for INT (bit 7) to be set. When INT is set, the Interrupt Status Register should be read to determine which interrupt was received.

3-10. COMMAND PASS THROUGH REGISTER



The command pass through register (CPT) stores the most recent 8-bit command received from the GPIB Controller-In-Charge (off board). This includes undefined as well as defined GPIB commands. (Note: The 8291A interprets any 8-bit data as a GPIB command whenever the data was accepted (DAC) while ATN was true). Table 3-4 lists all commands that are defined and undefined by the 8291A. Defined commands are acted upon automatically by the 8291A. Undefined commands have no effect on the 8291A operation but are stored (1 byte) in the CPT register for the host CPU to read and interpret.

Table 3-4. Defined/Undefined Commands Received

Mnemonic	Message
ATN	Attention
DCL	Device Clear
GET	Group Execute Trigger
GTL	Go To Local
LLO	Local Lock Out
MLA	My Listen Address
MTA	My Talk Address
MSA	My Secondary Address
*PPC	Parallel Poll Configure
*PPE	Parallel Poll Enable
*PPD	Parallel Poll Disable
*PPU	Parallel Poll Unconfigure
SDC	Selective Device Clear
SPD	Serial Poll Disable
SPE	Serial Poll Enable
*TCT	Take Control
UNL	Unlisten

NOTE:
* = This command is undefined to the 8291A but is stored in the CPT register for read access.

An APT or CPT interrupt may be sent to the host CPU and the GPIB handshake (SH and AH) halted indicating the availability of a secondary address or an undefined command in the CPT register. This feature is discussed in section 3-11 under Auxiliary Register B and section 3-9 under Mode 3 addressing.

An added feature of the 8291A is its ability to handle undefined secondary commands following undefined primaries. Thus, the number of available commands for future IEEE-488 definition is increased.

The recommended use of the 8291A undefined command feature is for a remote configured Parallel Poll (subset PP1). The PPC message is an undefined primary command typically followed by PPE or PPD, an undefined secondary command. For details on this procedure, refer to the section on Parallel Poll Protocol.

NOTE

All commands received are held in the CPT register but a Handshake Hold Off on CPT Interrupt is not generated unless it is an undefined command. The content of this register is destroyed when a new command is received.

3-11. AUXILIARY MODE REGISTER

CNT2	CNT1	CNT0	COM4	COM3	COM2	COM1	COM0

AUX MODE

The auxiliary mode register contains a three-bit control field linked with a five-bit command field. It is used for the following purposes:

a. To issue commands and GPIB local messages from the host board.

b. To load auxiliary registers.

c. To preset an internal counter.

Code		Command
Control Field	Command Field	
CNT2-CNT0	COM4-COM0	
000	$0C_3C_2C_1C_0$	Execute auxiliary command CCCC
001	$0F_4F_2F_1F_0$	Preset internal counter for T1 SH delay from 0.167 to 2.75 microseconds. FFFF (base 2 is the scalar.)
100	$A_4A_3A_2A_1A_0$	Write $A_4A_3A_2A_1A_0$ into auxiliary register A
101	$B_4B_3B_2B_1B_0$	Write $B_4B_3B_2B_1B_0$ into auxiliary register B
011	$USP_3P_2P_1$	Parallel Poll Enable (PPE) or Parallel Poll Disable (PPD) either in response to remote messages (PPC followed by PPE or PPD) or as a local Ipe message. (Enable if U=0, disable if U=1.)

Internal Counter

The internal counter is used to generate the T1 delay in the source handshake function, as defined in IEEE 488. By writing $0010F_3F_2F_1F_0$ into the Auxiliary Mode Register, the counter is preset to match a 6.0 MHz clock input, where $F_3F_2F_1F_0$ is the binary representation of N_F ($1 \leqslant N_F \leqslant 8$). Example: when $N_F = 6$ ($F_3F_2F_1F_0$=0110), a 2 μsec T_1 delay will be generated before each DAV asserted.

Equation 1 $T_1(\mu sec) = \frac{N_F}{3}(\mu sec) + tsync\ (\mu sec)$

tsync is a synchronization error $\leqslant 0.083$

If $T_1 = 2\mu s$ is not suitable, N_F may be set to a value other than 6. In this manner, data transfer rates may be programmed for a given system. In small systems, for example, where transfer rates exceeding GPIB specifications are required, one may set $N_F < 6$ to decrease T_1. The maximum T_1 is 2.75 μsec (N_F=8).

Since the 8293 devices are Tri-State Drivers (during non Parallel Poll operation) a faster transfer (Lower T_1) may be achieved by enabling the high speed transfer bit in Auxiliary Register B. (See Auxiliary Register B description.) In the aforementioned case, setting N_F=6 causes a T_1 delay of 2μs (from equation 1) for the first byte transmitted, and a delay of 500ns (see equation 2) for all subsequent bytes transmitted.

Equation 2 $T_1(\text{High Speed})\ \mu sec = \frac{N_F}{12}(\mu sec) + tsync\ (\mu sec)$

Thus, the shortest T_1 is achieved by setting N_F=1 (T1 = 1/12 + 0.083 = 0.167 μsec max.)

Auxiliary Commands

Auxiliary commands are issued by the 8291A device whenever the "execute auxiliary command" command is asserted. There are twelve different auxiliary commands, each with its own 4-bit code. These commands are identified in Table 3-5.

Auxiliary registers A and B are "hidden" 5-bit registers used to enable various iSBX 488 Multimodule Board features.

Auxiliary commands are executed by the 8291A whenever $0000C_3C_2C_1C_0$ is written into the Auxiliary Mode Register, where $C_3C_2C_1C_0$ is the 4-bit command code.

Table 3-5. Auxiliary Command Identification

4-Bit Code ($C_3C_2C_1C_0$)	Description
0000	Immediate Execute pon — This command is used to release the "initialize" state generated by either an external reset pulse or the Reset (0010) command. This command forces the 8291A to enter idle states (SIDS, AIDS, TIDS, LIDS, NPRS, LOCS, PPIS, DCIS and DTIS).
0010	Chip Reset (Initialize) — This command has the same effect as a pulse applied to the Reset pin. The 8291A is reset to an initialization state in which case the 8291A does not participate in any GPIB bus activity. The following registers are cleared: Interrupt Status, Auxiliary A and B and Serial Poll Mode. The following bits are cleared: Parallel Poll Flag and the EOI bit in the Address Status Register. The Internal Counter, N_F, is reset to 8, thus causing a T_1 delay of 2.75 μsec (see section on Internal Counter.) This state is released by an "immediate execute pon" command.
0011	Finish Handshake — This command finishes a handshake that was stopped because of a holdoff on RFD Command. (Refer to Auxiliary Register A.)
0100	Trigger — A "Group Execute Trigger" is forced by this command. It has the same effect as a GET command received from the Controller-In-Charge of the GPIB, but does not cause a GET interrupt.
0101, 1101	Clear/Set rtl — These commands correspond to the local rtl (return to local) message as defined by the IEEE-488. The 8291A will go into local mode when a Set rtl Auxiliary Command (0101) is received if local lockout is not in effect. The 8291A will exit local mode after receiving a Clear rtl Auxiliary Command (1101) if the 8291A is addressed to listen.
0110	Send EOI — The 8291A EOI line may be asserted with this command. The command causes EOI to go true with the next byte transmitted. The EOI line is then cleared upon completion of the handshake for that byte.
0111, 1111	Non-Valid/Valid Secondary Address or Command (VSCMD) — This auxiliary command is used in three separate instances where the host processor is responding to the 8291A: response to an APT interrupt, response to a CPT interrupt and a response to a GET or DEC interrupt. See section 3-6 for a description of these interrupts. When an APT interrupt occurs the host processor must read and interpret the secondary address (Mode 3 Addressing only) and respond to the 8291A with a valid (1111) or invalid (0111) command. Either response will release the RFD holdoff on the GPIB. Only a valid response will release the 8291A into a TADS or LADS state. When a CPT interrupt occurs and the Undefined Command Pass Through (bit BO in Auxiliary Register B) feature has been enabled, the host processor must read and interpret the undefined command and respond to the 8291A with a valid or invalid command to release the RFD holdoff. This operation is similar for a response to the GET or DEC interrupt when RFD holdoff is enabled by bit B4 in Auxiliary Register B.
1000	pon — This command puts the 8291A into the pon (power on) state and holds it there. It is similar to a Chip Reset except none of the Auxiliary Mode Registers are cleared. In this state, the 8291A does not participate in any bus activity. An Immediate Execute pon following this command releases the 8291A from the pon state and permits the device to participate in bus activity.
0001, 1001	Parallel Poll Flag (local "ist" message) — This command sets (1001) or clears (0001) the parallel poll flag. A "1" is sent over the assigned data line (PPR-Parallel Poll Response true) only if the parallel poll flag matches the sense bit from the lpe local message (or indirectly from the PPE message). For a more complete description of the Parallel Poll features and procedures refer to the section on Parallel Poll Protocol.

Auxiliary Register A

Auxiliary Register A is a "hidden" 5-bit register which is used to enable some of the 8291A features. Whenever a 100 $A_4A_3A_2A_1A_0$ byte is written into the Auxiliary Mode Register, it is loaded with the data $A_4A_3A_2A_1A_0$. Setting the respective bits to "1" enables the following features.

A_0 — RFD Holdoff on all Data: When the 8291A is listening (in LADS or LACS) RFD will be held false after a data byte has been accepted (DAC) to holdoff further data transfers across the GPIB. This feature is enabled for each data byte accepted until disabled. To continue the handshake a "finish handshake" auxiliary command is required.

A_1 — RFD Holdoff on End: This feature enables the holdoff on EOI or EOS (if enabled). However, no holdoff will be in effect on any other data bytes.

A_2 — End on EOS Received: Whenever the byte in the Data in Register matches the byte in the EOS Register, the END interrupt bit will be set in the Interrupt Status 1 Register.

A_3 — Output EOI on EOS Sent: Any occurrence of data in the Data Out Register matching the EOS Register causes the EOI line to be sent true along with the data.

A_4 — EOS Binary Compare: Setting this bit causes the EOS Register to function as a full 8-bit word. When it is not set, the EOS Register is a 7-bit word (for ASCII characters).

If A_0=A_1=1, a special "continuous Acceptor Handshake cycling" mode is enabled. This mode should be used only when the 8292 is the Controller-In-Charge of the GPIB. It provides a continuous cycling through the Acceptor Handshake diagram, requiring no local rdy messages from the microprocessor; the rdy local message is automatically generated when in ANRS. As such, the 8291A Acceptor Handshake serves as the 8292 controller Acceptor Handshake. Thus, the controller cycles through the Acceptor Handshake without delaying the data transfer in progress. When the tcs local message is executed, the 8291A should be taken out of the "continuous AH cycling" mode, the GPIB will hang up in ANRS, and a BI interrupt will be generated to indicate that control may be taken. A simpler procedure may be used when a "tcs on end of block" is executed; the 8291A may stay in "continuous AH cycling." Upon the end of a block (EOI or EOS received), a holdoff is generated, the GPIB halts in ANRS, and control may be taken.

Auxiliary Register B

Auxiliary Register B is a "hidden" 5-bit register which is used to enable some of the features of the 8291A. Whenever a 101 $B_4B_3B_2B_1B_0$ is written into the Auxiliary Mode Register, it is loaded with the data $B_4B_3B_2B_1B_0$. Setting the respective bits to "1" enables the following features:

B_0 — Enable Undefined Command Pass Through: This feature allows any commands not recognized by the 8291A to be handled in software. If enabled, this feature will cause the 8291A to holdoff the handshake when an undefined command is received. The microprocessor must then read the command from the Command Pass Through (CPT) Register and send the VSCMD auxiliary command. The handshake holdoff will be in effect until the VSCMD command is sent.

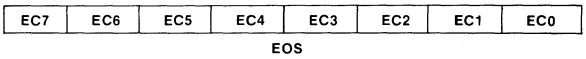
B_1 — Send EOI in SPAS: This bit enables EOI to be sent with the status byte; EOI is set true in Serial Poll Active State. Otherwise, EOI is sent false in SPAS.

B_2 — Enable High Speed Data Transfer: The data transfer rate is determined by the T_1 delay time generated in the Source Handshake function. When the "High Speed" feature is enabled, T_1 is determined by equation 1 (refer to Internal Counter in section 3-11) for the first byte transmitted after each true to false transition of ATN. For all subsequent bytes, T_1 is determined by equation 2 (which is equal to 1/4 of equation 1). Refer to the Internal Counter section for an explanation of T_1 duration as a function of B_2 and the clock frequency.

B_3 — Enable Active Low Interrupt: Setting this bit causes the polarity of the INT pin to be reversed.

