iSBC 80/10B™
SINGLE BOARD COMPUTER
HARDWARE REFERENCE MANUAL

Order Number: 9803119-02

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

REV.	REVISION HISTORY	PRINT DATE
-001	Original Issue	10/79
-002	Revised: Tables 1-1, 2-8, 2-10 Sections: 2-20 & 5-2 Appendix C.	4/81

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

and the combination of ICE, iCS, iRMX, iSBC, iSBX, MCS, 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 iSBC 80/10B Single Board Computer. Additional related information is provided in the following Intel documents:

- *Intel MCS-80 User's Manual,* Order Number 9800153.
- *Intel 8080/8085 Assembly Language Programming Manual,* Order Number 9800301.
- *Intel Multibus Specification,* Order Number 9800683.
- *Intel iSBX Bus Specification,* Order Numer 142686.
- *Designing iSBX Multimodule Boards,* Ap. Note AP-96.

CONTENTS

CONTENTS (Continued)

TABLES

ILLUSTRATIONS

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The iSBC 80/10B Single Board Computer is a Multibus compatible computer system on a single printed circuit assembly (figure 1-1). The iSBC 80/10B board includes a central processing unit (CPU), 1K bytes of static random access memory (RAM), one serial and six programmable 8-bit I/O ports and sockets for up to 16K bytes of read only memory (ROM). In addition, the board is Multimodule compatible and can be equipped with battery backup power to preserve RAM data during a power failure.

This manual provides all the information you will need to promptly install and set up the iSBC 80/10B board. Programming and service information are also provided.

1-2. DESCRIPTION

The iSBC 80/10B board is controlled by an Intel 8080A microprocessor in conjunction with an Intel 8238 system controller and bus driver. System access is provided through the Multibus connector and an auxiliary 60-pin connector. Direct I/O operations are handled through the board's 48 parallel lines or the serial I/O lines.

Up to 64K bytes of memory may be directly addressed by the iSBC 80/10B board. The first 16K bytes are reserved for on-board ROM or PROM devices, and the next 4K bytes for on-board RAM devices. The board may utilize low power PROM devices with a standby mode, and low power RAM devices.

Serial I/O operation is handled by an Intel 8251A Programmable Communications Interface (PCI) device. On-board circuitry allows a choice of either 20 mA current loop (TTY) or RS232C serial interface. Baud rates are jumper selected.

The iSBC 80/10B board utilizes two Intel 8255A Programmable Peripheral Interface (PPI) devices to control the six, 8-bit parallel I/O ports. These ports may be configured to a variety of dedicated or general purpose applications. Sockets are provided to accommodate line drivers or terminators for the two PPI devices.

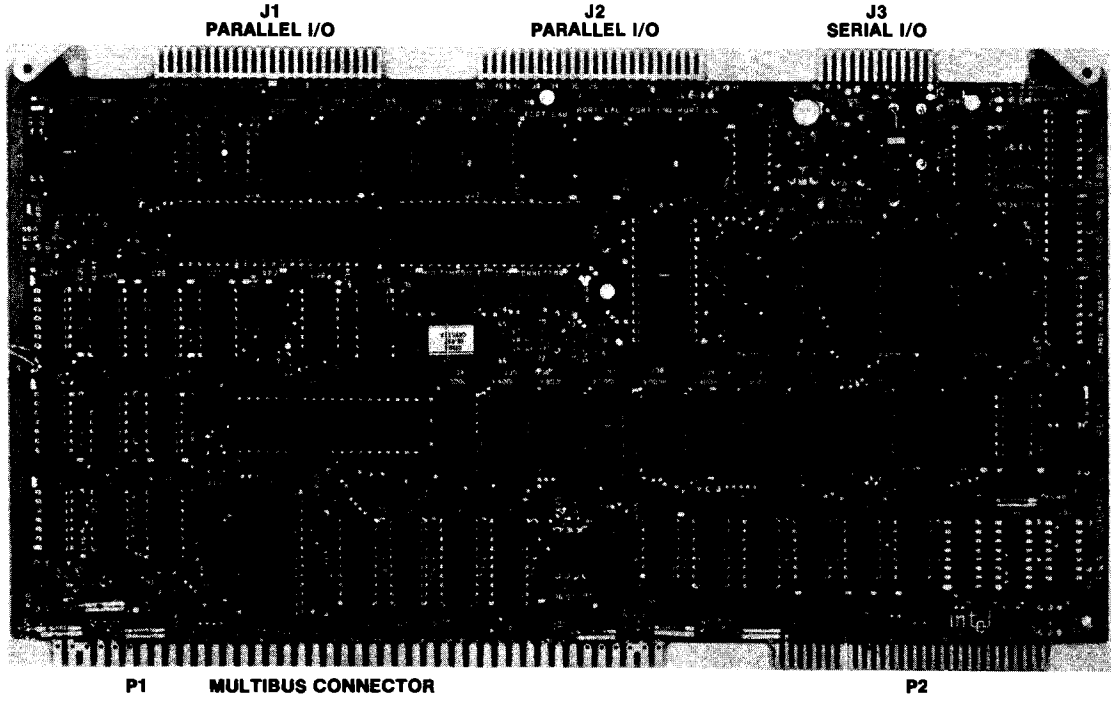


Figure 1-1. iSBC 80/10B™ Single Board Computer

A single iSBX Multimodule connector is provided which can accommodate a variety of special purpose modules, allowing direct access to the on-board CPU. Multimodule boards reside directly on the iSBC 80/10B board, and do not require external power.

The iSBC 80/10B board is designed to operate as a limited master in either the iSBC 655 System Chassis, the iSBC 660 System Chassis, or the iCS 80 Industrial Chassis. The board may also reside in your own custom designed chassis, using Multibus compatible connectors (refer to Chapter 2).

1-3. DOCUMENTATION SUPPLIED

Each iSBC 80/10B board is shipped with a corresponding set of schematic diagrams and a component layout drawing. Refer to Chapter 5 for related information.

1-4. ADDITIONAL EQUIPMENT REQUIRED

The iSBC 80/10B board requires few optional components for operation. Depending on your application, you may need to purchase up to three I/O connectors and cables. Any on-board ROM/PROM must also be purchased separately. Chapter 2 provides information for selecting these items, based on your specific needs.

1-5. SPECIFICATIONS

Specifications of the iSBC 80/10B board are provided in table 1-1.

Table 1-1. Specifications

CPU	Intel 8080A
WORD SIZE	
Address	16 bits
Data	8 bits
Instruction	8, 16, or 24 bits
CYCLE TIME	Basic Instruction Cycle — 1.95 μs
SYSTEM CLOCK	2.048 MHz ± 0.1%
MEMORY ADDRESSING (factory configuration)	
On-Board ROM/EPROM	0-0FFF (see section 2-8)
On-Board RAM	3C00-3FFF (see section 2-9)
MEMORY CAPACITY	
On-Board ROM/EPROM	16K bytes (sockets only)
On-Board RAM	4K bytes
Off-Board Expansion	Up to 64K bytes using user specified combinations of RAM, ROM, and EPROM.

I/O ADDRESSING
On-Board Programmable I/O

	8255A No. 1			8255A No. 2			8255A No. 1	8255A No.2	8251A	8251A	iSBX Multimodule	iSBX Multimodule
Port	A	B	C	A	B	C	Control	Control	Data	Control	MCS0	MCS1
Address	E4	E5	E6	E8	E9	EA	E7	EB	EC	ED	F0-F7	F8-FF

I/O CAPACITY	
Parallel	48 programmable lines
Serial	1 transmit, 1 receive

Table 1-1. Specifications (Cont'd)

SERIAL BAUD RATES

Frequency (kHz) (Jumper Selectable)	Baud Rate (Hz)		
	Synchronous	Asynchronous (Program Selectable)	
		÷ 16	÷ 64
307.2	—	19200	4800
153.6	—	9600	2400
76.8	—	4800	1200
38.4	38400	2400	600
19.2	19200	1200	300
9.6	9600	600	150
6.98	6960	—	110
4.8	4800	300	75

SERIAL COMMUNICATION CHARACTERISTICS

Synchronous — 5–8 bit characters; internal or external character synchronization; automatic sync insertion

Asynchronous — 5–8 bit characters; break character generation; 1, 1½, or 2 stop bits; false start bit detectors

Interrupts — Single-level with on-board logic that automatically vectors processor to location 38 (hex) using a restart instruction (RESTART 7). Interrupt requests may originate from user specified I/O (2); the programmable peripheral interface (2); Multimodule board (2); or the programmable communications interface (2).

MILLISECOND TIMER — Period 1.432 ms ± 0.1% with jumper 80-81 installed.
Period 2.084 ms ± 0.1% with jumper 80-81 removed

INTERFACES

Multibus Lines — All signals TTL compatible (P1 & P2)
Parallel I/O — All signals TTL compatible (J1 & J2)
Serial I/O — RS232C or a 20 mA current loop TTY interface (jumper selectable) (J3)
Interrupt Requests — All TTL compatible (active-low)
Multimodule Board (iSBX) — (J4)

PHYSICAL CHARACTERISTICS — Width — 12.00 in. (30.48 cm)
Length — 6.75 in. (17.15 cm)
Thickness — 0.50 in. (1.27 cm)
Weight — 14 oz. (484.4 gm)

LINE DRIVERS AND TERMINATORS

I/O Drivers — The following line drivers and terminators are all compatible with the I/O driver sockets on the iSBC 80/10B board:

Driver	Characteristic	Sink Current (mA)
7438	I,OC	48
7437	I	48
7432	NI	16
7426	I,OC	16
7409	NI,OC	16
7408	NI	16
7403	I,OC	16
7400	I	16

NOTE

I = inverting; NI = non-inverting; OC = open collector.

Port E4 has 28 mA totem pole drivers and 1 kΩ terminators.

I/O Terminators — 220Ω/330Ω divider or 1 kΩ pullup

Table 1-1. Specifications (Cont'd)

BUS DRIVERS

Function	Characteristic	Sink Current (mA)
Data	Tri-state	32
Address	Tri-state	34
Commands	Tri-state	32

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature 0°C to 55°C

ELECTRICAL CHARACTERISTICS

Voltage	Without EPROM[1]	With 2708 EPROM[2]	With 2758, 2716 2732 EPROM[3]	Power Down Requirements (RAM & Support Circuit)
V_{CC} +5V ±5%	I^4_{CC} = 2.0A	3.1A	3.46A	200 mA/K (2114)
V_{DD} +12V ±5%	I_{DD} = 150 mA	400 mA	150 mA	Not Required
V_{BB} - 5V ±5%	I_{BB} = 2 mA	200 mA	2 mA	Not Required
V_{AA} -12V ±5%	I_{AA} = 175 mA	175 mA	175 mA	Not Required

NOTES:
1. Does not include power required for optional ROM/EPROM, I/O drivers or I/O terminators.
2. With four Intel 2708 EPROMS and 220/330 for terminators, installed for 40 input lines. All terminator inputs low.
3. Same as #2 except with four 2758 or 2716 or 2732 installed.
4. I_{CC} shown without RAM supply current. For 2114 add .2A per K byte to a maximum of .8 amps for 4K.

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter provides specific information enabling you to install the iSBC 80/10B board into your system, with minimal effort. The board's default, or factory configuration for RAM addressing, ROM type, and other variables is described, followed by procedures for altering the default configuration. In this manner, the board will accommodate a variety of applications. To completely familiarize yourself with the flexibility of the iSBC 80/10B board, we recommend reading the entire chapter before installation and use.

2-2. UNPACKING AND INSPECTION

Inspect the shipping carton immediately upon receipt for evidence of mishandling during transit. If the shipping carton is severely damaged or waterstained, 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 of the carton are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment contact the Intel Technical Support Center (see section 5-2) to obtain a Return Authorization Number and further instructions. 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

There are several general requirements which should be considered, prior to board installation. These requirements are discussed in sections 2-4 through 2-6.

2-4. MINIMAL OPERATING REQUIREMENTS

The iSBC 80/10B board standard configuration is described in Chapter 1. In order to operate the board you will need additional equipment. For most applications this will typically be the following:

a. CPU software, residing in on-board ROM (section 2-8).

b. I/O connectors and cables (section 2-21).

c. Additional on-board RAM, if more than 1K bytes are required (section 2-9).

d. Line drivers or terminators for parallel I/O ports (section 2-10).

Instructions for installing these components are given in this chapter.

2-5. POWER REQUIREMENTS

Four voltages are required for operating the iSBC 80/10B board: +5 Vdc, −5 Vdc, +12 Vdc, and −12 Vdc. All must be within ±5.0% of absolute. Power requirements for the various board configurations are listed in table 2-1.

Table 2-1. Power Supply Requirements

Voltage	Without EPROM[1]	With 2708 EPROM[2]	With 2758, 2716 2732 EPROM[3]	Power Down Requirements (RAM & Support Circuit)
V_{CC} +5V ±5%	I^4_{CC} = 2.0A	3.1A	3.46A	200 mA/K (2114)
V_{DD} +12V ±5%	I_{DD} = 150 mA	400 mA	150 mA	Not Required
V_{BB} - 5V ±5%	I_{BB} = 2 mA	200 mA	2 mA	Not Required
V_{AA} -12V ±5%	I_{AA} = 175 mA	175 mA	175 mA	Not Required

NOTES:
1. Does not include power required for optional ROM/EPROM, I/O drivers or I/O terminators.
2. With four Intel 2708 EPROMS and 220/330 for terminators, installed for 40 input lines. All terminator inputs low.
3. Same as #2 except with four 2758 or 2716 or 2732 installed.
4. I_{CC} shown without RAM supply current. For 2114 add .2A per K byte to a maximum of .8 amps for 4K.

2-6. COOLING REQUIREMENTS

Operating temperature range for the iSBC 80/10B board is 0° to 55° Celsius. If the board is installed into an iSBC 655 or iSBC 660 System Chassis, or an iCS Industrial Chassis, adequate cooling is provided by the supplied fans. However, if the board is used in another chassis, ensure adequate cooling is provided by taking temperature readings inside the chassis, at the site environment.

2-7. COMPONENT INSTALLATION

Instructions for installing optional ROM/EPROM, RAM, line drivers or terminators are given in the following sections. When installing optional ROM/EPROM or RAM devices, ensure that pin 1 of the device corresponds to the pin 1 marking on the board (white dot). Refer to figure 2-1. Table 2-2 lists all empty I.C. sockets, and identifies their intended function.

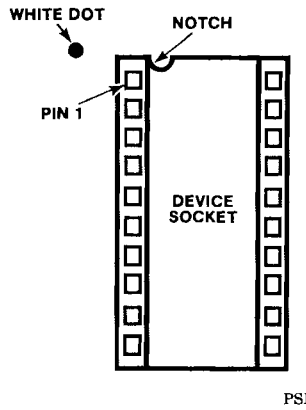


Figure 2-1. Pin Alignment

2-8. ROM/EPROM INSTALLATION

Sockets U19 through U22 are reserved for ROM, PROM, or EPROM devices. A maximum of 16K bytes may be installed. A summary of compatible device types, and capacity is provided in table 2-3. Device types may not be mixed, however empty sockets are allowed (in ascending order only). After selecting the ROM or EPROM which best suits your application, carefully insert each device into its socket. The following cautions are suggested:

a. Never install any device into a live board, that is, one with power applied.

b. Ensure pin 1 of the device is aligned to pin 1 of the socket.

c. Insert the device slowly; check all pins to ensure correct insertion.

Table 2-2. Empty I.C. Sockets

Line Driver/Terminator Sockets	U2 — U11
ROM/EPROM Sockets	U19 — U22
RAM Sockets	U34-36 and U38-40

Once your devices are installed, it is necessary to install header plug P3 into the corresponding device type socket. There are two sockets used for ROM/EPROM type selection: J6 and J7. Each of these sockets is divided into an upper half and a lower half. Plug P3 must be placed in one of the four halves (refer to table 2-3). Each half of sockets J6 and J7 are labeled on the board, as shown in figure 2-2.

After selecting the ROM/EPROM type, another header plug (P4) must be configured to indicate power down mode and access time.

When using devices with a low power standby mode, such as the Intel 2716 or 2732 PROM, the P4 plug may be placed into the PWR DN (upper) position, if ROM access is less than or equal to 350 ns. When using 2708 or ROMs with greater than 350 ns access P4 must be placed in the NORM (lower) position. When your application does not require any on-board ROM, header plug P3 should not be inserted into J6 or J7. Since the iSBC 80/10B board is shipped with the plug installed in the 2708 position, it would be necessary to remove it if no on-board ROM is used.

Wait State Jumper

If you are using ROM/PROM devices which have memory access times of > 550 nanoseconds and < 1020 nanoseconds, the following jumper modification is required:

Install jumper 108 to 109

Refer to figure 5-3, sheet 4 and figure 5-2. Installing this jumper imposes mandatory WAIT states on all CPU memory and I/O requests. The board will realize a 25% speed loss with this jumper installed.

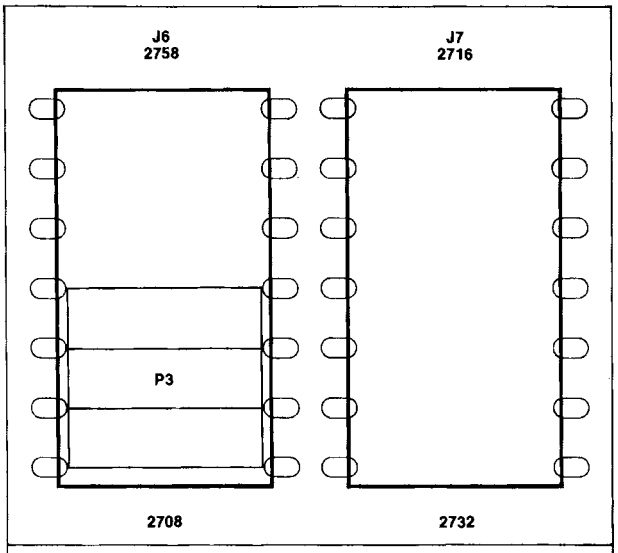
NOTE

Some Multimodules *require* the MWAIT/ signal for proper operation. The wait state jumper *cannot* be used when the MWAIT/ signal is required. Refer to your Multimodule Hardware Reference Manual for additional information.

Table 2-3. ROM/EPROM Reference Chart

EPROM[1] Type	ROM[1] Type	Capacity	Header Plug P3	Max ROM Address (Hexadecimal)
2708	2608[2]	1K x 8	J6 Lower	0FFF
2758	—	1K x 8	J6 Upper	0FFF
2716	2316E	2K x 8	J7 Upper	1FFF
2732	2332A	4K x 8	J7 Lower	3FFF

NOTES:
1. Refer to the *Intel Component Data Catalog* for device specifications.
2. Factory Programmable PROM.



Note: Shown with P3 in 2708 position.

PSI25

Figure 2-2. Device Type Sockets

Summary of the ROM installation procedure:

a. Install devices into sockets U19 through U22.

b. Configure board to correct device type, using header plug P3.

c. Select correct device access time using header plug P4.

The base or starting address of the ROM area is 0000 (Hexadecimal). The upper or maximum address of the ROM area will naturally depend on the type of devices installed. These values are provided in table 2-3. This scheme assumes that a device is residing in each of the four ROM sockets. You do not necessarily have to install a device into each socket, providing the empty sockets are not addressed. The upper address of each socket is provided in table 2-4.

2-9. RAM INSTALLATION

There are three important concepts to be considered when installing optional RAM devices:

a. RAM devices must always be installed in pairs;

b. Device placement determines address spacing;

c. Amount of ROM/EPROM installed determines RAM starting address.

Table 2-4. ROM Socket Address Ranges (Hexadecimal)

Device Type	U19	U20	U21	U22
1K x 8	0 - 03FF	0400 - 07FF	0800 - 0BFF	0C00 - 0FFF
2K x 8	0 - 07FF	0800 - 0FFF	1000 - 17FF	1800 - 1FFF
4K x 8	0 - 0FFF	1000 - 1FFF	2000 - 2FFF	3000 - 3FFF

The iSBC 80/10B board is shipped with 1K bytes of RAM residing in sockets U37 and U41. In this standard configuration the base address is 3C00 (Hex) when 1K or 2K ROM devices are installed. If 4K ROM devices are installed, the base address is 4C00 (Hex).

Additional RAM devices may be installed in 1K byte increments, with the base address being reduced accordingly (table 2-5). With the full 4K bytes of RAM, its space would cover 3000 to 3FFF with 1K or 2K ROM devices installed, or 4000 to 4FFF with 4K ROM devices installed.

Notice that with the 1K or 2K ROM devices installed, a memory gap will exist from the end of ROM space to the beginning of RAM space. More precisely, with the 1K ROM devices installed, the gap will be from 1000 to 2FFF, and with 2K ROM devices installed the gap will be from 2000 to 2FFF. If your application requires that you assign any of these addresses to memory, it will, of course, need to be off-board.

If installing additional RAM to the standard 1K byte configuration, the following jumper modification must be performed (refer to figure 5-2):

Install jumper 96 to 97

This modification indicates (to the CPU) that there are 2K to 4K bytes of RAM on-board. Without the jumper, as in the factory default configuration, 1K bytes are indicated.

2-10. LINE DRIVERS AND I/O TERMINATORS

Line drivers or I/O terminators may be installed in sockets U3 through U11. These sockets correspond to the 48 parallel I/O lines available on the iSBC 80/10B board. Table 2-6 identifies these lines and their corresponding ports. For additional information, refer to figure 5-3, sheets 7 and 8. Table 2-7 lists the types of I/O terminators and line drivers which are recommended for use with the iSBC 80/10B board. Notice that two driver or terminator devices are required for one port.

Table 2-5. RAM Socket Pairs

	Sockets	Addresses[1]
Pair 0	U37, U41	XC00 - XFFF
Pair 1	U36, U40	X800 - XBFF
Pair 2	U35, U39	X400 - X7FF
Pair 3	U34, U38	X000 - X3FF

Note:
1. X prefix will be 3 with 1K or 2K ROM devices; and 4 with 4K ROM devices installed.

Table 2-6. Parallel I/O Line Identification

I/O Port	Component Line Function	I/O Socket	PPI Device	I/O Connector Pin	Notes:
E4	Data 0	U1	U16 Port A	J1-43	Bus Transceiver is factory installed for this port (in socket U1).
	Data 1	U1		J1-41	
	Data 2	U1		J1-45	
	Data 3	U1		J1-47	
	Data 4	U1		J1-39	
	Data 5	U1		J1-37	
	Data 6	U1		J1-35	Schematic reference on figure 5-3, sheet 7.
	Data 7	U1		J1-33	

Table 2-6. Parallel I/O Line Identification (Cont.)

I/O Port	Component Line Function	I/O Socket	PPI Device	I/O Connector Pin	Notes:
E5	Data 0 Data 1 Data 2 Data 3	U5	U16 Port B	J1-7 J1-5 J1-3 J1-1	
	Data 4 Data 5 Data 6 Data 7	U4		J1-9 J1-11 J1-13 J1-15	Schematic reference on figure 5-3, sheet 7
E6	Data 7 Data 5 Data 1 Data 0	U2	U16 Port C	J1-31 J1-27 J1-29 J1-25	Table indicates factory configuration. Data lines may be altered. Refer to section 2-17 for instructions.
	Data 4 Data 6 Data 2 Data 3	113		J1-21 J1-23 J1-19 J1-17	Schematic reference on figure 5-3, sheet 7
E8	Data 0 Data 1 Data 2 Data 3	U6	U17 Port A	J2-43 J2-45 J2-47 J2-49	Schematic reference on figure 5-3, sheet 8
	Data 4 Data 5 Data 6 Data 7	U7		J2-41 J2-39 J2-37 J2-35	
E9	Data 0 Data 1 Data 2 Data 3	U11	U17 Port B	J2-5 J2-7 J2-9 J2-3	Schematic reference on figure 5-3, sheet 8
	Data 4 Data 5 Data 6 Data 7	U10		J2-11 J2-13 J2-15 J2-17	
EA	Data 0 Data 1 Data 2 Data 3	U9	U17 Port C	J2-25 J2-23 J2-21 J2-19	Schematic reference on figure 5-3, sheet 8
	Data 4 Data 5 Data 6 Data 7	U8		J2-27 J2-29 J2-31 J2-33	

2-11. JUMPER CONFIGURATIONS

Much of the flexibility of your iSBC 80/10B board is due to the use of jumper connections which may easily be altered from their factory configurations to suit your particular application. The following sections describe all the jumper connections relevant to the scope of this discussion. Locations of the referenced jumpers are shown in figure 5-2. A complete list of the jumpers with a brief functional description is provided in table 2-8.

NOTE

Jumper posts are shown on the schematic diagrams with an E prefix (e.g., E86). However, the prefix is not used on the board itself.

Table 2-7. Recommended Line Drivers and Terminators

Line Drivers	Current	I/O Terminators
7400 I	16 mA	iSBC 901
7403 I,OC	16 mA	
7408 NI	16 mA	iSBC 902
7409 NI,OC	16 mA	

I = inverting; NI = non-inverting; OC = open collector.

Table 2-8. Jumper Configurations

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
1 thru 4	Configure Port E6 bits	7 B5	2-17
5-10*	Disable E6 interrupt	7 B5	2-17
6 thru 9	Configure Port E6 bits	7 B5	2-17
11-X	Enable CTI for millisecond timer	7 C4	2-20
12-X	Enable PFSN/	7 C4	2-25
14-X	Enable PFSR/	7 C4	2-25
15-X	Enable millisecond timer (MST)	7 B4	2-20
16 thru 19	Configure Port E6 bits	7 B5	2-17
20-25*	Disable E6 interrupt	7 B5	2-17
21 thru 24	Configure Port E6 bits	7 B5	2-17
26-27*	GND to J2-1	8 B2	None
27-28	+5V to J2-1	8 B2	2-28
30-31*	Connects RTS/ to CTS/	6 A4	None
32-33*	Sets CTS driver to +12 volts	6 A4	None
34-35	Connects TxD to RS232C driver	6 B4	2-15
35-36*	Connects TxD to TTY driver	6 B4	None
37-38	Connects DTR to RS232C driver	6 B4	2-15
38-39*	Connects DTR to TTY driver	6 B4	None
41-42	Connects external clock to RxC	6 B5	2-16
41-46*	Connects TTY return to RxD	6 B5	None
42-47*	Connects internal clock to RxC	6 B5	None
43-48*	Connects internal clock to TxC	6 B5	None
44-49*	Connects DTR to DSR input	6 B5	None
45-46	Connects RS232C data to RxD	6 B5	2-15
48-49	Connects external clock to TxC	6 B5	2-16
50-54*	Selects 110 baud for PCI device (see table 2-9)	6 D4	2-13
59-60*	Sets Port E4 to output mode	7 C5	2-17
60-61	Port E4 mode programmed by Port E6 Bit 6	7 C5	2-17
63-64*	Connects HALT/ to P2 Connector	4 A4	2-21
67-68	Enables Multimodule interrupt 0 (MINTR0)	8 A6	2-24
68-69*	Disables Multimodule interrupt 0 (MINTR0)	8 A6	2-24
70-71	Enables Multimodule interrupt 1 (MINTR1)	8 A6	2-24
71-72*	Disables Multimodule interrupt 1 (MINTR1)	8 A6	2-24
73-74	Enable RxRDY interrupt	6 C4	2-14
74-75*	Disable RxRDY interrupt	6 C4	2-14
76-78*	Disable TxRDY interrupt	6 B4	2-14

Table 2-8. Jumper Configurations (Cont'd)

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
78-79	Enable TxRDY interrupt	6 B4	2-14
77-78	Enable TxE interrupt	6 B4	2-14
80-81*	Selects 110 baud for PCI device (see table 2-9)	6 D4	2-13
82-83*	Connects RESET to Multibus	2 C6	2-21
84-85*	Connects BPRN to board	2 C6	2-23
85-86	Implements BPRN/ (Multibus compatible)	2 C6	2-23
87-88*	Connects INTR0 to CPU INT	2 D7	2-21
88-89	Connects PFIN/ to CPU INT	2 D7	2-25
90-91	Connects AACK/ to board	4 B7	None
92-93*	Connects BCLK/ to Multibus	3 A3	2-21
94-95*	Connects CCLK/ to Multibus	3 A3	2-21
96-97	Specifies amount of on-board RAM	4 C7	2-9
98-99*	Reserved	5 B6	None
100-101*	Connects WAIT/ to P2 Connector	4 A2	2-21
103-104*	Connects SYNC to P2 Connector	4 C3	None
106-107*	Enables failsafe timer	4 B4	2-19
108-109	Wait State Jumper	4 A7	2-8
W1	-12V to J3-19	6 B2	2-27
W2	GND to J3-1	6 A7	None
W3	+12V to J3-22	6 C6	2-28

Note: * indicates default connection

2-12. PROGRAMMABLE COMMUNICATIONS INTERFACE HARDWARE CONFIGURATION

All serial data I/O operations are controlled by the Programmable Communications Interface (PCI) device (U18). Several hardware features of the PCI can be reconfigured, to match your application. The following features may be reconfigured:

a. Baud rate;
b. Interrupt request mechanism;
c. Interface type;
d. Transmit and Receive Clock source;
e. Control line configuration.

The iSBC 80/10B board is shipped in the following configuration:

a. Baud rate set to 110.
b. Interrupt not connected to CPU.
c. TTY (20 mA current loop) interface.
d. Internal Transmit and Receive clock connected.
e. RTS looped to CTS; DSR jumper installed.

The following sections give procedures for reconfiguring of the listed features.

2-13. BAUD RATE SELECTION. When shipped, the iSBC 80/10B board is configured for 110 baud. This rate may be reconfigured to another value, as shown in table 2-8. Physically, it will be necessary to remove the existing jumpers and then to install the jumper indicated in the table. Location of the jumpers on the board is shown in figure 5-2. Schematically, the jumper connections are shown in figure 5-3, sheet 6.

Notice that the baud rate is derived by hardware jumper placement *and* a software selected divide rate. The software baud rate factor is sent from the CPU to the PCI as part of the Mode Instruction Format byte. This procedure is described in Chapter 4.

CAUTION

A baud rate of 9600 may be obtained only with a divide factor of 16. See table 2-9.

Table 2-9. Baud Rate Selection

Jumper[1] Connection	Effective Baud Rate (Hz)		
	Synchronous Mode	Asynchronous Mode	
		Software Baud Rate Factor = 16 (see note 2)	Software Baud Rate Factor = 64 (see note 2)
57-50	—	19,200	4800
56-50	—	9600[3]	2400
55-50	—	4800	1200
51-50	38,400	2400	600
52-50	19,200	1200	300
53-50	9600	600	150
54-50	4800	300	75
54-50 80-81	6980	—	110 (TTY)

Note:
1. If jumper pair 80-81 is not connected, the frequency at jumper pole 54 is 4.8 KHZ. If jumper 80-81 is connected, however, the frequency at jumper pole 54 is 6.98 KHZ which, with a programmed baud rate factor of 64, provides an effective baud rate of approximately 110 baud for Teletype use.
2. Baud rate factor is software selectable, within above limits.
3. Caution: Baud Rate Factor = 16.
4. Jumper 58 not recommended for PCI use.

2-14. SERIAL INTERFACE INTERRUPTS. As shipped from the factory, the iSBC 80/10B board's PCI circuitry will not interrupt the CPU. Rather, its status will be read in a polled fashion, as directed by the CPU program. The board may be reconfigured to allow the PCI device to interrupt the CPU. The following list describes the three different interrupt mechanisms allowed, and indicates how to implement each type.

a. *Receiver Ready Interrupt.* Remove jumper 74 to 75; install jumper 74 to 73.

b. *Transmitter Ready Interrupt.* Remove jumper 76 to 78; install jumper 78 to 79.

c. *Transmitter Empty Interrupt.* Remove jumper 76 to 78; install jumper 78 to 77.

Figure 5-2, gives the physical location of the relevant jumper posts. They are also shown schematically in figure 5-3, sheet 6, zone C4.

2-15. CONVERSION TO RS232C INTERFACE. The iSBC 80/10B board is shipped with the serial interface channel configured to the TTY (20 mA current loop) mode. To convert this to the RS232C mode, the following modifications must be performed:

a. Remove jumper 35 to 36; install jumper 35 to 34.

b. Remove jumper 38 to 39; install jumper 38 to 37.

c. Remove jumper 46 to 41; install jumper 46 to 45.

If your application requires the use of external transmit and receive clocks in the RS232C mode, refer to section 2-16.

Figure 5-2, zone C4, gives the physical location of the relevant jumper posts. They are also shown schematically in figure 5-3, sheet 6.

2-16. SERIAL INTERFACE EXTERNAL CLOCK. In the standard configuration, the transmit and receive clocks (TxC and RxC) are driven by the on-board timer network. To use external clock sources for TxC and RxC, the following modifications are required:

a. *For TxC:* remove jumpers 43 to 48 and 44 to 49; install jumper 48 to 49.

b. *For RxC:* remove jumpers 42 to 47 and 41 to 46; install jumper 41 to 42.

Figure 5-2, zone C3, gives the physical location of the relevant jumper posts. They are also shown schematically in figure 5-3, sheet 6, zone C5.

2-17. PROGRAMMABLE PERIPHERAL INTERFACE CONFIGURATIONS

The iSBC 80/10B board utilizes two Intel 8255A

Programmable Peripheral Interface (PPI) devices to control the board's six 8-bit I/O ports. Three ports are interfaced through connector J1 and three are interfaced through connector J2. Table 2-6 provides port and line identification of the two PPI devices.

Since each port may be programmed to suit several applications, specific information pertaining to the use of each port is provided in Chapter 3. Using one port in a particular mode may impose restrictions on another port. There are also certain jumper connections which are required for certain applications. Refer to sections 3-13 through 3-22 for complete PPI programming instructions and jumper requirements.

Lines which are used as outputs will require line driver circuits for proper operation. Likewise, lines which are used as inputs require terminator networks. Recommended line/driver and terminator devices are provided in table 2-7.

Unlike the other PPI ports, each bit of port E6 is jumper connected to its driver/terminator socket pin (U2, U3). This allows greater flexibility for port configurations when using the on-board millisecond timer (section 2-20), a power-fail configuration (section 2-25) or an interrupt (section 3-20). The jumper can also be used to reconfigure the bit order of port E6.

Port E4 is default configured to the output mode, with a bus transceiver installed in socket U1. Alternatively, the bus transceiver mode (input or output) can be specified under program control by performing the following modifications:

a. Remove jumper 59 to 60.

b. Install jumper 60 to 61.

Port E6, bit 6 (PPI port C6) will then determine input mode or the output mode. Refer to section 3-18 for port C programming information.

To place the transciever in the input *only* mode, remove jumper 59-60.

2-18. OPTIONAL RAM CONFIGURATION

If you install additional RAM devices on-board, the following jumper modification is required for proper memory addressing:

Install jumper 96 to 97.

This modification indicates *more* than 1K bytes of RAM reside on-board.

This jumper is shown on figure 5-2, and schematically on figure 5-3, sheet 4, zone C7.

2-19. FAILSAFE TIMER JUMPER

The iSBC 80/10B board is equipped with an on-board failsafe timer. This timer is activated at the beginnig of every CPU machine cycle. If an acknowledge is not received within approximately 10 milliseconds, the failsafe timer expires, and issues an acknowledge signal, so the CPU may resume operation. If your application does not require this feature, the following modification must be performed:

Remove jumper 106 to 107

Figure 5-2, zone C7, gives the physical location of the relevant jumper posts. They are also shown schematically in figure 5-3, sheet 4, grid location B4.

2-20. MILLISECOND TIMER

A millisecond timer may be implemented on-board for use as a reference period. This timer does not affect the baud rate clock or the failsafe timer. The period of the millisecond timer, however, is affected by the 110 baud jumper 80-81 (Fig. 5-3, Sheet 6). With 80-81 installed, the timer signal is low for 0.7ms (1.4ms period). With 80-81 removed, the timer signal is low for 1.0ms (2.0ms period). The timer may be reset at any time by CTI, but if it is not reset it will continue as a symmetrical signal with the period described above. To implement the millisecond timer:

a. Install a jumper from timer output (post E15) to the appropriate location. For example, timer output may be connected to an 8255A port E6 input line, a port E6 output driver, or it may be used as an interrupt source. Refer to Figure 5-3, Sheet 7, zone C4.

b. Install a jumper from CTI (post E11) to the appropriate location which will control resetting the timer. For example, CTI may be connected to an 8255A port E6 output line, a port E6 input terminator, or it may be tied low. Refer to Figure 5-2, Sheet 7, zone C4.

The output of the timer may be verified at jumper post 15. The clock output is generated at U27, pin 13 and is identified as MST (Millisecond Timer). Refer to Figure 5-3, Sheet 6, zone C2. Jumper posts are located between sockets U2 and U3 on the board.

2-21. CONNECTOR INFORMATION

For systems applications, the iSBC 80/10B board is designed for installation into a standard Intel iSBC 604 or iSBC 614 Cardcage and Backplane assembly. Alternatively, the board may be interfaced with a design of your own choice, by means of an 86-pin connector. Refer to table 2-10 for a list of suggested manufacturers.

Multibus signal characteristics and methods of implementing a priority resolution scheme for bus

contention in a dual master system are described in the following sections.

CAUTION

Always turn off system power before installing or removing the board. Failure to observe this precaution can cause damage to the board.

Table 2-10 also provides a list of serial and parallel I/O connectors.

2-22. MULTIBUS SIGNAL CHARACTERISTICS

Connector P1 interfaces the iSBC 80/10B board to the Multibus connector. Pin assignments are listed in

Table 2-10. User Furnished Connector Details

Function	Pins	Centers (inches)	Connector Type	Vendor	Vendor Part Number
Multibus Connector P1	43/86	0.156	Solder PCB	ELFAB VIKING	BS1562043PBB 2KH43/9AMK12
			Wire Wrap (no ears)	EDAC ELFAB	337-086-0540-201 BW1562D-43PBB
			Wire Wrap (with 0.128 mounting holes)	EDAC ELFAB	337-086-540-202 BW1562A-43PBB
Auxiliary Connector P2	30/60	0.100	Wire Wrap	EDAC ELFAB	345-060-524-802 BS1020A-30PBB
			With 0.128 mounting holes	TI VIKING	H421121-30 3KH30/9JNK
			No Ears	EDAC ELFAB	345-060-540-201 BW1020D-30PBB
Parallel Port	35/50	0.100	Flat Crimp	3M 3M AMP ANSLEY SAE	3415-0001 (w/o ears) 3415-0000 (w/ears) 88083-1 609-5015 S06750 Series
			Soldered	GTE MASTERITE MICROPLASTICS VIKING	6AD01-25-1A1-DD NDD8GR25-DR-H-X MP-0100-25-DP-1 3KH25/9JN5
			Wire Wrap	VIKING TI ITT CANNON	3KH25/JND5 H421011-25 EC4A050A1A
Serial Port	13/26	0.100	PCB Soldered mounting holes	AMP EDAC	1-583715-1 345-026-520-202
			Flat Crimp	3M AMP	3462-0001 88373-5
			Soldered, pierced tail	EDAC	345-026-500-201
			Wire Wrap	EDAC	345-026-540-201

Notes:
1. Connector heights are not guaranteed to conform to OEM equipment.
2. Wire wrap pin lengths are not guaranteed to conform to OEM equipment.
3. Connector numbering convention may not agree with board connector.

table 2-11 and descriptions of the signal functions are provided in table 2-12.

The DC characteristics of the iSBC 80/10B board Multibus interface signals are provided in table 2-13. The AC characteristics with continuous bus control, are provided in table 2-14. AC characteristics with bus exchange are provided in table 2-15. Bus exchange timing is shown in figure 2-3. Memory and I/O read timing is shown in figure 2-4 and memory and I/O write timing is shown in figure 2-5.

Table 2-11. P1 and P2 Connector Pin Assignments

P1 Pin Assignments		
Pin*	**Signal**	**Function**
1	GND	Ground
2	GND	
3	+5V	Power Input
4	+5V	
5	+5V	
6	+5V	
7	+12V	
8	+12V	
9	-5V	
10	-5V	
11	GND	Ground
12	GND	
13	BCLK/	Bus Clock
14	INIT/	System Initialize
15	BPRN/	Bus Priority In
17	BUSY/	Bus Busy
19	MRDC/	Memory Read Command
20	MWTC/	Memory Write Command
21	IORC/	I/O Read Command
22	IOWC/	I/O Write Command
23	XACK/	Transfer Acknowledge
25	AACK/	Advanced Acknowledge
31	CCLK/	Constant Clock
33	INTA/	Interrupt Acknowledge
42	INT1/	Interrupt request on level 1

P2 Pin Assignments		
Pin*	**Signal**	**Function**
3	+5Vdc	Power
4	+5Vdc	Power
13	PFSR/	Power Fail Sense Return
17	PFSN/	Power Fail Sense
19	PFIN/	Power Fail Interrupt
20	MPRO/	Memory Protect
28	HALT/	
29	WAIT/	
32	SYNC/	
38	AUX RESET/	Auxiliary Reset

Note: All other P2 pins are reserved for future use by Intel.

*All odd-numbered pins (1, 3, 5 . . . 85) are on component side of the board. Pin 1 is the left-most pin when viewed from the component side of the board with the extractors at the top. All unassigned pins are reserved.

Table 2-12. Signal Functions Used By the iSBC 80/10B™ Board

Signal	Functional Description
ADR0/-ADRF/	*Address.* These 16 lines transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most-significant bit.
BCLK/	*Bus Clock.* Used to synchronize the bus contention logic on all bus masters. When generated by the iSBC 80/30, BCLK/ has a period of 108 nanoseconds (9.22MHz) with a 35-65 percent duty cycle.
BPRN	*Bus Priority In.* Indicates to a particular bus master that no higher priority bus master is requesting use of the bus.
BUSY/	*Bus Busy.* Indicates that the bus is in use and prevents all other bus masters from gaining control of the bus. BUSY/ is synchronized with BCLK/.
CCLK/	*Constant Clock.* Provides a clock signal of constant frequency for use by other system modules. When generated by the iSBC 80/10B board, CCLK/ has a period of 108 nanoseconds (9.22 MHz) with a 35-65 percent duty cycle.
DAT0/-DAT7/	*Data.* These eight bidirectional data lines transmit and receive data to and from the addressed memory location or I/O port. DAT7/ is the most-significant bit.
INIT/	*Initialization.* Resets the entire system to a known internal state.
IORC/	*I/O Read Command.* Indicates that the address of an I/O port is on the Multibus address lines and that the output of that port is to be read (placed) onto the Multibus data lines.
IOWC/	*I/O Write Command.* Indicates that the address of an I/O port is on the Multibus address lines and that the contents on the Multibus data lines are to be accepted by the addressed port.
MRDC/	*Memory Read Command.* Indicates that the address of a memory location is on the Multibus address lines and that the contents of that location are to be read (placed) on the Multibus data lines.
MWTC/	*Memory Write Command.* Indicates that the address of a memory location is on the Multibus address lines and that the contents on the Multibus data lines are to be written into that location.
XACK/	*Transfer Acknowledge.* Indicates that the addressed memory location has completed the specified read or write operation. That is, data has been placed onto or accepted from the Multibus data lines.

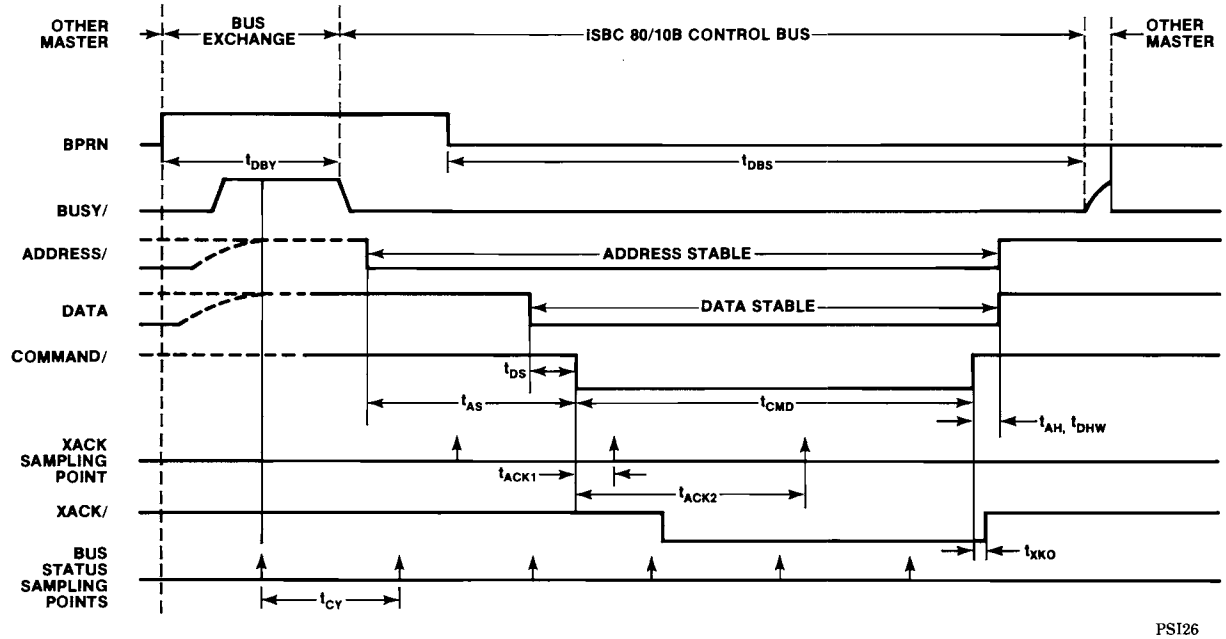


Figure 2-3. Bus Exchange Timing (Write)

Table 2-13. iSBC 80/10B DC Characteristics

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
ADR0/-ADRF	V_{OL}	Output Low Voltage	I_{OL} = 24 mA		0.4	V
ADDRESS	V_{OH}	Output High Voltage	I_{OH} = 15 mA	2.4		V
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-0.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5		20	μA
	*C_L	Capacitive Load			18	pF
MRDC/, MWTC/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.4	V
IORC/, IOWC/	V_{OH}	Output High Voltage	I_{OH} = -5.2 mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 2.4		40	μA
	I_{LL}	Output Leakage Low	V_O = 0.4		-40	μA
	*C_L	Capacitive Load			15	pF
DAT0/-DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = -5 mA	2.4		V
	V_{IL}	Input Low Voltage			0.9	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-0.2	mA
	I_{LH}	Output Leakage High	V_O = 5.25		50	μA
	I_{LL}	Output Leakage Low	V_O = 0.45		200	μA
	*C_L	Capacitive Load			18	pF
INT1/	V_{IL}	Input Low Voltage		0.8	V	
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-2.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5		1	mA
	*C_L	Capacitive Load			18	pF
BPRN/, XACK/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = .5		-2.1	μA
	I_{IH}	Input Current at High V	V_{IN} = 2.7		.3	mA
	*C_L	Capacitive Load			18	pF
BUSY/	V_{OL}	Output Low Voltage	I_{OL} = 48 mA		0.4	V
OPEN COLLECTOR	*C_L	Capacitive Load			18	pF
INT	V_{OL}	Output Low Voltage	I_{OL} = 48 mA		0.4	V
(SYSTEM RESET)	V_{OH}	Output High Voltage	Open Collector			
	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IH}	Input Current at High V	V_{IN} = 5.5		0.2	mA
	I_{IL}	Input Current at Low V	V_{IN} = 0.3		-0.9	mA
	*C_L	Capacitive Load			18	pF
BCLK/, CCLK/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = -5.2 mA	2.4		V
	*C_L	Capacitive Load			18	pF

*Capacitive values are approximations only.

Table 2-13. iSBC 80/10B DC Characteristics (Cont'd)

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
EXT INTR0/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		-6.8	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.5V		2	mA
	*C_L	Capacitive Load			18	pF
PORT E4 BIDIRECTIONAL DRIVERS	V_{OL}	Output Low Voltage	I_{OL} = 27 mA		.5	V
	V_{OH}	Output High Voltage	I_{OH} = -7 mA	2.4		V
	V_{IL}	Input Low Voltage			.9	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		-5.2	mA
	I_{LH}	Output Leakage High	V_O = 5.25		.3	mA
	*C_L	Capacitive Load			18	pF
8255A DRIVER/ RECEIVER	V_{OL}	Output Low Voltage	I_{OL} = 1.7 mA		.45	V
	V_{OH}	Output High Voltage	I_{OH} = -50μA	2.4		V
	V_{IL}	Input Low Voltage			.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		10	μA
	I_{IH}	Input Current at High V	V_{IN} = 5.0		10	μA
	*C_L	Capacitive Load			18	pF

*Capacitive values are approximations only.

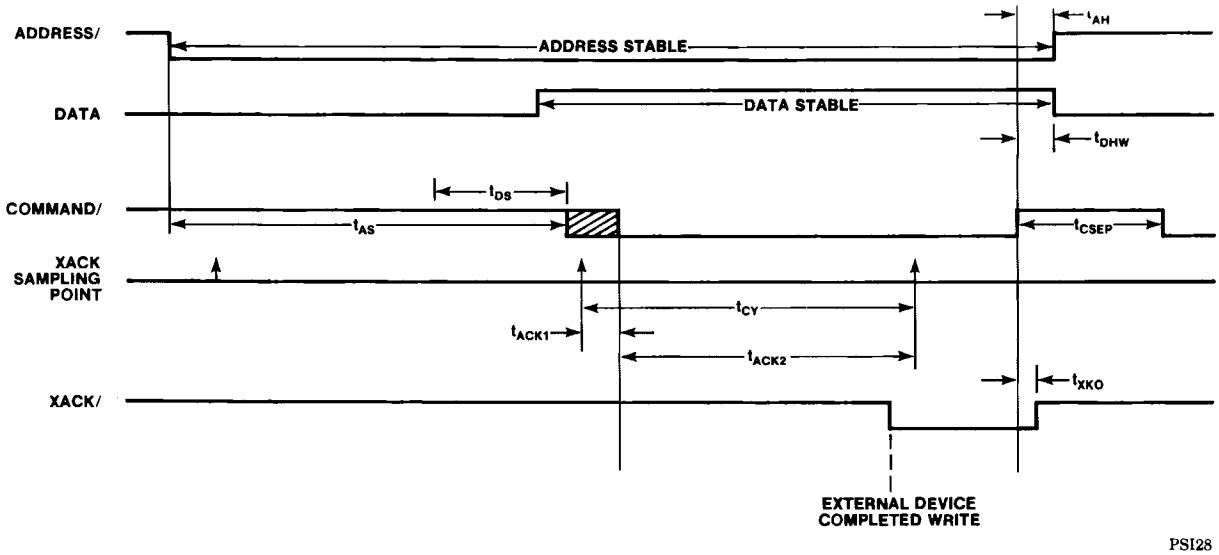


Figure 2-4. Memory and I/O Read Timing

Table 2-14. AC Characteristics with Continuous Bus Control

Multibus Standard Parameter	iSBC 80/10A Former Parameter	Overall		Continuous Bus Control				Description	Remarks
				Read		Memory Write			
		Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)		
t_{AS}	t_{AS}	82		82		658		Address Setup Time to Command	
t_{AH}	t_{AH}	79		0		79		Address Hold Time	
t_{DS}	t_{DS}	113		—		113		Data Setup Time to Command	
t_{DHR}, t_{DHW}	t_{DH}	79		0		79		Data Hold Time	
t_{XACK}	t_{ACK0}			63	191			1st ACK Sampling Point of Current Cycle	Generates 0 Wait States
	t_{ACK1}			546	684	-84	137	2nd ACK Sampling Point of Current Cycle	Generates 1 Wait State
	t_{ACK2}			1029	1177	399	630	3rd ACK Sampling Point of Current Cycle	Generates 2 Wait States
t_{CY}	t_{CY}	483	493					ACK & BPRN Sample Cycle Time	
t_{CSEP}	t_{SEP}	259		613		259	Note 2	Command Separation	
t_{CMD}	t_{WC}			596	840	1412	1516	Command Width	Read, 0 Wait States Write, 2 Wait States
t_{ACC}	t_{ACC}	344			344			Read Access Time	See Note 1
	t_{XKD}	0		0				XACK Delay from Valid Data or Write	
	t_{XKO}	0	100	0	100	0	100	XACK Turn Off Delay	
t_{BCY}	t_{BCY}	107	110					Bus Clock Cycle Time	80/10 Generator
t_{BW}	t_{BW}	25	85					Bus Clock Low or High Periods	80/10 Generator
t_{INT}	t_{INT}	3000						Initialization Width	After all voltages have stabilized

NOTES:
1. Max assumes no acknowledge delays.
2. Write Command to next Read Command separation.

Table 2-15. AC Characteristics with Bus Exchange

Multibus Standard Parameter	iSBC 80/10A Former Parameter	Overall		Continuous Bus Control				Description	Remarks
				Read		Memory Write			
		Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)		
t_{AS}	t_{AS}	82		82		658		Address Setup Time to Command	
t_{AH}	t_{AH}	61		0		61		Address Hold Time	
t_{DS}	t_{DS}	113		—		113		Data Setup Time to Command	
t_{DHR}, t_{DHW}	t_{DH}	61		0		61		Data Hold Time	
t_{ACK}	t_{ACK0}			63	191			1st ACK Sampling Point of Current Cycle	Generates 0 Wait States
	t_{ACK1}			546	684	-84	132	2nd ACK Sampling Point of Current Cycle	Generates 1 Wait State
	t_{ACK2}			1029	1174	399	630	Third ACK Sampling Point of Current Cycle	Generates 2 Wait States
t_{CY}	t_{CY}	483	493					ACK & BPRN Sample Cycle Time	
t_{CMD}	t_{WC}			596	840	1412	1516	Command Width	Read, 0 Wait States Write, 2 Wait States
t_{ACC}	t_{ACC}				344			Read Access Time	See Note 1
	t_{XKD}	0		0				XACK Delay from Valid Data or Write	
	t_{XKO}	0	100	0	100	0	100	XACK Turn Off Delay	
	t_{DBS}		3500					Bus Sample to Exchange Initiation	See Note 1
t_{DBY}	t_{DBY}	544	1217					Bus Busy Turn On Delay	

Note:
1. Memory and I/O access occurs without wait states.

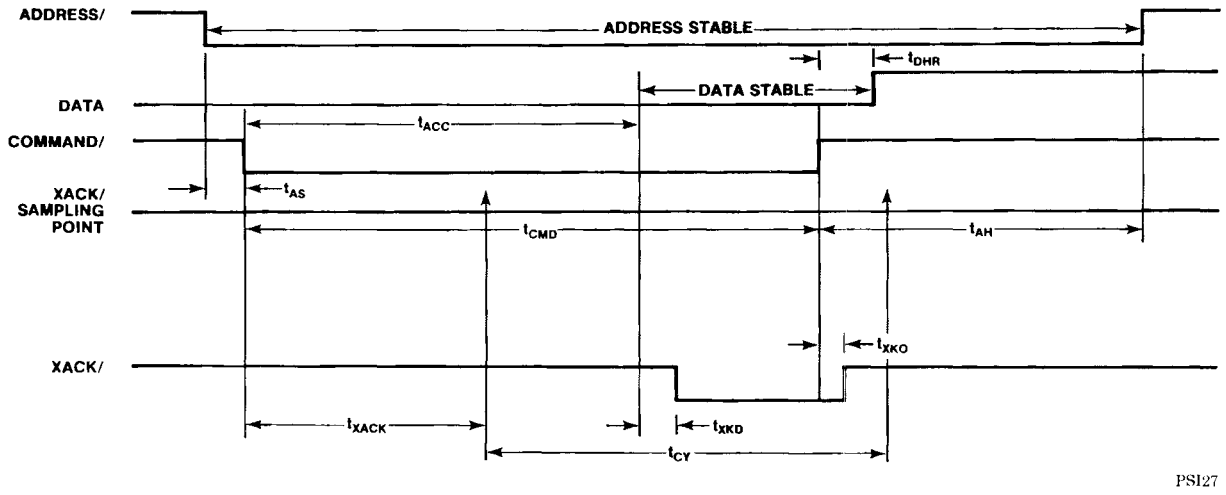


Figure 2-5. Memory and I/O Write Timing

2-23. PRIORITY RESOLUTION

If the iSBC 80/10B board is the only master in your system, it may be placed into any slot in the iSBC 604/614 Cardcage and Backplane which *does not* have pin 15 (BPRN/) grounded.

The iSBC 80/10B board bus structure allows you to interface one other Multibus compatible master to your system. In this configuration, the iSBC 80/10B board will *always* have the *lower* priority of the two masters. There are two methods of implementing the dual master configuration in the iSBC 604/614 Cardcage and Backplane. The first method is recommended for full Multibus line compatibility. The second method should be used only when the iSBC 80/10B board is being used as a direct replacement for the iSBC 80/10A board in an existing system.

The first method is illustrated in figure 2-6. In this configuration the other master is placed into any cardcage slot which has pin 15 (BPRN/) of the backplane grounded. Jumper pair 84 to 85 on the iSBC 80/10B board must then be removed and replaced with a jumper between 85 and 86. The iSBC 80/10B board would then be placed in an adjacent slot to the other master for proper operation. To illustrate, the first method is implemented in the following example:

a. Place other master in cardcage slot J2.
b. Ground J2 pin 15 (BPRN/) by installing a jumper wire between wire wrap posts B and N (figure 2-6).
c. Remove jumper 84 to 85 on the iSBC 80/10B board. Install a jumper between 85 and 86.
d. Place the iSBC 80/10B board in slot J3.

The second method of implementing the dual master priority configuration is shown in figure 2-7. In this configuration the other master is also placed into any cardcage slot which has pin 15 (BPRN/) grounded. A jumper must then be installed between pin 18 of the other master and pin 15 of the iSBC 80/10B board. A slot must also be skipped in this scheme. To illustrate, the second method is implemented in the following example:

a. Place other master in cardcage slot J2.
b. Add a jumper wire between wire wrap posts A and E on the backplane (figure 2-7).
c. Install the iSBC 80/10B board into slot J4.
d. Alternatively, the iSBC 80/10B board could be placed in slot J5. In this case, the jumper wire would be installed between wire wrap posts A and H on the backplane.

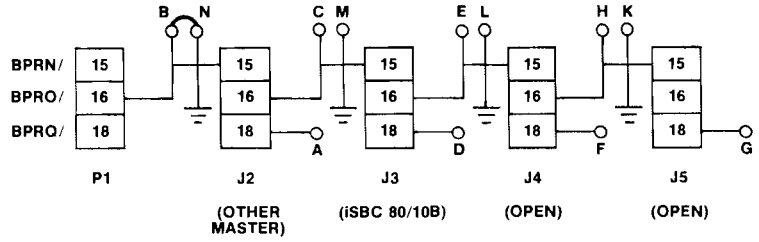


Figure 2-6. Multibus™ Compatible Priority Resolution

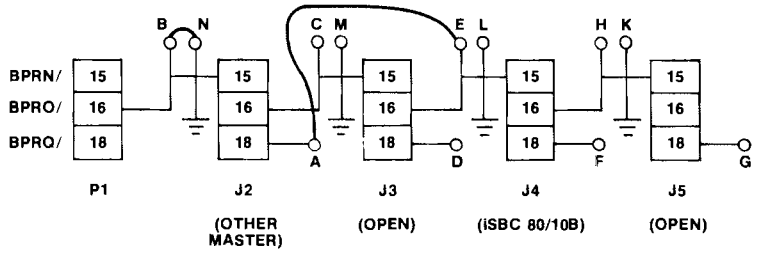


Figure 2-7. Replacement Method Priority Resolution

2-24. MULTIMODULE CONFIGURATION

The iSBC 80/10B board is Multimodule compatible. Multimodule boards are special purpose, add-on circuit boards which reside directly on the component side of the iSBC 80/10B board and are interfaced through connector J4. For specific Multimodule installation instructions, refer to the corresponding Multimodule hardware reference manual.

2-25. POWER FAIL AND MEMORY PROTECT CONFIGURATION

A power-fail and memory protect scheme may be implemented to preserve RAM contents in the event of an AC line failure. There are many ways to implement such a scheme, however Intel does not recommend a specific method. A typical memory protect scheme could be implemented as follows:

NOTE

Detailed timing information for a typical power fail routine is provided in the *Intel Multibus Specification,* Order Number 9800683.

a. Connect +5 volt battery supply leads to P2-4 and P2-3 connectors. Connect returns to P2-1 and P2-2.

b. Remove or cut W4 connection shown on figure 5-3, sheet 1, zone D6.

c. Remove jumper 87 to 88 and install jumper 88 to 89. Refer to figure 5-3, sheet 2, zone D7. This connection enables PFIN/ to interrupt the CPU.

d. Connect MPR0/ to P2-20. Refer to figure 5-3, sheet 4, zone D7. MPR0/ must be generated by an external, battery operated latch, which is triggered by ACLO and PFIN/.

e. Connect PFSN/ to P2-17 and PFSR/ to P2-13. Refer to figure 5-3, sheet 7. The PFSN/ input and the PFSR/ output should be connected to Port C on PPI circuit U16, using jumper posts 12 and 14, respectively. This enables the PPI to provide status to the CPU, to identify the power-fail interrupt as such, and subsequently reset the PFSR/ signal upon restart.

NOTE

Ensure that the upper and lower portions of port C of the PPI circuit are configured to the appropriate mode to allow operation with PFSN/ and PFSR/. Refer to Chapter 3 for PPI programming information.

2-26. USING RMX-80 SOFTWARE

If your iSBC 80/10B board is used with Intel RMX-80 software, some hardware modifications may be required for proper operation. Depending on the nature of your application and the types of extensions used, the iSBC 80/10B board may require re-configuration of the following:

a. Baud rate selection jumpers.

b. Clock source jumper.

c. Interrupt jumper.

d. PCI control line jumpers.

e. Interface jumpers.

In addition, user software may require modification to initialize and format the Programmable Peripheral Interface devices and the Programmable Communication Interface device.

Refer to the *RMX/80 User's Guide,* Manual Order Number 9800522, for complete instructions.

2-27. SERIAL I/O CABLING

Pin assignments and signal definitions for RS232C serial I/O interface are listed in table 2-16. An Intel iSBC 955 Cable Set is recommended for RS232C interfacing. One cable assembly consists of a 25-conductor flat cable with a 26-pin PC connector at one end and an RS232C interface connector at the other end. The second cable assembly includes an RS232C connector at one end and has spade lugs at the other end; the spade lugs are used to interface to a teletypewriter. (See Appendix B for ASR-33 TTY interface instructions.)

Table 2-16. Pin Assignments for Connector J3 (Serial I/O Interface)

Pin	Signal Name	Pin	Signal Name
1	CHASSIS GND	2	—
3	TRANSMITTED DATA	4	—
5	RECEIVED DATA	6	TTY RD CONTROL
7	REQ TO SEND	8	—
9	CLEAR TO SEND	10	—
11	DATA SET READY	12	—
13	GND	14	Tx CLK/DATA TERMINAL RDY
15	DATA CARRIER RETURN	16	TTY RD CONTROL RETURN
17	—	18	—
19	-12 Volts	20	—
21	—	22	RECEIVE CLK/TTY Rx DATA RETURN/+12
23	TTY Rx DATA	24	TTY Tx DATA RETURN
25	TTY Tx DATA	26	GND

Note:
Even numbered pins are on the component side of the board. Pin number 1 is on extreme right of solder side (as viewed from component side).

+12 and -12 Volt Outputs

A +12 volt level may be applied to J3-22 if this pin is unused in your configuration (refer to figure 5-3, sheet 6), by installing jumper W3. Likewise, a -12 volt level will be applied to J3-19 if jumper W1 is installed.

CAUTION

Ensure connector J3 is properly installed and the pins are connected to appropriate loads. Damage may result from improper connection.

For OEM applications where cables will be made for the iSBC 80/10B board, it is important to note that the mating connector for J3 has 26 pins whereas the RS232C connector has 25 pins. Consequently, when connecting the 26-pin mating connector to 25-conductor flat cable, be sure that the cable makes contact with pins 1 and 2 of the mating connector and not with pin 26. Table 2-17 provides pin correspondence between connector J3 and the RS232C connector. When attaching the cable to J3, be sure that the PC connector is oriented properly with respect to pin 1 on the edge connector. (Refer to the footnote in table 2-16.)

2-28. PARALLEL I/O CABLING

Parallel I/O ports E4, E5, E6, E8, E9, and EA are controlled by two Intel 8255A Programmable Peripheral Interface (PPI) devices and interfaced via edge connector J1 and J2. Pin assignments for J1 and J2 are listed in table 2-6; dc characteristics of the parallel I/O signals are given in table 2-13. Table 2-10 lists some 50-pin edge connectors that can be used for interface to J1 and J2; flat crimp, solder, and wirewrap connector types are listed.

An Intel iSBC 956 Cable Set, consisting of two cable assemblies, is recommended for parallel I/O interfacing.

Both cable assemblies consist of a 50-conductor flat cable with a 50-pin PC connector at one end. When attaching the cable to J1 or J2, be sure that the connector is oriented properly with respect to pin 1 on the edge connector. (Refer to the footnote in table 2-16.)

The transmission path from the I/O source to the iSBC 80/10B board should be limited to 3 meters (10 feet) maximum.

The following bulk cable types (or equivalent) are recommended for interfacing with the parallel I/O ports:

a. Cable, flat, 50-conductor, 3M 3306-50.

b. Cable, flat, 50-conductor (with ground plane), 3M 3380-50.

c. Cable, woven, 25-pair, 3M 3321-25.

Table 2-17. J3/RS232C Connector Pin Correspondence

Signal Name	J3 Connector Pin No.	RS232C Connector Pin No.
CHASSIS GND	1	1
	2*	14
TRANSMITTED DATA	3	2
	4*	15
RECEIVED DATA	5	3
TTY RD CONTROL	6	16
REQ TO SEND	7	4
	8	17
CLEAR TO SEND	9	5
	10*	18
DATA SET READY	11	6
	12*	19
GND	13	7
Tx CLK/DATA TERMINAL RDY	14	20
DATA CARRIER RETURN	15	8
TTY RD CONTROL RETURN	16	21
	17*	9
	18*	22
-12 VOLTS	19	—
	20*	23
	21*	11
RECEIVE CLK/TTY Rx DATA RETURN/+12V	22	24
TTY Rx DATA	23	12
TTY Tx DATA RETURN	24	25
TTY Tx DATA	25	13
	26*	—

Note:
* = Not Used on iSBC 80/10B board.

5 Volt Output

In the standard factory configuration, pin J2-1 is grounded (refer to figure 5-3, sheet 8). As an optional connection, you may apply +5 volts to this pin by removing jumper pair 26-27 and installing 27-28.

CAUTION

Ensure connector J2 is properly installed and pin 1 is connected to an appropriate load. Damage may result from improper connection.

2-29. BOARD INSTALLATION

CAUTION

Always turn off the computer system power supply before installing or removing the iSBC 80/10B board and before installing or removing interface cables. Failure to take these precautions can result in damage to the board.

In an iSBC Single Board Computer based system, install the iSBC 80/10B board in any slot that has not been wired for a dedicated function. In an Intellec System, install the board in any slot except slot 1 or 2. Make sure that auxiliary connector P2 (if used) mates correctly with the user-installed connector. Attach the appropriate cable assemblies to connectors J1 through J3.

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

This chapter provides programming instructions for the two Programmable Peripheral Interface devices (U16, U17) and the Programmable Communications Interface device (U18). Memory addressing and system initialization information is also provided.

3-2. MEMORY ADDRESSING

The iSBC 80/10B board can accommodate up to 16K bytes of ROM/EPROM on-board. Additionally, up to 4K bytes of RAM may be installed on-board. However, it is only in this maximum configuration that memory will be continuous, from 0000 to 4FFF (Hexadecimal). In this maximum configuration, ROM space runs from 0000 to 3FFF, and RAM space runs from 4000 to 4FFF. None of this memory may be addressed by another board or processor.

If you are using the iSBC 80/10B board in another configuration, less than the maximum, a gap in continous memory appears between the end of ROM space and the beginning of RAM space. This information is summarized in table 3-1. Memory addresses which fall into the gap zone may be assigned off-board, and accessed by the iSBC 80/10B board via the Multibus lines.

If non-existant memory is addressed, the on-board failsafe timer will expire in approximately 10 milliseconds, sending an acknowledge signal to the CPU so that it may resume processing. For more information on the failsafe timer, refer to section 2-18.

Table 3-1. Memory Addressing

ROM Devices	Address Space
1K x 8	0000 - 0FFF
2K x 8	0000 - 1FFF
4K x 8	0000 - 3FFF
RAM Devices	**Address Space**
1K	XC00 - XFFF
2K	X800 - XBFF
3K	X400 - X7FF
4K	X000 - X3FF

NOTE:
X prefix will be 3 with 1K or 2K ROM devices installed; and 4 with 4K ROM devices installed. Refer to Chapter 2.

When the CPU is addressing *on-board* memory (ROM/PROM or RAM), an internal PROM or RAM Acknowledge (ACK/) is automatically generated to prevent imposing a CPU wait state. When the CPU is addressing *system* memory via the Multibus lines, the CPU must first gain control of the Multibus lines and, after the Memory Read or Memory Write Command is given, must wait for a Transfer Acknowledge (XACK/) to be received from the addressed memory device.