B_4 — Enable RFD holdoff on GET or DEC: Setting this bit causes RFD to be held false until the VSCMD is written to the 8291A after any GET, SDC or DCL commands are received. This allows the host CPU to hold off the bus until it has completed a clear or trigger routine.

3-12. END OF SEQUENCE REGISTER



The end of sequence (EOS) register and its features offer an alternative to the "Send EOI" auxiliary command. A seven bit ASCII character or eight bit byte may be placed in the register to flag the end of a block or read. The type of EOS byte to be used is selected in Auxiliary Register A bit A_4.

If the 8291A is a listener, and the "End on EOS Received" is enabled with Auxiliary Register A bit A_2, then an END interrupt is generated in the Interrupt Status 1 Register whenever the byte in the Data-In Register matches the byte in the EOS Register.

If the 8291A is a talker, and the "Output EOI on EOS Sent" is enabled with Auxiliary Register A bit A_3, then the EOI line is sent true with the next byte whenever the contents of the Data Out Register match the EOS register.

Parallel Poll Protocol

Writing a $011USP_3P_2P_1$ into the Auxiliary Mode Register will enable (U=0) or disable (U=1) the 8291A for a parallel poll. When U=0, this command is the "lpe" (local poll enable) local message as defined in IEEE-488. The "S" bit is the sense in which the 8291A is enabled; only if the Parallel Poll Flag ("ist" local message) matches this bit will the Parallel Poll Response, PPR_N, be sent true. (Response = 1 only if "S" = "ist"). The bits $P_3P_2P_1$ specify which of the eight data lines PPR_N will be sent.

P3	P2	P1	PPR message	GPIB Response Line
0	0	0	PPR1	DIO0
0	0	1	PPR2	DIO1
0	1	0	PPR3	DIO2
0	1	1	PPR4	DIO3
1	0	0	PPR5	DIO4
1	0	1	PPR6	DIO5
1	1	0	PPR7	DIO6
1	1	1	PPR8	DIO7

Thus, once the 8291A has been configured for Parallel Poll, whenever it senses both EOI and ATN true, it will automatically compare its PP flag with the sense bit and send PPR_N true or false according to the comparison.

If a subset PP2* implementation is desired, the "lpe" and "ist" local messages are all that are needed. Typically, the 8291A can be configured for Parallel Poll immediately after initialization. During normal operation the microprocessor will set or clear the Parallel Poll flag (ist). The 8291A will be set up to give the proper response to IDY (EOI and ATN) without involving the microprocessor.

If a subset PP1* implementation is desired, the undefined command features of the 8291A must be used. In PP1, the 8291A is indirectly configured for Parallel Poll by the active controller on the GPIB. The sequence of the 8291A being enabled or disabled remotely is as follows:

1. The PPC message is received and is loaded into the Command Pass Through Register as an undefined command. A CPT Interrupt is sent to the microprocessor; the handshake is automatically held off.
2. The microprocessor reads the CPT Register and sends VSCMD to the 8291A, releasing the handshake.
3. Having received an undefined primary command, the 8291A is set up to receive an undefined secondary command (the PPE or PPD message). This message is also received into the CPT Register, the handshake is held off, and the CPT interrupt is generated.
4. The microprocessor reads the PPE or PPD message and writes this message command into the Auxiliary Mode Register (bit 7 should be cleared first). Finally, the microprocessor sends VSCMD and the handshake is released.

*This subset of the Parallel Poll Function is defined in the IEEE Std. 488.

3-13. 8292 PROGRAMMING

The 8292 controller contains eight read registers, and five write registers. Register bit identification and their address assignments are provided in Table 3-6.

Table 3-6. 8292 Registers & Port Addresses

READ REGISTERS								PORT # (HEX)	WRITE REGISTERS							
INTERRUPT STATUS									COMMAND FIELD							
SYC	ERR	SRQ	EV	X	IFCR	IBF	OBF	'X' B or 'X' F	1	1	1	OP	C	C	C	C
D_7							D_0									
ERROR FLAG*									INTERRUPT MASK**							
X	X	USER	X	X	$TOUT_3$	$TOUT_2$	$TOUT_1$	'X' A or 'X' E	1	SPI	TCI	SYC	OBFI	IBFI	0	SRQ
									D_7							D_0
CONTROLLER STATUS*									ERROR MASK**							
CSBS	CA	X	X	SYCS	IFC	REN	SRQ	'X' A or 'X' E	0	0	USER	0	0	$TOUT_3$	$TOUT_2$	$TOUT_1$
GPIB BUS STATUS*									EVENT COUNTER*							
REN	DAV	EOI	X	SYC	IFC	ATN	SRQ	'X' A or 'X' E	D	D	D	D	D	D	D	D
EVENT COUNTER STATUS*									TIME OUT*							
D	D	D	D	D	D	D	D	'X' A or 'X' E	D	D	D	D	D	D	D	D
TIME OUT STATUS*																
D	D	D	D	D	D	D	D	'X' A or 'X' E								

NOTES:

*THIS REGISTER IS ACCESSED (READ OR WRITTEN) BY FIRST WRITING A SPECIFIC BYTE TO THE COMMAND FIELD REGISTER.
**THIS REGISTER MAY ALSO BE ACCESSED FOR A READ OPERATION BY FIRST WRITING A SPECIFIC BYTE TO THE COMMAND FIELD REGISTER.

3-14. 8292 REGISTERS

The 8292 registers may be used in numerous ways for controlling the GPIB, monitoring activity on the GPIB, reporting timeout bus errors, and counting bus data transfers on the bus. The registers are each described in the following paragraphs.

3-15. INTERRUPT STATUS REGISTER

INTERRUPT STATUS

SYC	ERR	SRQ	EV	X	IFCR	IBF	OBF
D7							D0

The 8292 can be configured to interrupt the microprocessor for one of several conditions. Upon receipt of the interrupt the microprocessor must read the 8292 interrupt status register to determine which event caused the interrupt before the appropriate subroutine can be performed. With the exception of OBF and IBF, these interrupts are enabled or disabled by the SPI interrupt mask. OBF and IBF have their own bits (OBFI and IBFI) in the interrupt mask register. Each bit of the Interrupt Status Register is described below.

DO OBF=1 Output Buffer Full. A byte is waiting to be read by the microprocessor. This flag is cleared when the data byte is read. The byte to be read originates from any of the registers: Error Flag, Controller Status, GPIB Bus Status, Event Counter Status, Time Out Status, Interrupt Mask and Error Mask.

D1 IBF=1 Input Buffer Full. A byte has been written by the microprocessor to the 8292. If another byte is written to the 8292 before this flag clears, data will be lost. IBF is cleared when the 8292 accepts the data byte. The destination of the byte is to any of the registers: Command Field, Inter rupt Mask, Error Mask, Event Counter and Time Out.

D2 IFCR=1 Interface Clear Received. The GPIB system controller has set IFC. The 8292 has become idle and is no longer in charge of the bus. The flag is cleared when the IACK command is issued. This situation assumes that the 8292 is not the system controller of the GPIB.

D4 EV=1 Event Counter Interrupt. The requested number of blocks or data bytes has been transferred. The EV interrupt flag is cleared by the IACK command.

D5 SRQ=1 Service Request. Notifies the 8292 that a service request (SRQ) message has been received. It is cleared by the IACK command.

D6 ERR=1 Error occurred. The type of error can be determined by reading the Error Flag Register. This interrupt flag is cleared by the IACK command.

D7 SYC=1 System Controller Jumper Change. Notifies the processor that the state of the system controller jumper (E8-E16) has changed. The actual state is contained in the GPIB Status Register. This flag is cleared by the IACK command.

NOTE

The IACK command is a utility type of command written to the Command Field Register for clearing the status bits IFCR, EV, SRQ, ERR and SYC. See section on IACK Utility Command.

3-16. INTERRUPT MASK REGISTER

INTERRUPT MASK

1	SPI	TCI	SYC	OBFI	IBFI	0	SRQ
D7							D0

The Interrupt Mask Register is written to directly and is used to enable or disable (mask out) the interrupt pins OBFI, $\overline{\text{IBFI}}$, TCI and SPI. It is also used to enable or disable two distinct conditions SRQ and SYC which may generate a SPI interrupt. The desired interrupt or condition may be enabled by writing a logic "1" to the appropriate register bit. Note that the Interrupt Status Register is not affected by the masking or unmasking of the Interrupt Mask Register. The Interrupt Mask Register may also be read by first writing a utility command RINM, (E5 Hex), to the Command Field Register. When the register is read bits D1 and D7 are undefined.

DO SRQ=1 Enable interrupts on SRQ received.

D2 $\overline{\text{IBFI}}$=1 Enable $\overline{\text{IBF}}$ interrupt pin for input buffer empty. Note that this interrupt is true when the $\overline{\text{IBF}}$ bit in the Interrupt Status Register is False (0), that is, the 8292 is ready to accept of another byte from the host processor.

D3 OBFI=1 Enable OBF interrupt pin for output buffer full. Note that this interrupt is true when the OBF bit in the Interrupt Status Register is true (1). That is, a byte is waiting to be read from the host processor.

D4 SYC=1 Enable interrupt on a change of state of the system controller jumper E8-E16.

D5 TCI=1 Enable TCI interrupt pin for tasks completed. Certain commands executed by the 8292 return a TCI. These commands are discussed in section 3-26 and 3-27.

D6 SPI=1 Enable SPI interrupt pin for special occuring events. These events (IFCR, EV, SRQ, ERR, SYC) are discussed in the Interrupt Status Register.

NOTE

The remaining conditions for generating a SPI interrupt are IFCR, EV and ERR. The IFCR condition cannot be masked (always enabled). The EV condition is enabled by first writing an operation command, GSEC (F4 Hex), to the Command Field Register discussed in section 3-15. The ERR condition is masked by writing to the Error Mask Register (discussed later in this section).

3-17. CONTROLLER STATUS REGISTER

CONTROLLER STATUS

CSBS	CA	X	X	SYCS	IFC	REN	SRQ
D_7							D_0

The Controller Status Register is used to determine the controller state of the 8292 and to monitor four 8292 lines. This register is read by first writing a utility command, RCST (E6 Hex) to the Command Field Register.

D0 SRQ=1 SRQ Line is active low

D1 REN=1 REN Line is active low

D2 IFC=1 IFC Line is active low

D3 SYCS=1 System Controller Line is high (i.e., E8-E16 not jumpered). The 8292 is configured for a System Controller.

D6 CA=1 8292 is the Controller-In-Charge and is one of the three states, CACS or CAWS or CSWS.

D7 CSBS=1 8292 is the Controller-In-Charge and is in the state, CSBS.

NOTE

If both CA and CSBS are not set the 8292 will be in the CIDS state. These states are defined by the IEEE std. 488.

3-18. GPIB STATUS REGISTER

GPIB BUS STATUS

REN	DAV	EOI	X	SYC	IFC	ANTI	SRQ
D_7							D_0

The GPIB Status Register is used to monitor the state of the bus lines. Note that lines SRQ, IFC, REN and SYC are duplicated in the Controller Status Register. This register is read by first writing a utility command, RBST (E7 Hex) to the Command Field Register.

D0 SRQ=1 SRQ Line is active low

D1 ATN=1 ATN Line is active low

D2 IFC=1 IFC Line is active low

D3 SYC=1 SYC Line is active high

D5 EOI=1 EOI Line is active low

D6 DAV=1 DAV Line is active low

D7 REN=1 REN Line is active low

3-19. EVENT COUNTER REGISTER

EVENT COUNTER

D_7	D	D	D	D	D	D	D_0

The Event Counter Register contains the initial value loaded into the event counter. The counter decrements the count on every high to low signal transitions on pin 39 (COUNT) of the 8292. It may be connected to the EOI (default jumpered E18-E19) or NDAC (E17-E18) buffered line to count blocks or bytes respectively during controller standby state. The minimum count period is ***approximately*** 7.5 microseconds, the minimum high pulse width is ***approximately*** 500ns and the minimum low pulse width is ***approximately*** 3.0 μsec. This register cannot be read and is written to by first writing the utility command, WEVC (E2 Hex), to the Command Field Register. A value of 00 Hex = 256 counts, FF_H = 255, FE_H = 254,... etc...and 01_H = 1.

3-20. EVENT COUNTER STATUS REGISTER

EVENT COUNTER STATUS

D_7	D	D	D	D	D	D	D_0

This register contains the current value in the event counter. The event counter decrements from the initial value stored in the Event Counter Register to zero and then generates an Event Counter Interrupt.

This register is read by first writing the utility command, REVC (E3 Hex), to the Command Field Register. See Figure 3-1 showing the block diagram of the Event Counter Function.