3-3. I/O ADDRESSING

The on-board 8080A microprocessor (CPU) communicates with the programmable devices through a sequence of I/O Read and I/O Write Commands. As shown in table 3-2, each of these devices recognizes several separate hexadecimal I/O addresses that are used to control the various programmable functions. Where two hexadecimal addresses are listed for a single function, either address may be used. For example, an I/O Read Command to ED or EF will read the status of the 8251A PCI.

NOTE

The on-board I/O functions are not accessible to another bus master via the Multibus lines.

Table 3-2. I/O Addressing

I/O Address	Device	Function
E4	U16 PPI	Read/Write Port A (J1)
E5		Read/Write Port B (J1)
E6		Read/Write Port C (J1)
E7		Write: Control Byte
E8	U17 PPI	Read/Write Port A (J2)
E9		Read/Write Port B (J2)
EA		Read/Write Port C (J2)
EB		Write: Control Byte
EC or EE	U18 PCI	Write: Data (J3)
		Read: Data (J3)
ED or EF		Write: Mode or Command
		Read: Status
F0-F7	If Multimodule is installed	Multimodule MCS0/ (J4)
F8-FF		Multimodule MCS1/ (J4)

3-4. SYSTEM INITIALIZATION

When power is initially applied to the system, an Initialize (INIT/) signal is automatically generated that clears the internal Program Counter, Instruction Register, and Interrupt Enable flip-flop and "resets" the 8251A PCI, and the 8255A PPI devices as follows:

a. The 8251A PCI is set to an "idle" mode, waiting for a set of Command Words to program the desired function.

b. All three ports of each 8255A PPI are set to the input mode.

The INIT/ signal is also gated onto the Multibus lines to set the remainder of the system components to a known internal state.

The INIT/ signal can also be generated by an auxiliary RESET switch. Pressing and releasing the RESET switch produces the same effect as the INIT/ signal described above.

3-5. 8251A PCI (USART) PROGRAMMING

The PCI converts parallel output data into virtually any serial output data format (including IBM Bi-Sync) for half- or full-duplex operation. The PCI also converts serial input data into parallel data format.

Prior to starting transmitting or receiving data, the PCI must be loaded with a set of control words. These control words, which define the complete functional operation of the PCI, must immediately follow a reset (internal or external). The control words are either a Mode instruction or a Command instruction.

3-6. MODE INSTRUCTION FORMAT

The Mode instruction word defines the general characteristics of the PCI and must follow a reset operation. Once the Mode instruction word has been written into the PCI, sync characters or command instructions may be inserted. The Mode instruction word defines the following:

a. For Sync Mode:
   (1) Character length
   (2) Parity enable
   (3) Even/odd parity generation and check
   (4) External sync detect (not supported by iSBC 80/10B
   (5) Single or double character sync

   (4) Even/odd parity generation and check
   (5) Number of stop bits

b. For Async Mode:
   (1) Baud rate factor (X1, X16, or X64)
   (2) Character length
   (3) Parity enable

Instruction word and data transmission formats for synchronous and asynchronous modes are shown in figures 3-1 through 3-4.

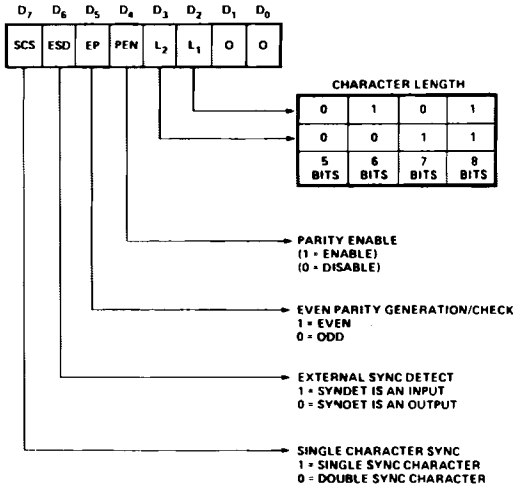


Figure 3-1. PCI Synchronous Mode Instruction Word Format

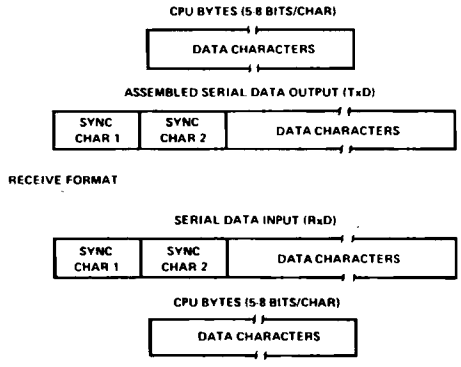


Figure 3-2. PCI Synchronous Mode Transmission Format

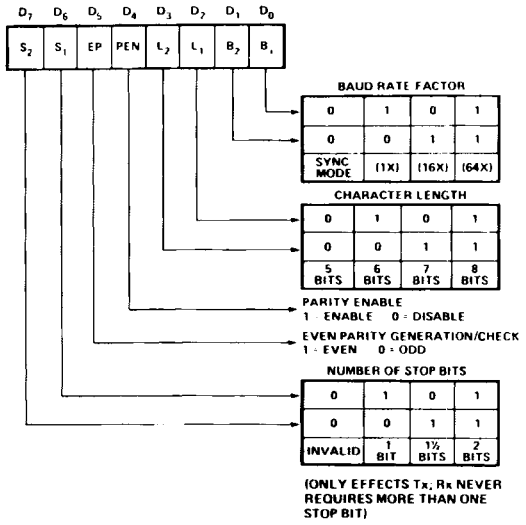


Figure 3-3. PCI Asynchronous Mode Instruction Word Format

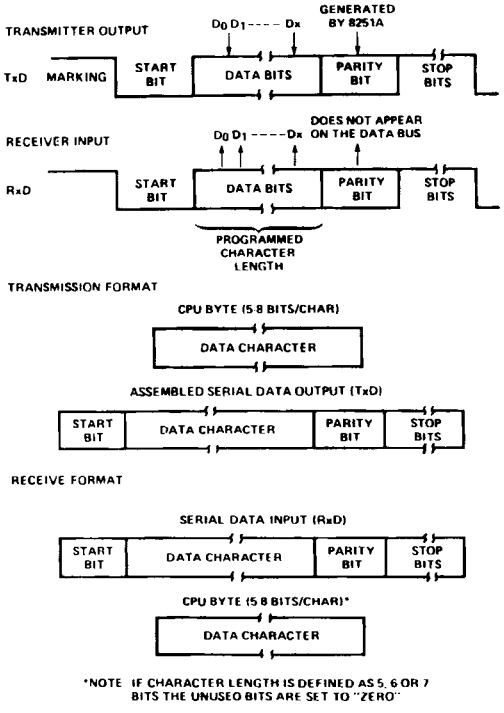


Figure 3-4. PCI Asynchronous Mode Transmission Format

3-7. SYNC CHARACTERS

Sync characters are written to the PCI in the synchronous mode only. The PCI can be programmed for either one or two sync characters; the format of the sync characters is at the option of the programmer.

3-8. COMMAND INSTRUCTION FORMAT

The Command instruction word shown in figure 3-5 controls the operation of the addressed PCI. A Command instruction must follow the mode and/or sync words and, once the Command instruction is written, data can be transmitted or received by the PCI.

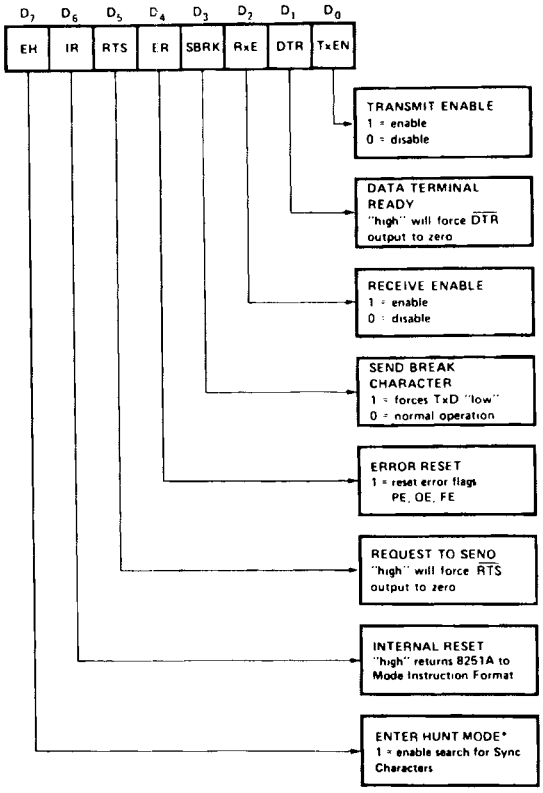


Figure 3-5. PCI Command Instruction Word Format

It is not necessary for a Command instruction to precede all data transactions; only those transactions that require a change in the Command instruction. An example is a change in the enable transmit bit or enable receive bit. Command instructions can be written to the PCI at any time after one or more data operations.

After initialization, always read the chip status and check for the TxRDY bit prior to writing either data or command words to the PCI. This ensures that any prior input is not overwritten and lost. Note that issuing a Command instruction with bit 6 (IR) set will return the PCI to the Mode instruction format.

3-9. RESET

To change the Mode instruction word, the PCI must receive a Reset command. The next word written to the PCI after a Reset command is assumed to be a Mode instruction. Similarly, for sync mode, the next word after a Mode instruction is assumed to be one or more sync characters. All control words written into the PCI after the Mode instruction (and/or the sync character) are assumed to be Command instructions.

3-10. ADDRESSING

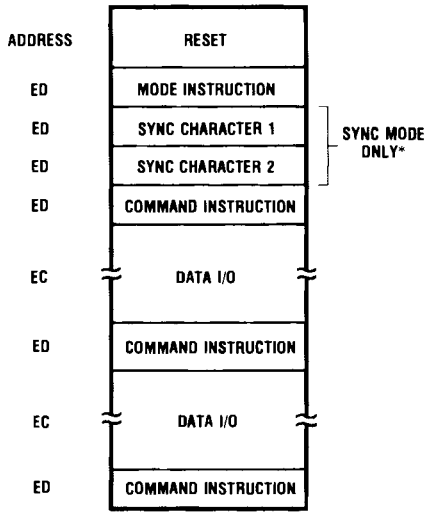
The PCI device uses two consecutive pairs of addresses. The lower of the two addresses in each pair is used to read and write I/O data; the upper address in each pair is used to write mode and command words and to read the PCI status. (Refer to table 3-2.)

3-11. INITIALIZATION

A typical PCI initialization and I/O data sequence is presented in figure 3-6. The PCI device is initialized in four steps:

a. Reset PCI to Mode instruction format.

b. Write Mode instruction word. One function of mode word is to specify synchronous or asynchronous operation.



*The second sync character is skipped if Mode instruction has programmed PCI to single character internal sync mode. Both sync characters are skipped if Mode instruction has programmed PCI to async mode.

Figure 3-6. Typical PCI Initialization and Data I/O Sequence

c. If synchronous mode is selected, write one or two sync characters as required.

d. Write Command instruction word.

To avoid spurious interrupts during PCI initialization, disable the PCI interrupt. This can be done by disabling the CPU interrupts by executing a DI instruction.

First, reset the PCI device by writing a Command instruction to location ED (or EF). The Command instruction must have bit 6 set (IR6=1); all other bits are immaterial.

NOTE

This reset procedure should be used only if the PCI has been completely initialized, or if the initialization procedure has reached the point that the PCI is ready to receive a Command word. For example, if the reset command is written when the initialization sequence calls for a sync character, then subsequent programming will be in error.

Next, write a Mode instruction word to the PCI. (See figures 3-1 through 3-4.) A typical subroutine for writing both Mode and Command instructions is given in table 3-3.

If the PCI is programmed for the synchronous mode, write one or two sync characters depending on the transmission format.

Finally, write a Command instruction word to the PCI. Refer to figure 3-5 and table 3-3.

3-12. OPERATION

Normal operating procedures use data I/O read and write, status read, and Command instruction write operations. Programming and addressing procedures for the above are summarized in the following paragraphs.

NOTE

After the PCI has been initialized, always check the status of the TxRDY bit *prior* to writing data or writing a new command word to the PCI. The TxRDY bit *must* be *true* to prevent overwriting and subsequent loss of command or data words. The TxRDY bit is inactive until initialization has been completed; therefore, do not check TxRDY until after the command word, which concludes the initialization procedure, has been written.

Table 3-3. Typical PCI Mode or Command Instruction Subroutine

```
;CMD2 OUTPUTS CONTROL WORD OR MODE WORD TO PCI.
;USES-STAT; DESTROYS-NOTHING.

        EXTRN   STAT

CMD2:   PUSH    PSW             ;SAVE DATA AND CPU STATUS
LP:     CALL    STAT            ;GET PCI STATUS
        ANI     1               ;CHECK TXRDY
        JZ      LP              ;TXRDY MUST BE TRUE
        POP     PSW             ;RESTORE DATA AND CPU STATUS;
        OUT     OEDH            ;SEND COMMAND/MODE WORD TO PCI
        RET

        END
```

Prior to any operating change, a new command word must be written with command bits changed as appropriate. (Refer to figure 3-5 and table 3-3.)

For data receive or transmit operations perform a read or write, respectively, to the PCI. Tables 3-4 and 3-5 provide examples of typical character read and write subroutines.

During normal transmit operation, the PCI generates a Transmit Ready (TxRDY) signal that indicates that the PCI is ready to accept a data character for transmission. TxRDY is automatically reset when the CPU loads a character into the PCI.

Similarly, during normal receive operation, the PCI generates a Receive Ready (RxRDY) signal that indicates that a character has been received and is ready for input to the CPU. RxRDY is automatically reset when a character is read by the CPU.

The TxRDY and RxRDY outputs of the PCI can be used to interrupt the CPU. Refer to section 2-14 for instructions.

Table 3-4. Typical PCI Data Character Read Subroutine

```
;RX1 READS DATA CHARACTER FROM PCI.
;USES-STAT; DESTROYS-A, FLAGS.

        EXTRN   STAT

RX1:    CALL    STAT            ;GET PCI STATUS
        ANI     2               ;CHECK FOR RXRDY
        JZ      RX1             ;RXRDY MUST BE TRUE
        IN      0ECH            ;READ DATA FROM PCI
        RET

        END
```

Table 3-5. Typical PCI Data Character Write Subroutine

```
;TX1 WRITES DATA CHARACTER FROM REG A TO PCI
;USES-STAT; DESTROYS NOTHING.

        EXTRN   STAT

TX1:    PUSH    PSW             ;SAVE DATA AND CPU STATUS
TX11:   CALL    STAT            ;GET PCI STATUS
        ANI     1               ;CHECK FOR TXRDY
        JZ      TX11            ;TXRDY MUST BE TRUE
        POP     PSW             ;RESTORE DATA AND CPU STATUS
        OUT     0ECH            ;SEND DATA TO PCI
        RET

        END
```

The CPU can determine the status of the serial I/O port by issuing an I/O Read Command to the upper address (ED or EF) of the PCI device. The format of the status word is shown in figure 3-7. A typical status read subroutine is given in table 3-6.

3-13. 8255A PPI PROGRAMMING

The iSBC 80/10B board has a total of 48 parallel I/O lines. Half of these use connector J1 and the other half use connector J2. One 8255A PPI device is used to control the 24 lines (3 ports) on each of the two connectors. Line identification is provided in table 2-6.

Each of the six ports (three on each 8255A device) may be independently programmed for a different I/O configuration. All the possible configurations are summarized in table 3-7.

Notice that port operation is not identified for each PPI device, although all six device ports may be

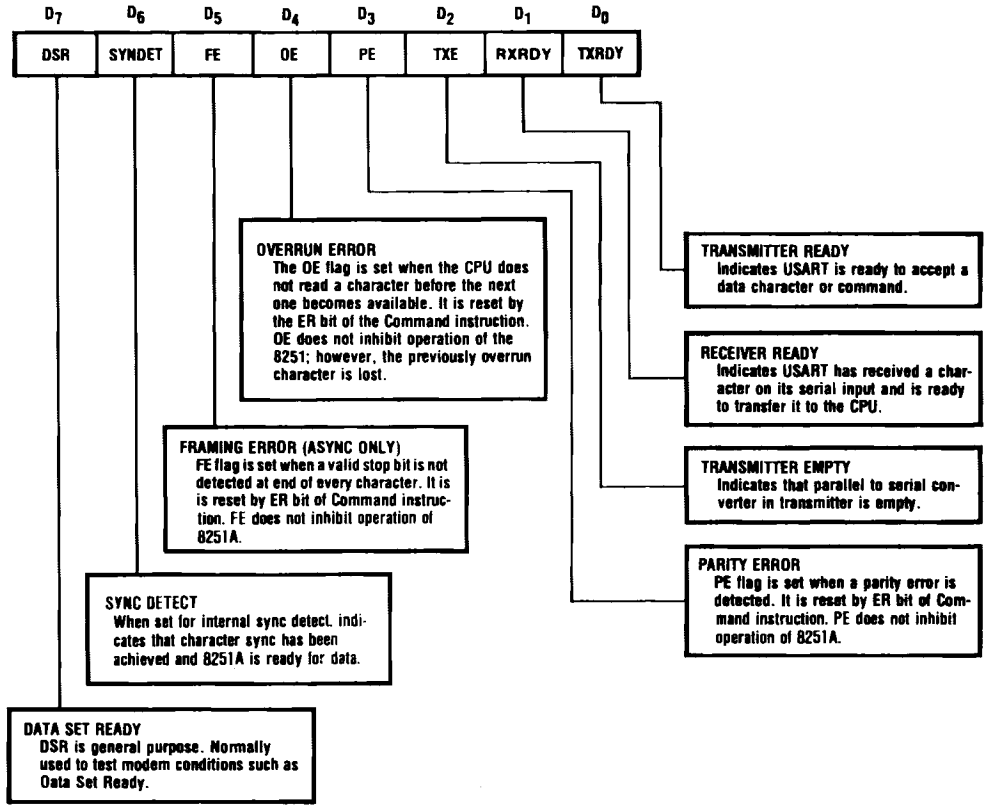


Figure 3-7. PCI Status Read Format

Table 3-6. Typical PCI Status Read Subroutine

```
;STAT READS PCI STATUS
;USES-NOTHING; DESTROYS-A.

;STAT    IN      0EDH            ;GET PCI STATUS
         RET

         END
```

Table 3-7. Parallel I/O Interface Configurations

Port E4 (Group 1 Port A J1)	
Mode 0	Input
Mode 0	Output (Latched)
Mode 1	Input (Strobed)
Mode 1	Output (Latched)
Mode 2	Bidirectional
Port E5 (Group 1 Port B J1)	
Mode 0	Input
Mode 0	Output (Latched)
Mode 1	Input (Strobed)
Mode 1	Output (Latched)
Port E6 (Group 1 Port C J1)	
Mode 0	8 Bit Input
Mode 0	8 Bit Output (Latched)
Note:	Control mode dependent upon Port A and B mode.
Ports E8 and E9 (Group 2 Port A,B J2)	
Mode 0	Input
Mode 0	Output (Latched)
Port EA (Group 2 Port C J2)	
Mode 0	8 Bit Input
Mode 0	8 Bit Output
Mode 0	4 Bit Input/4 Bit Output (Unlatched/Latched)
Mode 0	4 Bit Output/4 Bit Input (Latched/Unlatched)

programmed as either inputs or outputs. Also, the J1 ports can fully utilize the PPI modes and external interrupt capabilities (section 4-30). However, the J2 ports may only operate in mode 0.

A default jumper sets the J1, port E4 bidirectional data buffers to the output mode. An optional jump connection allows the data buffers for this port to be set to the input mode or to allow bit 6 of Port E6 to set the data buffers to either the input or the output mode. Refer to section 2-17 for instructions on this conversion.

A description of 8255A operation is provided in Chapter 4.

3-14. CONTROL WORD FORMAT

The control word format shown in figure 3-8 is used to initialize each PPI port. Note that the three ports on each PPI device are separated into two groups. Group A (control word bits 3 through 6) defines the operating mode for Port A and the upper four bits of Port C. Group B (control word bits 0 through 2) defines the operating mode for Port B and the lower four bits of Port C. Bit 7 of the control word controls the mode set flag. Control words are sent to Port E7 for the J1 PPI device, or the Port EB for the J2 PPI device (table 3-2). There are restrictions associated with the use of certain ports. Refer to table 3-7 and sections 3-17 through 3-22.

3-15. ADDRESSING

The J1 PPI (U16) uses three consecutive data addresses: E4 through E6, plus command port E7. The J2 PPI (U17) also uses three consecutive data addresses: E8 through EA, plus command port EB. Refer to table 3-2.

3-16. INITIALIZATION

To initialize a PPI, write a control word to its control port (E7 or EB). Refer to figure 3-8 and table 3-8 and assume that the control word is 92 (hexadecimal). This initializes the PPI as follows:

a. Mode Set Flag active
b. Port A (E4 or E8) set to Mode 0 Input
c. Port C (EA or E6) upper set to Mode 0 Output
d. Port B (E9 or E5) set to Mode 0 Input
e. Port C (EA or E6) lower set to Mode 0 Output

Table 3-8. Typical PPI Initialization Subroutine

```
;INTPAR INITIALIZES PARALLEL PORTS.
;USES NOTHING; DESTROYS-A.

INTPAR:  MVI    A,92H          ;MODE WORD (PPI PORT A&B IN, C OUT).
         OUT    0EBH           ;SEND MODE WORD TO PPI
         RET

         END
```

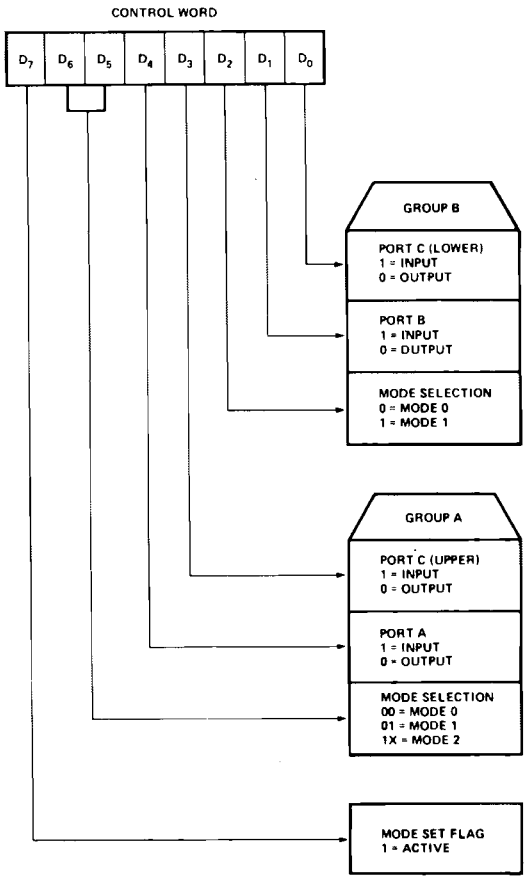


Figure 3-8. PPI Control Word Format

3-17. OPERATION

The primary considerations in determining how to operate each of the six I/O ports are:

a. choice of operating mode (as defined in table 3-7),

b. direction of data flow (input, output or bidirectional),

c. choice of driver/termination networks.

In the following paragraphs, we will define the capabilities of each port and summarize, in tables, information necessary to use the port in each of its potential configurations. Each table will list the port I/O address, the control register address and the format for the control word which is sent to the PPI by the CPU and which specifies the particular configuration to be used. Each table will also summarize the relevant information concerning the choice and use of driver/termination networks, the data polarity, the connecting of jumpers and what they enable, and any restrictions on the use of the other two ports in each group.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single output instruction (see figure 3-9). This feature reduces software requirements in Control-based applications.

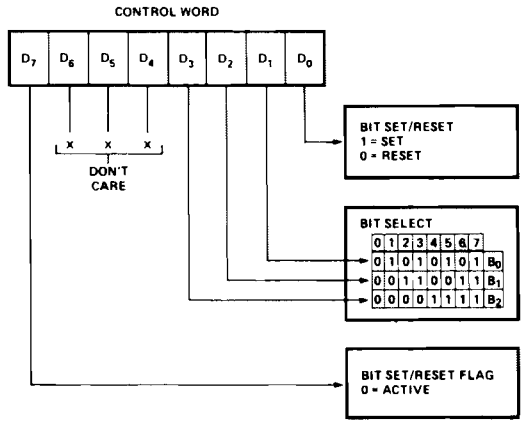


Figure 3-9. PPI Port C Bit Set/RESET Control Word Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the PPI is programmed to operate in Mode 1 or Mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from Port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the Bit Set/Reset function of Port C.

This function allows the Programmer to disallow or allow specific I/O devices to interrupt the CPU without effecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET - Interrupt enable
(BIT-RESET) — INTE is RESET - Interrupt disable

All Mask flip-flops are automatically reset during mode selection and device Reset.

3-18. PORT E4. Port E4 is the only port that already includes a socketed bidirectional driver/

termination network (8287 Octal Bus Transciever). Port E4 is also the only port which can be programmed to function in any one of the three PPI operating modes. Before Port E4 is programmed for input or output in any one of the three modes, certain jumper connections must be made to allow the port to function properly in the chosen mode. Other jumper connections must be made to enable interrupts when Port E4 is in mode 1 or mode 2. In all, there are five potential configurations for Port E4. All of the necessary information for implementing each configuration has been summarized in the following tables:

PORT E4 CONFIGURATIONS		Table
Mode	Direction	
1. Mode 0	Input	Table 3-9
2. Mode 0	Output (Latched)	Table 3-10
3. Mode 1	Input (Strobed)	Table 3-11
4. Mode 1	Output (Latched)	Table 3-12
5. Mode 2	Bidirectional	Table 3-13

3-19. PORT E5. Port E5 can be programmed for input or output in either mode 0 or mode 1. If Port E5 is to be used for input, in either mode, terminator networks must be installed in sockets U4 and U5. If Port E5 is to be used for output, in either mode, driver networks must be installed in sockets U4 and U5. When Port E5 is programmed for mode 1, interrupts can be enabled by connecting jumper pair 9-10. The four potential configurations for Port E5 are summarized in the following tables:

Port E5 Configurations		Table
Mode	Direction	
1. Mode 0	Input	Table 3-14
2. Mode 0	Output (Latched)	Table 3-15
3. Mode 1	Input (Strobed)	Table 3-16
4. Mode 1	Output (Latched)	Table 3-17

3-20. PORT E6. The use of Port E6 is dependent on the modes programmed for Ports E4 and E5 (refer to tables 3-9 to 3-17). While certain Port E6 bits are available if Port E4 is in mode 1 or if Port E5 is in mode 0, the use of Port E6 as an 8-bit data path is restricted to those configurations that have both Port E4 and Port E5 programmed for mode 0. In this case, all eight bits of Port E6 can be programmed for mode 0 input (see table 3-18) or output (see table 3-19). A 4-bit input/4-bit output configuration is never possible for Port E6.

NOTE

If Ports E4 and E5 are not both in mode 0, then a driver network must be installed in the sockets in U2 and a termination network must be installed in U3 so that the Port E6 control lines can function properly.

3-21. PORTS E8 AND E9. Ports E8 and E9 can be programmed for input or output but only in mode 0. The two potential configurations for each port are summarized in the following tables:

Configurations			Table
Port	Mode	Direction	
1. Port E8	Mode 0	Input	Table 3-20
2. Port E8	Mode 0	Output (Latched)	Table 3-21
1. Port E9	Mode 0	Input	Table 3-22
2. Port E9	Mode 0	Output (Latched)	Table 3-23

3-22. PORT EA. All eight bits of Port EA can be programmed for mode 0 input or output, or four bits can be programmed for mode 0 input while the other four bits are programmed for mode 0 output. The four potential configurations for Port EA are summarized in the following tables:

PORT EA CONFIGURATIONS		
Mode	Direction	Table
1. Mode 0	8-bit Input	Table 3-24
2. Mode 0	8-bit Output (Latched)	Table 3-25
3. Mode 0	Upper 4-bit Input/ Lower 4-bit Output	Table 3-26
4. Mode 0	Upper 4-bit Output/ Lower 4-bit Input	Table 3-27

After the PPI has been initialized, operation is accomplished by performing a read or write operation to the appropriate port. A typical read subroutine is shown in table 3-28. A typical write subroutine is shown in table 3-29.

Table 3-9. Port E4, Mode 0 Input Configuration

Port Address: E4, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	1	X	X	X	X

Driver/Termination Networks: 8287 Transceiver installed in socket U1.

Data Polarity: Negative-true.

Jumper Connections: Remove 59-60.

Port E5 Restrictions: None; Port E5 can be programmed for mode 0 or mode 1, input or output (see Section 3-19).

Port E6 Restrictions: None; Port E6 can be programmed for mode 0, 8-bit input or output, unless Port E5 is in mode 1 (see Section 3-20).

Table 3-10. Port E4, Mode 0 Latched Output Configuration

Port Address: E4, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	0	X	X	X	X

Driver/Termination Networks: 8287 Transceiver installed in socket U1.

Data Polarity: Negative-true.

Jumper Connections: Factory Default: 59-60 installed.

Port E5 Restrictions: None; Port E5 can be programmed for mode 0 or mode 1, input or output (see Section 3-19).

Port E6 Restrictions: None; Port E6 can be programmed for mode 0, input or output, unless Port E5 is in mode 1 (see Section 3-20).

Table 3-11. Port E4, Mode 1 Strobed Input Configuration

Port Address: E4, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	1	1	0/1	X	X	X

Driver/Termination Networks: 8287 Transceiver permanently installed in U1. A driver network must be installed in U2 and a termination network must be installed at U3.

Data Polarity: Negative-true. The polarity of Port E6 control outputs is dependent on the type of driver installed in U2.

Jumper Connections: 60-61 to enable input 8287, connect 19-20 to enable interrupt request via INT55/. Remove 59-60; 20-25; 19-24.

Port E5 Restrictions: None; Port E5 can be programmed for mode 0 or mode 1, input or output (see Section 3-19).

Port E6 Restrictions: Port E6 bits perform the following dedicated functions:

**Bits 0, 1 and 2* — dedicated to control of Port E5 if it is in mode 1 (see tables 4-9 to 4-12).

**Bit 3* — INTR (interrupt request) output for Port E4.

**Bit 4* — STB/ (strobe) input for Port E4.

**Bit 5* — IBF (input buffer full) output for port E4.

**Bit 6 and 7* — Only one bit can be used. If input, use bit 6; do not use bit 7. Bit 3 of Control Word=1. If output, use bit 7, do not use bit 6. Bit 3 of Control Word=0.

Table 3-12. Port E4, Mode 1 Latched Output Configuration

Port Address: E4, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	1	0	X	X	X	X

Driver/Termination Networks: 8287 Transceiver permanently installed in U1. A driver network must be installed in U2 and a termination network must be installed at U3.

Data Polarity: Negative-true. The polarity of Port C control outputs is dependent on the type of driver installed in U3.

Jumper Connections: 59-60 (default) to enable output at 8287, connect 19-20 to enable interrupt request via INT55/. Remove 20-25; 19-24.

Port E5 Restrictions: None; Port E5 can be programmed for mode 0 or mode 1, input or output (see Section 3-19).

Port E6 Restrictions: Port E6 bits perform the following dedicated functions:

**Bits 0, 1 and 2* — dedicated to the control of Port E5 if it is in mode 1 (see tables 3-16 and 3-17).
**Bit 3* — INTR (interrupt request) output for Port E4.
**Bit 4* — can be used for input if bit 3 of control word=1.
Bit 5 — cannot be used if PC4 is used; can be used for output if control word bit 3=0 (PC4 cannot be used then).
**Bit 6* — ACK/ (acknowledge) input for Port E4.
**Bit 7* — OBF/ (output buffer full) output for Port E4.

Table 3-13. Port E4, Mode 2 Bidirectional Configuration

Port Address: E4, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	1	X	X	X	X	X	X

Driver/Termination Networks: 8287 Transceiver permanently installed in U1. A driver network must be installed in U2 and a termination network must be installed in U3.

Data Polarity: Negative-true. The polarity of Port C control outputs is dependent on the type of driver installed in U3.

Jumper Connections: 60-61 to allow ACK/ input on PC6 to dynamically change data direction at 8287 (input when ACK/=1 and output when ACK/=0); connect 19-20 to enable interrupt request via INT55/. Remove 59-60; 20-25; 19-24.

Port E5 Restrictions: None.

Port E6 Restrictions: Port E6 bits perform the following dedicated functions:

**Bits 0 and 1* — can be used for output if bit 3 of control word=0.
**Bit 2* — cannot be used if PC0 and PC1 are used; can be used for input if control word bit 3=1 (PC0 and PC1 cannot be used then).
**Bit 3* — INTR (interrupt request) output for Port E4.
**Bit 4* — STB/ (strobe) input for Port E4.
**Bit 6* — ACK/ (acknowledge) input for Port E4.
**Bit 7* — OBF/ (output buffer full) output for Port E4.

Table 3-14. Port E5, Mode 0 Input Configuration

Port Address: E5, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	X	X	X	X	0	1	X

Driver/Termination Networks: Termination networks must be installed in U4 and U5.

Data Polarity: Positive-true.

Jumper Connections: None.

Port E4 Restrictions: None (see Section 3-18).

Port E6 Restrictions: None; Port E6 can be programmed for mode 0, input or output, unless Port E4 is in mode 1 or mode 2 (see Section 3-20).

Table 3-15. Port E5, Mode 0 Latched Output Configuration

Port Address: E5, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	X	X	X	X	0	0	X

Driver/Termination Networks: Driver networks must be installed in U4 and U5.

Data Polarity: Negative-true, assuming that inverting drivers are installed.

Jumper Connections: None.

Port E4 Restrictions: None (see Section 3-18)

Port E6 Restrictions: None; Port E6 can be programmed for mode 0, 8-bit input or output, unless Port E4 is in mode 1 or mode 2 (see Section 3-20).

Table 3-16. Port E5, Mode 1 Strobed Input Configuration

Port Address: E5, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	X	X	X	1	1	X

Driver/Termination Networks: Termination networks must be installed in U4 and U5. A driver network must be installed in U2 and a termination network must be installed in U3.

Data Polarity: Positive-true. The polarity of Port C control outputs is dependent on the type of driver installed in U2.

Jumper Connections: 9-10 to enable interrupt request via INT55/. Remove 9-4; 10-5.

Port E4 Restrictions: None.