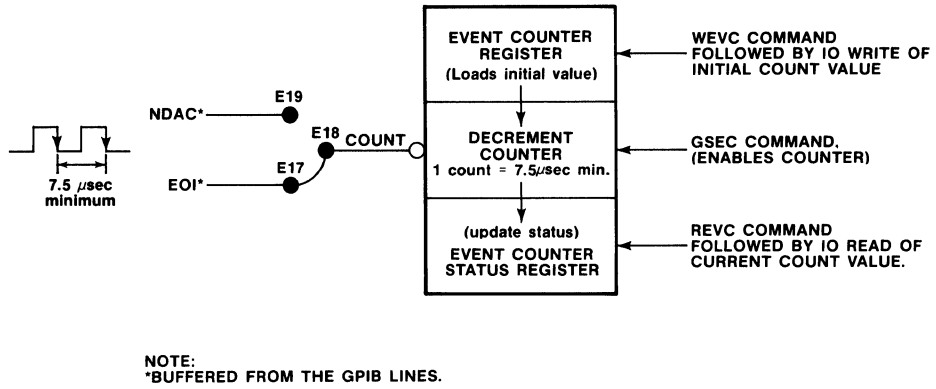


Figure 3-1. 8292 Event Counter Block Diagram

3-21. TIME OUT REGISTER

TIME OUT*							
D_7	D	D	D	D	D	D	D_0

The Time Out Register is used to store the count value used for the time out error function. See the individual timeouts (TOUT1, 2, 3) to determine the units of this counter.

This register cannot be read and it is written to by first writing the utility command WTOUT (E1 Hex) to the Command Field Register. A vlaue of 00HEX = 256 counts, FF_H = 255, FE_H = 254....etc...01_H = 1.

3-22. TIME OUT STATUS REGISTER

TIME OUT STATUS							
D_7	D	D	D	D	D	D	D_0

This register contains the current value in the time out counter. The time out counter decrements from the original value stored in the Time Out Register. When zero is reached, the appropriate error interrupt is generated. If the register is read while none of the time out functions are active, the register will contain the value reached during the previous active Time Out function. This register is read from by first writing the Utility Command RTOUT (E9 Hex) to the Command Field Register. See Figure 3-2 for a block diagram of the Time Out Function.

3-23. ERROR FLAG REGISTER

ERROR FLAG							
X	X	USER	X	X	$TOUT_3$	$TOUT_2$	$TOUT_1$
D_7							D_0

The Error Flag Register shows the status of the three TIME OUT errors and the USER error. Each of these flags can be enabled by writing a 1 to the corresponding bit in the Error Mask Register. This Register cannot be written and it should be read by first writing the RERF (E4 Hex) command to the Command Field Register after the ERR bit is set in the Interrupt Status Register.

TOUT1. Time out Error 1 is used by the GPIB Multimodule board when attempting to receive control of the Bus, becoming CIC (Controller in Charge). This error occurs when the controller that is passing control of the bus has not released the ATN line for the time period specified in the Time Out Register. Each count in the Time Out Register is approximately 4.5 milliseconds. The count is started when the TCNTR command (FAH) is issued to the Command Field Register. When this command is written, the content of the Time Out Register is transferred to the Time Out Status Register and the count-down begins. When the count equals zero the TOUT1 bit and the ERR bit in the Interrupt Status Register are set. After flagging the error, the 8292 will remain in a loop trying to take control until the current controller stops sending ATN or a new command is issued by the microprocessor. If a new command is issued, the 8292 will execute the command and return to the loop checking the ATN line. If a RSTI command is issued the 8292 will stop checking ATN and clear the ERR and TOUT bits.

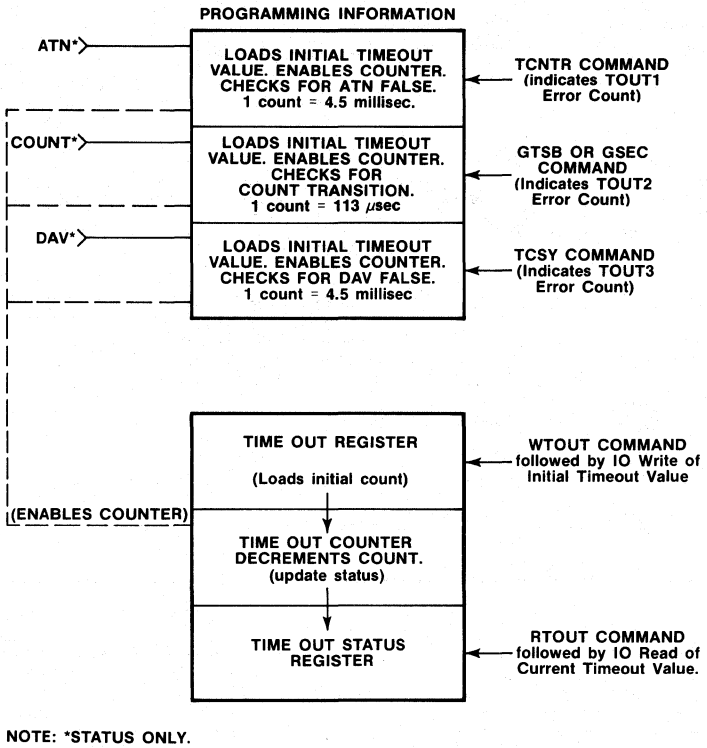


Figure 3-2. 8292 Timeout Counter Block Diagram

TOUT2. The TOUT2 Error function checks for bus activity, i.e., EOI (factory jumpered) or NDAC high to low transitions at the COUNT input to the 8292 after the GTSB or GSEC commands are written to the Command Field Register. The count in the Time Out Status is decremented until there is either an EOI transition or a count of 00H detected. IF EOI occurs before a count of zero is reached, the count is reinitialized and begins counting down again until the next EOI is encountered. If a count of zero is reached before NDAC or EOI occurs, the TOUT2 and ERR bits will be set. Thereafter, the number of counts will be 256 until the next EOI transition. Each count in the Time Out Register is approximately 113μsec. Thus, for a count of 20Hex (32Decimal) the Time Out will wait 3.6 milliseconds for EOI to become active on the bus. IF NDAC is jumpered to COUNT instead of EOI, the byte transfer rate for a count of 20H must be at least 1/3.6 milliseconds or 275 bytes per second.

TOUT3. Time Out Error 3 occurs when the TCSY Command (FD Hex) is written to the 8292 and the 8292 has not succeeded in taking control because DAV is held low (Active) longer than the value in the Time Out Register. The TOUT3 flag will be set when the count in the Time Out Status Register is decremented to 00H. Each count in the Time Out Register is approximately 4.5 milliseconds. The 8292 will continue checking DAV until it becomes false, or a new command is received. After executing the last command, the 8292 will return to checking DAV. A RSTI command will stop the 8292 from checking ATN and will clear the ERR and TOUT bits.

USER. User error occurs when the host CPU requests the 8292 to assert IFC or REN and the 8292 is not the system controller.

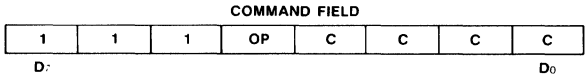
3-24. ERROR MASK REGISTER

ERROR MASK*

0	0	USER	0	0	$TOUT_3$	$TOUT_2$	$TOUT_1$
D_7							D_0

The Error Mask Register is used to mask the ERR interrupt from a particular type of error. Each type of error interrupt is enabled by setting the corresponding bit in the Error Mask Register. This register can be written to directly (bits D_3, D_4, D_6, and D_7 must be low [0]. To read this register, first write the REFM command (EA Hex) to the Command Field Register. When reading the Error Mask Register bits D_3, D_4, D_6 and D_7 are undefined.

3-25. COMMAND FIELD REGISTER



There are two categories of commands distinguished by the OP (operation) bit (D4). The first category consists of the operation command (OP=1). These commands initiate some activity on the GPIB. The second category is the utility commands (OP=0). These commands are used to aid communication between the host board processor and the 8292.

3-26. 8292 OPERATION COMMANDS

This category contains 14 commands (identified by hex bytes F0, F1, F2, F3, F4, F5, F6, F7, F8, F9, FA, FC, FD, and FE) for either resetting the 8292, enabling/disabling the counter interrupts or initiating controller actions on the GPIB. These commands are discussed below in numerical order.

SPCNI — Stop Counter Interrupts
Command = 0F0H)

The 8292 will not generate an EV interrupt when the counter reaches 0. Note that the counter will continue counting. TCI will not be set.

GIDL — Go to Idle (Command = 0F1H)

If the 8292 is not the Controller-In-Charge and in CACS, it will ignore this command and not set TCI. Otherwise it sets ATN FALSE, enters CIDS, and sets TCI TRUE. This command is used after the TCT command has been sent to and accepted by another GPIB controller, during a pass control operation.

RST — Reset (Command 0F2H)

This command has the same effect as an external reset applied to the 8292 (pin #4). The resulting actions are:

a. 8092 interrupt lines SPI, TCI, OBFI and $\overline{\text{IFBI}}$ will go high (TRUE) for approximately 17.5 μsec before returning low. The host processor should mask any interrupt inputs during this period.

b. These registers will be cleared: Interrupt Status, Interrupt Mask, Error Mask, Time Out, Event Counter and Error Flag.

c. If the 8292 is the System Controller (SYC jumper is open), then IFC will be set TRUE for approximately 100 μsec and the 8292 will be in charge of the bus (CACS state). If the 8292 is not the System Controller, it will enter the Idle state (CIDS state).

RSTI — Reset Interrupts (Command = 0F3H)

This command clears all pending interrupts and error flags. The 8292 will stop waiting for actions to occur (e.g., waiting for ATN to go FALSE in a TCNTR command or waiting for the proper handshake state in a TCSY command). TCI will not be set.

GSEC — Go to Standby and Enable Counting
(Command = 0F4H)

The 8292 COUNT input is jumpered to the buffered EOI line but may be jumpered to the buffered NDAC. When the counter reaches zero, it sets EV (and SPI if enabled) in the Interrupt Status Register and will set EV every 256 counts thereafter. Note that there is a potential loss of count information if the CPU does not respond to the EV before another EV has occurred. TCI will be set at the end of the command.

EXPP — Execute Parallel Poll
(Command = 0F5H)

If the 8292 is not Controller-In-Charge, it will ignore this command. TCI will not be set. If it is the Controller-In-Charger then it sets IDY (EOI & ATN) TRUE. If the 8291A is configured as a listener, it will capture the Parallel Poll Response byte in its data register. Since TCI is not generated, the CPU must detect the BI (Byte In) from the 8291A which serves as a task complete indicator.

GTSB — Go To Standby (Command = 0F6H)

If the 8292 is not the Controller-In-Charge, it will ignore this command and not set TCI TRUE. Otherwise, it enters the Controller Standby State (CSBS), sets ATN FALSE and TCI TRUE.

If a data transfer does not start within the specified Time-Out, the 8292 sets TOUT2 TRUE in the Error Flag Register and sets SPI (if enabled). The controller continues waiting for a new command. The CPU must decide to wait longer or to regain control and take corrective action.

SLOC — Set Interface to Local Mode
(Command = 0F7H)

If the 8292 is the System Controller, it will set REN FALSE for at least 100μs and TCI TRUE. Otherwise, it only sets the User Error Flag.

SREM — Set Interface to Remote Control
(Command = 0F8H)

If the 8292 is the System Controller, it will set REN and TCI TRUE. Otherwise it only sets the User Error Flag.

ABORT — Abort all operations and Clear Interface (Command = 0F9H)

If the 8292 is not the System Controller this command will be ignored and the USER ERROR flag will be set in the Error Flag Register. No TCI will occur.

If the 8292 is the System Controller then IFC is set TRUE for 100 μsec minimum, and the 8292 becomes the Controller-In-Charge and asserts ATN. TCI will be set, only if the 8292 was not the Controller-In-Charge.

TCNTR — Take Control (Command = 0FAH)

If the 8292 is not in CIDS it ignores this command and does not set TCI. Otherwise it waits for the current Controller-In-Charge to set ATN FALSE. If this does not occur within the specified Time Out, the 8292 sets TOUT1 in the Error Flag Register and sets SPI (if enabled). It will not proceed until ATN goes false or it receives an RSTI command. Note that the Controller-In-Charge must previously have sent this controller (via the 8291A's command pass through register) a TCT message. When ATN goes FALSE the 8292 sets ATN and TCI TRUE and enters CACS.

TCAS — Take Control Asynchronously (Command = 0FCH)

If the 8292 is not in Standby, it ignores this command and does not set TCI. Otherwise, it arbitrarily sets ATN TRUE and TCI TRUE. Note that this action may cause devices on the bus to lose a data byte or cause them to interpret a data byte as a command byte. Both actions can result in anomalous behavior. TCAS should be used cautiously. If TCAS fails, the System Controller may have to issue an IFC and/or DCL.