Port E6 Restrictions: Port E6 bits perform the following dedicated functions:

**Bit 0* — INTR (interrupt request) output for Port E5.
**Bit 1* — IBF (input buffer full) output for Port E5.
**Bit 2* — STB/ (strobe) input for Port E5.
**Bit 3 to 7* — dedicated to control of Port E4 if it is in mode 1 (see tables 3-9 to 3-12).

Table 3-17. Port E5, Mode 1 Latched Output Configuration

Port Address: E5, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	X	X	X	1	0	X

Driver/Termination Networks: Driver networks must be installed in U4 and U5. A driver network must be installed in U2 and a termination network must be installed in U3.

Data Polarity: Negative-true, assuming that inverting drivers are in U4 and U5. The polarity of Port C control outputs is dependent on the type of driver installed in U2.

Jumper Connections: 9-10 to enable interrupt request via INT55/. Remove 9-4; 5-10.

Port E4 Restrictions: None.

Port E6 Restrictions: Port E6 bits perform the following dedicated functions:

**Bit 0* — INTR (interrupt request) output for Port E5.
**Bit 1* — OBF/ (output buffer full) output for Port E5.
**Bit 2* — ACK/ (acknowledge) input for Port E5.
**Bit 3* — PC7 - dedicated to control of Port E4 if it is in mode 1 (see tables 3-9 to 3-12).

Table 3-18. Port E6, Mode 0, 8-bit Input Configuration

Port Address: E6, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	X	1	0	X	1

Driver/Termination Networks: Termination networks must be installed in U2 and U3.

Data Polarity: Positive-true.

Jumper Connections: 5-10 and 20-25 to disable interrupts and enable Port E6, bits 0 and 3.

Port E4 and E5 Restrictions: Both Ports E4 and E5 must be in mode 0.

Table 3-19. Port E6, Mode 0, 8-bit Latched Output Configuration

Port Address: E6, **Control Register Address:** E7

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	X	0	0	X	0

Driver/Termination Networks: Driver networks must be installed in U2 and U3.

Data Polarity: Negative-true, assuming that inverting drivers are installed in U2 and U3.

Jumper Connections: 5-10, and 20-25 to disable interrupts and enable Port E6, bits 0 and 3.

Port E4 and E5 Restrictions: Both ports must be in mode 0.

Table 3-20. Port E8, Mode 0, Input Configuration

Port Address: E8, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	1	X	X	X	X

Driver/Termination Networks: Termination networks must be installed in U6 and U7.

Data Polarity: Positive-true.

Jumper Connections: None.

Port E9 and EA Restrictions: None; both ports can be programmed for mode 0, input or output (also see Section 3-22).

Table 3-21. Port E8, Mode 0 Latched Output Configuration

Port Address: E8, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	0	X	X	X	X

Driver/Termination Networks: Driver networks must be installed in U6 and U7.

Data Polarity: Negative-true, assuming that inverting drivers are installed in U6 and U7.

Jumper Connections: None.

Port E9 and EA Restrictions: None; both ports can be programmed for mode 0, input or output (also see Section 3-22).

Table 3-22. Port E9, Mode 0 Input Configuration

Port Address: E9, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	X	X	X	X	0	1	X

Driver/Termination Networks: Termination networks must be installed in U10 and U11.

Data Polarity: Positive-true.

Jumper Connections: None.

Port E8 and EA Restrictions: None; both ports can be programmed for mode 0, input or output (see Section 3-22).

Table 3-23. Port E9, Mode 0 Latched Output Configuration

Port Address: E9, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	X	X	X	X	0	0	X

Driver/Termination Networks: Driver networks must be installed in U10 and U11.

Data Polarity: Negative-true, assuming that inverting drivers are installed in U10 and U11.

Jumper Connections: None.

Port E8 and EA Restrictions: None; both ports can be programmed for mode 0, input or output (also see Section 3-22).

Table 3-24. Port EA, Mode 0, 8-bit Input Configuration

Port Address: EA, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	X	1	0	X	1

Driver/Termination Networks: Termination networks must be installed in U8 and U9.

Data Polarity: Positive-true.

Jumper Connections: None.

Port E8 and E9 Restrictions: None (see Section 3-21).

Table 3-25. Port EA, Mode 0, 8-bit Latched Output Configuration

Port Address: EA, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	X	0	0	X	0

Driver/Termination Networks: Driver networks must be installed in U8 and U9.

Data Polarity: Negative-true, assuming that inverting drivers are installed in U8 and U9.

Jumper Connections: None.

Port E8 and E9 Restrictions: None (see Section 3-21).

Table 3-26. Port EA, Mode 0 Upper 4-bit Input/Lower 4-bit Latched Output Configuration

Port Address: EA, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	X	1	0	X	0

Driver/Termination Networks: A termination network must be installed In U8 and a driver network must be installed in U9.

Data Polarity: The upper 4-bits will be in positive-true form; however, the lower four bits will be in negative-true form if an inverting driver is installed in U9.

Jumper Connections: None.

Port E8 and E9 Restrictions: None (see Section 3-21).

Table 3-27. Port EA, Mode 0 Upper 4-bit Latched Output/Lower 4-bit Input Configuration

Port Address: EA, **Control Register Address:** EB

Control Word Format:

7	6	5	4	3	2	1	0
1	0	0	X	0	0	X	1

Driver/Termination Networks: A driver network must be installed and a termination network must be installed in U9.

Data Polarity: The lower 4-bits will be in positive-true form; however, the upper 4-bits will be in negative-true form if an inverting driver is installed in U8.

Jumper Connections: None.

Port E8 and E9 Restrictions: None (see Section 3-21).

Table 3-28. Typical PPI Port Read Subroutine

```
;AREAD READS A BYTE FROM PORT A INTO REG A.
;USES NOTHING, DESTROYS-A.

AREAD:  IN      0E8H            ;READ PORT EBH
        RET

        END
```

Table 3-29. Typical PPI Port Write Subroutine

```
;COUT OUTPUTS A BYTE FROM REG A TO PORT C.
;USES NOTHING, DESTROYS NOTHING.

COUT:   OUT     0EAH            ;OUTPUT TO PORT EAH
        RET

        END
```

CHAPTER 4
PRINCIPLES OF OPERATION

4-1. INTRODUCTION

This chapter provides a summarized description of each functional block of the iSBC 80/10B board in sections 4-2 through 4-10. Detailed descriptions of board operation and circuit analysis are provided in sections 4-11 through 4-26. Circuit diagrams (figure 5-3) used for reference purposes are located in Chapter 5.

4-2. FUNCTIONAL DESCRIPTION

The following sections provide a brief description of each functional block of the iSBC 80/10B board. All circuit locations refer to figures 5-1 and 5-3. A functional block diagram of the board is provided in figure 4-1.

4-3. CLOCK CIRCUITS

All timing originates from the 8224 (U29) device. Its 18.432 MHz crystal is used as a reference to derive the two CPU timing waveforms (ϕ1, ϕ2), the ϕ2TTL waveform, and the OSC waveform.

The OSC signal is used to drive the baud rate timer network (figure 5-3, sheet 6, zone D5). Additional timing pulses are derived from this network, including the 1 millisecond timer (MST) and the master clock (MCLK). The MCLK signal is subsequently split into BCLK/ and CCLK/ for Multibus use (figure 5-3, sheet 3, zone A4).

4-4. CENTRAL PROCESSING UNIT (CPU) GROUP

The CPU group consists of the 8080A microprocessor (U33), the 8224 clock generator (U29) and the 8238 system controller (U49).

The CPU group is the heart of the iSBC 80/10B board. It performs all system processing functions and provides a stable timing reference for all other

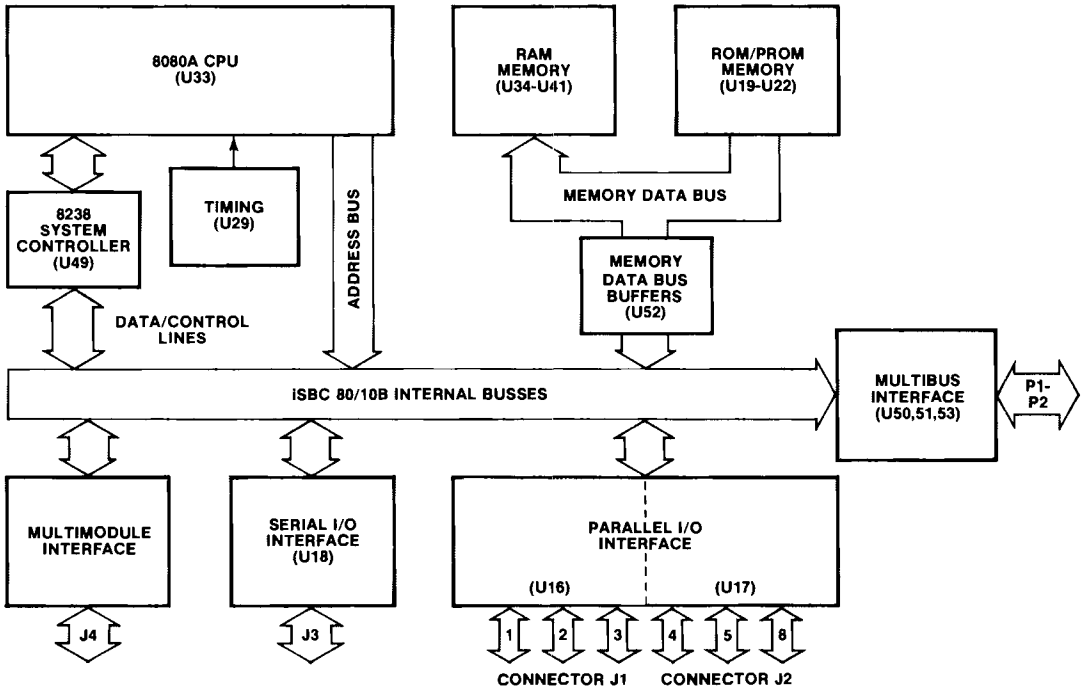


Figure 4-1. iSBC 80/10B™ Block Diagram

circuitry in the system. The CPU group generates all of the address and control signals necessary to access memory and I/O ports both on the iSBC 80/10B and external to the board. The CPU group is capable of fetching and executing any of the 8080A instructions. The CPU group responds to interrupt requests originating both on and off the iSBC 80/10B board, HOLD requests from modules wishing to acquire control of the Multibus lines, and to WAIT requests from memory or I/O devices having an access time which is slower than the 8080A cycle time.

4-5. SERIAL I/O INTERFACE

The serial I/O interface, using Intel's 8251A PCI (U18) device, provides a bi-directional serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols. Synchronous or asynchronous mode, baud rate, character length, number of stop bits and the choice of even, odd or no parity are all program selectable. You also have the option of configuring the serial I/O interface as an EIA RS232C interface or as a teletypewriter-compatible current loop interface.

4-6. PARALLEL I/O INTERFACE

The parallel I/O interface, using two Intel 8255A Programmable Peripheral Interface (PPI) devices (U16 and U17), provides 48 signal lines for the transfer and control of data to or from peripheral devices. Port E4 already has a bidirectional driver and termination network installed. This bidirectional network allows these eight lines to be inputs, outputs, or bidirectional (selected via jumpers). The remaining 40 lines (five ports) are uncommitted. Sockets are provided for the installation of drivers or termination networks as required to meet the specific needs of your system.

4-7. ROM/PROM MEMORY

The memory area occupies the space from 0000 to 3FFF (hexadecimal). Four sockets are provided for ROM/PROM devices, accommodating up to 16K bytes (refer to section 2-8). ROM/PROM type is indicated by header plug placement, which performs the necessary addressing and voltage connections for proper operation. Data is read from the ROM/PROM devices in complete bytes and is placed onto the internal data bus for subsequent use.

4-8. RAM MEMORY

The iSBC 80/10B board is shipped with 1K bytes of RAM on-board, in sockets U37 and U41. Each device is a 2114 type with 1K x 4 bits. Refer to section 2-9 for RAM installation and addressing information. Six additional sockets are provided for optional on-board RAM, providing a maximum of 4K bytes. Bits D0 through D3 are read/written by the device in U37, while bits D4 through D7 are handled by the device in U41. Data is placed on or written from the internal data bus drivers (U52); which is also used by the ROM/PROM array.

4-9. MULTIMODULE INTERFACE

Multimodule boards are special purpose, add-on circuit boards which reside directly on the component side of the iSBC 80/10B board, and are interfaced through connector J4.

The Multimodule connector is shown schematically in figure 5-3, sheet 8.

4-10. MULTIBUS INTERFACE

Multibus signals include all off-board signals which are handled by connectors P1 and P2. These include address, data, and control lines. The Multibus signals used by the iSBC 80/10B board are described in table 2-11. For additional Multibus information refer to the *Intel Multibus Specifications*, order number 9800683.

4-11. CIRCUIT ANALYSIS

The iSBC 80/10B board schematic diagram is provided in figure 5-3, sheets 1 through 9. Many signals traverse from one sheet to another. These are identified by a single or double alpha character within a box (e.g., AF) along with the signal mnemonic. The source sheet number is generally on the left side of the drawing, while the destination sheet is on the right side. *Conversely, signals which enter or exit the board are not boxed,* and carry the connector and pin designation along with the signal mnemonic (e.g., P1-19 MRDC/).

Both active-high and active-low signals are used. A signal mnemonic which ends with a slash (e.g, WAIT/) denotes that the signal is active-low ($\leq$0.4V).

Conversely, a signal mnemonic without the slash (e.g., OSC) indicates that the signal is active-high (≥2.0V).

4-12. CPU GROUP

The Central Processing Unit (CPU) group consists of three Intel integrated circuit devices:

- 8080A Microprocessor
- 8224 Clock Generator
- 8238 System Controller

An 18.432 MHz crystal establishes the frequency of oscillation for the 8224 device via a 10pF capacitor. Together, the elements in the CPU group perform all central processing functions. The following paragraphs describe how the elements within the CPU group interact with all other logic on the iSBC 80/10B board. The interaction between the devices within the CPU group, however, is not described. For additional information on these devices, refer to the *Intel MCS-80 User's Manual,* Order Number 9800153.

4-13. INSTRUCTION TIMING. The activities of the CPU group are cyclical. The CPU fetches an instruction, performs the operations required, fetches the next instruction, and so on. This orderly sequence of events requires precise timing. The 8224 Clock Generator, provides the primary timing reference for the CPU group. The crystal in conjunction with a 10 pF capacitor tunes an oscillator within the 8224 to precisely 18.432 MHz. The 8224 "divides" the oscillations by nine to produce two-phase timing inputs (ϕ1 and ϕ2) for the 8080A. The ϕ1 and ϕ2 signals define a cycle of approximately 488 ns duration. A TTL level phase 2 (ϕTTL) signal is also derived and made available to external logic. In addition, the output of the oscillator is buffered and brought out on OSC so that other system timing signals can be derived from this stable, crystal controlled source (e.g., the serial I/O baud rate is derived from OSC). All processing activities of the CPU group are referred to the period of the ϕ1 and ϕ2 clock signals.

Within the 8080A CPU group, an *instruction cycle* is defined as the time required to fetch and execute an instruction. During the fetch, a selected instruction (one, two or three bytes) is extracted from memory and deposited in the CPU operating registers. During the execution part, the instruction is decoded and translated into specific processing activities.

Every instruction cycle has at least one reference to memory, during which the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of instruction requires no further references to memory. The first machine cycle in every instruction cycle is therefore a FETCH. Beyond that, there are no fast rules. The input (IN) and output (OUT), instructions each require three machine cycles: a FETCH, to obtain the instruction; a MEMORY READ, to obtain the address of the object peripheral; and an INPUT or an OUTPUT machine cycle, to complete the transfer.

Each machine cycle consists of three, four or five states. A *state* is the smallest unit or processing activity and is defined as the interval between two successive positive-going transitions of the ϕ1 clock pulse.

There are three exceptions to the defined duration of a state. They are the WAIT state, the hold (HLDA) state and the halt (HLTA) state. Because the WAIT, the HLDA, and the HLTA states depend upon external events, they are by their nature of indeterminate length. Even these exceptional states, however, must be synchronized with the pulses of the driving clock. Thus the duration of all states, including these, are integral multiples of the clock pulse.

To summarize, each *clock period* marks a *state*; three to five *states* comprise a *machine cycle*; and one to five *machine cycles* comprise an *instruction cycle.* A full instruction cycle requires anywhere from four to seventeen states for its completion, depending on the kind of instruction involved (refer to Appendix A).

There is just one consideration that determines how many machine cycles are required in any given instruction cycle: the number of times that the processor must reference a memory address or an I/O address, in order to fetch and execute the instruction. Like many processors, the 8080A is designed so that it transmits one address per machine cycle. Thus, if the fetching and execution of an instruction requires two memory references, then the instruction cycle associated with that instruction consists of two machine cycles. If five such references are called for, then the instruction cycle contains five machine cycles.

Every instruction cycle has at least one reference to memory, during which the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of instruction requires no further references to memory. The first machine cycle in every instruction cycle is therefore a FETCH. Beyond that, there are no fast rules. The input (IN) and output (OUT), instructions each require three machine cycles: a FETCH, to obtain the instruction; a MEMORY READ, to obtain the address of the object peripheral; and an INPUT or an OUTPUT machine cycle, to complete the transfer.

Every machine cycle within an instruction cycle consists of three to five active states (referred to a T1, T2, T3, T4, and T5). The actual number of states depends upon the instruction being executed, and on the particular machine cycle within the greater instruction cycle. Figure 4-2 shows the timing relationships in a typical FETCH machine cycle. Events that occur in each state are referred to transitions of the $\phi 1$ and $\phi 2$ clock pulses.

At the beginning of each machine cycle (in state T1), the 8080A activates its SYNC output and issues status information on its data bus. The 8224 accepts SYNC and generates an active-low status strobe (STSTB/) as soon as the status data is stable on the data bus. The status information indicates the type of machine cycle in progress. The 8238 system controller accepts the status bits from the 8080A and STSTB/ from the 8224, and uses them to generate the appropriate control signals (MEMR/, MEMW/, IOR/, and IOWR/) for the current machine cycle.

The rising edge of $\phi 2$ during T1 loads the processor's address lines (A0-A15). These lines become stable within a brief delay of the $\phi 2$ clocking pulse, and they remain stable until the first $\phi 2$ pulse after state T3. This gives the processor ample time to read the data returned from memory.

Once the processor has sent an address to memory, there is an opportunity for the memory to request a WAIT. This is done by pulling the 8224 RDYIN line low. As long as the RDYIN line remains low, the CPU group will idle, giving the memory time to respond to the addressed data request. The 8224 synchronizes RDYIN with internal processor timing and applies the result to the 8080A READY input. The processor responds to a wait request by entering an alternative state (TW) at the end of T2, rather than proceeding directly to the T3 state. A wait period may be of indefinite duration. The 8080A remains in the waiting condition until its READY line again goes high. The cycle may then proceed, beginning

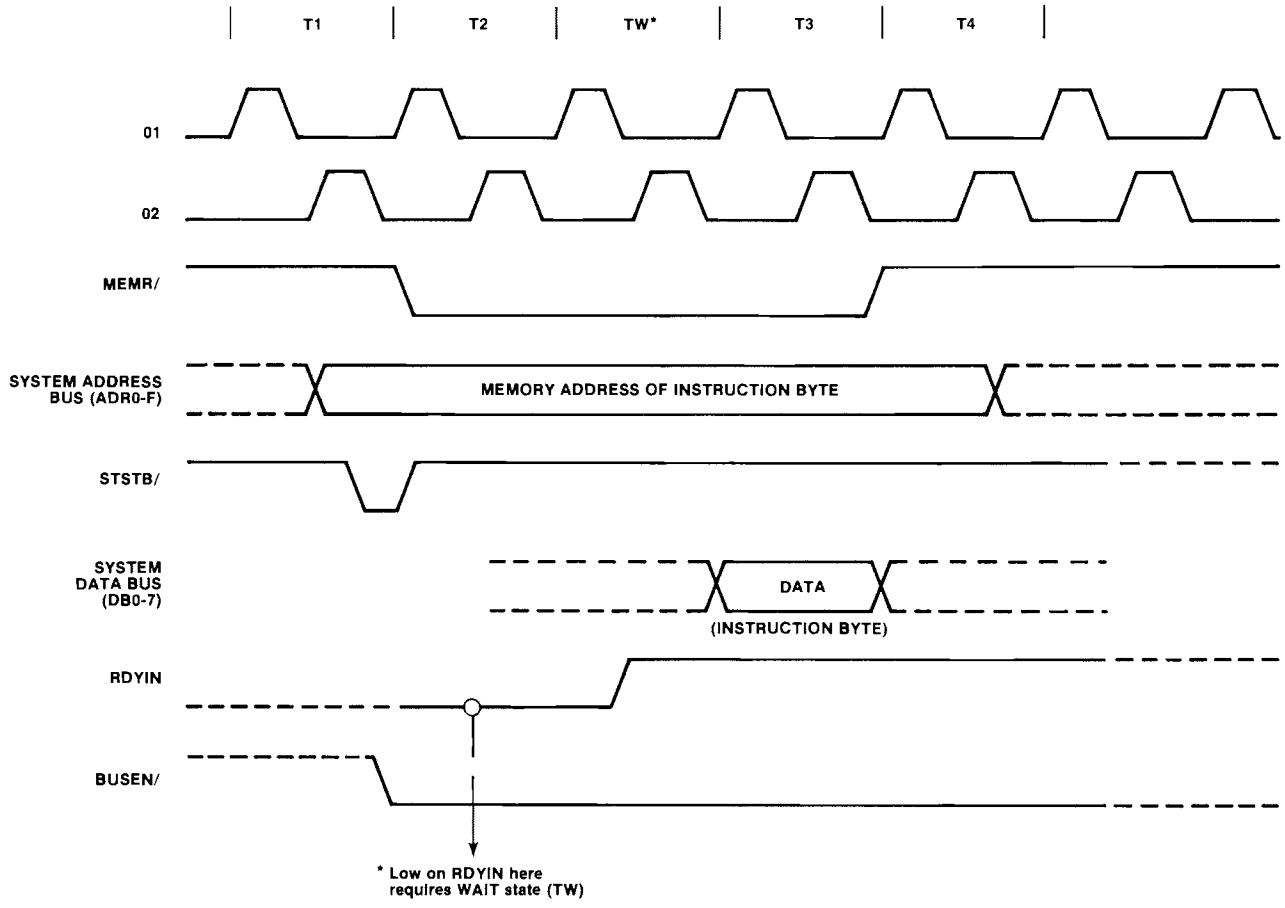


Figure 4-2. Typical FETCH Machine Cycle

with the rising edge of the next $\phi 1$ clock. A WAIT interval will therefore consist of an integral number of TW states and will always be a multiple of the clock period.

The events that take place during the T3 state are determined by the kind of machine cycle in progress. In a FETCH machine cycle, the CPU interprets the data on its data bus as an instruction. During a MEMORY READ, signals on the same bus are interpreted as a data word. The CPU group itself writes out data on this bus during a MEMORY WRITE machine cycle. And during I/O operations, the CPU group may either transmit or receive data, depending on whether an INPUT or an OUTPUT operation is involved. Consider the following two examples.

Figure 4-3 illustrates the timing that is characteristic of an input instruction cycle. During the first machine cycle (M1), the first byte of the two-byte IN instruction is fetched from memory. The 8080A places the 16-bit memory address on the system bus near the end of state T1. The 8238 activates the memory read control signal (MEMR/) during states T2 and T3 (and any intervening wait states, if required). During the next machine cycle (M2), the second byte of the instruction is fetched. During the next machine cycle (M2), the second byte of the instruction is fetched. During the third machine cycle (M3), the IN instruction is executed. The 8080A duplicates the 8-bit I/O address on address lines ADR0-7 and ADR8-F. The 8238 activates the I/O read control signal (IOR/) during states T2 and T3 of this cycle. In all cases the system bus enable input (BUSEN/) to the 8238 alows for normal operation of the data bus buffers and the read/write control signals. If BUSEN/ goes high the data bus output buffers and control signal buffers are forced into a high-impedance state.

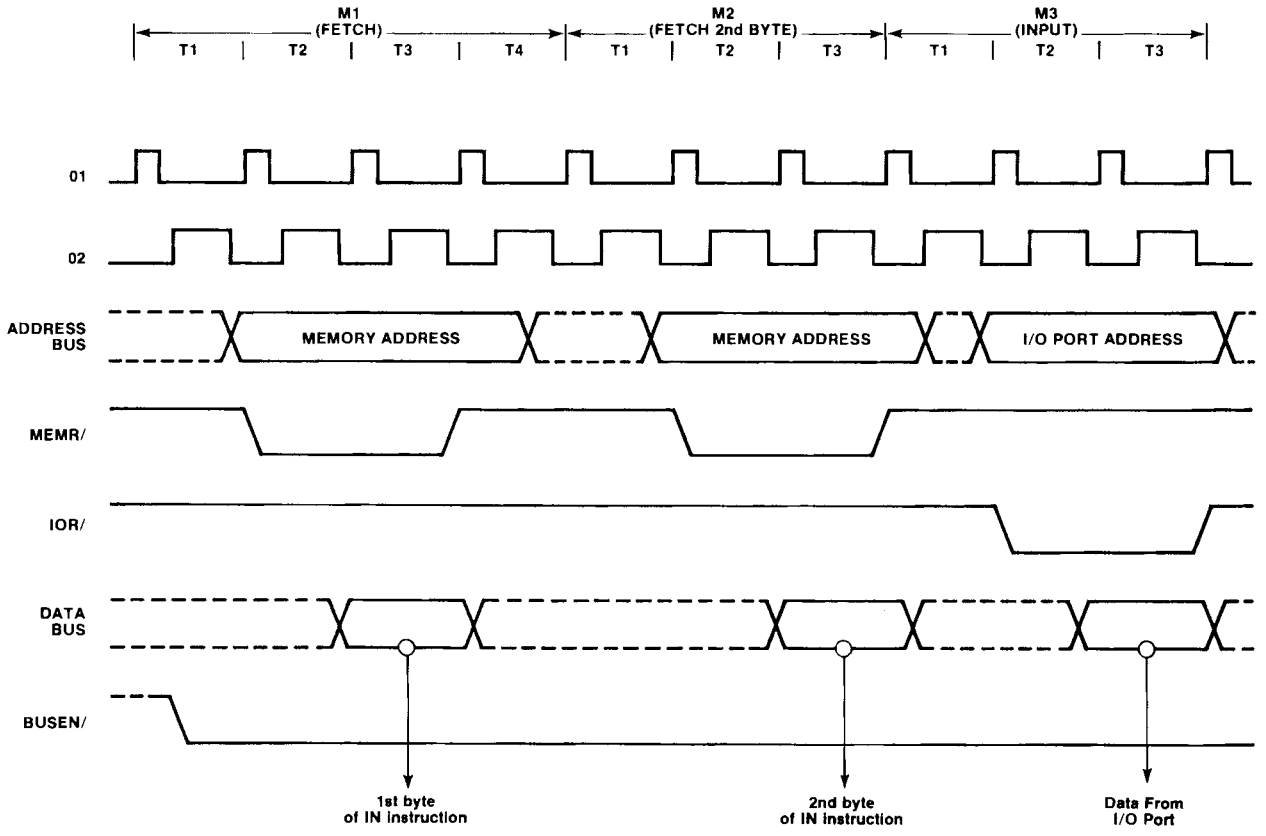


Figure 4-3. Input Instruction Cycle

Figure 4-4 illustrates an instruction cycle during which the CPU group writes data. During the first two machine cycles (M1 and M2), the CPU fetches the two-byte OUT instruction. During the third machine cycle (M3), the OUT instruction is executed. The 8080A duplicates the 8-bit I/O address on lines ADR0-7 and ADR8-F. The 8238 activates *an advanced* I/O write control signal (IOWR/) at the beginning of state T2 of this cycle. The 8238 outputs the data onto the system bus at the end of state T2. Data on the bus remains stable throughout the remainder of the machine cycle. BUSEN/ must be low to prevent the output and control buffers from being forced into the high-impedance state.

Observe that a RDYIN signal is necessary for completion of an output machine cycle. Unless such an indication is present, the processor enters the TW state, following the T2 state. Data on the output lines remains stable in the interim, and the processing cycle will not proceed until the RDYIN line again goes high.

The 8080A generates a WR/ output for qualification of the advanced I/O write (IOWR/) and memory write (MEMW/) control signals from the 8238, during those machine cycles in which the CPU group outputs data. The negative-going leading edge of WR/ is referred to the rising edge of the first $\phi 1$ clock pulse following T2. WR/ remains low until re-triggered by the leading edge of $\phi 2$, during the state following T3. Note that any TW states intervening between T2 and T3 of the output machine cycle will necessarily extend WR/.

All processor machine cycles consist of at least three states: T1, T2, and T3 as just described. If the CPU group has to wait for a RDYIN response, then the machine cycle may also contain one or more TW states. During the three basic states, data is transferred to or from the CPU group.

After the T3 state, however, it becomes difficult to generalize. T4 and T5 states are available, if the execution of a particular instruction requires them. But not all machine cycles make use of these states. It depends upon the kind of instruction being executed, and on the particular machine cycle within the instruction cycle. The processor will terminate any machine cycle as soon as its processing activities are completed, rather than proceeding through the T4 and T5 states every time. Thus the 8080A may

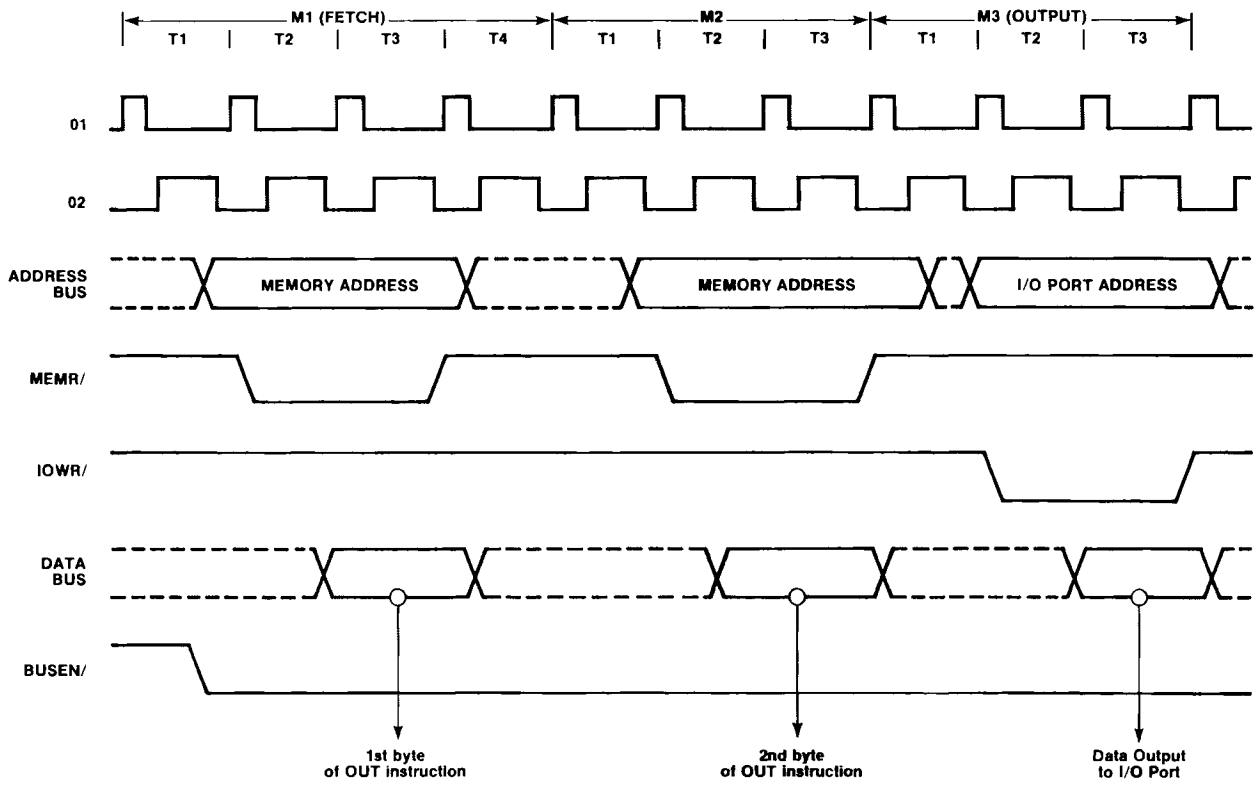


Figure 4-4. Output Instruction Cycle

exit a machine cycle following the T3, the T4, or the T5 state and proceed directly to the T1 state of the next machine cycle.

4-14. INTERRUPT SEQUENCES. The 8080A has a built-in capacity to handle external interrupt requests. Peripheral logic can initiate an interrupt simply by driving the processor's interrupt (INT) line high. The interrupt (INT) input is asynchronous, and a request may therefore originate at any time during any instruction cycle. Internal logic re-clocks the external request, so that a proper correspondence with the driving clock is established. An interrupt request (INT) arriving during the time that the interrupt enable line (INTE) is high, acts in coincidence with the $\phi 2$ clock to set the internal interrupt latch. This event takes place during the last state of the instruction cycle in which the request occurs, thus ensuring that any instruction in progress is completed before the interrupt can be processed.

The INTERRUPT machine cycle which follows the arrival of an enabled interrupt request resembles an ordinary FETCH machine cycle in most respects. The contents of the program counter are latched onto the address lines during T1, but the counter itself is not incremented during the INTERRUPT machine cycle, as it otherwise would be. In this way, the pre-interrupt status of the program counter is preserved, so that data in the counter may be saved in the stack. This in turn permits an orderly return to the interrupted program after the interrupt request has been processed.

Because the 8238 INTA/ output (pin 23) is tied to +12 volts, the 8238 blocks incoming data and automatically inserts a Restart (RST 7) instruction onto the 8080A data bus during state T3, when the interrupt is acknowledged by the 8080A. RST is a special one-byte call instruction that facilitates the processing of interrupts (the ordinary program call instruction is three bytes long). The RST 7 instruction causes the 8080A to branch program control to the instruction being stored in memory location 38 (hex).

4-15. HOLD SEQUENCES. By activating the 8080A HOLD input, an external device can cause the CPU group to suspend its normal operations and relinquish control of the address and data busses. The CPU group responds to a request of this kind by floating its address and data outputs, so that these exhibit a high impedance to other devices sharing the busses. At the same time, the processor acknowledges the HOLD by placing a high on its HLDA output pin. During an acknowledged HOLD, the address and data busses are under control of the peripheral which originated the request, enabling it to conduct off-board memory transfers without processor intervention.