TCSY — Take Control Synchronously (Command = 0FDH)

If the 8292 is not in Standby, it ignores this command and does not set TCI. Otherwise, it waits for the proper handshake state (DAV line high) and sets ATN TRUE. The 8292 will set TOUT3 if the handshake never assumes the correct state and will remain in this loop waiting until the handshake is proper or a RSTI command is issued. If the 8292 successfully takes control, it sets TCI TRUE.

This is the typical way to regain control at the end of a Send, Receive, Transfer or Serial Poll System Command. If TCSY is not successful, then the controller must try TCAS. Refer to the description of the continuous acceptor handshake mode for the 8291A, (section 3-11; Auxiliary Register A.)

STCNI — Start Counter Interrupts (Command = 0FEH)

Enables the EV Counter Interrupt. TCI will not be set. Note that the counter must be enabled by a GSEC command.

3-27. 8292 UTILITY COMMANDS

This category contains 10 commands (identified by Hex bytes E1, E2, E3, E4, E5, E6, E7, E9, EA, and the IACK command.) These commands are used to read from or write to the 8292 registers that are not directly accessible.

NOTE

The Registers that are directly accessible are the Command Field (write), Interrupt Mask (write), Error Mask (write), and the Interrupt Status (read).

For writing into registers the general sequence is:

1. Check for IBF = 0 in Interrupt Status Register.
2. Write the appropriate utility command to the 8292 Command Field Register (port XB or XF).
3. Write the desired register value to the 8292 (port XA or XE) with no other writes intervening.

For reading a register the general sequence is:

1. Wait for IBF = 0 in Interrupt Status Register.
2. Write the appropriate utility command to the 8292.
3. Wait for a TCI (Task Complete Interrupt).
4. Read the value of the accessed register from the 8292 register (port XA or XE).

WTOUT — Write to Time Out Register (Command = 0E1H)

The byte written following this command will be loaded into the Time Out Register. Because the register is 8 bits, the maximum count is 256 time increments. When the command is complete $\overline{IBF}$ will be set to a "0" and will cause an $\overline{IBF}$I interrupt if masked on.

WEVC — Write to Event Counter (Command = 0E2H)

Following this command the byte written will be loaded into the Event Counter Register for event counting. The counter is decremented on a high to low transition of the COUNT input. The counter is an 8 bit register and therefore can count up to 256 maximum.

When the 8292 has completed the command, IBF will become a "0" and will cause an $\overline{IBFI}$ interrupt if masked.

REVC — Read Event Counter Status
(Command = 0E3H)

This command enables the content of the Event Counter Status Register to be read. The 8292 then sets the TCI interrupt (if not masked). The CPU may then read the value from the 8292.

RERF — Read Error Flag Register
(Command 0E4H)

This command enables the content of the Error Flag Register to be read. TCI will be set.

RINM — Read Interrupt Mask Register
(Command = 0E5H)

This command enables the content of the Interrupt Mask Register to be read. TCI will be set.

RCST — Read Controller Status Register
(Command = 0E6H)

This command enables the content of the Controller Status Register to be read. TCI will be set.

RBST — Read Bus Status Register
(Command 0E7H)

This command enables the status of the GPIB management lines to be read. TCI will be set.

RTOUT — Read Time Out Status Register
(Command = 0E9H)

This command enables the content of the Time Out Status Register to be read. TCI will be set.

If this register is read while a time-out function is in process the value will be the current time-out count. If it is read after a time-out, the content will be zero. If it is read when no time-out function is in process the content will contain the value reached in the previous time-out function.

RERM — Read Error Mask Register
(Command 0EAH)

This command enables the content of the Error Mask Register to be read. TCI will be set.

IACK — Interrupt Acknowledge

SYC	ERR	SRQ	EV	1	IFCR	1	1
D7	D6	D5	D4	D3	D2	D1	D0

This command is used to acknowledge and reset any condition of the five SPI interrupts: SYC, ERR, SRQ, EV, and IFCR. Each bit (D_2, D_4-D_7) is an individual acknowledgement to the corresponding bit in the Interrupt Status Register. The command clears SPI. SPI will be set again if not all interrupts were acknowledged.

If a pending User or Time Out Error occurs while an IACK command was written, the ERR bit (D6) will be set and thus activate the SPI interrupt (if enabled). The TCI interrupt will also be set, indicating that the Error Flag Register may be read directly (Read XA or XE) without having to issue an RERF command.

An example (Figure 3-3) shows how to write and read from registers that do not require utility commands. The example is based on the status polling method.

This assembly language program shows how to enable the SPI interrupt and SRQ interrupt condition.

```
IMR     EQU     80H     Intrpt Mask Register, Base Content
SRQ     EQU     01H     Intrpt Mask Register, SRQ Bit D0
SPI     EQU     40H     Intrpt Mask Register, SPI Bit D6
IMP     EQU     XAH     Intrpt Mask Port No. XA or XE
ISP     EQU     XBH     Intrpt Status Port No. XB or XF
IBF     EQU     02H     Intrpt Status Register, IBF Bit D1
BACK:   IN      ISP     Check IBF = 0
        ANI     IBF     If true continue
        JNZ     BACK    If false, Jump Back.
CONT:   MVI     A, IMR
        ORI     SRQ     Unmask SRQ & SPI bits
        ORI     SPI
        OUT     IMP
```

Figure 3-3. Register Read Without Utility Command

An example (Figure 3-4) shows how to read from a register requiring a utility command. This particular example shows how to read the GPIB Bus Status Register and presumes that the TCI interrupt has been previously unmasked.

```
RBST    EQU     0E7H    "Read Bus Status" Utility
                        Command
CFP     EQU     XBH     Command Field Port No.
                        XB or XF
BSP     EQU     XAH     Bus Status Port No. XA or
                        XE
GPP     EQU     X8H     8282 General Purpose Port
                        No.
*TCIS   EQU     80H     8282 Port, TCI Line Status
BACK:   IN      ISP     Check IBF = 0
        ANI     IBF     If true, continue
        JNZ     BACK    If false, Jump Back

CONT:   MVI     A, RBST Write RBST utility com-
                        mand
        OUT     CFP     to command Field Register.
WAITL:  IN      GPP     Wait for TCI Line
        ANI     TCIS    to transition high
        JNZ     WAITL   to low. (i.e., command
                        accepted)
WAITH:  IN      GPP     Wait for TCI Line
        ANI     TCIS    to transition low to
        JZ      WAITH   high (i.e., task complete)
        IN      BSP     Host Processor may read
                        the bus Status Register
```

*TCI is default jumpered to line D7 of the 8282 port (X8H)

Figure 3-4. Read GPIB Bus Status Register

Table 3-7. Summary of 8292 Operation and Utility Commands

Code	Name	TCI[2]	IBFI
F0	SPCNI - Stop Counter Interrupts	No	Yes
F1	GIDL - Go To Idle	Yes	Yes
F2	RST - Reset	No	Yes
F3	RSTI - Reset Interrupts	No	Yes
F4	GSEC - Go To Standby and Enable Counting	No	Yes
F5	EXPP - Execute Parallel Poll	No	Yes
F6	GTSB - Go To Standby	Yes	Yes
F7	SLOC - Set Interface to Local Mode	Yes	Yes
F8	SREM - Set Interface to Remote Control	Yes	Yes
F9	ABORT - Abort all operations & Clear Interface	Note[1]	Yes
FA	TCNTR - Take Control	Yes	Yes
FC	TCAS - Take Control Asynchronously	Yes	Yes
FD	TCSY - Take Control Synchronously	Yes	Yes
FE	STCNI - Start Counter Interrupts	No	Yes
—	IACK - Interrupt Acknowledge	No	No
E1	WTOUT - Write to Time Out Register	No	Yes
E2	WEVC - Write To Event Counter	No	Yes
E3	REVC - Read Event Counter Status	Yes	Yes
E4	RERF - Read Error Flag Register	Yes	Yes
E5	RINM - Read Interrupt Mask Register	Yes	Yes
E6	RCST - Read Controller Status Register	Yes	Yes
E7	RBST - Read Bus Status Register	Yes	Yes
E9	RTOUT - Read Time Out Status Register	Yes	Yes
EA	RERM - Read Error Mask Register	Yes	Yes

NOTE:

1 If the 8292 is not the System Controller, no TCI will be generated. If the 8292 is the System Controller and not Controller-In-Charge, TCI will be generated after IFC has been asserted for 100 μsec.

2 TCI will reset approximately 1.2 μsec after a command has been written (trailing eldge of IOWR/). TCI will remain low for approximately 100 μsec to 525 μsec, depending on the command written, before it is set.

3-28. 8292 INTERRUPTS

The 8292 controller can issue four different interrupts:

OBFI	Output buffer full interrupt.
$\overline{\text{IBFI}}$	Input buffer NOT full interrupt.
TCI	Task completed interrupt
SPI	Special interrupt

The SPI interrupt line is connected to the MINTR1 line on the iSBX bus. The other three interrupts are jumper connected as follows:

Interrupt	Jumper Pair	Destination
OBFI	E22 - E25	P1-30 (OPT0)
$\overline{\text{IBFI}}$	E21 - E26	P1-28 (OPT1)
TCI	E20 - E27	DI7 on 8282 Latch
SPI	—	P1-12 (MINTR1)

The OBFI output is asserted whenever the 8292 output buffer is full, waiting to be read. Similarly, the $\overline{\text{IBFI}}$ output is asserted whenever the input buffer is empty, waiting for the next data byte to be written. The SPI output becomes active when the following special events occur:

1. System Controller Jumper Change.
2. Event Counter decrements to zero.
3. Service Request received.
4. Time Out or User error occurred.
5. Interface clear received

See Figure 3-5 for a representation of the SPI interrupt logic.

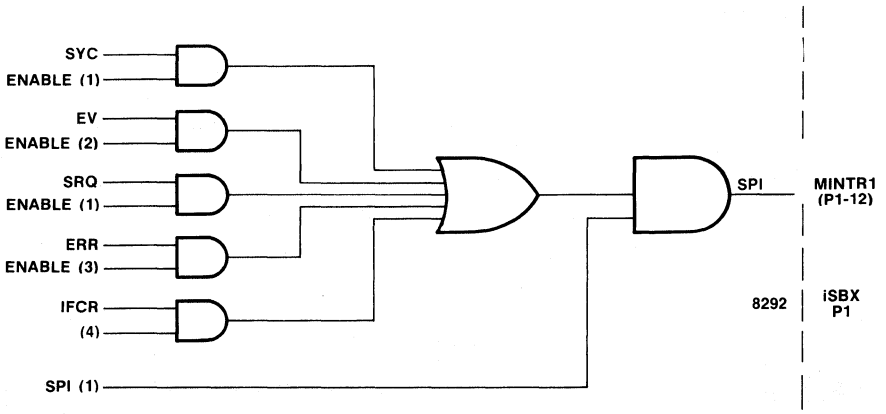
3-29. 8282 GENERAL PURPOSE PORT

The 8282 is configured as a transparent latch, and the contents may be read by an I/O Read to the port hex address X8, X9, XC OR XD. Bits D0 through D6 represent the status of jumpers E1-E9, E2-E10, E3-E11, E4-E12, E5-E13, E6-E14, and E7-E15 respectively. Bit D7 represents the status of the 8292 TCI output (default jumpered E20-E27). Although the jumper inputs to bits D0-D6 are general purpose it is a software convenience to assign specific meanings to these bits. That is, bits D0-D4 may represent the 5-bit GPIB address while bits D5 and D6 may represent the listen-only or talk-only mode for the iSBX 488 Multimodule Board.

3-30. BOARD POWER ON/RESET

After power-on or after RESET/ at P1-5 is asserted, the following events occur:

- GPIB Signal Lines. These lines are buffered by the 8293 transceivers U4 and U6.
  a. Receiver Mode. The following GPIB lines will be in a receiver mode.
     DIO1-DIO8
     DAV
     ATN (momentarily)
     EOI (momentarily)
     IFC (If 8292 ≠ System Controller)
     REN (If 8292 ≠ System Controller)



NOTES:
1. This condition is enabled through the use of the Interrupt Mask Register.
2. This condition is enabled through the use of the FSEC (F4H) or disabled by the STCNI (FEH) Operation Command.
3. This condition is enabled through the use of the Error Mask Register.
4. This condition is always enabled.