4-16. HALT SEQUENCE. When a halt instruction (HLT) is executed, the 8080A enters the halt state after state T2 of the next machine cycle. There are only three ways in which the 8080A can exit the halt state:

a. A high on the 8224 reset input (RESIN/) will always reset the 8080A to state T1; reset also clears the program counter.

b. A HOLD input will cause the 8080A to enter the hold state, as previously described. When the HOLD line goes low, the 8080A re-enters the halt state on the rising edge of the next $\phi 1$ clock pulse.

c. An interrupt (i.e., INT goes high while INTE is enabled) will cause the 8080A to exit the halt state and enter state T1 on the rising edge of the next $\phi 1$ clock pulse.

NOTE

The interrupt enable (INTE) flag must be set when the halt state is entered; otherwise, the 8080A will only be able to exit via a reset signal.

4-17. START-UP SEQUENCE. When power is initially applied to the 8080A the processor begins operating immediately. The contents of its program counter, stack pointer, and the other working registers are naturally subject to random factors and cannot be specified. For this reason, the CPU group power-up sequence begins with a reset. An external RC network is connected to the 8224 RESIN/ input. The slow transition of the power supply rise is sensed by an internal Schmitt Trigger which converts the slow transition into a clean, fast edge on the RESIN/ line when the input level reaches a predetermined value.

An active RESIN/ input to the 8224 produces a synchronoized RESET signal which restores the processor's internal program counter to zero. Program execution thus begins with memory location zero, following a reset. Systems which require the processor to wait for an explicit start-up signal will store a halt instruction (HLT) after enabling interrupts in this location. A manual or an automatic INTERRUPT will be used for starting. In other systems, the processor may begin executing its stored program immediately. Note, however, that the reset has no effect on status flags, or on any of the processor's working registers (accumulator, indices, or stack pointer). The contents of these registers remain indeterminate, until initialized by the program.

In addition to generating a RESET signal, the RESIN/ input causes the 8224 status strobe (STSTB/) output to remain true (low). This allows both the 8080A and 8238 to be reset by a power-up sequence or an externally generated RESIN/ condition.

4-18. READ/WRITE SIGNAL GENERATION

The I/O and memory read/write control signals are derived from the 8238 system controller (U49) depending on the status of the WR pin and the command data. The I/O commands (IOW/ and IOR/) are routed to the on-board PCI (U18) and the two PPI devices (U16 and U17) and to off-board peripherals via the Multibus connector (figure 5-3, sheet 3).

The memory read and write signals (MEMR/ and MEMW/) are gated off-board by the OFFBD/ signal (figure 5-3, sheet 3). The MEMW/ signal is gated to on-board RAM memory in the form of ADV MEMW/. The MEMR/ signal is used to gate the outputs of decoder U56 which is used for ROM and RAM chip select (figure 5-3, sheet 4).

Memory and I/O read timing is shown in figure 4-5; write timing is shown in figure 4-6.

4-19. ROM/PROM OPERATION

Sockets U19 through U22 are available for installation of up to 16K bytes of ROM/PROM memory. The types of devices which may be installed are listed in table 2-3. Device chip select signals (ROMCS0/ through ROMCS3/) are generated by address decode PROM U54 and decoder U56, using address bits ADRA through ADRF. The address of the byte to be read within the selected ROM/PROM is selected with address bits ADR0 through ADR9. Data is buffered by a bus transceiver (U52) and placed on the board data bus (DB0-DB7).

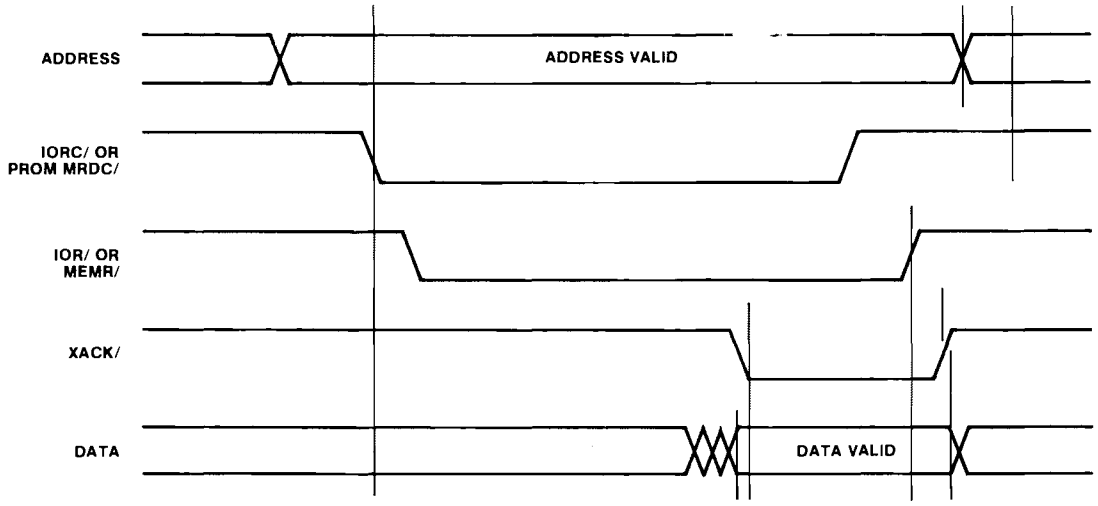


Figure 4-5. Memory and I/O Read Timing

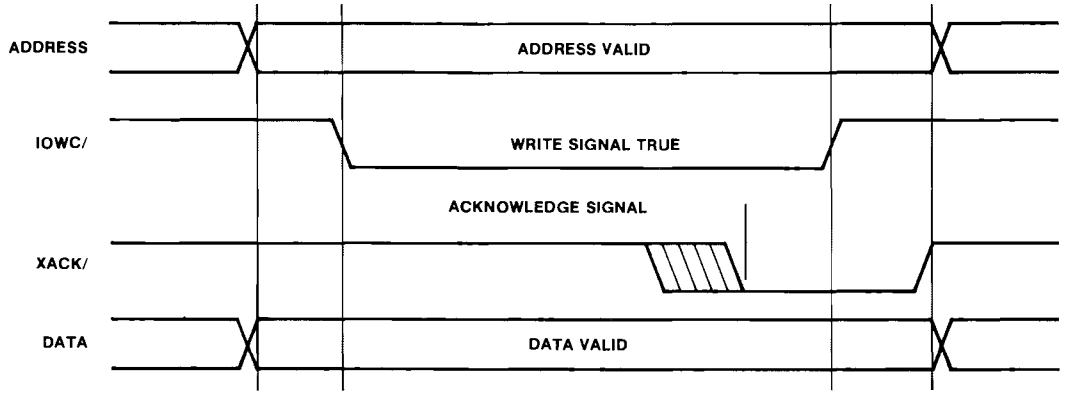


Figure 4-6. Memory and I/O Write Timing

4-20. RAM OPERATION

Sockets U34 through U41 are available for installing up to 4K bytes of RAM in 1K byte increments. Section 2-18 described the various RAM configurations. Since 4-bit RAM devices are utilized, two devices are always selected during a RAM read or write operation. RAM chip select signals (RAMCS0/ through RAMCS3/) are derived from address bits ADRA through ADRF using decode PROM U54 and decoder U56. Address bits ADR0 through ADR9 are used to select the cell locations of the data to be read or written. A write operation requires the ADV MEMW/ signal to be true (low); a read operation requires this signal to be false (high). Data is transmitted or received from the board data bus (DB0-DB7) by buffering transceiver (U52).

4-21. I/O OPERATION

The following sections describe on-board and system (off-board) I/O operations.

4-22. ON-BOARD I/O OPERATION. On-board iSBC 80/10B operations include all data and/or command byte communications between the two PPI (U16, U17) devices and the CPU; between the serial PCI device (U18) and the CPU; and between optional Multimodule devices and the CPU. Addresses for these devices are provided in table 3-1.

Chip selection for these devices is accomplished by decoding the address bits with PROM decoder U54 and U56/U57 (figure 5-3, sheet 4, zone D5). The following signals may be produced for on-board chip selection:

IOCS0/	Enables U16 PPI device
IOCS1/	Enables U17 PPI device
IOCS2/	Enables U18 PCI device
MCS0/	Enables Multimodule device 0
MCS1/	Enables Multimodule device 1

The ACK/ output of the PROM decoder (U54) will be true for any on-board address, whether it is memory or I/O (refer to Appendix C). The ACK/ signal gates U57 at pin 15, to produce either an IOACK/ or MACK/ (Figure 5-3, sheet 4). Flip-flop U26-16 will indicate whether the address is I/O (IO/M=1) or memory (IO/M=0). Data bits D4 and D6 are read during 8080A status time to set or reset this flip-flop. The resultant signal is connected to U57, pin 13. The third qualifier for U57 will be ADRF. ADRF is true for all on-board I/O addresses and false for on-board memory addresses.

The IOACK/ signal must therefore be true to properly acknowledge an on-board I/O request. Likewise, the MACK/ signal would have to be true to properly acknowledge an on-board memory request.

If both MACK/ and IOACK/ are false, the request is assumed to be off-board. For a discussion of off-board I/O operation, refer to section 4-23.

4-23. OFF-BOARD I/O OPERATION. Off-board I/O operations are handled in a similar manner to the on-board operations (section 4-21). In the off-board operations, all of the chip selects mentioned in the on-board discussion will be inactive (figure 5-3, sheet 4).

In addition, the OFFBD/ signal, generated by the acknowledge gate (U57), will be true for an off-board operation. This signal is used to turn the Multibus data drivers (U53) on, allowing the off-board data to be read from the Multibus lines or written to the Multibus lines.

4-24. SERIAL I/O INTERFACE (PCI)

The serial I/O interface logic provides the iSBC 80/10B board with a serial data communications channel that can be programmed to operate with most of the current serial data transmission protocols, synchronous or asynchronous. Baud rate, character length, number of stop bits and even/odd parity are program selectable. In addition, the serial I/O interface can be configured (through jumper connections) as an EIA RS232C interface or as a teletypewriter-compatible current loop interface.

The serial I/O interface logic consists primarily of an Intel 8251A PCI device (U18) and a counting network for baud rate selection (figure 5-3, sheet 6) and associated driver circuitry.

The PCI presents a parallel, eight-bit interface to the CPU group via the system data bus (DB0-DB7) and presents an EIA RS232C or TTY current loop interface to a peripheral (via edge connector J3). The PCI interface with the CPU group is enabled by a low level on its chip select (CS/) pin. CS/ is low when the I/O address on the system address bus is between EC and EF (hexadecimal). Address bits A through F are decoded (at U54) to produce the CS/ input. The least significant adress bit, ADR0, is applied to the PCI's C/D input (pin 12) thus indicating a control (if set) or data (if reset) byte on the data bus.

An output instruction (IOW/ is true) to port ED or EF (CS/ is low and ADR0 is high) causes the PCI to accept a control byte through its data bus pins. The control byte can be either a mode instruction or a command instruction, depending on the sequence in which it is sent. The various bits in the mode control word specify the baud rate multiplexer, character length, parity and the number of stop bits as described in Chapter 3. Note that the actual baud rate selected is dependent on the configuration of the baud rate jumper network (refer to section 2-13). The

various bits in the command control word instruct the PCI to enable/disable the receiver and transmitter, to reset errors, to reset internal control and return to the mode control cycle, and to set/clear the Data Terminal Ready output.

An output instruction to Port EC or EE (CS/ and ADR0 are low) causes the PCI to accept a data byte through its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit. The PCI will subsequently transmit the data byte (if the transmitter is enabled), in serial fashion, to the peripheral equipment.

An input instruction (IOR/ is true) to Port ED or EF (CS/ is low and ADRO is high) causes the PPI to place a status byte onto the system bus. The status bits are the result of status and error checking functions performed within the PCI.

An input instruction (IOR/ is true) to port EC or EE (CS/ and ADRO are low) causes the PCI to output a data byte (previously received from the external device) from its data bus pins. Bit 0 is the least significant bit and bit 7 is the most significant bit.

Timing for the PCI internal function is provided by the $\phi 2$TTL signal. The PCI is reset by the occurrence of a high level on the RESET line.

The PCI transmits and receives serial data, synchronously or asynchronously, as described in sections 4-25 and 4-26. By jumper-connecting the PCI pins the serial I/O logic can present either a teletypewriter-compatible current loop interface or an EIA RS232C interface to an external device (refer to Chapter 2).

Once programmed, the PCI is ready to perform its communication functions. The TxRDY output is raised high to signal the CPU that the PCI is ready to receive a character. This output (TxRDY) is reset automatically when the CPU writes a character to the PCI. On the other hand, the PCI receives serial data from the moden or I/O peripheral; upon receiving an entire character the RxRDY output is raised high to signal the CPU that the PCI has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU read operation.

The PCI cannot begin transmission until the TxEN (Transmitter Enable) bit is set in the command instruction and it has received a Clear To Send (CTS) input. The TxD output will be held in the marking state upon Reset.

4-25. ASYNCHRONOUS MODE (TRANSMIT). Whenever a data character is sent by the CPU the PCI automatically adds a start bit (low level) and the programmed number of stop bits to each character. Also, an even or odd parity bit is inserted prior to the stop bit(s), as defined by the mode instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of TxC at a rate equal to 1, 1/16 or 1/64 that of the TxC, as defined by the mode instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the PCI the TxD output remains high (marking) unless a BREAK (continuously low) has been programmed.

ASYNCHRONOUS MODE (RECEIVE). The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of RxC. If a low level is detected as the STOP bit, the framing error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the PCI. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the PCI.

4-26. SYNCHRONOUS MODE (TRANSMIT). The TxD output is continuously high until the CPU sends its first character to the PCI which usually is a SYNC character. When the CTS line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of TxC. Data is shifted out at the same rate as the TxC.

Once transmission has started, the data stream at TxD output must continue at the TxC rate. If the CPU does not provide the PCI with a character before the PCI becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin will momentarily go high to signal that the PCI is empty and SYNC characters are being sent out. The TxEMPTY pin is intenally reset by the next character being written into the PCI.

SYNCHRONOUS MODE (RECEIVE). In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has

been programmed, the receiver starts in a HUNT mode. Data on the RxD pin is then sampled in on the rising edge of RxC. The content of the Rx buffer is continuously compared with the first SYNC character until a match occurs. If the PCI has been programmed for two SYNC characters, the subsequent received character is also compared. When both SYNC characters have been detected, the PCI ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ.

Parity error and overrun error are both checked in the same way as in the Asynchronous receive mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.

4-27. SERIAL I/O INTERRUPTS. The serial I/O logic can be configured with different forms of an interrupt request mechanism. By connecting jumpers as described in section 2-14 the receiver ready (RxRDY) output (pin 14) can be used to generate an interrupt request (INT51/) to the CPU. RxRDY goes high whenever the receiver enable bit of the command word has been set and the PCI contains a character that is ready to be input to the CPU. You can also choose to have the transmitter ready (TxRDY) or the transmitter empty (TxE) output activate the INT51/ interrupt request. TxE goes high when the PCI has no characters to transmit. TxRDY is high when the PCI is ready to accept a character from the CPU. Both TxE and TxRDY are enabled by setting the transmit enable bit of the command word.

Upon receiving an interrupt, the CPU program can determine the actual condition which is responsible for the interrupt (RxRDY, TxRDY, or TxE) by reading the status of the PCI device. The interrupt request will be removed when the data is transferred to/from the PCI, as required. Note that the TxE or TxRDY output will be high, and consequently maintain an interrupt request, during all idle periods, since the PCI transmit buffer will remain empty. To disable the transmitter, and the resultant interrupt request, the CPU program can issue a command instruction to the PCI with the TxEN bit (bit 0) equal to zero. The transmitter should not be disabled until TxE is high. For additional information on the Intel 8251A PCI device, refer to the *Intel MCS-80 User's Manual,* order number 9800153.

4-28. PPI OPERATION

The iSBC 80/10B board utilizies two PPI devices, one located at U16, the other at U17. Each device has three eight-bit ports. Port addresses are provided in table 4-1.

Table 4-1. Parallel I/O Port Addresses

PPI Device Location		Eight-Bit Address (Hexadecimal)	
U16	Port (A)	E4	
	Port (B)	E5	
	Port (C)	E6	
	Control	E7	For I/O write only
U17	Port (A)	E8	
	Port (B)	E9	
	Port (C)	EA	
	Control	EB	For I/O write only

* Note: If address = 111001xx, IOCS0/ is activated.
If address = 111010xx, IOCS1/ is activated.

The PPI devices both communicate with the CPU using the same signal lines: the 8-bit data bus, DB0-DB7, and seven control/address lines; ADR0, ADR1, RESET, IOR/, IOW/. Control line IOCS0/ enables U16 while control line IOCS1/ enables U17. The data lines bring control bytes or data bytes to a PCI device or deliver data from the PCI to the CPU. The chip select control signals (IOCS0 and IOCS1) select the two devices, respectively, when the proper I/O address appears on the address bus. IOCS0 and IOCS1 are the result of decoding ACK/, IO/M, A15 and D0 and D1 of U54. The two least significant address bits select the control register (when programming an 8255A) or one of the three I/O ports (when reading or writing data). IOR/ (PPI → CPU) and IOW/ (CPU → PPI) indicate the direction of data flow, as summarized in table 4-2.

A high on the RESET line clears all internal PPI registers including the control register; all ports (A, B and C) are set for input.

Though both PPI devices maintain the same interface (at different I/O addresses) with the CPU, the interface between the U16 device and edge connector J1 is significantly different than the interface between the U17 device and its associated edge connector (J2). This gives you a great deal of flexibility when configuring the system's external parallel I/O devices. Because of those flexible external interfaces, however, not all ports are capable of operating in each PPI mode, though all ports can be programmed as either input or output. The U16 ports (E4, E5, and E6) can fully utilize the PPI multimode and external interrupt capabilities as described in section 3-17. The U17 ports (E8, E9, and EA) however, are limited to a single mode of operation. The allowable port configurations for both PPI devices are summarized in table 3-7.

Table 4-2. PPI Basic Operation

A1	A0	IOR/	IOW/	CS/	Input Operation (Read)
0	0	0	1	0	Port A — Data Bus
0	1	0	1	0	Port B — Data Bus
1	0	0	1	0	Port C — Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus — Port A
0	1	1	0	0	Data Bus — Port B
1	0	1	0	0	Data Bus — Port C
1	1	1	0	0	Data Bus — Control
					Disable Function
x	x	x	x	1	Data Bus — High-Impedance
1	1	0	1	0	Illegal

4-29. MULTIBUS INTERFACE

The iSBC 80/10B board Multibus interface circuitry is shown in figure 5-3, sheet 3. Data is gated on or off the board by the 8287 bus transceiver (U53). Address drivers U50 and U51 handle the off-board addresses. Multibus interface signals used by the iSBC 80/10B board are defined in table 2-11.

4-30. INTERRUPT HANDLING

The iSBC 80/10B board has no dedicated interrupt controller other than the CPU itself. The board accepts two external (off-board) interrupts via J1-49 and P1-42. All interrupts, internal and external, are of the same priority, and are wired to one input on the CPU. For additional information on interrupt handling by the Intel 8080A CPU, refer to the *MCS-80 User's Manual*, order number 9800153.

4-31. MULTIMODULE I/O BOARD OPERATION

The iSBC 80/10B board CPU may communicate with one or two I/O devices which reside on a Multimodule board. Two chip select signals are used for Multimodule applications: MCS0/ and MCS1/. Multimodule ports F0 through F7 are enabled by MCS0/ and Multimodule ports F8 through FF are enabled by MCS1/. Multimodule interrupts (MINTR1, MINTR2) must be enabled by jumper connection. Refer to figure 5-3, sheet 8.

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; and

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 MCSD 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 MCSD products, the serial number is usually stamped on a label.

d. Shipping and 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 agreement.

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.

5-3. REPLACEMENT PARTS

A complete list of replacement parts is provided in table 5-1. This list provides the part number,

Table 5-1. Replaceable Parts

Reference Designation	Description	Mfr. Part No.	Mfr. Code	Qty.
U1, U53	Bus Transceiver, Octal	8287	INTEL	2
U12	Line Driver, Quad	75188N	TI	1
U13	Line Receiver, Quad	75189AN	TI	1
U14, U15	Counter, Synchronous 4-Bit	74LS161	TI	2
U16, U17	Programmable Peripheral Interface	8255A	INTEL	2
U18	Programmable Communications Interface	8251A	INTEL	1
U23	Binary Counter, 4-Bit	74S163	TI	1
U24, U45	Hex Inverter	74LS04	TI	2
U25	One Shot Multivibrator	9602PC	Fairchild	1
U26	Bistable Latch, 4-Bit	74LS75	TI	1
U27	Counter, Dual Decade	74LS390	TI	1
U28	Buffer Gate, 3-State	74126	TI	1
U29	Clock Generator/Driver	D8224	INTEL	1
U30	Nand Gate, Positive 2-Input	74LS00	TI	1
U31	Flip-Flop, D-Type	74LS74	TI	1

Table 5-1. Replaceable Parts (Cont.)

Reference Designation	Description	Mfr. Part No.	Mfr. Code	Qty.
U32	Or Gate, Positive 2-Input	74LS32	TI	1
U33	Microprocessor, 8-Bit	8080A	INTEL	1
U37, U41	RAM, Static 1024 x 4-Bit	2114	INTEL	2
U46	Nand Gate, Positive 4-Input	7420	TI	1
U47	Nand Buffer, Positive 2-Input	7438	TI	1
U48	Buffer, Tri-State Hex	8097	National	1
U49	System Controller/Driver	C8238	INTEL	1
U50, U51	Buffer/Driver/Receiver, Octal	74LS240	TI	2
U52	Bus Transceiver, Octal	8286	INTEL	1
U54	PROM (pre-programmed)	91-00268	INTEL	1
U54	PROM (not programmed)	3625A	INTEL	1
U55	Exclusive Or Gate	74LS86	TI	1
U56	Decoder, 3-to-8 Line	74LS138	TI	1
U57	Decoder, 2-to-4 Line	74LS139	TI	1
U58	Or Gate, Positive 2-Input	74S32	TI	1
U59	Nor Gate, Positive 2-Input	74LS02	TI	1
U60	Nand Gate, Positive 3-Input	74S10	TI	1
C1-7	Capacitor, Tantalum, 22 uf, 10%, 15V	OBD	CML	7
C8	Capacitor, Mica, 10 pf, 5%, 500V	OBD	CML	1
C9	Capacitor, Mica, 330 pf, 5%, 500V	OBD	CML	1
C10-59	Capacitor, Ceramic, .1 uf, +80% -20%, 50V	OBD	CML	50
CR1	Zener Diode	1N5231A	CML	1
CR2	Diode	1N914	CML	1
R1, R2, R16, R18, R28	Resistor, 10KΩ, ±5%, 1/4W	OBD	CML	5
R3	Resistor, 220Ω, ±5%, 1/4W	OBD	CML	1
R4, R9, R10	Resistor, 2.7KΩ, ±5%, 1/4W	OBD	CML	3
R5, R6	Resistor, Wirewound, 250Ω, ±5%, 1W	OBD	CML	2
R7	Resistor, Wirewound, 47Ω, ±5%, 2W	OBD	CML	1
R8	Resistor, 47Ω, ±5%, 1/2W	OBD	CML	1
R11	Resistor, 2.4KΩ, ±5%, 1/4W	OBD	CML	1
R14	Resistor, 330KΩ, ±5%, 1/4W	OBD	CML	1
R15	Resistor, 33KΩ, ±5%, 1/4W	OBD	CML	1
R19	Resistor, 100KΩ, ±5%, 1/4W	OBD	CML	1
RP1	Resistor Pack, 1KΩ, 10-Pin, SIP	OBD	CML	1
RP2, RP3, RP4	Resistor Pack, 10KΩ, 10-Pin, SIP	OBD	CML	3
Q1	Transistor	2N2905A	CML	1
Q2	Transistor	2N2907	CML	1
Q3	Transistor	2N3904	CML	1
Y1	Crystal, 18.432 MHz	OBD	CML	1
XU1	I.C. Socket, 20-Pin	OBD	CML	1
XU2-11, XJ6, XJ7	I.C. Socket, 14-Pin	OBD	CML	13
XJ5	I.C. Socket, 16-Pin	OBD	CML	1
XU19-22	I.C. Socket, 24-Pin	OBD	CML	4
XU33	I.C. Socket, 40-Pin	OBD	CML	1
J4	Connector, 36-Pin (Multimodule)	68-358	INTEL	1
P3, P4	Connector Shorting Plug (Header), 4-Position	OBD	CML	2
—	Card Ejector	S203	Scanbe	2

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 description. These items are called out as CML in the table rather than listing the part number. Figure 5-1 shows the location of each referenced part in table 5-1.

Table 5-2. List of Manufacturers' Codes

Mfr. Code	Manufacturer	Address	Mfr. Code	Manufacturer	Address
INTEL	Intel Corporation	Santa Clara, CA	Fairchild	Fairchild Electronics	
National	National Semiconductor Corporation	Santa Clara, CA	TI	Texas Instruments	Dallas, TX
			Scanbe	Scanbe, Incorporated	El Monte, CA
			CML	Available from any commercial source; order by description (OBD)	

5-4. SERVICE DIAGRAMS

Schematic diagrams of the iSBC 80/10B board are provided in figure 5-3, sheets 1 through 8. Notice that a functional description of each jumper connection on a particular schematic sheet is referenced to the left of the fold-out sheet.

The schematic diagrams in figure 5-3 are current when the manual is printed. However, minor revisions to the diagrams may occur between manual printings. Therefore, Intel provides copies 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 included in the manual.

5-5. INTERNAL SIGNALS

Internal board signals which traverse from one sheet to another in figure 5-3 are identified by a single or double alpha character within a box (e.g., C or AN). The signal mnemonic is shown adjacent to the boxed character, along with the source or destination sheet number (e.g., SH 2 ADRB BF). Signals coming into the sheet are shown on the left side of the diagram. Conversely, signals leaving the sheet are shown on the right side.

To follow a signal from one sheet to another, read the sheet number and boxed character, then look for the same boxed character on the indicated sheet. For example, if you are going to trace the path of MSEL/ when it exits sheet 4, the first step would be to turn to the indicated sheet. Since MSEL/ will be entering sheet 5, as indicated on sheet 4, look for the AJ symbol on the left side of the sheet. Notice that the inputs also list the source sheet number (sheet 4 in this example).

Each signal will keep the same boxed character throughout figure 5-3. This will enable you to trace the signal to any sheet with minimal effort.

The internal board signal mnemonics are listed and defined in table 5-3. The signals are listed according to boxed code alphabetical order.

Table 5-3. Glossary of Internal Signal Mnemonics

Code	Signal Mnemonic	Description
A	INT55	Parallel Port or Multimodule Interrupt
B	INT51	Serial Port Interrupt
C	RDYIN	Ready Input
D	RESET	Reset CPU (INIT/)
E	ADR0	Address Line 0
F	D0, D1, D3, D4, D6	Data Lines (Internal)
G	WAIT/	Acknowledges that CPU is in a wait state (Output)
H	IOW/	I/O Write
J	IOR/	I/O Read
K	MEMR/	Memory Read
L	ADV MEM W/	Advance Memory Write
M	DB0-DB7	Data Bus Lines
N	P.U.	1K Ohm Pull-Up Resistor
P	DHLDA	Delayed Hold Acknowledge
R	ϕ2(TTL)'	Clock Pulse
S	OSC	Oscillator Output
T	ϕ2(TTL)	Clock Pulse
U	OFFBD/	Off-Board; Turns Multibus Drivers On/Off
V	RDCMD/	Read Command
W	MCLK	Master Clock Signal; Generates BCLK & CCLK
X	TO	Test Point
Y	ROM SEL 0	Identifies size of on-board ROM array for decode PROM (U54)
AA	MWAIT/	Multimodule Wait (See Table 2-11)
AB	ROMCS3/	Selects U22 ROM
AC	ROMCS2/	Selects U21 ROM
AD	ROMCS1/	Selects U20 ROM
AE	ROMCS0/	Selects U19 ROM
AF	RAMCS0/	Selects U34, U38 RAM Pair
AG	RAMCS1/	Selects U35, U39 RAM Pair
AH	RAMCS2/	Selects U36, U40 RAM Pair
AJ	MSEL/	Memory Select
AK	IOCS0/	Selects U16 PPI Device

Table 5-3. Glossary of Internal Signal Mnemonics (Cont'd)

Code	Signal Mnemonic	Description
AL	IOCS1/	Selects U17 PPI Device
AM	IOCS2/	Selects U18 PCI Device
AN	MCS0/	Multimodule Chip Select 0
AP	MPST/	Multimodule Present
AS	MST	Millisecond Timer Output
AT	INT55A	See INT55 (A)
AU	MEMW/	Memory Write
AV	SYNC	Timing Signal Derived from CPU SYNC Output
AW	DHLDA/	Complement of DHLDA (P)
AX	ROM SEL 1	See ROM SEL 0 (Y)
AY	RAM CS3/	Selects U37, U41 RAM Pair
BA	MCS1/	Multimodule Chip Select 1
BB	CTI	Clear Millisecond Timer Input
BC	ADR1	Address Line 1
BD	ADR8,9	Address Line 8, 9
BE	ADRA	Address Line A
BF	ADRB-ADRF	Address Line B-F
BG	ADR8-ADR9	Address Line 8, 9
BH	WAIT	Wait State
BI	ADRB	Address Line B

Sheet 4 Signals —	ACK/	On-Board Address Acknowledge
	MACK/	On-Board Memory Acknowledge
	I/O ACK/	On-Board I/O Acknowledge
	ONBD	Qualified On-Board Address
	IO/M	IO/Memory Qualifier Signal

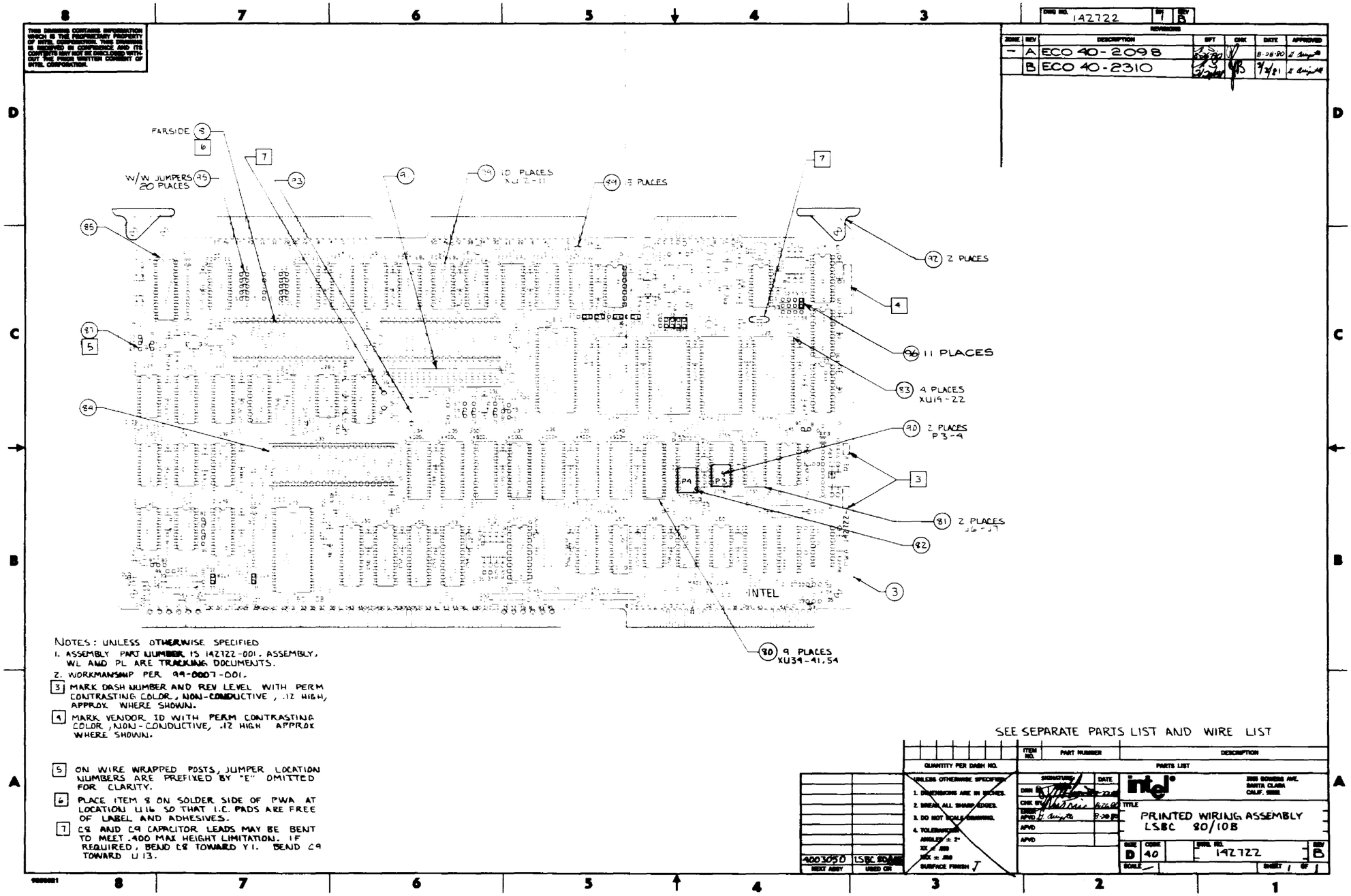


Figure 5-1. iSBC 80/10B™ Parts Location Diagram

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

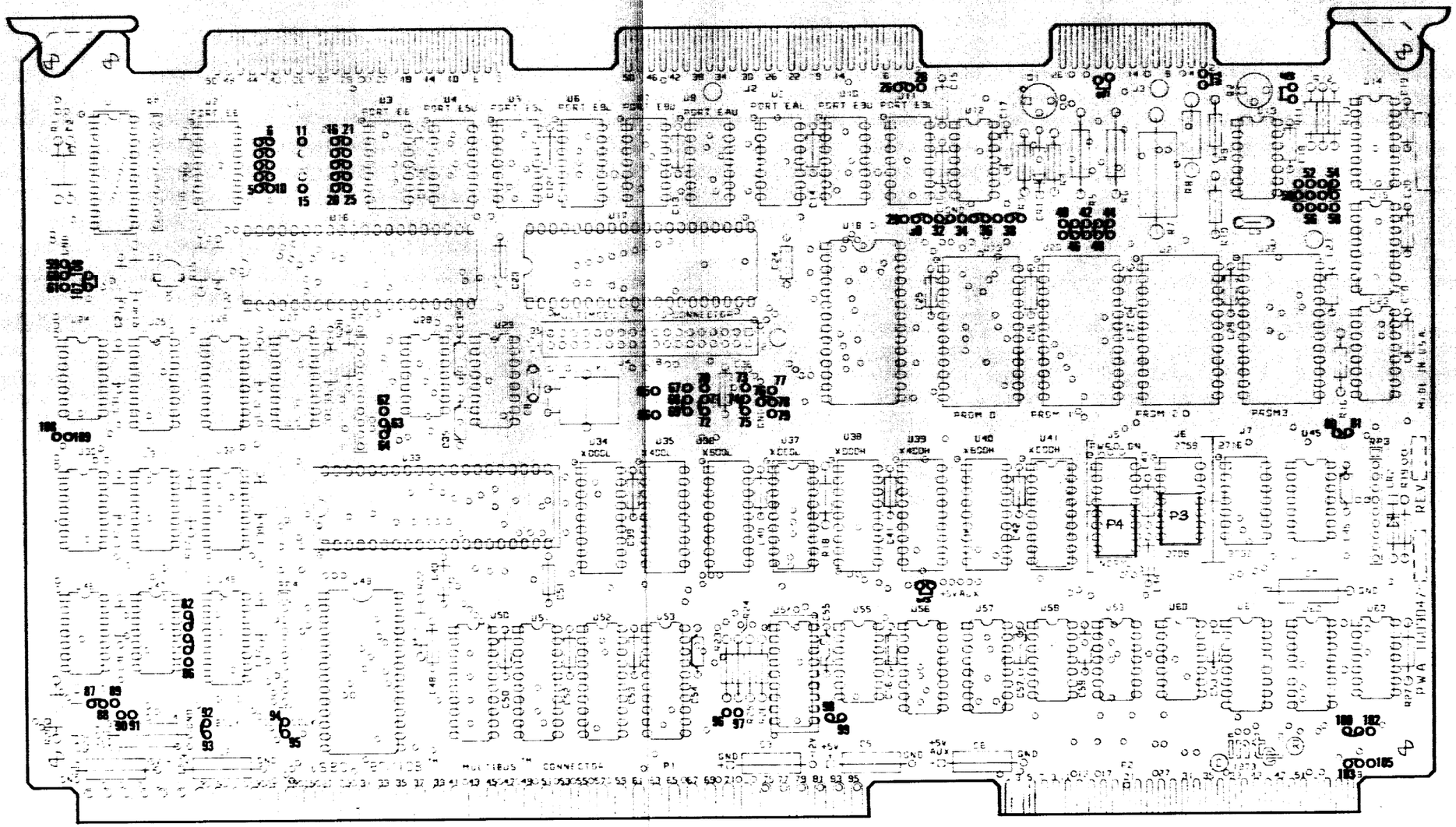
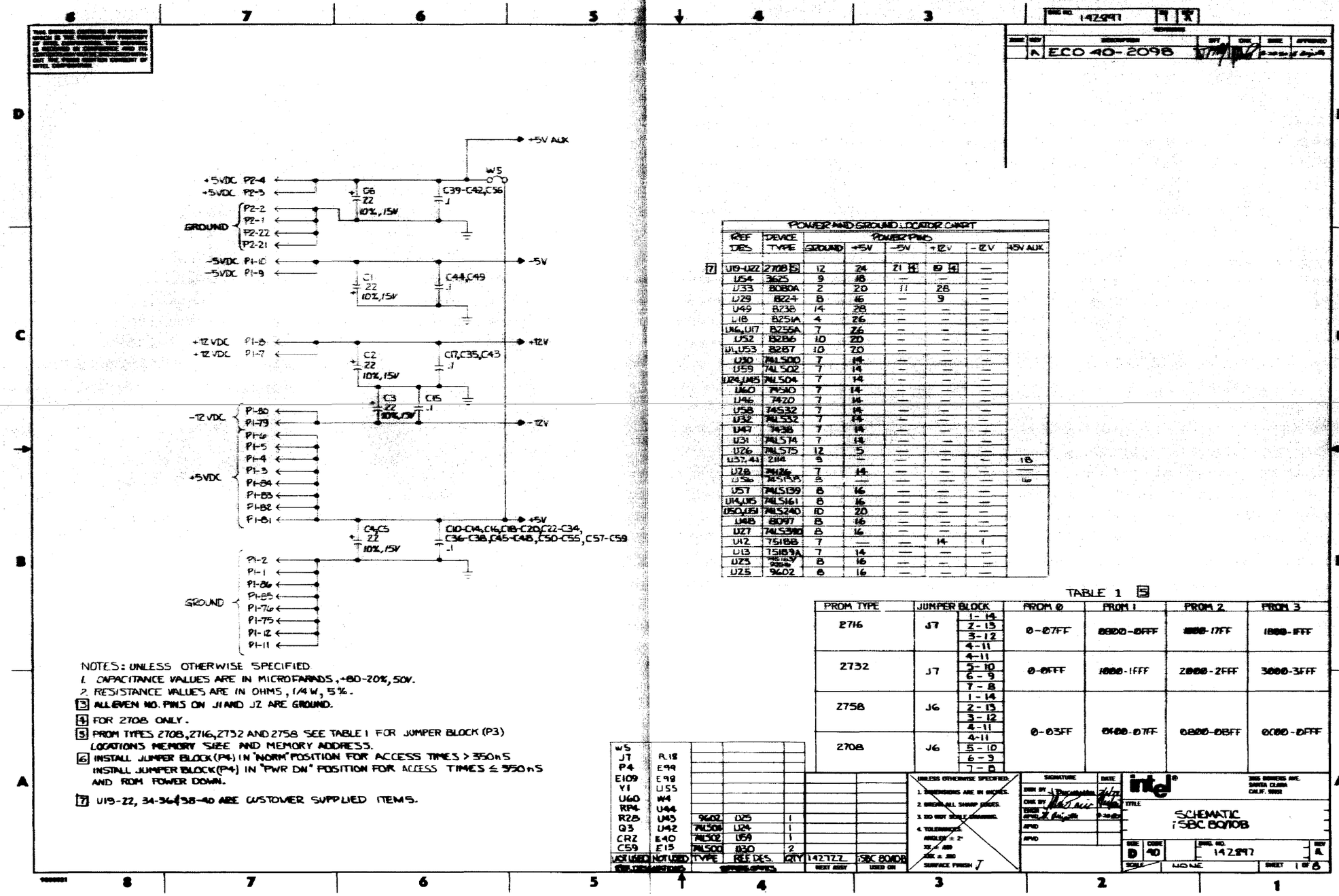


Figure 5-2. iSBC 80/10B™ Jumper Post Locations



NOTES: UNLESS OTHERWISE SPECIFIED

1. CAPACITANCE VALUES ARE IN MICROFARADS, +80-20%, 50V.
2. RESISTANCE VALUES ARE IN OHMS, 1/4 W, 5%.
3. ALL EVEN NO. PINS ON J1 AND J2 ARE GROUND.
4. FOR 2708 ONLY.
5. PROM TYPES 2708, 2716, 2732 AND 2758 SEE TABLE 1 FOR JUMPER BLOCK (P3) LOCATIONS MEMORY SIZE AND MEMORY ADDRESS.
6. INSTALL JUMPER BLOCK (P4) IN "NORM" POSITION FOR ACCESS TIMES > 350nS
   INSTALL JUMPER BLOCK (P4) IN "PWR DN" POSITION FOR ACCESS TIMES ≤ 350nS AND ROM POWER DOWN.
7. U19-22, 34-36, 38-40 ARE CUSTOMER SUPPLIED ITEMS.

POWER AND GROUND LOCATOR CHART

REF DES	DEVICE TYPE	POWER PINS: GROUND	+5V	-5V	+12V	-12V	+5V AUX
7 U19-U22	2708 5	12	24	21 4	8 4	—	
U54	3625	9	18	—	—	—	
U33	8080A	2	20	11	28	—	
U29	8224	8	16	—	9	—	
U49	8238	14	28	—	—	—	
U18	8251A	4	26	—	—	—	
U16, U17	8256A	7	26	—	—	—	
U52	8286	10	20	—	—	—	
U1, U53	8287	10	20	—	—	—	
U30	74LS00	7	14	—	—	—	
U59	74LS02	7	14	—	—	—	
U24, U45	74LS04	7	14	—	—	—	
U60	74S10	7	14	—	—	—	
U46	7420	7	14	—	—	—	
U58	74S32	7	14	—	—	—	
U32	74LS32	7	14	—	—	—	
U47	7438	7	14	—	—	—	
U31	74LS74	7	14	—	—	—	
U26	74LS75	12	5	—	—	—	
U37, 41	2114	9	—	—	—	—	18
U28	74126	7	14	—	—	—	—
U56	74LS138	8	—	—	—	—	16
U57	74LS139	8	16	—	—	—	
U14, U15	74LS161	8	16	—	—	—	
U50, U51	74LS240	10	20	—	—	—	
U48	8097	8	16	—	—	—	
U27	74LS390	8	16	—	—	—	
U12	75188	7	—	—	14	1	
U13	75189A	7	14	—	—	—	
U23	9334	8	16	—	—	—	
U25	9602	8	16	—	—	—	

TABLE 1 5

PROM TYPE	JUMPER	BLOCK	PROM 0	PROM 1	PROM 2	PROM 3
2716	J7	1-14, 2-13, 3-12, 4-11	0-07FF	0800-0FFF	1000-17FF	1800-1FFF
2732	J7	4-11, 5-10, 6-9, 7-8	0-0FFF	1000-1FFF	2000-2FFF	3000-3FFF
2758	J6	1-14, 2-13, 3-12, 4-11	0-03FF	0400-07FF	0800-0BFF	0C00-0FFF
2708	J6	4-11, 5-10, 6-9, 7-8				

WS					
J7	R18				
P4	E99				
E109	E98				
Y1	U55				
U60	W4				
RP4	U44				
R28	U43	9602	U25	1	
Q3	U42	74LS08	U24	1	
CR2	E40	74LS02	U59	1	
C59	E13	74LS00	B30	2	

USE AS REQUIRED	NOT USED	TYPE	REF DES.	QTY	142722 NEXT ASSY	iSBC 80/10B USED ON

UNLESS OTHERWISE SPECIFIED:
1. DIMENSIONS ARE IN INCHES.
2. BREAK ALL SHARP EDGES.
3. DO NOT SCALE DRAWING.
4. TOLERANCES: ANGLES ± 2°; XX ± .010; XXX ± .005
SURFACE FINISH 7

SIGNATURE	DATE
DRWN BY	
CHK BY	
ENGR APPD	
APVD	
APVD	

intel 3065 BOWERS AVE. SANTA CLARA CALIF. 95051

TITLE: SCHEMATIC iSBC 80/10B

SIZE D | CODE 80 | DWG NO. 142297 | REV A

SCALE NONE | SHEET 1 OF 8

REF DES	DESCRIPTION	DATE	BY	CHK	APPROVED
A	ECO 40-2098				

142897 A

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

Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 1 of 8)

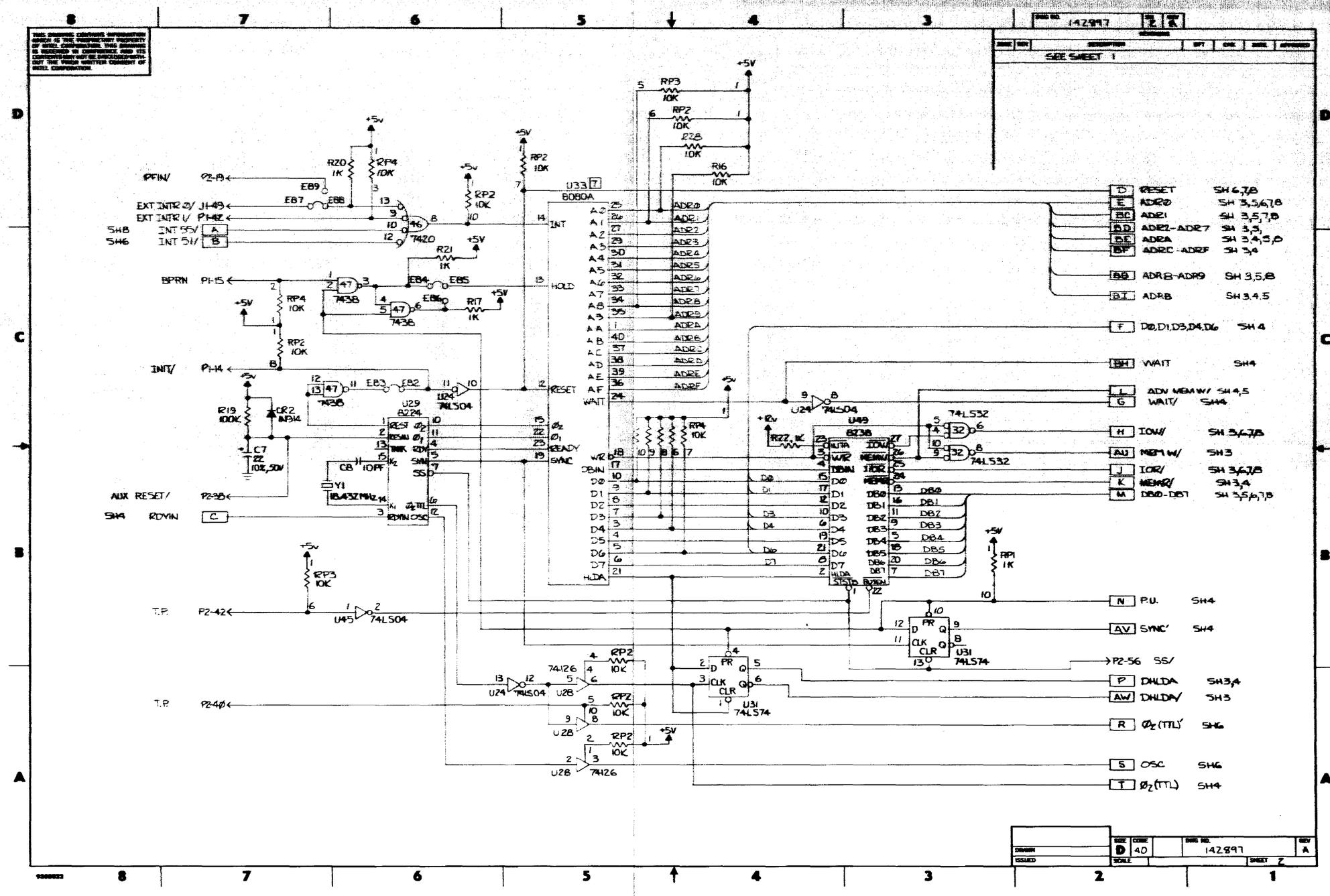


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 2 of 8)

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

Jumper Configuration

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
92-93*	Connects BCLK/ to Multibus	3 A3	2-21
94-95*	Connects CCLK/ to Multibus	3 A3	2-21
Note: * indicates default connection			

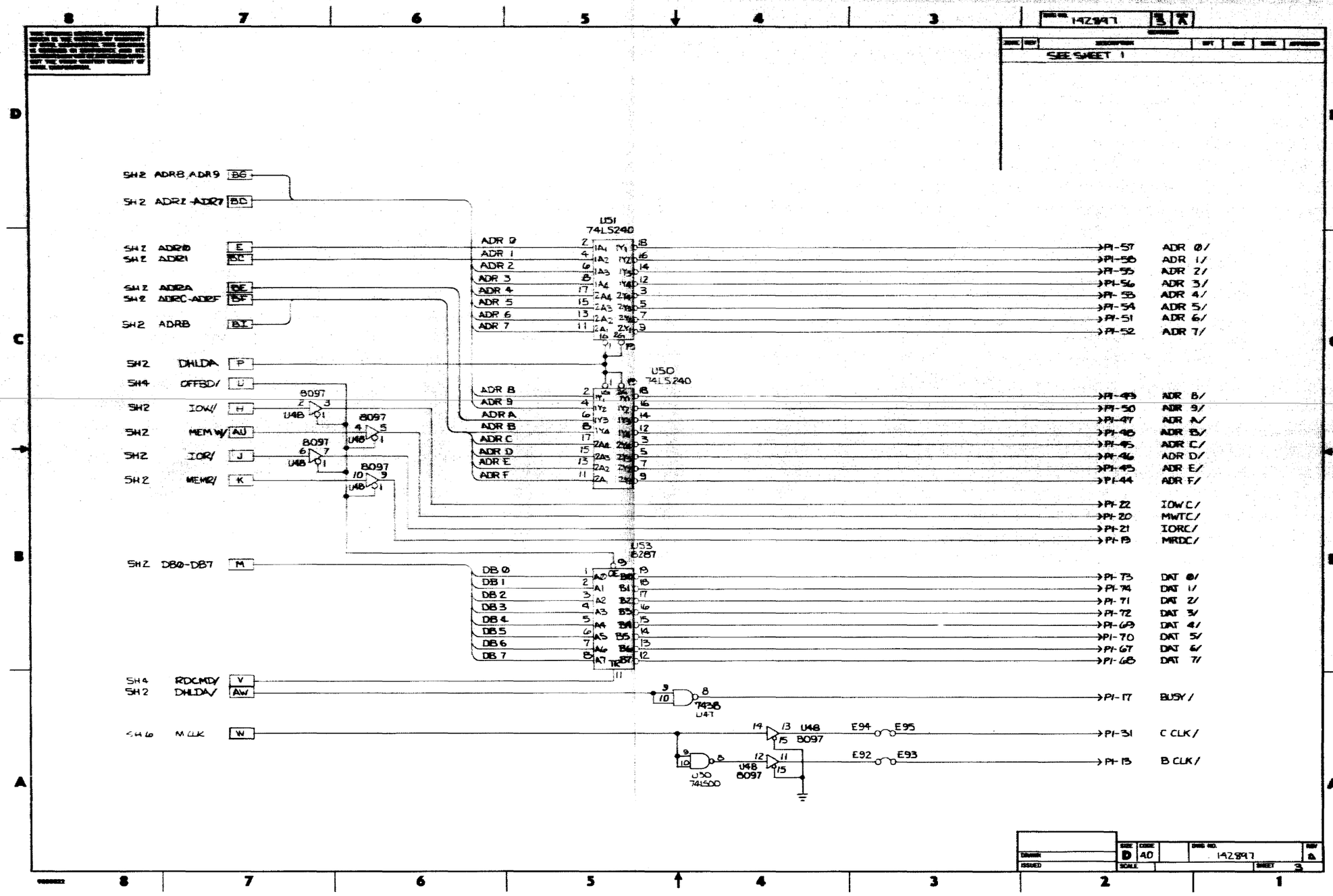


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 3 of 8)

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

Jumper Configuration

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
82-83*	Connects RESET to Multibus	2 C6	2-21
84-85*	Connects BPRN to board	2 C6	2-23
85-86	Implements BPRN/ (Multibus compatible)	2 C6	2-23
87-88*	Connects INTR0 to CPU INT	2 D7	2-21
88-89	Connects PFIN/ to CPU INT	2 D7	2-25

Note: * indicates default connection

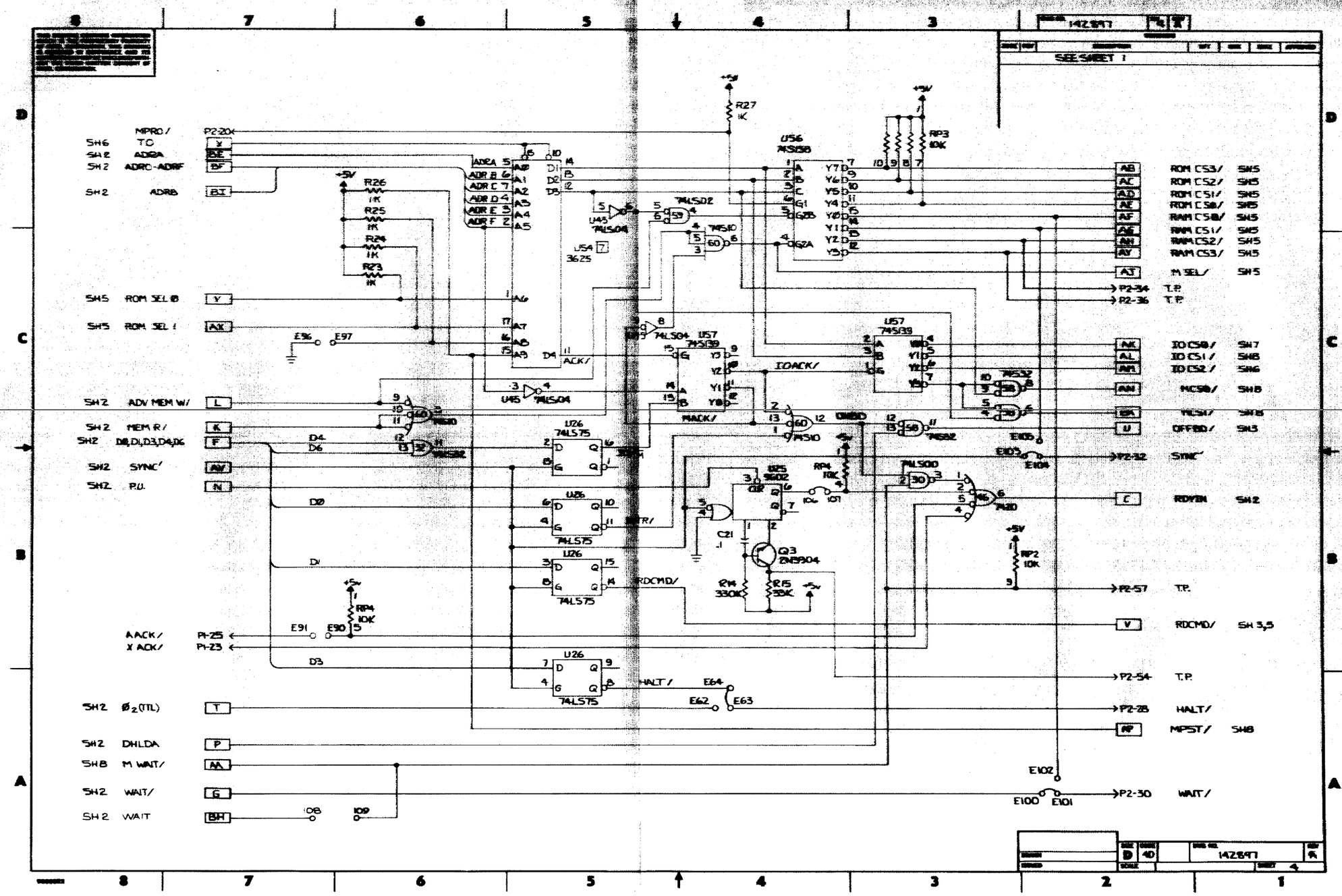


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 4 of 8)

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

Jumper Configuration

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
63-64*	Connects HALT/ to P2 Connector	4 A4	2-21
90-91	Connects AACK/ to board	4 B7	None
96-97	Specifies amount of on-board RAM	4 C7	2-9
100-101*	Connects WAIT/ to P2 Connector	4 A2	2-21
103-104*	Connects SYNC to P2 Connector	4 C3	None
106-107*	Enables failsafe timer	4 B4	2-19
108-109	Wait State Jumper	4 A7	2-8

Note: * indicates default connection

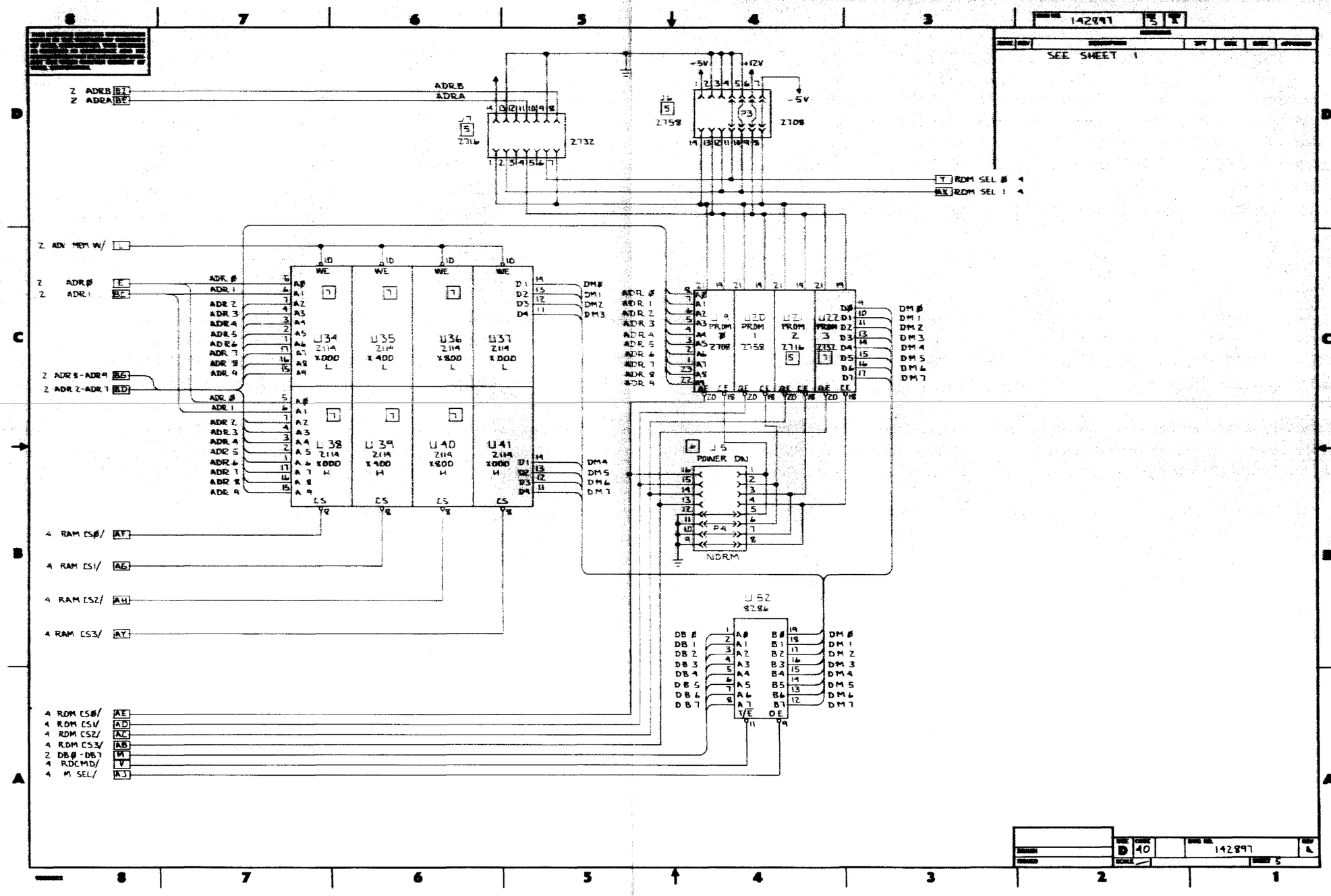


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 5 of 8)

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

Jumper Configuration

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
30-31*	Connects RTS/ to CTS/	6 A4	None
32-33*	Sets CTS driver to +12 volts	6 A4	None
34-35	Connects TxD to RS232C driver	6 B4	2-15
35-36*	Connects TxD to TTY driver	6 B4	None
37-38	Connects DTR to RS232C driver	6 B4	2-15
38-39*	Connects DTR to TTY driver	6 B4	None
41-46*	Connects TTY return to RxD	6 B5	None
41-42	Connects external clock to RxC	6 B5	2-16
42-47*	Connects internal clock to RxC	6 B5	None
45-46	Connects RS232C data to RxD	6 B5	2-15
43-48*	Connects internal clock to TxC	6 B5	None
44-49*	Connects DTR to DSR input	6 B5	None
48-49	Connects external clock to TxC	6 B5	2-16
50-54* 80-81*	Selects 110 baud for PCI device (see table 2-9)	6 D4	2-13
74-75*	Disable RxRDY interrupt	6 C4	2-14
73-74	Enable RxRDY interrupt	6 C4	2-14
76-78*	Disable TxRDY interrupt	6 B4	2-14
78-79	Enable TxRDY interrupt	6 B4	2-14
77-78	Enable TxE interrupt	6 B4	2-14
W1	-12V to J3-19	6 B2	2-27
W2	GND to J3-1	6 A7	None
W3	+12V to J3-22	6 C6	2-28

Note: * indicates default connection

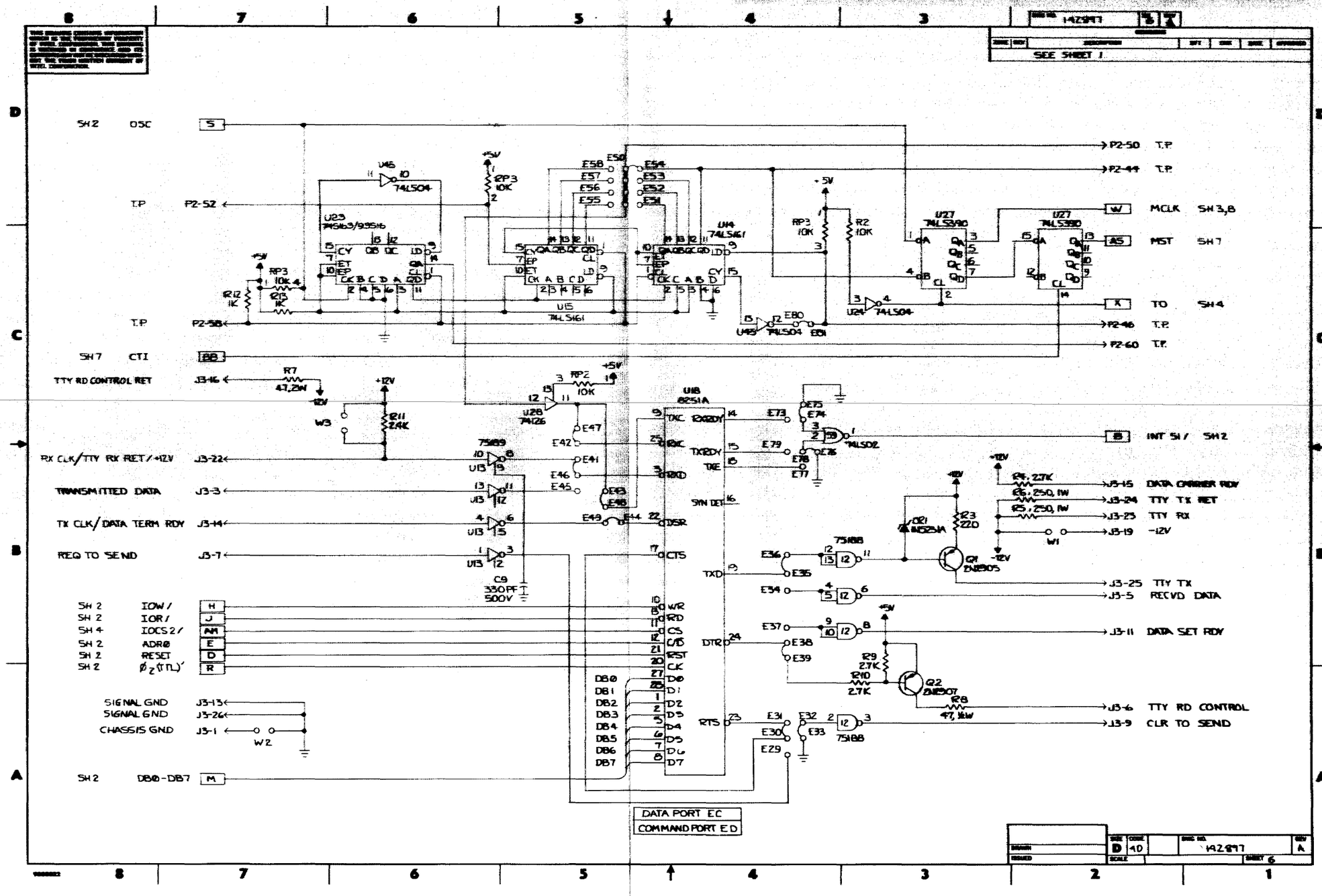


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 6 of 8)

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

Jumper Configuration

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
59-60*	Sets Port E4 to output mode	7 C5	2-17
60-61	Port E4 mode programmed by Port E6 Bit 6	7 C5	2-17
5-10*	Disable E6 interrupt	7 B5	2-17
20-25*	Disable E6 interrupt	7 B5	2-17
11-X	Enable CTI for millisecond timer	7 C4	2-20
14-X	Enable PFSR/	7 C4	2-25
12-X	Enable PFSN/	7 C4	2-25
15-X	Enable millisecond timer (MST)	7 B4	2-20
1 thru 4 6 thru 9 21 thru 24 16 thru 19	Configure port E6 bits	7 B5	2-17

Note: * indicates default connection

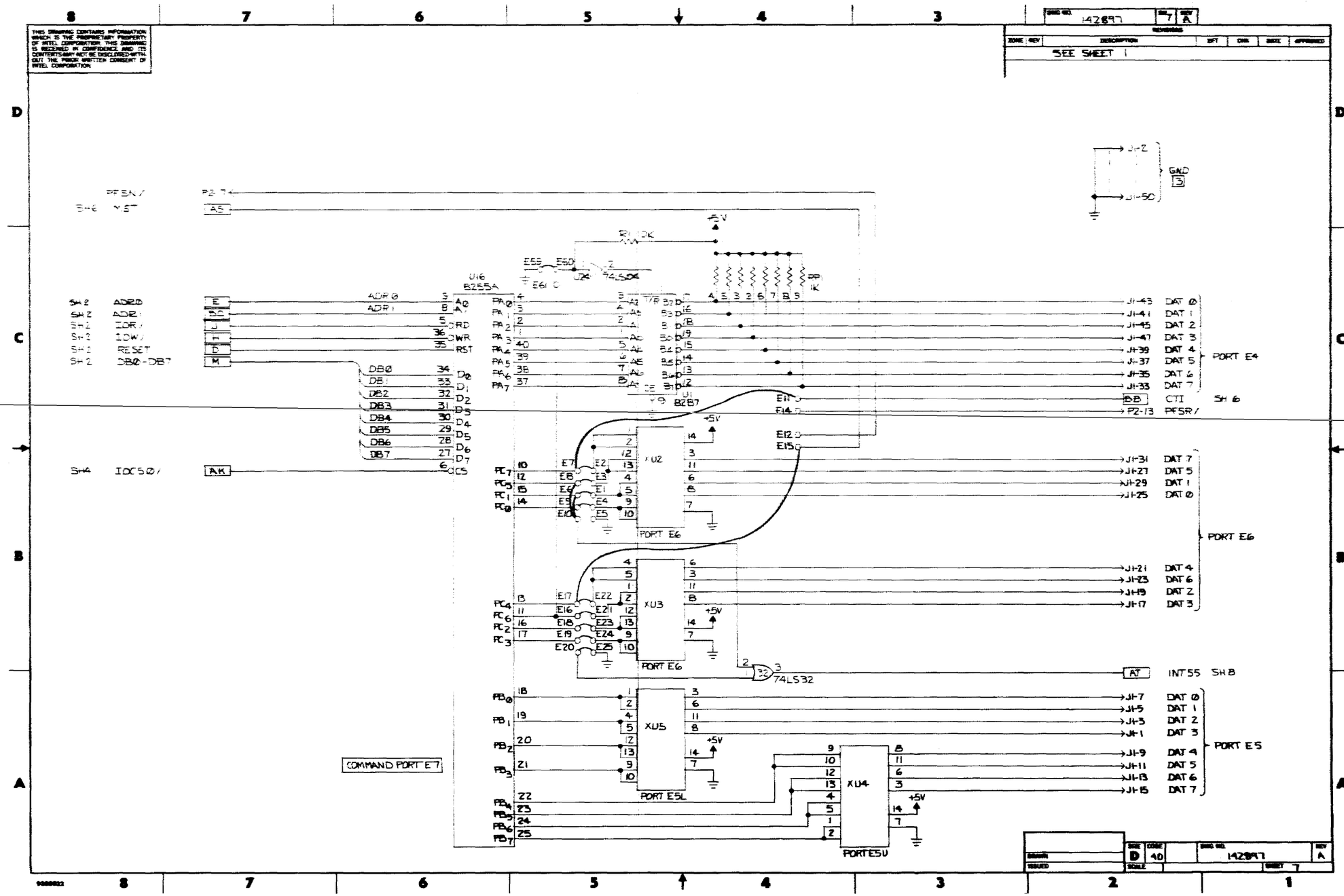


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 7 of 8)