Figure 3-5. SPI Interrupt Logic

b. Driver Mode (High Output State) The following GPIB lines will be drivers:

NDAC (Open Collector off state)

NRFD (Open Collector off state)

SRQ (Momentarily)

IFC (If 8292 = System Controller)

REN (If 8292 = System Controller)

- 8292

The following 8292 events occur:

NOTE

The following events also take place following a RST command.

a. Interrupt Outputs

TCI, SPI, OBFI & IBFI outputs will become High during Reset active and go low immediately after Reset is inactive.

b. Registers Cleared:

Interrupt Status

Interrupt Mask

Error Flag

Error Mask

Time Out

Event Counter (counter disabled)

c. If 8292 = System Controller (i.e., Jumper off) The 8292 ABORT command executes; and the 8292 becomes Controller-In-Charge (Controller Active State).

d. ATN transitions from receiver to driver mode at the GPIB and is asserted active low. EOI is reconfigured from receiver to transceiver mode. SRQ is reconfigured from driver to receiver mode.

e. If 8292 $\neq$ System Controller

8292 remains in Controller Idle State. The ATN, EOI, & SRQ will not change from their previous driver or receiver mode.

- 8291A

The following 8291A events occur:

NOTE

The following events also take place following an 8291A Auxiliary Reset Command.

A "pon" local message as defined by IEEE-488 is held true until the initialization state is released.

The Interrupt Status Registers are cleared (not Interrupt Mask Registers).

Auxiliary Registers A and B are cleared.

The Serial Poll Mode Register is cleared.

The Parallel Poll Flag is cleared.

The EOI bit in the Address Status Register is cleared.

N_F in the Internal Counter is set to 8. This setting causes the longest possible T_1 delay to be generated in the Source Handshake (2.75 μsec).

The rdy local message is sent. (I.e., 8292 is ready to accept a command from the host processor.)

The initialization state is released by an "immediate execute pon" command (00H written into the Auxiliary Command Register).

Software Command Reset

The 8291A and 8292 may be independently reset by program control. The RST (F2H) command written to the 8292 Command Field Register forces the reset events discussed earlier. The auxiliary command chip reset (02H) written to the 8291A forces events discussed earlier.

3-31. BOARD INITIALIZATION

The initialization process is application dependent (i.e., whether or not the 8292 is configured as a System Controller.

CAUTION

If the 8291A is to be used solely without the 8292 central functions, the 8292 should be configured as a non-system controller (i.e., install jumper E8-E16).

- 8292

If the iSBX 488 board is to have some controller capability (subsets C1-C28, of Table 1-1 in Chapter 1) it is necessary to enable the TCI interrupt via the 8292 Interrupt Mask Register. The TCI interrupt or its status may be used for interrupt driven or polling routines respectively. The TCI status may be read from the 8282 general purpose port.

- 8291A

  Set the initial conditions for the appropriate subsets desired (see Table 1-1 for subset list), by writing into the Interrupt Mask, Serial Poll Mode, Address Mode, Address 0/1, and EOS Registers. The Auxiliary Mode Registers should be written to configure/unconfigure the parallel poll, preset the internal counter for setting time T1, and to enable the desired functions in Auxiliary Registers A & B. Next, send the "immediate execute pon" command to the Auxiliary Mode Register to release the 8291A from the "initialize" state.

  If the iSBX 488 board is configured as a System Controller, the 8291A will not need its own talk or listen address. The Address Mode Register should be set up with Talk Only or Listen Only capability.

  If the iSBX 488 board is configured as a non-System Controller, the 8291A will have to be set up with a Talk and/or Listen Address in the Address 0/1 Register. This address may be set and dynamically changed by using general purpose jumpers as described in section 3-29.

  Figures 3-6 and 3-7 depict examples of a minimal initialization sequence program for a system controller and non-system controller.

3-32. SOFTWARE DRIVERS

Following initialization, the host board program may contain individual drivers which support GPIB functions (Talker, Listener, Device Trigger, Parallel Enable, LOCAL, Service Request, Interface Clear, etc.) to the capability level (subset) desired. The Talker Routine may, for example, call a module to send a block of data to several GPIB devices. This module may be called SEND and is described in Appendix A as a PL/M 80 program.

```
SEQ         SOURCE      STATEMENT
  1         NAME        SINIT
  2 ;
  3 ;       THIS PROGRAM INITIALIZES THE iSBX 488 BOARD AS A SYSTEM
  4 ;       CONTROLLER. BASE PORT ADDRESS (20H) IS logical OR'ed WITH THE
  5 ;       8291A, 8292 and 8282 PORT ADDRESS.
  6         PUBLIC      SINIT
  7         CSEG
  8 SINIT:  MVI         A, 0A0H
  9         OUT         2AH           ;ENABLE TCI, 8292
 10 BACK:   IN          OFBH          ;WAIT FOR 8292 IBF LOW
 11         ANI         02            ;BEFORE CONTINUING
 12         JNZ         BACK
 13         XRA         A             ;CLEAR A
 14         OUT         21H           ;8291A CLEAR INTERRUPT MASK 1,
 15         OUT         22H           ;CLEAR INTERRUPT MASK 2
 16         MVI         A,80H         ;SET UP
 17         OUT         24H           ;TALK ONLY MODE, 8291A
 18         MVI         A,06
 19         OUT         26H           ;DISABLE ADDR 0, 8291A
 20         MVI         A,0E0H
 21         OUT         26H           ;DISABLE ADDR 1, 8291A
 22         MVI         A,26H         ;SET NF TO 6, T1 TO 2 USEC
 23         OUT         25H           ;SET INTERNAL COUNTER, 8291A
 24         XRA         A             ;CLEAR MESSAGE
 25         OUT         25H           ;PON MESSAGE, 8291A
 26         RET
 27         END
```

Figure 3-6. System Controller

```
SEQ         SOURCE      STATEMENT
  1         NAME        CINIT
  2 ;
  3 ;       THIS PROGRAM INITIALIZES THE iSBX 488 BOARD AS A NON SYSTEM
  4 ;       CONTROLLER WITH ADDRESSING MODE 1 BASE ADDRESS IS 20H
  5 ;
  6         PUBLIC      CINIT
  7         CSEG
  8
  9 CINIT:  MVI         A,0A0H
 10         OUT         2AH         ;ENABLE TCI, 8292
 11 BACK:   IN          OFBH        ;WAIT FOR 8292 IBF LOW
 12         ANI         02          ;BEFORE CONTINUING
 13         JNZ         BACK
 14         XRA         A           ;CLEAR A
 15         OUT         21H         ;CLEAR INTERRUPT MASK 1
 16         OUT         22H         ;CLEAR INTERRUPT MASK 2, 8291A
 17         MVI         A,01        ;SET UP
 18         OUT         24H         ;ADDR MODE 1, 8291A
 19         IN          28H         ;INPUT JUMPER ADDRESS FROM 8282
 20         ANI         1FH         ;MASK FOR TALK/LISTEN ADDRESS
 21         OUT         26H         ;ENABLE ADDR 0, 8291A
 22         MVI         A,0E0H
 23         OUT         26H         ;DISABLE ADDR 1, 8291A
 24         MVI         A,26H       ;SET NF TO 6, T1 to 2 USEC
 25         OUT         25H         ;SET INTERNAL COUNTER
 26         XRA         A           ;CLEAR A, 8291A
 27         OUT         25H         ;PON MESSAGE to 8291A
 28         RET
 29         END
```

Figure 3-7. Non-System Controller

CHAPTER 4
PRINCIPLES OF OPERATION

4-1. INTRODUCTION

This chapter provides a brief operational description of the iSBX 488 Multimodule Board. The functional description of the iSBX interface, the GPIB interface and the board's internal architecture are included in this chapter. This chapter assumes the reader is familiar with the published IEEE Standard 488 GPIB (1978 Version) and the Intel iSBX Bus Specification, Manual Order Number 142686.

4-2. FUNCTIONAL DESCRIPTION

The iSBX 488 Multimodule Board is designed to interface a host iSBC Single Board Computer to the IEEE 488 General Purpose Interface Bus (GPIB). The board is divided into several hardware components as shown in Figure 4-1. Board operation is determined by programming the 8291A and 8292 components and is application dependent. The iSBX 488 Multimodule Board in conjunction with the iSBC host hoard may be conceptually partitioned into the following three major areas:

a. ***Device Functions:*** Defined by the iSBC host board functionality (including software). The interface is the iSBX connection.

b. ***Interface Functions:*** programmable functions defined by the IEEE 488 Standard and implemented by the 8291A and 8292.

c. ***Message Coding Logic:*** communication to and from the interface functions defined in terms of messages and state linkages as implemented by the 8291A and 8292.

The iSBX 488 Multimodule Board combines the interface functions and the message coding within the 8291A and 8292 microcomputers. The iSBX Bus connector signal lines connect the device functions to the interface functions. The connection from the message coding logic to the driver/receiver unit is implemented within the two 8293 transceivers. The remaining logic supports iSBX address decoding, interrupt line routing, clock generation, jumper selections and a general purpose jumper input port.

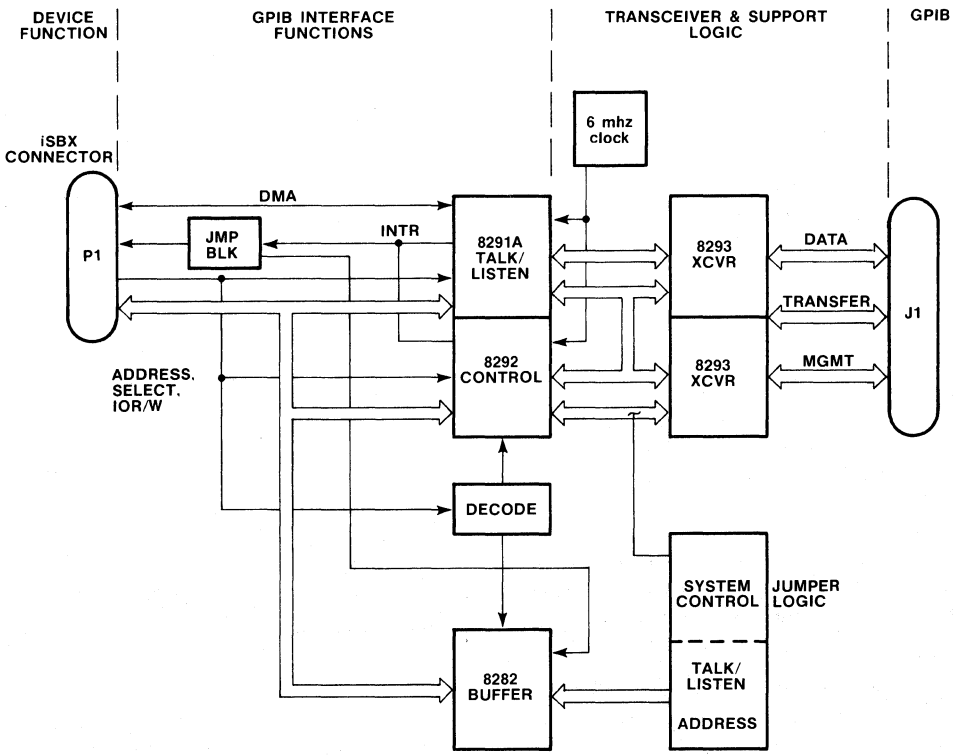


Figure 4-1. iSBX 488™ Multimodule Board Block Diagram

4-3. iSBX BUS INTERFACE

The iSBX bus interface is grouped into six functional classes:

a. Control Lines
b. Address and Chip Select Lines
c. Data Lines
d. Interrupt Lines
e. Option Lines
f. Power Lines

4-4. Control Lines. The control lines provide the host iSBC board with a means to communicate with the iSBX 488 board. This communication link is provided by four unique groups of lines (Command, DMA Control, Initialize, and System Control) and are described in the following text.

4-5. Command Lines. The Command Lines (IORD/, IOWRT/) are active low signals that provide the communications link between the base board and the iSBX 488 board. An active command line, conditioned by Chip Select (MCS0/), indicates to the iSBX 488 board that the address lines are valid and the board should perform the specified I/O operation.

4-6. DMA Control Lines. The DMA Lines (MDRQT, MDACK/) are the handshake control between the DMA controller device on the host board and the iSBX 488 board. MDRQT is an active high signal output from the iSBX 488 board to the DMA controller on the host iSBC board, requesting a DMA cycle. MDACK/ is an active low input signal to the iSBX 488 board from the DMA controller, acknowledging that the requested DMA cycle has been granted. One byte of data is transferred between the iSBX 488 board and the host board for each DMA cycle.

4-7. Initialize Line. The Initialize Line (RESET) sent to the iSBX 488 board is generated by the base board to put the iSBX 488 board in a known reset state.

4-8. System Control Line. The System Control Line (MPST/) is an output signal from the iSBX 488 board to the base board. The signal, identified as Multimodule Board Present, is active low and indicates to the base board I/O decode logic that an iSBX 488 board is installed. The MPST/ signal is electrically grounded on the iSBX 488 board.

4-9. Address And Chip Select Lines. The iSBX connector provides three address lines (MA0, MA1, MA2), and two chip select lines (MCS0/, MCS1/). All address and chip select lines are utilized by the iSBX 488 Multimodule board to decode the selection of the 8291A, 8292 and 8282 components.

The base board decodes I/O addresses and generates the chip select signals for one to three multimodule boards. The base board decodes all but the lower order three address bits in generating the multimodule board chip select signals. Thus, a base board would normally reserve two blocks of 8 I/O ports for each iSBX board used.

4-10. Address Lines. Address lines MA0-MA2 are used to select the 8 read and 8 write registers in the 8291A component. MA1 and MA2 selects the 8 read and 5 write registers in the 8292 component. MA1 is used to select the single read register of the 8282 component. See Table 3-1 of Chapter 3.

4-11. Chip Select Line. Chip select line MCS0/ is used to select the 8291A component. MCS1/ is used to enable selection of the 8292 and 8282 components while address line MA1 determines which component is actually selected.

4-12. Data Lines. Eight bidirectional data lines (MD0-MD7) are used to transmit or receive information to or from the iSBX port. These lines are bussed with other I/O ports on the iSBC host board and other iSBX boards. The 8291A, 8292 and 8282 ports are in a high impedance state unless accessed for an IO read operation.

4-13. Interrupt Lines. The Interrupt Lines (MINTR0, MINTR1) are active high output lines used to make interrupt requests to the host iSBC board. The MINTR0 line is generated from the 8291A INT output. The MINTR1 originates from the SPI output of the 8292 component.

4-14. Option Lines. There are two option lines (OPT0, OPT1) provided as reserve lines, that connect to wire wrap posts on both the base board and the iSBX 488 board. They are used for optional interrupt lines or the TRIG line originating from the iSBX 488 board.

4-15. Power Lines. The iSBX connector provides for the base board to supply +5, +12 Volts and ground. However, the iSBX 488 board requires only +5 Volts and ground.

4-16. I/O Command Operations. The I/O command lines from the base board are driven by tri-state drivers with pull-up resistors or standard TTL totem pole drivers. These lines indicate (to the iSBX 488 board) what action is being requested.

4-17. I/O Read. The I/O Read command timing is shown in Figure 4-2. The base board generates a valid port address and chip select for the iSBX 488 board. After the set up timings are met the host iSBC board activates the IORD/ line. The iSBX 488 board must put valid data on the data bus (MD0-MD7) within 250nsec. The host iSBC board reads the data and removes the read command, address, and chip select signals.

4-18. I/O Write. The I/O Write command timing is shown in Figure 4-3. The host iSBC board generates a valid I/O address and chip select for the iSBX 488 board. After the set up timings are met, the base board activates the IOWRT/ line. The IOWRT/ line remains active (low) for 300ns and the data is valid for 250ns before IOWRT/ is removed. The host iSBC board then removes the data address and chip select signals.

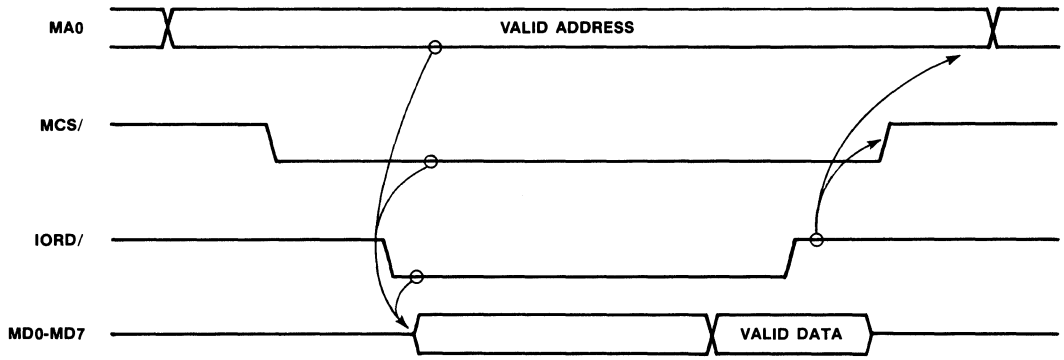


Figure 4-2. I/O Read Timing

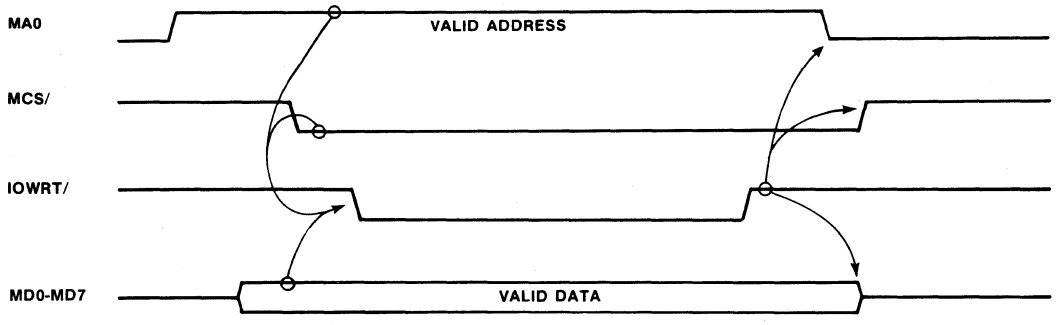


Figure 4-3. I/O Write Timing

4-19. DIRECT MEMORY ACCESS (DMA)

The iSBX 488 board can be operated in DMA or non-DMA mode. When the base board is equipped with a DMA controller the iSBX bus will support DMA operation, permitting the host board processor to perform other tasks while data is being transferred. The following timing example shows the interface lines in their operational sequence. Because of the similarity between a DMA read and DMA write, only the DMA Read is illustrated, Figure 4-4.

A DMA cycle is initiated when the iSBX 488 board activates MDREQ to the DMA controller on the base board. Once the DMA controller gains control of the iSBX bus, it acknowledges back to the iSBX 488 board with MDACK/. The DMA controller then activates an I/O Read cycle and the iSBX 488 board puts valid data on the data bus (MD0-MD7) within 250 nsec from the leading edge of IORD/. The DMA controller then activates MEM WRITE/ to load the Read data into the host iSBC board memory. The MDACK/ signal acts as a chip select and address to the Multimodule board (the MCS and MA0-MA1 signals are undetermined as they are driven by the memory address). The iSBX 488 board removes the MDRQT during the cycle to stop the DMA cycle. Once the read operation is complete the DMA controller deactivates the read command providing a data hold time. If the DMA request signal was removed, the DMA controller will release the iSBX bus back to the host processor and remove MDACK/. If the request is not removed, the DMA controller will proceed to another DMA cycle.

4-20. GPIB INTERFACE FUNCTIONS

There are ten (10) interface functions specified by the IEEE 488 standard. Not all devices will have all functions and some may only have partial subsets. The ten functions are summarized below with the relevant section number from the IEEE 488 document given at the beginning of each paragraph. For further information please see the IEEE standard.

1. SH — Source Handshake (IEEE section 2-3) This function provides a device with the ability to properly transfer data from a Talker to one or more Listeners using the three handshake lines.
2. AH — Acceptor Handshake (IEEE section 2-4) This function provides a device with the ability to properly receive data from the Talker using the three handshake lines. The AH function may also delay the beginning (NRFD) or end (NDAC) of any transfer.
3. T — Talker (IEEE section 2-5) This function allows a device to send status and data bytes when addressed to talk. An address consists of one (Primary) or two (Primary and Secondary) bytes. The latter is called an extended Talker.
4. L — Listener (IEEE section 2-6) This function allows a device to receive data when addressed to listen. There can be extended Listeners (analogous to extended Talkers above).
5. SR — Service Request (IEEE section 2-7) This function allows a device to request service (interrupt) the Controller. The SRQ line may be asserted asynchronously.
6. RL — Remote Local (IEEE section 2-8) This function allows a device to be operated in two modes: Remote via the GPIB or Local via the manual front panel controls.
7. PP — Parallel Poll (IEEE section 2-9) This function allows a device to present one bit of status to the Controller-In-Charge. The device need not be addressed to talk and no handshake is required.

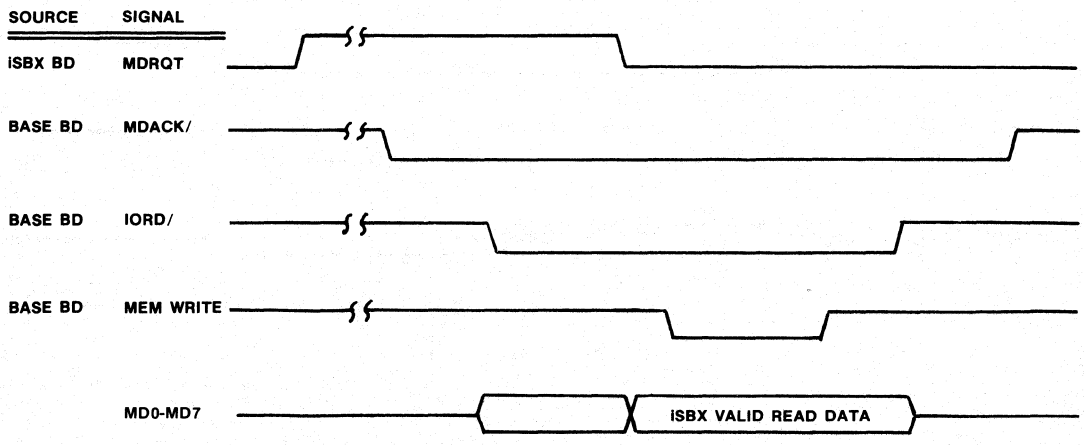


Figure 4-4. DMA Read Timing

8. DC — Device Clear (IEEE section 2-10) This function allows a device to be cleared (initialized) by the Controller. Note that there is a difference between DC (*device* clear) and the IFC line (*interface* clear).
9. DT — Device Trigger (IEEE section 2-11) This function allows a device to have its basic operation started either individually or as part of a group. This capability is often used to synchronize several instruments.
10. C — Controller (IEEE section 2-12) This function allows a device to send addresses, as well as universal and addressed commands to other devices. There may be more than one controller on a system, but only one may be the Controller-In-Charge at any one time.

At power-on time the controller that is wired to be the System Controller becomes the active Controller-In-Charge. The System Controller has several unique capabilities including the ability to send Interface Clear (IFC — clears all device interfaces and returns control to the System Controller) and to send Remote Enable (REN — allows devices to respond to bus data once they are addressed to listen). The System Controller may optionally Pass Control to another controller, if the system software has the capability to do so.

4-21. CODING LOGIC

All GPIB related signals (i.e., DIO1-DIO8, DAV, EOI, ATN, SRQ, IFC, NDAC, NRFD, and REN) at internal nodes have a slash (/) suffix to indicate that the low voltage state equals a logical 1. These signals are buffered by U4 and U6, 8293 non-inverting transceivers, for GPIB interfacing. The / suffix is removed from the signals at the GPIB interface, however the logical definition does not change. Refer to the schematic diagram in Figure 5-2.

DIO1/-DIO8/, DAV/, T/R1

The DATA (1-8)/ and DAV/ lines are bidirectional. T/R1 is an output of U5 (8291A) which controls the transmit/receive direction for these signals at U4 and U6 (8293). The 8291A sends and receives data (over DIO lines) for both the Talker/Listener (8291A) functions and the Controller (8292) function. The DAV/ is sent by the 8291A during a Source Handshake and received during an Acceptor Handshake function. The 8292 sends a DAV/ during a Parallel Poll and monitors DAV/ during a "Take Control Synchronously" function.

EOI/, ATN/, T/R2

EOI/ is bidirectional to U5 (8291A), bidirectional to U6 (8293), and input to U4 (8293), and an input to U3 (8292). ATN/ is an output from U6 and inputs to U5 and U4. T/R2 is an output from U5 and an input to U6. The 8291A may send EIO/ as an END remote message over the GPIB indicating the end of a multiple byte transfer sequence, or in conjunction with ATN/, receive an EOI/ during a parallel poll sequence. T/R2 controls the send/receive direction of EOI/ at U6. EOI/ will normally be received from the GPIB whenever another GPIB device is sending an END message and the 8291A is addressed to listen. ATN/ is monitored by the 8291A to interpret the data on the DIO/ lines. ATN/ and EOI/ are ANDed within U4 to control the type of output (tristate or open collector) on DIO1-DIO8 and DAV lines. These lines will have tristate outputs at all times except when both ATN/ and EOI/ are low (logical "1"). Whenever the 8292 is the Controller-In-Charge of the GPIB the ATN/ line is forced high by U6. If the 8292 is not the Controller-In-Charge the ATN/ level will be determined by ATN from the GPIB.