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

Jumper Configuration;

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
26-27*	GND to J2-1	8 B2	None
27-28	+5V to J2-1	8 B2	2-28
68-69*	Disables Multimodule interrupt 0 (MINTR0)	8 A6	2-24
67-68	Enables Multimodule interrupt 0 (MINTR0)	8 A6	2-24
71-72*	Disables Multimodule interrupt 1 (MINTR1)	8 A6	2-24
70-71	Enables Multimodule interrupt 1 (MINTR1)	8 A6	2-24
Note: * indicates default connection			

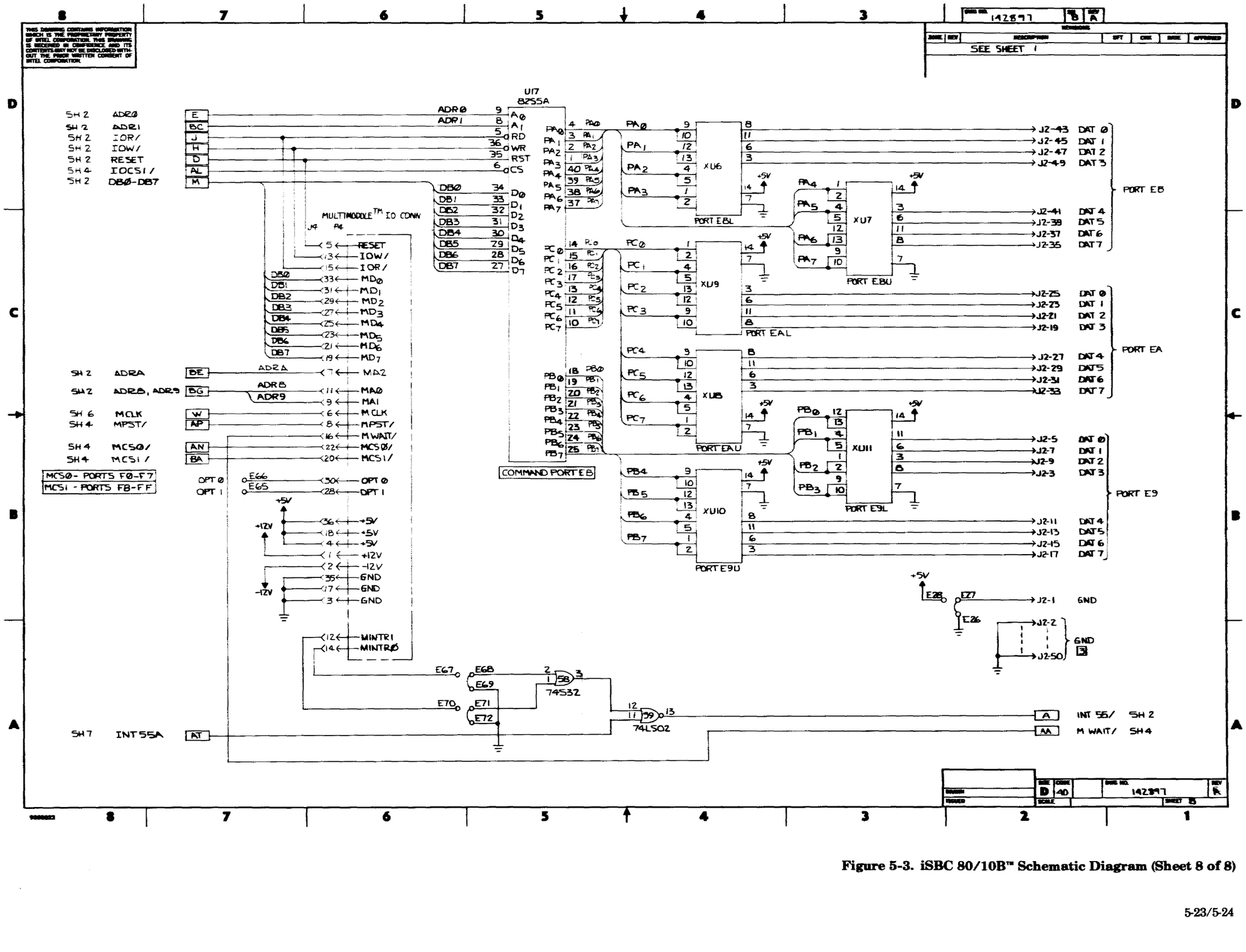


Figure 5-3. iSBC 80/10B™ Schematic Diagram (Sheet 8 of 8)

APPENDIX A
8080 INSTRUCTION SET

A computer, no matter how sophisticated, can only do what it is "told" to do. One "tells" the computer what to do via a series of coded instructions referred to as a **Program**. The realm of the programmer is referred to as **Software**, in contrast to the **Hardware** that comprises the actual computer equipment. A computer's software refers to all of the programs that have been written for that computer.

When a computer is designed, the engineers provide the Central Processing Unit (CPU) with the ability to perform a particular set of operations. The CPU is designed such that a specific operation is performed when the CPU control logic decodes a particular instruction. Consequently, the operations that can be performed by a CPU define the computer's **Instruction Set**.

Each computer instruction allows the programmer to initiate the performance of a specific operation. All computers implement certain arithmetic operations in their instruction set, such as an instruction to add the contents of two registers. Often logical operations (e.g., OR the contents of two registers) and register operate instructions (e.g., increment a register) are included in the instruction set. A computer's instruction set will also have instructions that move data between registers, between a register and memory, and between a register and an I/O device. Most instruction sets also provide **Conditional Instructions**. A conditional instruction specifies an operation to be performed only if certain conditions have been met; for example, jump to a particular instruction if the result of the last operation was zero. Conditional instructions provide a program with a decision-making capability.

By logically organizing a sequence of instructions into a coherent program, the programmer can "tell" the computer to perform a very specific and useful function.

The computer, however, can only execute programs whose instructions are in a binary coded form (i.e., a series of 1's and 0's), that is called **Machine Code**. Because it would be extremely cumbersome to program in machine code, programming languages have been developed. There are programs available which convert the programming language instructions into machine code that can be interpreted by the processor.

One type of programming language is **Assembly Language**. A unique assembly language mnemonic is assigned to each of the computer's instructions. The programmer can write a program (called the **Source Program**) using these mnemonics and certain operands; the source program is then converted into machine instructions (called the **Object Code**). Each assembly language instruction is converted into one machine code instruction (1 or more bytes) by an **Assembler** program. Assembly languages are usually machine dependent (i.e., they are usually able to run on only one type of computer).

THE 8080 INSTRUCTION SET

The 8080 instruction set includes five different types of instructions:

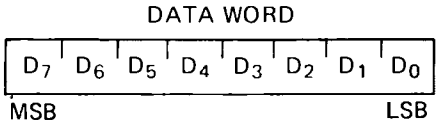
- **Data Transfer Group**—move data between registers or between memory and registers
- **Arithmetic Group** – add, subtract, increment or decrement data in registers or in memory
- **Logical Group** – AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory
- **Branch Group** – conditional and unconditional jump instructions, subroutine call instructions and return instructions
- **Stack, I/O and Machine Control Group** – includes I/O instructions, as well as instructions for maintaining the stack and internal control flags.

Instruction and Data Formats:

Memory for the 8080 is organized into 8-bit quantities, called Bytes. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

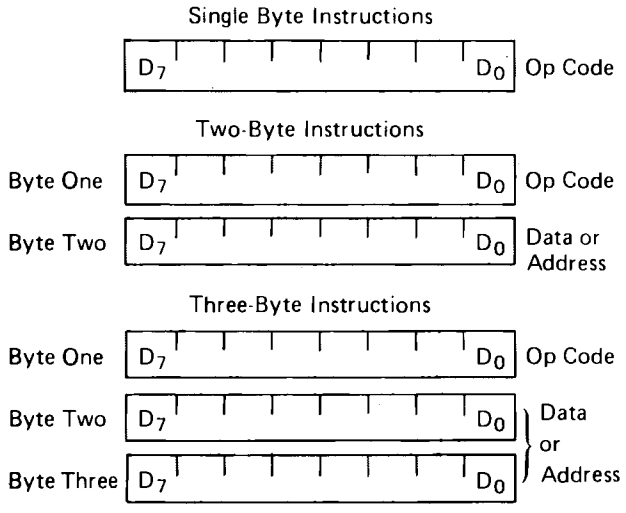
The 8080 can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8080 is stored in the form of 8-bit binary integers:



When a register or data word contains a binary number, it is necessary to establish the order in which the bits of the number are written. In the Intel 8080, BIT 0 is referred to as the **Least Significant Bit (LSB)**, and BIT 7 (of an 8 bit number) is referred to as the **Most Significant Bit (MSB).**

The 8080 program instructions may be one, two or three bytes in length. Multiple byte instructions must be stored in successive memory locations; the address of the first byte is always used as the address of the instructions. The exact instruction format will depend on the particular operation to be executed.



Addressing Modes:

Often the data that is to be operated on is stored in memory. When multi-byte numeric data is used, the data, like instructions, is stored in successive memory locations, with the least significant byte first, followed by increasingly significant bytes. The 8080 has four different modes for addressing data stored in memory or in registers:

- Direct – Bytes 2 and 3 of the instruction contain the exact memory address of the data item (the low-order bits of the address are in byte 2, the high-order bits in byte 3).
- Register – The instruction specifies the register or register-pair in which the data is located.
- Register Indirect – The instruction specifies a register-pair which contains the memory address where the data is located (the high-order bits of the address are in the first register of the pair, the low-order bits in the second).
- Immediate – The instruction contains the data itself. This is either an 8-bit quantity or a 16-bit quantity (least significant byte first, most significant byte second).

Unless directed by an interrupt or branch instruction, the execution of instructions proceeds through consecutively increasing memory locations. A branch instruction can specify the address of the next instruction to be executed in one of two ways:

- Direct – The branch instruction contains the address of the next instruction to be executed. (Except for the 'RST' instruction, byte 2 contains the low-order address and byte 3 the high-order address.)
- Register indirect – The branch instruction indicates a register-pair which contains the address of the next instruction to be executed. (The high-order bits of the address are in the first register of the pair, the low-order bits in the second.)

The RST instruction is a special one-byte call instruction (usually used during interrupt sequences). RST includes a three-bit field; program control is transferred to the instruction whose address is eight times the contents of this three-bit field.

Condition Flags:

There are five condition flags associated with the execution of instructions on the 8080. They are Zero, Sign, Parity, Carry, and Auxiliary Carry, and are each represented by a 1-bit register in the CPU. A flag is "set" by forcing the bit to 1; "reset" by forcing the bit to 0.

Unless indicated otherwise, when an instruction affects a flag, it affects it in the following manner:

Zero: If the result of an instruction has the value 0, this flag is set; otherwise it is reset.

Sign: If the most significant bit of the result of the operation has the value 1, this flag is set; otherwise it is reset.

Parity: If the modulo 2 sum of the bits of the result of the operation is 0, (i.e., if the result has even parity), this flag is set; otherwise it is reset (i.e., if the result has odd parity).

Carry: If the instruction resulted in a carry (from addition), or a borrow (from subtraction or a comparison) out of the high-order bit, this flag is set; otherwise it is reset.

Auxiliary Carry: If the instruction caused a carry out of bit 3 and into bit 4 of the resulting value, the auxiliary carry is set; otherwise it is reset. This flag is affected by single precision additions, subtractions, increments, decrements, comparisons, and logical operations, but is principally used with additions and increments preceding a DAA (Decimal Adjust Accumulator) instruction.

Symbols and Abbreviations:

The following symbols and abbreviations are used in the subsequent description of the 8080 instructions:

SYMBOLS	MEANING
accumulator	Register A
addr	16-bit address quantity
data	8-bit data quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L
DDD,SSS	The bit pattern designating one of the registers A,B,C,D,E,H,L (DDD=destination, SSS=source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

rp	One of the register pairs: B represents the B,C pair with B as the high-order register and C as the low-order register; D represents the D,E pair with D as the high-order register and E as the low-order register; H represents the H,L pair with H as the high-order register and L as the low-order register; SP represents the 16-bit stack pointer register.
RP	The bit pattern designating one of the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

rh	The first (high-order) register of a designated register pair.
rl	The second (low-order) register of a designated register pair.
PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).
Z,S,P,CY,AC	The condition flags: Zero, Sign, Parity, Carry, and Auxiliary Carry, respectively.
( )	The contents of the memory location or registers enclosed in the parentheses.
←	"Is transferred to"
$\wedge$	Logical AND
$\forall$	Exclusive OR
$\vee$	Inclusive OR
+	Addition
−	Two's complement subtraction
*	Multiplication
↔	"Is exchanged with"
—	The one's complement (e.g., $(\overline{A})$)
n	The restart number 0 through 7
NNN	The binary representation 000 through 111 for restart number 0 through 7 respectively.

Description Format:

The following pages provide a detailed description of the instruction set of the 8080. Each instruction is described in the following manner:

1. The MAC 80 assembler format, consisting of the instruction mnemonic and operand fields, is printed in **BOLDFACE** on the left side of the first line.
2. The name of the instruction is enclosed in parenthesis on the right side of the first line.
3. The next line(s) contain a symbolic description of the operation of the instruction.
4. This is followed by a narative description of the operation of the instruction.
5. The following line(s) contain the binary fields and patterns that comprise the machine instruction.

6. The last four lines contain incidental information about the execution of the instruction. The number of machine cycles and states required to execute the instruction are listed first. If the instruction has two possible execution times, as in a Conditional Jump, both times will be listed, separated by a slash. Next, any significant data addressing modes (see Page 4-2) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

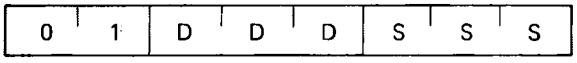
Data Transfer Group:

This group of instructions transfers data to and from registers and memory. **Condition flags are not affected** by any instruction in this group.

MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

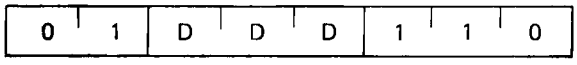


Cycles: 1
States: 5
Addressing: register
Flags: none

MOV r, M (Move from memory)

(r) ← ((H) (L))

The content of the memory location, whose address is in registers H and L, is moved to register r.

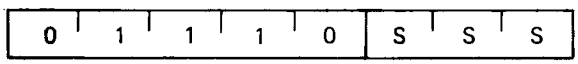


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MOV M, r (Move to memory)

((H) (L)) ← (r)

The content of register r is moved to the memory location whose address is in registers H and L.

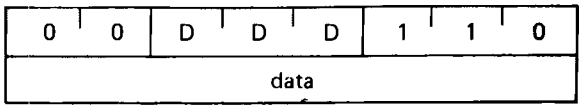


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move Immediate)

(r) ← (byte 2)

The content of byte 2 of the instruction is moved to register r.

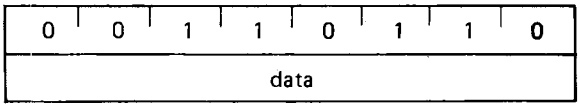


Cycles: 2
States: 7
Addressing: immediate
Flags: none

MVI M, data (Move to memory immediate)

((H) (L)) ← (byte 2)

The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.



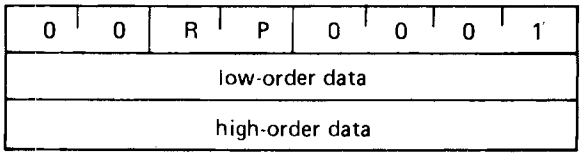
Cycles: 3
States: 10
Addressing: immed./reg. indirect
Flags: none

LXI rp, data 16 (Load register pair immediate)

(rh) ← (byte 3),

(rl) ← (byte 2)

Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.

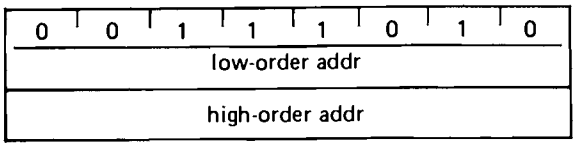


Cycles: 3
States: 10
Addressing: immediate
Flags: none

LDA addr (Load Accumulator direct)

(A) ← ((byte 3)(byte 2))

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

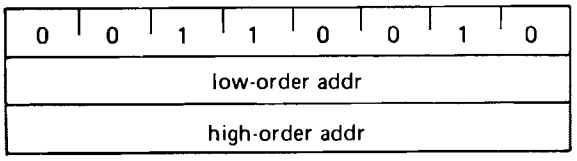


Cycles: 4
States: 13
Addressing: direct
Flags: none

STA addr (Store Accumulator direct)

((byte 3)(byte 2)) ← (A)

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.



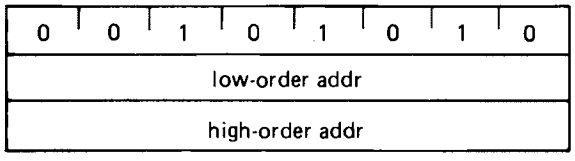
Cycles: 4
States: 13
Addressing: direct
Flags: none

LHLD addr (Load H and L direct)

(L) ← ((byte 3)(byte 2))

(H) ← ((byte 3)(byte 2) + 1)

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register L. The content of the memory location at the succeeding address is moved to register H.



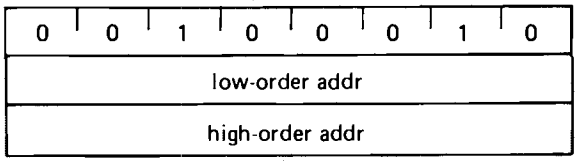
Cycles: 5
States: 16
Addressing: direct
Flags: none

SHLD addr (Store H and L direct)

((byte 3)(byte 2)) ← (L)

((byte 3)(byte 2) + 1) ← (H)

The content of register L is moved to the memory location whose address is specified in byte 2 and byte 3. The content of register H is moved to the succeeding memory location.

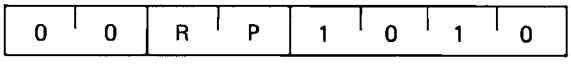


Cycles: 5
States: 16
Addressing: direct
Flags: none

LDAX rp (Load accumulator indirect)

(A) ← ((rp))

The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

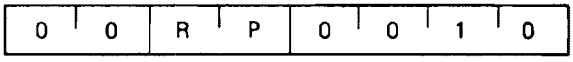


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

STAX rp (Store accumulator indirect)

((rp)) ← (A)

The content of register A is moved to the memory location whose address is in the register pair rp. Note: only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.



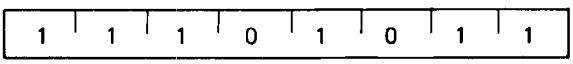
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

XCHG (Exchange H and L with D and E)

(H) ←→ (D)

(L) ←→ (E)

The contents of registers H and L are exchanged with the contents of registers D and E.



Cycles: 1
States: 4
Addressing: register
Flags: none

Arithmetic Group:

This group of instructions performs arithmetic operations on data in registers and memory.

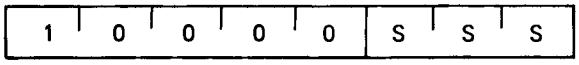
Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

All subtraction operations are performed via two's complement arithmetic and set the carry flag to one to indicate a borrow and clear it to indicate no borrow.

ADD r (Add Register)

(A) ← (A) + (r)

The content of register r is added to the content of the accumulator. The result is placed in the accumulator.

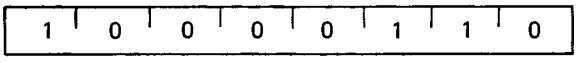


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADD M (Add memory)

(A) ← (A) + ((H) (L))

The content of the memory location whose address is contained in the H and L registers is added to the content of the accumulator. The result is placed in the accumulator.

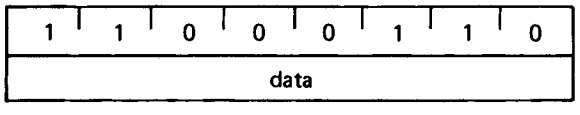


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ADI data (Add immediate)

(A) ← (A) + (byte 2)

The content of the second byte of the instruction is added to the content of the accumulator. The result is placed in the accumulator.

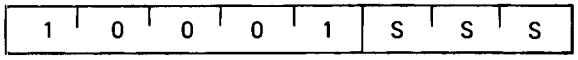


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ADC r (Add Register with carry)

(A) ← (A) + (r) + (CY)

The content of register r and the content of the carry bit are added to the content of the accumulator. The result is placed in the accumulator.

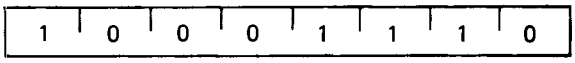


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADC M (Add memory with carry)

(A) ← (A) + ((H) (L)) + (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are added to the accumulator. The result is placed in the accumulator.

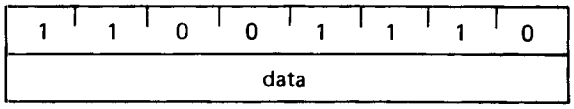


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ACI data (Add immediate with carry)

(A) ← (A) + (byte 2) + (CY)

The content of the second byte of the instruction and the content of the CY flag are added to the contents of the accumulator. The result is placed in the accumulator.

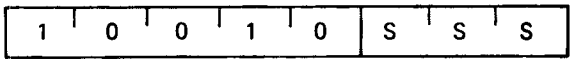


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SUB r (Subtract Register)

(A) ← (A) − (r)

The content of register r is subtracted from the content of the accumulator. The result is placed in the accumulator.

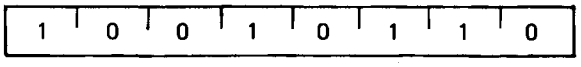


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SUB M (Subtract memory)

(A) ← (A) – ((H) (L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.

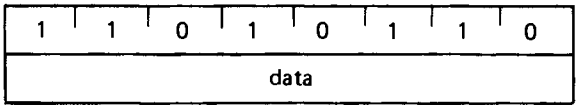


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SUI data (Subtract immediate)

(A) ← (A) – (byte 2)

The content of the second byte of the instruction is subtracted from the content of the accumulator. The result is placed in the accumulator.

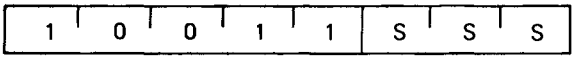


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with borrow)

(A) ← (A) – (r) – (CY)

The content of register r and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

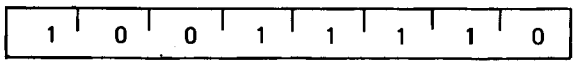


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

SBB M (Subtract memory with borrow)

(A) ← (A) – ((H) (L)) – (CY)

The content of the memory location whose address is contained in the H and L registers and the content of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

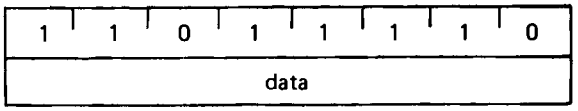


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

SBI data (Subtract immediate with borrow)

(A) ← (A) – (byte 2) – (CY)

The contents of the second byte of the instruction and the contents of the CY flag are both subtracted from the accumulator. The result is placed in the accumulator.

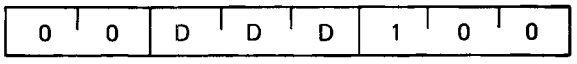


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

INR r (Increment Register)

(r) ← (r) + 1

The content of register r is incremented by one. Note: All condition flags **except CY** are affected.

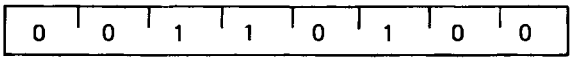


Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

INR M (Increment memory)

((H) (L)) ← ((H) (L)) + 1

The content of the memory location whose address is contained in the H and L registers is incremented by one. Note: All condition flags **except CY** are affected.

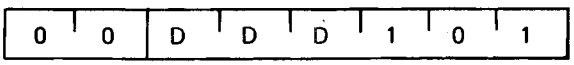


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

DCR r (Decrement Register)

(r) ← (r) – 1

The content of register r is decremented by one. Note: All condition flags **except CY** are affected.

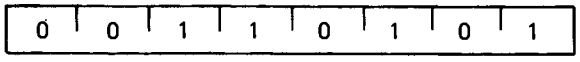


Cycles: 1
States: 5
Addressing: register
Flags: Z,S,P,AC

DCR M (Decrement memory)

((H) (L)) ← ((H) (L)) – 1

The content of the memory location whose address is contained in the H and L registers is decremented by one. Note: All condition flags **except CY** are affected.



Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

INX rp (Increment register pair)

(rh) (rl) ← (rh) (rl) + 1

The content of the register pair rp is incremented by one. Note: **No condition flags are affected.**

Cycles: 1
States: 5
Addressing: register
Flags: none

DCX rp (Decrement register pair)

(rh) (rl) ← (rh) (rl) – 1

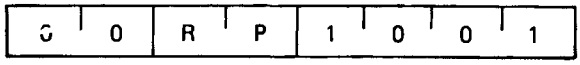
The content of the register pair rp is decremented by one. Note: **No condition flags are affected.**

Cycles: 1
States: 5
Addressing: register
Flags: none

DAD rp (Add register pair to H and L)

(H) (L) ← (H) (L) + (rh) (rl)

The content of the register pair rp is added to the content of the register pair H and L. The result is placed in the register pair H and L. Note: **Only the CY flag is affected.** It is set if there is a carry out of the double precision add; otherwise it is reset.



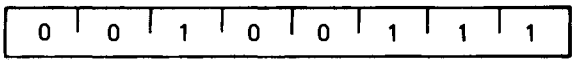
Cycles: 3
States: 10
Addressing: register
Flags: CY

DAA (Decimal Adjust Accumulator)

The eight-bit number in the accumulator is adjusted to form two four-bit Binary-Coded-Decimal digits by the following process:

1. If the value of the least significant 4 bits of the accumulator is greater than 9 or if the AC flag is set, 6 is added to the accumulator.
2. If the value of the most significant 4 bits of the accumulator is now greater than 9, or if the CY flag is set, 6 is added to the most significant 4 bits of the accumulator.

NOTE: All flags are affected.



Cycles: 1
States: 4
Flags: Z,S,P,CY,AC

Logical Group:

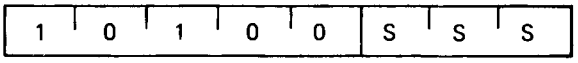
This group of instructions performs logical (Boolean) operations on data in registers and memory and on condition flags.

Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Auxiliary Carry, and Carry flags according to the standard rules.

ANA r (AND Register)

(A) ← (A) ∧ (r)

The content of register r is logically anded with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared.**

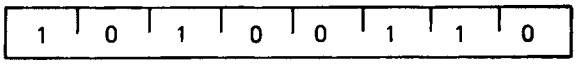


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ANA M (AND memory)

(A) ← (A) ∧ ((H) (L))

The contents of the memory location whose address is contained in the H and L registers is logically anded with the content of the accumulator. The result is placed in the accumulator. **The CY flag is cleared.**

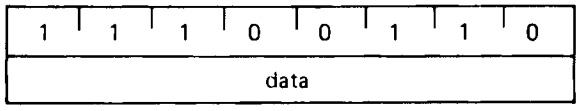


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ANI data (AND immediate)

(A) ← (A) ∧ (byte 2)

The content of the second byte of the instruction is logically anded with the contents of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

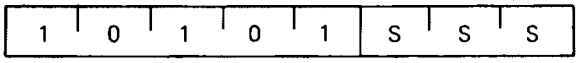


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

XRA r (Exclusive OR Register)

(A) ← (A) ∀ (r)

The content of register r is exclusive-or'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

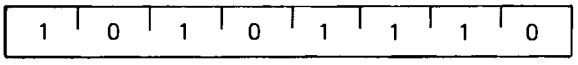


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

XRA M (Exclusive OR Memory)

(A) ← (A) ∀ ((H) (L))

The content of the memory location whose address is contained in the H and L registers is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

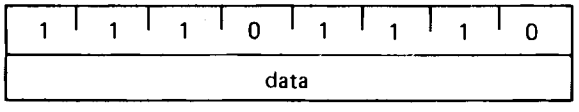


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XRI data (Exclusive OR immediate)

(A) ← (A) ∀ (byte 2)

The content of the second byte of the instruction is exclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

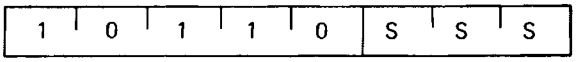


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

ORA r (OR Register)

(A) ← (A) V (r)

The content of register r is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

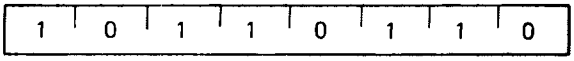


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ORA M (OR memory)

(A) ← (A) V ((H) (L))

The content of the memory location whose address is contained in the H and L registers is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

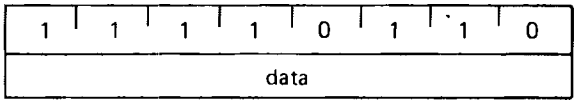


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORI data (OR Immediate)

(A) ← (A) V (byte 2)

The content of the second byte of the instruction is inclusive-OR'd with the content of the accumulator. The result is placed in the accumulator. **The CY and AC flags are cleared.**

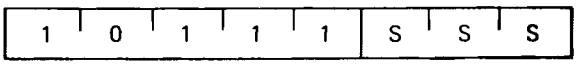


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

CMP r (Compare Register)

(A) − (r)

The content of register r is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. **The Z flag is set to 1 if (A) = (r). The CY flag is set to 1 if (A) < (r).**

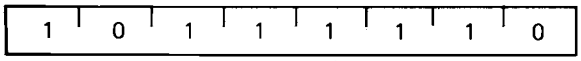


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

CMP M (Compare memory)

(A) – ((H) (L))

The content of the memory location whose address is contained in the H and L registers is subtracted from the accumulator. The accumulator remains unchanged. The condition flags are set as a result of the subtraction. The Z flag is set to 1 if (A) = ((H) (L)). The CY flag is set to 1 if (A) < ((H) (L)).

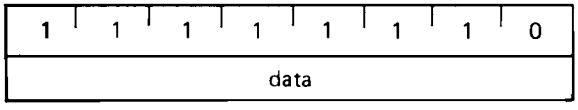


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

CPI data (Compare immediate)

(A) – (byte 2)

The content of the second byte of the instruction is subtracted from the accumulator. The condition flags are set by the result of the subtraction. The Z flag is set to 1 if (A) = (byte 2). The CY flag is set to 1 if (A) < (byte 2).



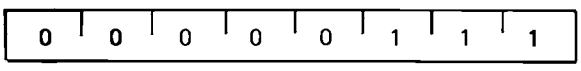
Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

RLC (Rotate left)

$(A_{n+1}) \leftarrow (A_n)$; $(A_0) \leftarrow (A_7)$

$(CY) \leftarrow (A_7)$

The content of the accumulator is rotated left one position. The low order bit and the CY flag are both set to the value shifted out of the high order bit position. **Only the CY flag is affected.**



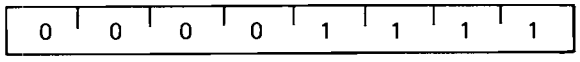
Cycles: 1
States: 4
Flags: CY

RRC (Rotate right)

$(A_n) \leftarrow (A_{n+1})$; $(A_7) \leftarrow (A_0)$

$(CY) \leftarrow (A_0)$

The content of the accumulator is rotated right one position. The high order bit and the CY flag are both set to the value shifted out of the low order bit position. **Only the CY flag is affected.**



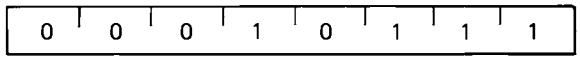
Cycles: 1
States: 4
Flags: CY

RAL (Rotate left through carry)

$(A_{n+1}) \leftarrow (A_n)$; $(CY) \leftarrow (A_7)$

$(A_0) \leftarrow (CY)$

The content of the accumulator is rotated left one position through the CY flag. The low order bit is set equal to the CY flag and the CY flag is set to the value shifted out of the high order bit. **Only the CY flag is affected.**



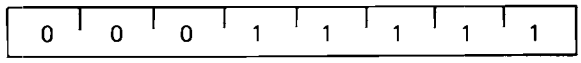
Cycles: 1
States: 4
Flags: CY

RAR (Rotate right through carry)

$(A_n) \leftarrow (A_{n+1})$; $(CY) \leftarrow (A_0)$

$(A_7) \leftarrow (CY)$

The content of the accumulator is rotated right one position through the CY flag. The high order bit is set to the CY flag and the CY flag is set to the value shifted out of the low order bit. **Only the CY flag is affected.**

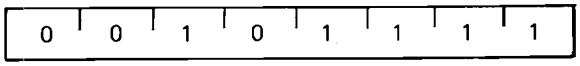


Cycles: 1
States: 4
Flags: CY

CMA (Complement accumulator)

$(A) \leftarrow (\overline{A})$

The contents of the accumulator are complemented (zero bits become 1, one bits become 0). **No flags are affected.**



Cycles: 1
States: 4
Flags: none

CMC (Complement carry)

(CY) ← $(\overline{CY})$

The CY flag is complemented. **No other flags are affected.**

0	0	1	1	1	1	1	1

Cycles: 1
States: 4
Flags: CY

STC (Set carry)

(CY) ← 1

The CY flag is set to 1. **No other flags are affected.**

0	0	1	1	0	1	1	1

Cycles: 1
States: 4
Flags: CY

Branch Group:

This group of instructions alter normal sequential program flow.