COUNT INPUT TO 8292

This input monitors EOI/ or NDAC/ (EOI/ by default jumper). Every low to high transition of EOI/ (or NDAC/) increments a counter internal to the 8282. A count of EOI/ (or NDAC/) transitions represent the number of data blocks (or bytes) that have taken place. The minimum period allowed for consecutive transitions is 7.5 microseconds. If the 8292 is he active Controller of the GPIB, a GSEC command written to the 8292 will force the 8292 to standby state, CSBS, and then enable the internal counter and corresponding Event Counter Interrupt. When the 8292 is not the active Controller a GSEC command will exit immediately. The interrupt may be disabled by a SPCNI command written to the 8292 or when the 8292 exits CSBS. However the counter will continue to count transitions.

REN/, IFC/, SYC

REN/ is an output of U3 (8292), an input to U5 (8291A), and bidirectional to U6 (8293). IFC/ is bidirectional to both U3 and U6 and an input to U5. SYC is pulled high by RP1 (12 K ohm) or shorted low by a jumper, and it is an input to U3 and U6. The 8292 monitors SYC to determine its initialization sequence at power on. If the 8292 input is the System Controller (SYC = High = 1) it will execute an ABORT Command, become the Controller-In-Charge, and enter the CACS state. If it is not the System Controller, it will remain in CIDS. SYC also controls the transmit/receive direction of REN/ and IFC/ at U6. When SYC is High both IFC/ and REN/ is transmitted on the GPIB from U6, where both are controlled by the 8292. The 8291A monitors IFC/ and REN/ whether it be sourced from the 8292 or from the GPIB. The 8292 only monitors IFC/ when it is not the System Controller.

NRFD/, NDAC/, T/R1

NRFD/ and NDAC/ are both bidirectional to U5 (8291A) and U6 (8293). The transmit/receive direction of both signals are controlled by T/R1 (at 8291A). When T/R1 is High, both signals are received from the GPIB while the 8291A is in a Source Handshake sequence. When T/R1 is Low, both signals are transmitted to the GPIB while the 8291A is in an Acceptor Handshake sequence.

SRQ/, ATNI/, ATNO/, IFCL/, CLTH/, CIC/

SRQ/ is an input to U3 (8292), an output from U5 (8291A) and bidirectional to U6 (8293). ATNI/ is an output from U6 and an input to U3. ATNO/ is an output from U3 and an input to both U6 and U4 (8293). IFCL/ is an output from U6 and an input to both U3 and U4. CLTH and CIC/ are both outputs from U3 and both inputs to U6. The 8291A may send SRQ/ during a Serial Poll sequence to the 8292 and/or out to the GPIB through U6. The 8292 receives SRQ/ from the 8291A and/or from the GPIB through U6. The transmit/receive direction for SRQ/ at U6 is determined by several other signals as discussed later. The ATNI/ signal is monitored by the 8292. Its source is either from ATN off the GPIB or from the ATNO/ signal controller by the 8292. The source for generating ATNI/ is determined by several other signals as discussed later. IFCL/ is monitored by the 8292 when it is not the System Controller (SYC = Low). IFCL/ is a latched low output from U6 whenever an off-board System Controller sends IFC. The latch is cleared High after the 8292 sends CLTH (High pulse) to U6. ATNO/, IFCL/, and T/R1 are gated in U4 to control the direction of DAV/. If the 8292 is the Controller-In-Charge conducting a Parallel Poll sequence the 8291A must capture the Parallel Poll Response for the 8292. The 8291A becomes a listener while the 8292 asserts DAV/. T/R1, IFCL/, and ATNO/ are gated such that the DIO/ lines are received by the 8291A while DAV/ is neither transmitted nor received by U4. The direction control for SRQ/ and the source for generating ATNI/ is determined by CIC/ and IFCL/. If IFCL/ is not latched Low or has just been cleared and CIC/ is Low, then SRQ/ is a buffered output from SRQ, and ATNO/ is the source for both ATNI/ and ATN outputs. If IFCL/ is latched Low or CIC/ is High, then SRQ/ is an input buffer driving SRQ, and ATN is the source for both ATN/ and ATNI/.

EOI2/

EOI2/ is bidirectional to both U3 (8292) and U6 (8293). The 8292 sends or monitors EOI2/ during a Parallel Poll sequence. The transmit/receive direction of EOI2/ is controlled by ATNO/ and IFCL/. If the 8292 is conducting a Parallel Poll both ATNO/ and EOI2/ is asserted Low by the 8292. ATNO/ Low enables EOI2/ to be transmitted as EOI on the GPIB except when IFCL/ is latched Low by the System Controller.

CHAPTER 5
SERVICE INFORMATION

5-1. INTRODUCTION

This chapter provides the following service related information:

a. Repair assistance information.
b. Replacement parts list and diagram.
c. Jumper post location diagram.
d. Schematic diagrams.

5-2. SERVICE AND REPAIR ASSISTANCE

United States Customers can obtain service and repair assistance by contacting the Intel Product Service Hotline in Phoenix, Arizona. Customers outside the United States should contact their sales source (Intel Sales Office or Authorized Distributor) for service information and repair assistance.

Before calling the Product Service Hotline, you should have the following information available:

a. Date you received the product.
b. Complete part number of the product (including dash number). On boards, this number is usually silk-screened onto the board. On other products, it is usually stamped on a label.
c. Serial number of product. On boards, this number is usually stamped on the board. On other products, the serial number is usually stamped on a label.
d. Shipping & billing addresses.
e. If your Intel product warranty has expired, you must provide a purchase order number for billing purposes.
f. If you have an extended warranty agreement, be sure to advise the Hotline personnel of this greement.

Use the following numbers for contacting the Intel Product Service Hotline:

TELEPHONE

All U.S. locations, except Alaska, Arizona, & Hawaii:

(800) 528 - 0595

All other locations: (602) 869 - 4600

TWX NUMBER: 910 - 951 - 1330

Always contact the Product Service Hotline before returning a product to Intel for repair. You will be given a repair authorization number, shipping instructions, and other important information which will help Intel provide you with fast, efficient service. If you are returning the product because of damage sustained during shipment or if the product is out of warranty, a purchase order is required before Intel can initiate the repair.

In preparing the product for shipment to the Repair Center, use the original factory packing material, if possible. If this material is not available, wrap the product in a cushioning material such as Air Cap TH - 240, manufactured by the Sealed Air Corporation, Hawthorne, N.J. Then enclose in a heavy duty corrugated shipping carton, and label "FRAGILE" to ensure careful handling. Ship only to the address specified by Product Service Hotline personnel.

5-3. REPLACEMENT PARTS

A complete list of replacement parts is provided in Table 5-1. This list provides the part number, manufacturer, description and quantity of the item. Notice that each item is referenced in the parts location diagram. Table 5-2 provides the full name of the manufacturer which is abbreviated in Table 5-1. Some of the parts are available from any normal commercial source, and should be ordered by their generic description. These items are called out as CML, rather than listing a specific part number. Figure 5-1 shows the location of each iSBX 488 referenced part in Table 5-1.

5-4. SERVICE DIAGRAMS

Figure 5-2 provides a schematic diagram of the iSBX 488 Multimodule Board. The schematic diagrams are current when the manual is printed. However, minor revisions to the diagrams may occur between manual printings. Therefore, Intel provides photocopies of the current schematic diagrams with the board, when it is shipped from the factory. These diagrams should be inserted into this manual for future reference. In most instances, the diagrams shipped with the board will be identical to those printed in the manual.

Table 5-1. Replacement Parts List

Ref	Description	Part Number	Manufacturer	Qty
C1-6	Capacitor, Cer. Z5U AXL .10 uf	OBD	CML	6
C7	Capacitor, Tant. 22 uf 15v, 10%	OBD	CML	1
G1	Oscillator, Crystal 6MHz	HY-4550-6	Hytek	1
P1	Connector, iSBX Multimodule 36 pin	292-0001	Viking	1
R1-2	Resistor, 470 ohm, 1/8W, 5%	OBD	CML	2
RP1	Resistor Pack, 10-pin 12K	OBD	CML	1
U1	Octal Latch	8282	Intel	1
U2	Hex Inverter	74LS04	TI	1
U3	GPIB Controller	8292	Intel	1
U4, 6	GPIB Transceiver	8293	Intel	2
U5	GPIB Talker/Listener	8291A	Intel	1
U7	2 Input Positive OR Gate	74LS32	TI	1

Table 5-2. Manufacturers' Names

Abbreviation	Description
Hytek	Hytek Microsystems, Inc.
Viking	Viking Connectors, Inc.
TI	Texas Instruments, Inc.
CML	Any commercial source
OBD	Order by description

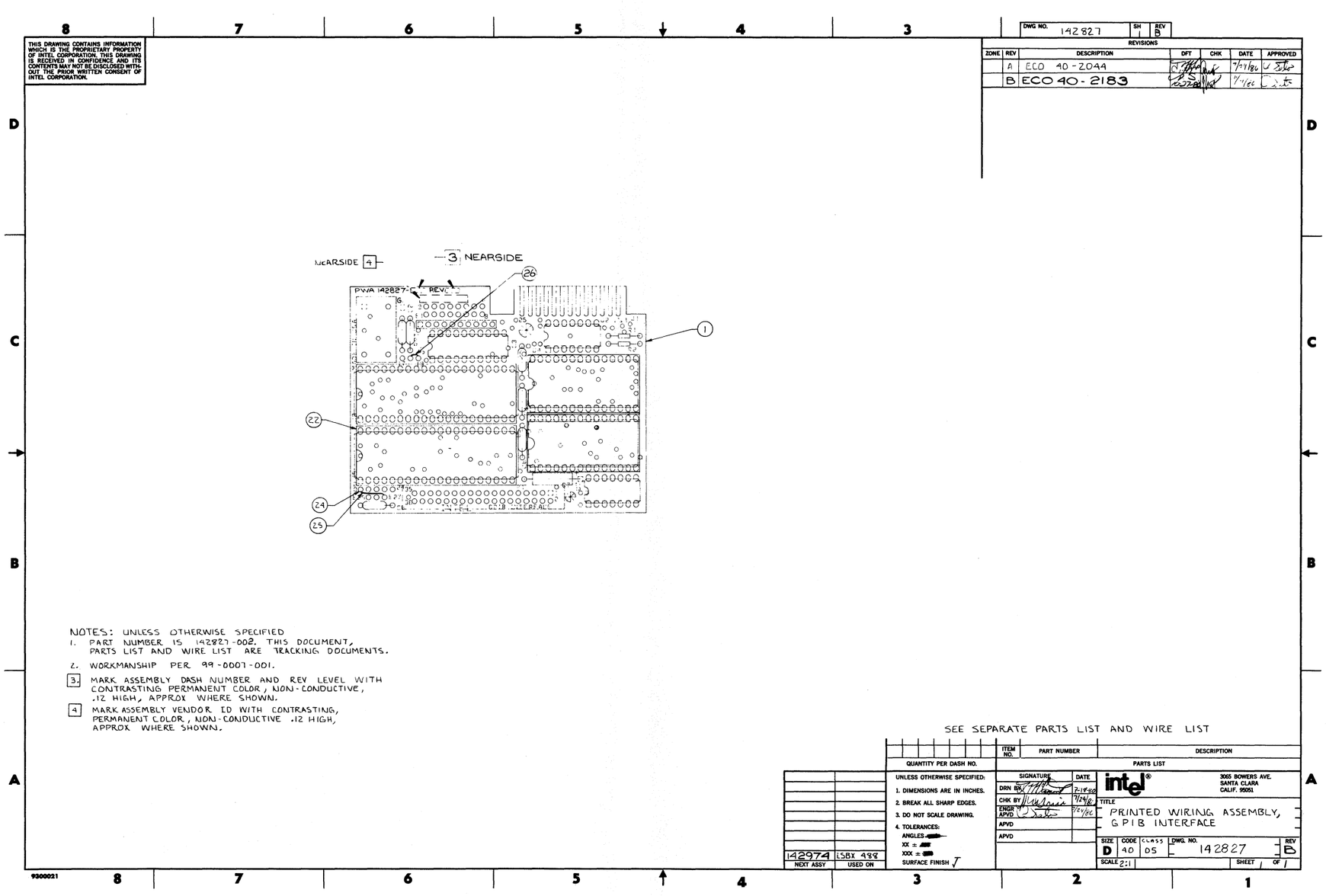


Figure 5-1. iSBX 488™ Multimodule Board Parts Location Diagram

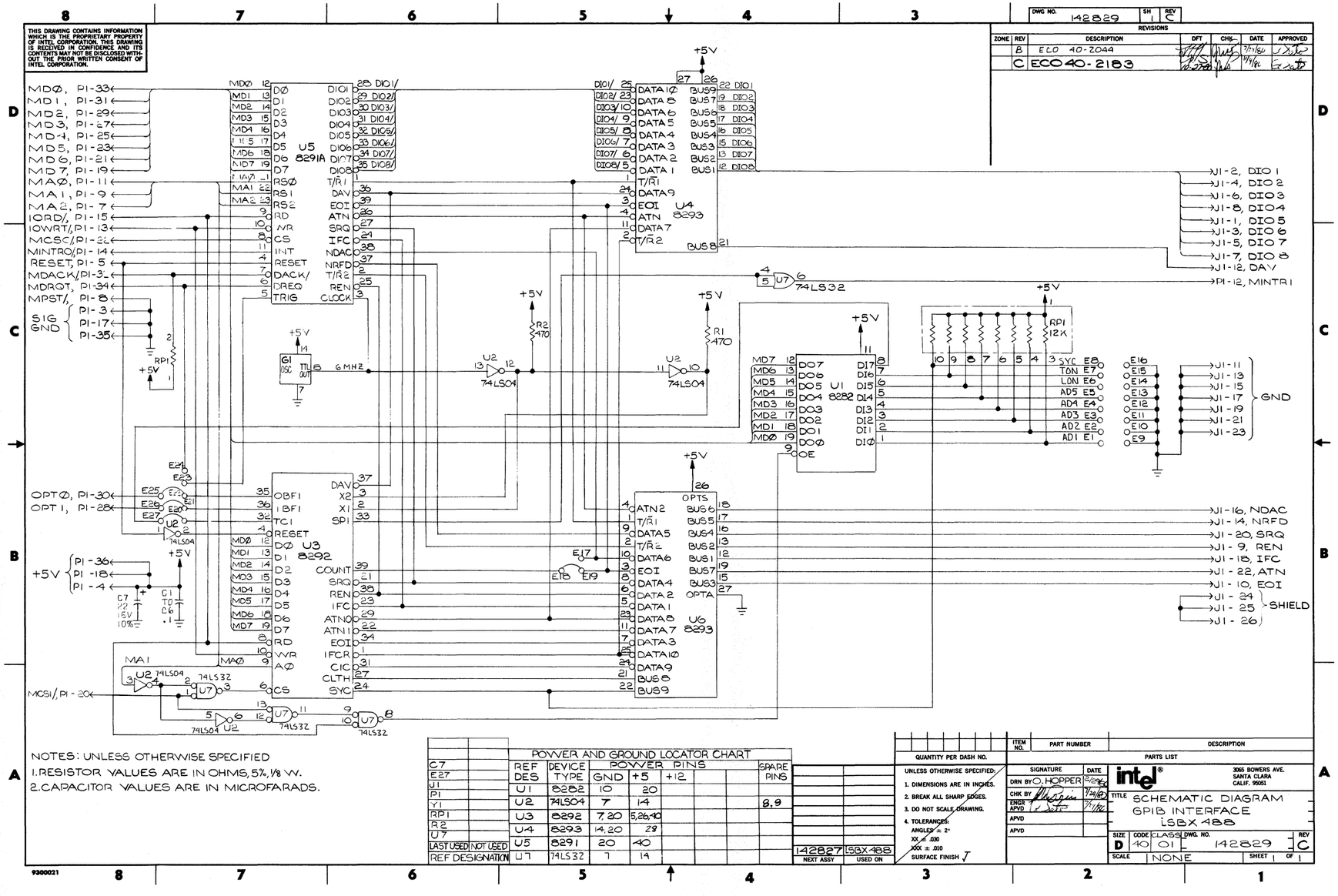


Figure 5-2. iSBX 488™ Multimodule Board Schematic Diagram

CAUTION: These schematic diagrams may have been revised. See "Service Information" chapter for details.

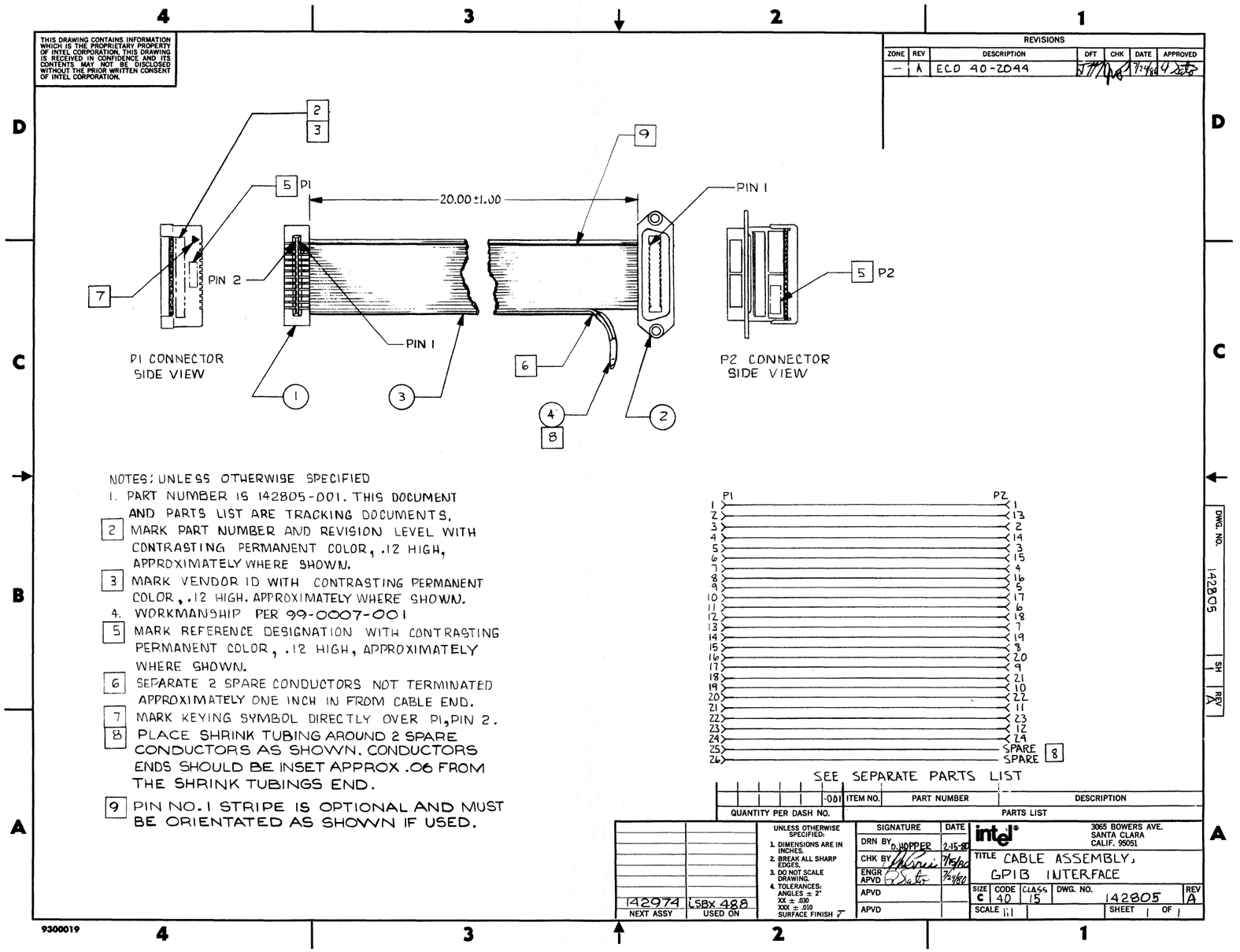


Figure 5-3. Cable Assembly

CAUTION: These schematic diagrams may have been revised. See "Service Information" chapter for details.

APPENDIX A
PL/M 80 SOFTWARE DRIVER LISTING EXAMPLE

This program assumes that the iSBX 488 board has been initialized as a System Controller in Talk-Only (ton) mode, ATN is asserted, and no holdoffs are in effect or enabled.

Four parameters must be passed to this routine: LISTEN$PTR, DATA$PTR, Count, and EOS. LISTEN$PTR and DATA$PTR are the beginning addresses of the Listen list and data list, respectively. Count is the maximum number of bytes to be sent. EOS is the character which will assert the EOI line on the GPIB 488 bus. When Count is decremented to zero, or EOI is sent, the procedure will stop sending data and return to the calling program.

```
 1          PLM$EX: DO;
 2    1         WAIT$BO:PROCEDURE EXTERNAL;
 3    2         END WAIT$BO;
                /*  */
 4    1         WAIT$T:PROCEDURE EXTERNAL;
 5    2         END WAIT$T;
                /*  */
 6    1         SEND:PROCEDURE (LISTEN$PTR,DATA$PTR,COUNT,EOS) PUBLIC;
 7    2             DECLARE (LISTEN$PTR,DATA$PTR) ADDRESS;
 8    2             DECLARE (COUNT,EOS) BYTE;
 9    2             DECLARE LISTENER BASED LISTEN$PTR BYTE;
10    2             DECLARE DATUM BASED DATA$PTR BYTE;
                /*  */
11    2             DECLARE DATA$OUT LITERALLY '0F0H';
12    2             DECLARE EOS$REG LITERALLY '0F7H';
13    2             DECLARE CMD$REG LITERALLY '0FBH';
14    2             DECLARE AUX$REG LITERALLY '0F5H';
                /*  */
15    2             DECLARE (MTA,UNL,GTSB,TCSY,EOIS,NO$EOI) BYTE
                    DATA ('A','?',0F6H,0FDH,088H,080H);
                /*  */
                /* BEGIN PROGRAM HERE */
                /*  */
16    2             OUTPUT(DATA$OUT) = MTA;          /* SEND MY TALK ADDRESS */
17    2             CALL WAIT$BO;                 /* WAIT FOR BO BIT SET IN 8291A */
18    2             OUTPUT(DATA$OUT) = UNL;          /* SEND UNIVERSAL UNLISTEN */
19    2             CALL WAIT$BO;
20    2             OUTPUT(EOS$REG) = EOS;           /* LOAD EOS IN 91A EOS REG */
                /* ADDRESS LISTENERS */
21    2             DO WHILE ((LISTENER >= 20H) AND (LISTENER <= 3EH));
22    3                 OUTPUT(DATA$OUT) = LISTENER;
23    3                 CALL WAIT$BO;
24    3                 LISTEN$PTR = LISTEN$PTR + 1;
25    3             END;
                /* THEN GO TO STANDBY */
26    2             OUTPUT(CMD$REG) = GTSB;
27    2             CALL WAIT$T;                  /* WAIT FOR TCI LOW THAN HIGH*/
28    2             OUTPUT(AUX$REG) = EOIS;
                /* SEND DATA TO LISTENER */
29    2             DO WHILE ((COUNT <> 0) AND (DATUM <> EOS));
30    3                 OUTPUT(DATA$OUT) = DATUM;
31    3                 CALL WAIT$BO;
32    3                 COUNT = COUNT - 1;
33    3                 DATA$PTR = DATA$PTR +1;
34    3             END;
                /* SEND LAST BYTE */
35    2             OUTPUT(DATA$OUT) = DATUM;
36    2             CALL WAIT$BO;
                /* TAKE CONTROL SYNCHRONOUSLY */
37    2             OUTPUT(CMD$REG) = TCSY;
38    2             CALL WAIT$T;
```

PL/M-80 COMPILER

```
39    2             OUTPUT(AUX$REG) = NO$EOI;        /* DISABLE EOI ON EOS */
40    2             OUTPUT(DATA$OUT) = UNL;
41    2             CALL WAIT$BO;
42    2         END SEND;
43    1     END PLM$EX;
```

INDEX

REQUEST FOR READER'S COMMENTS

Intel Corporation attempts to provide documents that meet the needs of all Intel product users. This form lets you participate directly in the documentation process.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this document.

1. Please specify by page any errors you found in this manual.

2. Does the document cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of document for your needs? Is it at the right level? What other types of documents are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this document on a scale of 1 to 10 with 10 being the best rating. _____

NAME _____ DATE _____

TITLE _____

COMPANY NAME/DEPARTMENT _____

ADDRESS _____

CITY _____ STATE _____ ZIP CODE _____

Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

This document is one of a series describing Intel products. Your comments on the back of this form will help us produce better manuals. Each reply will be carefully reviewed by the responsible person. All comments and suggestions become the property of Intel Corporation.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
5200 N.E. Elam Young Pkwy.
Hillsboro, Oregon 97123

O.M.S. Technical Publications

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) **987-8080**

Printed in U.S.A.