Condition flags are not affected by any instruction in this group.

The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed. The conditions that may be specified are as follows:

CONDITION		CCC
NZ	– not zero (Z = 0)	000
Z	– zero (Z = 1)	001
NC	– no carry (CY = 0)	010
C	– carry (CY = 1)	011
PO	– parity odd (P = 0)	100
PE	– parity even (P = 1)	101
P	– plus (S = 0)	110
M	– minus (S = 1)	111

JMP addr (Jump)

(PC) ← (byte 3) (byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

1	1	0	0	0	0	1	1
low-order addr							
high-order addr							

Cycles: 3
States: 10
Addressing: immediate
Flags: none

Jcondition addr (Conditional jump)

If (CCC),

(PC) ← (byte 3) (byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.

1	1	C	C	C	0	1	0
low-order addr							
high-order addr							

Cycles: 3
States: 10
Addressing: immediate
Flags: none

CALL addr (Call)

((SP) – 1) ← (PCH)

((SP) – 2) ← (PCL)

(SP) ← (SP) – 2

(PC) ← (byte 3) (byte 2)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.

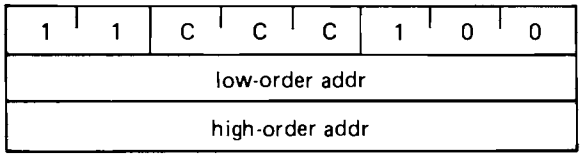
1	1	0	0	1	1	0	1
low-order addr							
high-order addr							

Cycles: 5
States: 17
Addressing: immediate/reg. indirect
Flags: none

Ccondition addr (Condition call)

If (CCC),
(SP) – 1) ← (PCH)
((SP) – 2) ← (PCL)
(SP) ← (SP) – 2
(PC) ← (byte 3) (byte 2)

If the specified condition is true, the actions specified in the CALL instruction (see above) are performed; otherwise, control continues sequentially.

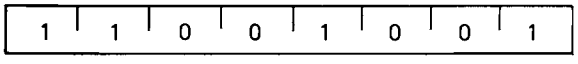


Cycles: 3/5
States: 11/17
Addressing: immediate/reg. indirect
Flags: none

RET (Return)

(PCL) ← ((SP));
(PCH) ← ((SP) + 1);
(SP) ← (SP) + 2;

The content of the memory location whose address is specified in register SP is moved to the low-order eight bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order eight bits of register PC. The content of register SP is incremented by 2.

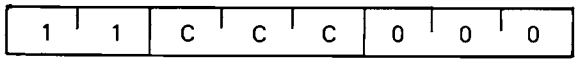


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

Rcondition (Conditional return)

If (CCC),
(PCL) ← ((SP))
(PCH) ← ((SP) + 1)
(SP) ← (SP) + 2

If the specified condition is true, the actions specified in the RET instruction (see above) are performed; otherwise, control continues sequentially.

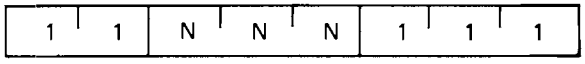


Cycles: 1/3
States: 5/11
Addressing: reg. indirect
Flags: none

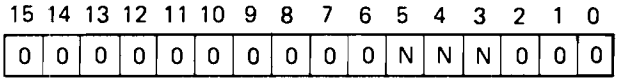
RST n (Restart)

((SP) – 1) ← (PCH)
((SP) – 2) ← (PCL)
(SP) ← (SP) – 2
(PC) ← 8 * (NNN)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two. Control is transferred to the instruction whose address is eight times the content of NNN.



Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

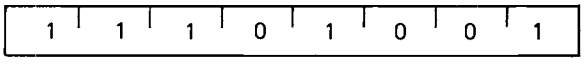


Program Counter After Restart

PCHL (Jump H and L indirect – move H and L to PC

(PCH) ← (H)
(PCL) ← (L)

The content of register H is moved to the high-order eight bits of register PC. The content of register L is moved to the low-order eight bits of register PC.



Cycles: 1
States: 5
Addressing: register
Flags: none

Stack, I/O, and Machine Control Group:

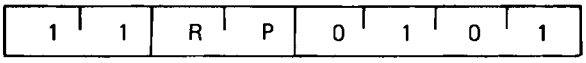
This group of instructions performs I/O, manipulates the Stack, and alters internal control flags.

Unless otherwise specified, **condition flags are not affected by any instructions in this group.**

PUSH rp (Push)

((SP) – 1) ← (rh)
((SP) – 2) ← (rl)
(SP) ← (SP) – 2

The content of the high-order register of register pair rp is moved to the memory location whose address is one less than the content of register SP. The content of the low-order register of register pair rp is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by 2. **Note: Register pair rp = SP may not be specified.**



Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

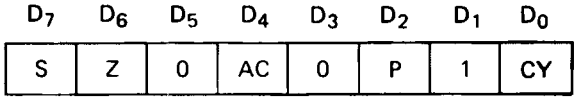
PUSH PSW (Push processor status word)

((SP) – 1) ← (A)
$((SP) - 2)_0$ ← (CY), $((SP) - 2)_1$ ← 1
$((SP) - 2)_2$ ← (P), $((SP) - 2)_3$ ← 0
$((SP) - 2)_4$ ← (AC), $((SP) - 2)_5$ ← 0
$((SP) - 2)_6$ ← (Z), $((SP) - 2)_7$ ← (S)
(SP) ← (SP) – 2

The content of register A is moved to the memory location whose address is one less than register SP. The contents of the condition flags are assembled into a processor status word and the word is moved to the memory location whose address is two less than the content of register SP. The content of register SP is decremented by two.

Cycles: 3
States: 11
Addressing: reg. indirect
Flags: none

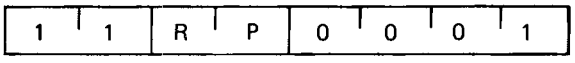
FLAG WORD



POP rp (Pop)

(rl) ← ((SP))
(rh) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location, whose address is specified by the content of register SP, is moved to the low-order register of register pair rp. The content of the memory location, whose address is one more than the content of register SP, is moved to the high-order register of register pair rp. The content of register SP is incremented by 2. **Note: Register pair rp = SP may not be specified.**

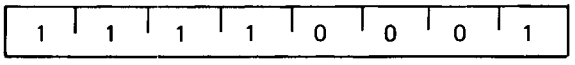


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: none

POP PSW (Pop processor status word)

(CY) ← $((SP))_0$
(P) ← $((SP))_2$
(AC) ← $((SP))_4$
(Z) ← $((SP))_6$
(S) ← $((SP))_7$
(A) ← ((SP) + 1)
(SP) ← (SP) + 2

The content of the memory location whose address is specified by the content of register SP is used to restore the condition flags. The content of the memory location whose address is one more than the content of register SP is moved to register A. The content of register SP is incremented by 2.



Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

XTHL (Exchange stack top with H and L)

(L) ↔ ((SP))
(H) ↔ ((SP) + 1)

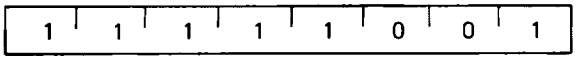
The content of the L register is exchanged with the content of the memory location whose address is specified by the content of register SP. The content of the H register is exchanged with the content of the memory location whose address is one more than the content of register SP.

Cycles: 5
States: 18
Addressing: reg. indirect
Flags: none

SPHL (Move HL to SP)

(SP) ← (H) (L)

The contents of registers H and L (16 bits) are moved to register SP.

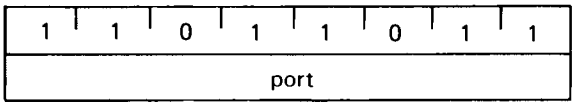


Cycles: 1
States: 5
Addressing: register
Flags: none

IN port (Input)

(A) ← (data)

The data placed on the eight bit bi-directional data bus by the specified port is moved to register A.

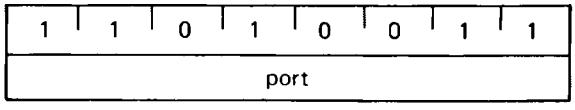


Cycles: 3
States: 10
Addressing: direct
Flags: none

OUT port (Output)

(data) ← (A)

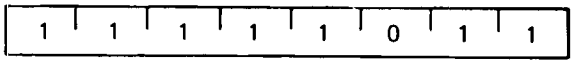
The content of register A is placed on the eight bit bi-directional data bus for transmission to the specified port.



Cycles: 3
States: 10
Addressing: direct
Flags: none

EI (Enable interrupts)

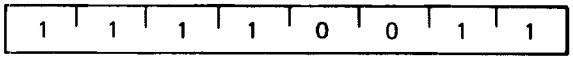
The interrupt system is enabled **following the execution of the next instruction.**



Cycles: 1
States: 4
Flags: none

DI (Disable interrupts)

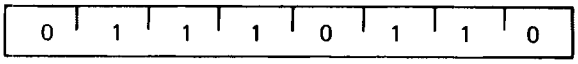
The interrupt system is disabled **immediately following the execution of the DI instruction.**



Cycles: 1
States: 4
Flags: none

HLT (Halt)

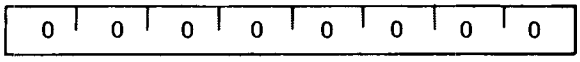
The processor is stopped. The registers and flags are unaffected.



Cycles: 1
States: 7
Flags: none

NOP (No op)

No operation is performed. The registers and flags are unaffected.



Cycles: 1
States: 4
Flags: none

8080 INSTRUCTION SET

Summary of Processor Instructions

Mnemonic	Description	Instruction Code[1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
MOVE, LOAD, AND STORE										
MOVr1,r2	Move register to register	0	1	D	D	D	S	S	S	5
MOV M,r	Move register to memory	0	1	1	1	0	S	S	S	7
MOV r,M	Move memory to register	0	1	D	D	D	1	1	0	7
MVI r	Move immediate register	0	0	D	D	D	1	1	0	7
MVI M	Move immediate memory	0	0	1	1	0	1	1	0	10
LXI B	Load immediate register Pair B & C	0	0	0	0	0	0	0	1	10
LXI D	Load immediate register Pair D & E	0	0	0	1	0	0	0	1	10
LXI H	Load immediate register Pair H & L	0	0	1	0	0	0	0	1	10
STAX B	Store A indirect	0	0	0	0	0	0	1	0	7
STAX D	Store A indirect	0	0	0	1	0	0	1	0	7
LDAX B	Load A indirect	0	0	0	0	1	0	1	0	7
LDAX D	Load A indirect	0	0	0	1	1	0	1	0	7
STA	Store A direct	0	0	1	1	0	0	1	0	13
LDA	Load A direct	0	0	1	1	1	0	1	0	13
SHLD	Store H & L direct	0	0	1	0	0	0	1	0	16
LHLD	Load H & L direct	0	0	1	0	1	0	1	0	16
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
STACK OPS										
PUSH B	Push register Pair B & C on stack	1	1	0	0	0	1	0	1	11
PUSH D	Push register Pair D & E on stack	1	1	0	1	0	1	0	1	11
PUSH H	Push register Pair H & L on stack	1	1	1	0	0	1	0	1	11
PUSH PSW	Push A and Flags on stack	1	1	1	1	0	1	0	1	11
POP B	Pop register Pair B & C off stack	1	1	0	0	0	0	0	1	10
POP D	Pop register Pair D & E off stack	1	1	0	1	0	0	0	1	10
POP H	Pop register Pair H & L off stack	1	1	1	0	0	0	0	1	10
POP PSW	Pop A and Flags off stack	1	1	1	1	0	0	0	1	10
XTHL	Exchange top of stack, H & L	1	1	1	0	0	0	1	1	18
SPHL	H & L to stack pointer	1	1	1	1	1	0	0	1	5
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
INX SP	Increment stack pointer	0	0	1	1	0	0	1	1	5
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
JUMP										
JMP	Jump unconditional	1	1	0	0	0	0	1	1	10
JC	Jump on carry	1	1	0	1	1	0	1	0	10
JNC	Jump on no carry	1	1	0	1	0	0	1	0	10
JZ	Jump on zero	1	1	0	0	1	0	1	0	10
JNZ	Jump on no zero	1	1	0	0	0	0	1	0	10
JP	Jump on positive	1	1	1	1	0	0	1	0	10
JM	Jump on minus	1	1	1	1	1	0	1	0	10
JPE	Jump on parity even	1	1	1	0	1	0	1	0	10
JPO	Jump on parity odd	1	1	1	0	0	0	1	0	10
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
CALL										
CALL	Call unconditional	1	1	0	0	1	1	0	1	17
CC	Call on carry	1	1	0	1	1	1	0	0	11/17
CNC	Call on no carry	1	1	0	1	0	1	0	0	11/17
CZ	Call on zero	1	1	0	0	1	1	0	0	11/17
CNZ	Call on no zero	1	1	0	0	0	1	0	0	11/17
CP	Call on positive	1	1	1	1	0	1	0	0	11/17
CM	Call on minus	1	1	1	1	1	1	0	0	11/17
CPE	Call on parity even	1	1	1	0	1	1	0	0	11/17
CPO	Call on parity odd	1	1	1	0	0	1	0	0	11/17
RETURN										
RET	Return	1	1	0	0	1	0	0	1	10
RC	Return on carry	1	1	0	1	1	0	0	0	5/11
RNC	Return on no carry	1	1	0	1	0	0	0	0	5/11
RZ	Return on zero	1	1	0	0	1	0	0	0	5/11
RNZ	Return on no zero	1	1	0	0	0	0	0	0	5/11
RP	Return on positive	1	1	1	1	0	0	0	0	5/11
RM	Return on minus	1	1	1	1	1	0	0	0	5/11
RPE	Return on parity even	1	1	1	0	1	0	0	0	5/11
RPO	Return on parity odd	1	1	1	0	0	0	0	0	5/11
RESTART										
RST	Restart	1	1	A	A	A	1	1	1	11
INCREMENT AND DECREMENT										
INR r	Increment register	0	0	D	D	D	1	0	0	5
DCR r	Decrement register	0	0	D	D	D	1	0	1	5
INR M	Increment memory	0	0	1	1	0	1	0	0	10
DCR M	Decrement memory	0	0	1	1	0	1	0	1	10
INX B	Increment B & C registers	0	0	0	0	0	0	1	1	5
INX D	Increment D & E registers	0	0	0	1	0	0	1	1	5
INX H	Increment H & L registers	0	0	1	0	0	0	1	1	5
DCX B	Decrement B & C	0	0	0	0	1	0	1	1	5
DCX D	Decrement D & E	0	0	0	1	1	0	1	1	5
DCX H	Decrement H & L	0	0	1	0	1	0	1	1	5
ADD										
ADD r	Add register to A	1	0	0	0	0	S	S	S	4
ADC r	Add register to A with carry	1	0	0	0	1	S	S	S	4
ADD M	Add memory to A	1	0	0	0	0	1	1	0	7
ADC M	Add memory to A with carry	1	0	0	0	1	1	1	0	7
ADI	Add immediate to A	1	1	0	0	0	1	1	0	7
ACI	Add immediate to A with carry	1	1	0	0	1	1	1	0	7
DAD B	Add B & C to H & L	0	0	0	0	1	0	0	1	10
DAD D	Add D & E to H & L	0	0	0	1	1	0	0	1	10
DAD H	Add H & L to H & L	0	0	1	0	1	0	0	1	10
DAD SP	Add stack pointer to H & L	0	0	1	1	1	0	0	1	10

NOTES: 1. DDD or SSS: B 000, C 001, D 010, E 011, H 100, L 101, Memory 110, A 111.
2. Two possible cycle times, (6/12) indicate instruction cycles dependent on condition flags.

8080 INSTRUCTION SET

Summary of Processor Instructions (Cont.)

Mnemonic	Description	Instruction Code[1] D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Clock[2] Cycles
SUBTRACT										
SUB r	Subtract register from A	1	0	0	1	0	S	S	S	4
SBB r	Subtract register from A with borrow	1	0	0	1	1	S	S	S	4
SUB M	Subtract memory from A	1	0	0	1	0	1	1	0	7
SBB M	Subtract memory from A with borrow	1	0	0	1	1	1	1	0	7
SUI	Subtract immediate from A	1	1	0	1	0	1	1	0	7
SBI	Subtract immediate from A with borrow	1	1	0	1	1	1	1	0	7
LOGICAL										
ANA r	And register with A	1	0	1	0	0	S	S	S	4
XRA r	Exclusive Or register with A	1	0	1	0	1	S	S	S	4
ORA r	Or register with A	1	0	1	1	0	S	S	S	4
CMP r	Compare register with A	1	0	1	1	1	S	S	S	4
ANA M	And memory with A	1	0	1	0	0	1	1	0	7
XRA M	Exclusive Or memory with A	1	0	1	0	1	1	1	0	7
ORA M	Or memory with A	1	0	1	1	0	1	1	0	7
CMP M	Compare memory with A	1	0	1	1	1	1	1	0	7
ANI	And immediate with A	1	1	1	0	0	1	1	0	7
XRI	Exclusive Or immediate with A	1	1	1	0	1	1	1	0	7
ORI	Or immediate with A	1	1	1	1	0	1	1	0	7
CPI	Compare immediate with A	1	1	1	1	1	1	1	0	7
ROTATE										
RLC	Rotate A left	0	0	0	0	0	1	1	1	4
RRC	Rotate A right	0	0	0	0	1	1	1	1	4
RAL	Rotate A left through carry	0	0	0	1	0	1	1	1	4
RAR	Rotate A right through carry	0	0	0	1	1	1	1	1	4
SPECIALS										
CMA	Complement A	0	0	1	0	1	1	1	1	4
STC	Set carry	0	0	1	1	0	1	1	1	4
CMC	Complement carry	0	0	1	1	1	1	1	1	4
DAA	Decimal adjust A	0	0	1	0	0	1	1	1	4
INPUT/OUTPUT										
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	10
CONTROL										
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupt	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
HLT	Halt	0	1	1	1	0	1	1	0	7

NOTES: 1. DDD or SSS: B=000. C=001. D=010. E=011. H=100. L=101. Memory=110. A=111.
2. Two possible cycle times. (6/12) indicate instruction cycles dependent on condition flags.

APPENDIX B
TELETYPEWRITER MODIFICATIONS

B-1. INTRODUCTION

This appendix provides information required to modify a Model ASR-33 Teletypewriter for use with certain Intel iSBC 80 computer systems.

B-2. INTERNAL MODIFICATIONS

WARNING

Hazardous voltages are exposed when the top cover of the teletypewriter is removed. To prevent accidental shock, disconnect the teleprinter power cord before proceeding beyond this point.

Remove the top cover and modify the teletypewriter as follows:

a. Remove blue lead from 750-ohm tap on current source resistor, reconnect this lead to 1450-ohm tap. (Refer to figures B-1 and B-2.)

b. On terminal block, change two wires as follows to create an internal full-duplex loop (refer to figures B-1 and B-3):
   1. Remove brown/yellow lead from terminal 3; reconnect this lead to terminal 5.
   2. Remove white/blue lead from terminal 4; reconnect this lead to terminal 5.

c. On terminal block, remove violet lead from terminal 8; reconnect this lead to terminal 9. This changes the receiver current level from 60 mA to 20 mA.

A relay circuit card must be fabricated and connected to the paper tape reader drive circuit. The relay circuit card to be fabricated requires a relay, a diode, a thyractor, a small 'vector' board for mounting the components, and suitable hardware for mounting the assembled relay card.

A circuit diagram of the relay circuit card is included in figure B-4; this diagram also includes the part numbers of the relay, diode, and thyractor. (Note that a 470-ohm resistor and a 0.1 F capacitor may be substituted for the thyractor.) After the relay circuit card has been assembled, mount it in position as shown in figure B-5. Secure the card to the base plate using two self-tapping screws. Connect the relay circuit to the distributor trip magnet and mode switch as follows:

a. Refer to figure B-4 and connect a wire (Wire 'A') from relay circuit card to terminal L2 on mode switch. (See figure B-6.)

b. Disconnect brown wire shown in figure B-7 from plastic connector. Connect this brown wire to terminal L2 on mode switch. (Brown wire will have to be extended.)

c. Refer to figure B-4 and connect a wire (Wire 'B') from relay circuit board to terminal L1 on mode switch.

B-3. EXTERNAL CONNECTIONS

Connect a two-wire receive loop, a two-wire send loop, and a two-wire tape reader control loop to the external device as shown in figure B-4. The external connector pin numbers shown in figure B-4 are for interface with an RS232C device.

B-4. iSBC 530 TTY ADAPTER

The iSBC 530 TTY adapter, which converts RS232C signal levels to an optically isolated 20 mA current loop interface, provides signal translation for transmitted data, received data, and a paper tape reader relay. The iSBC 530 TTY adapter interfaces an Intel iSBC 80 computer system to a teletypewriter as shown in figure A-8.

The iSBC 530 TTY adapter requires +12V at 98 mA and -12V at 98 mA. An auxiliary supply must be used if the iSBC 80 system does not supply this power. A schematic diagram of the iSBC 530 TTY adapter is supplied with the unit. The following auxiliary power connector (or equivalent) must be procured by the user:

Connector, Molex 09-50-7071
Pins, Molex 08-50-0106
Polarizing Key, Molex 15-04-0219

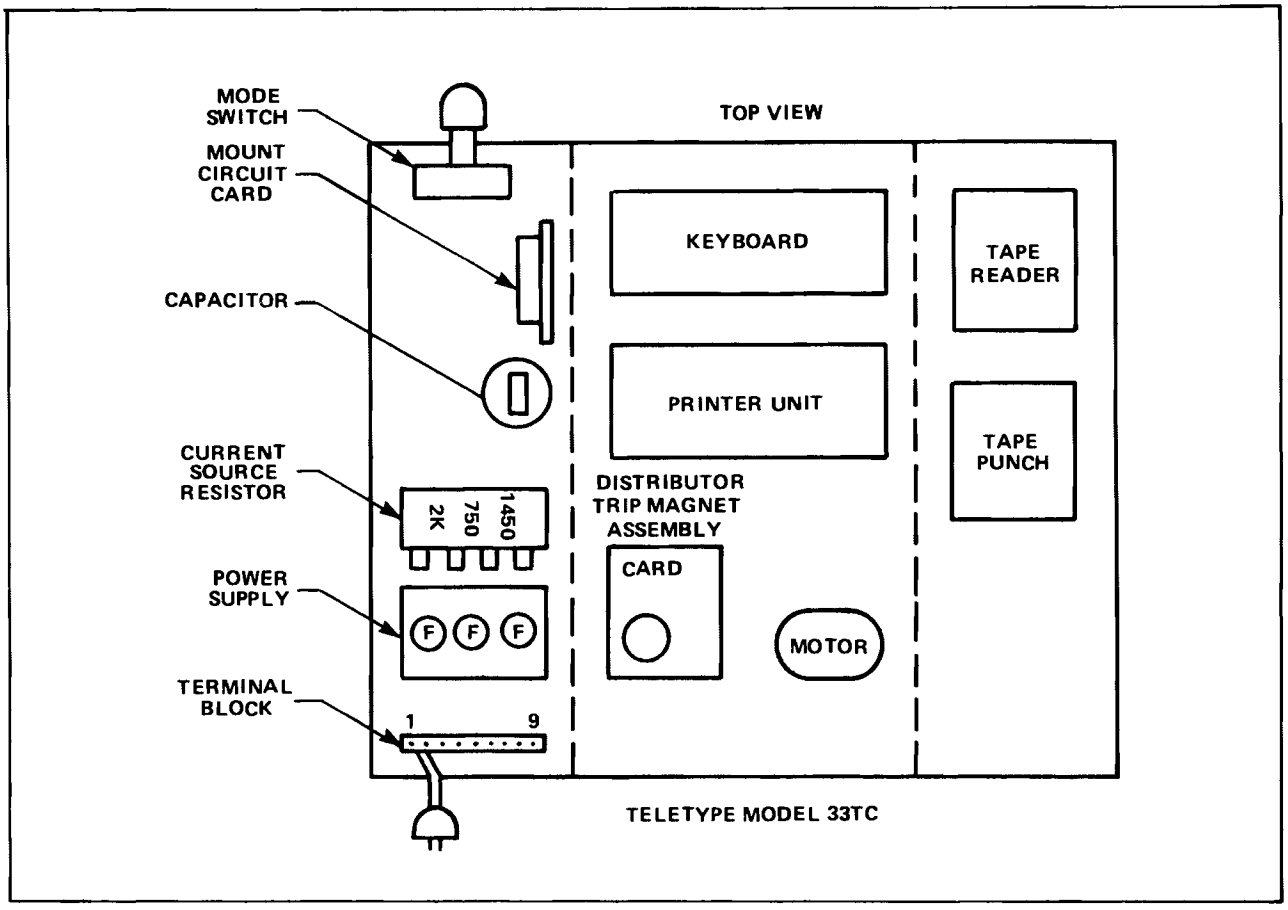


Figure B-1. Teletype Component Layout

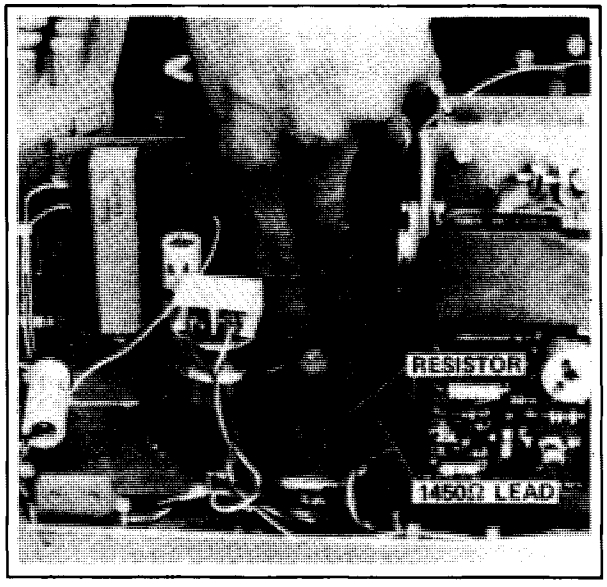


Figure B-2. Current Source Resistor

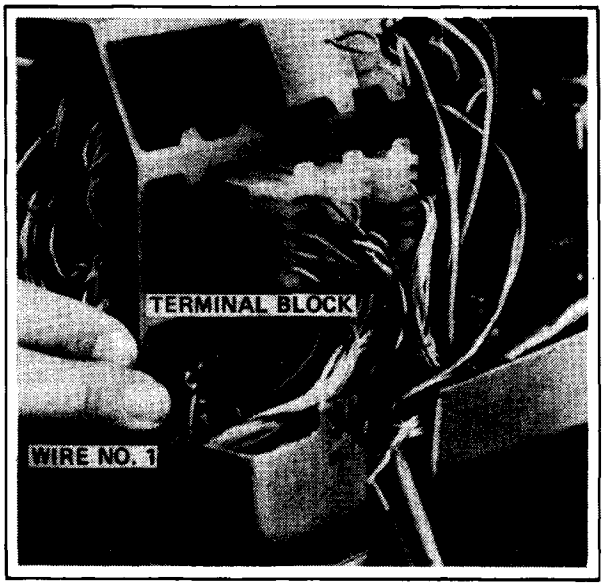


Figure B-3. Terminal Block

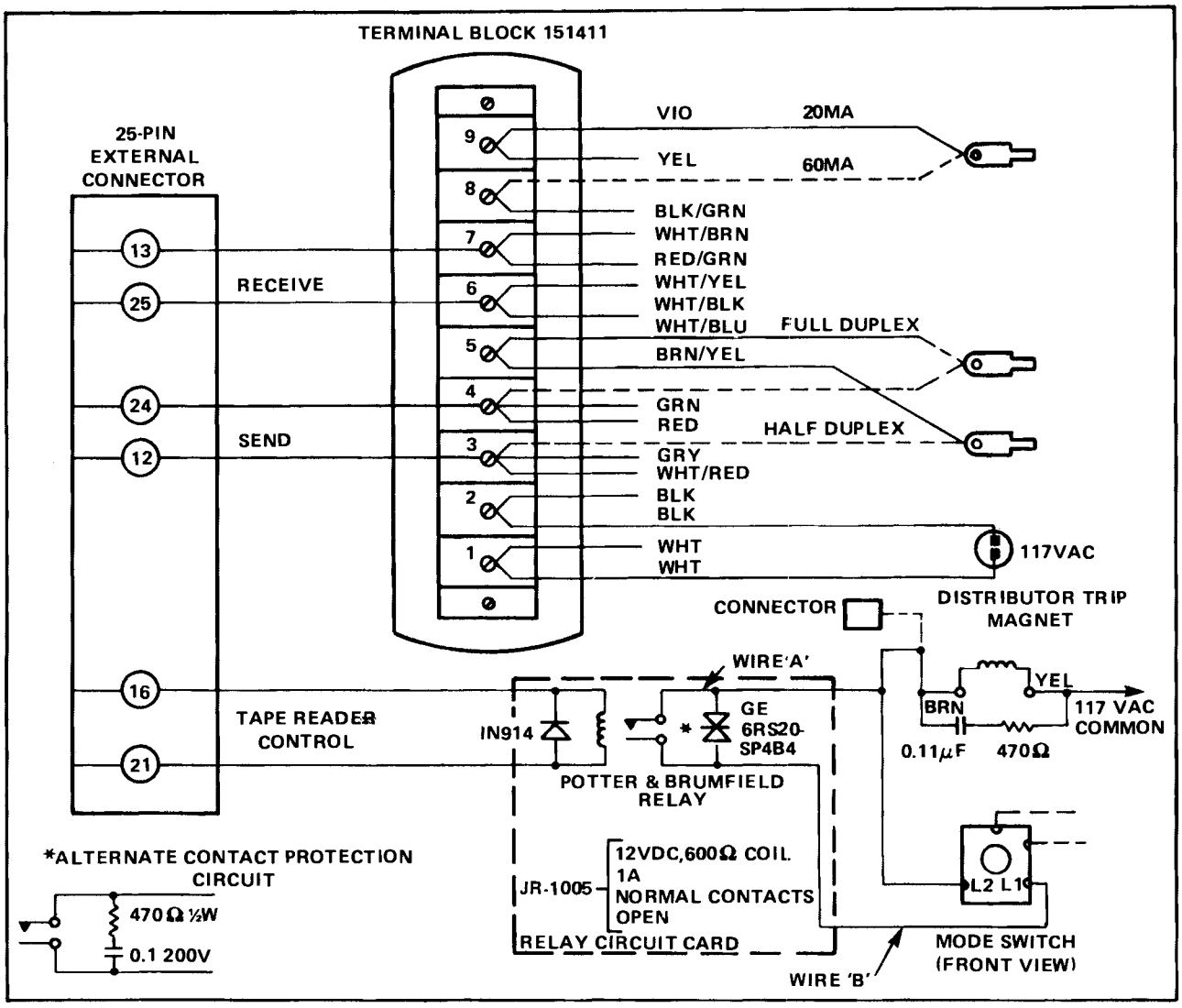


Figure B-4. Teletypewriter Modifications

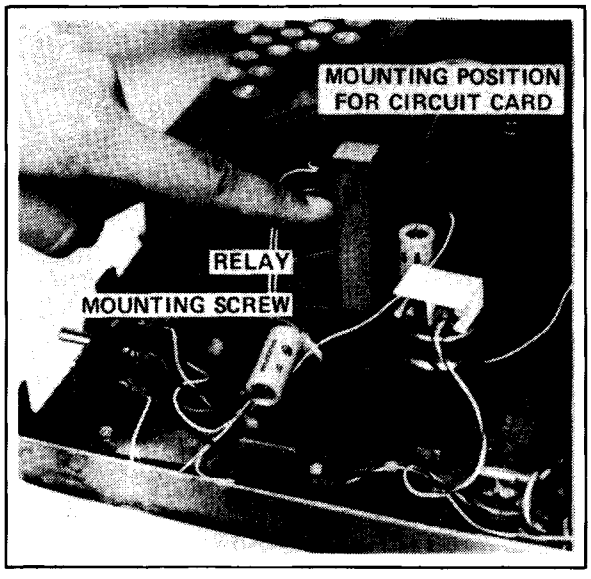


Figure B-5. Relay Circuit

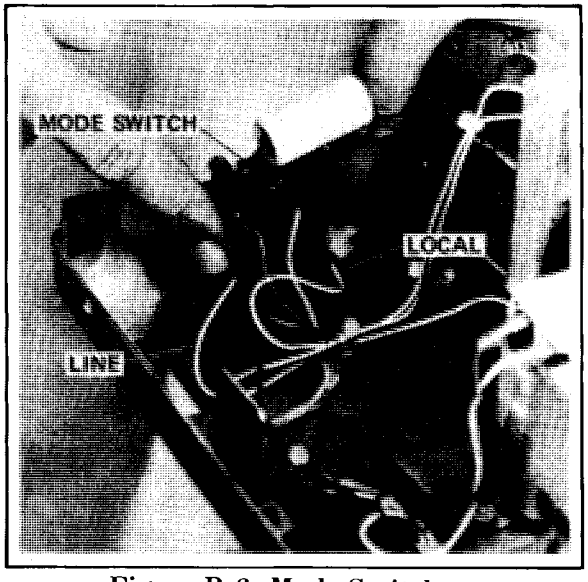


Figure B-6. Mode Switch

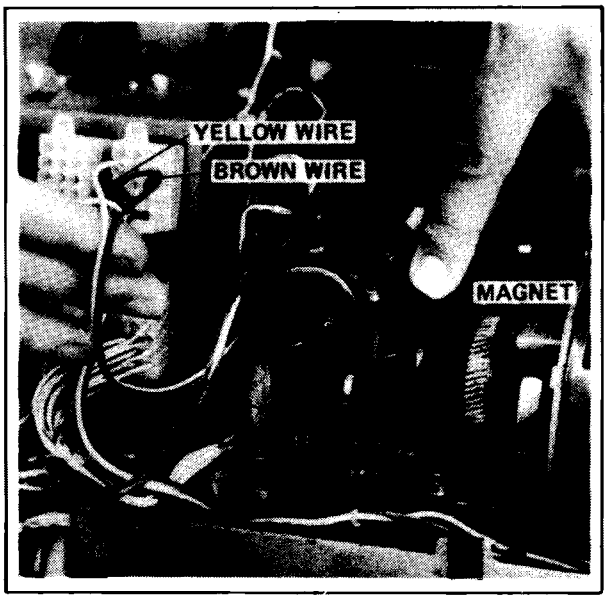


Figure A-7. Distributor Trip Magnet

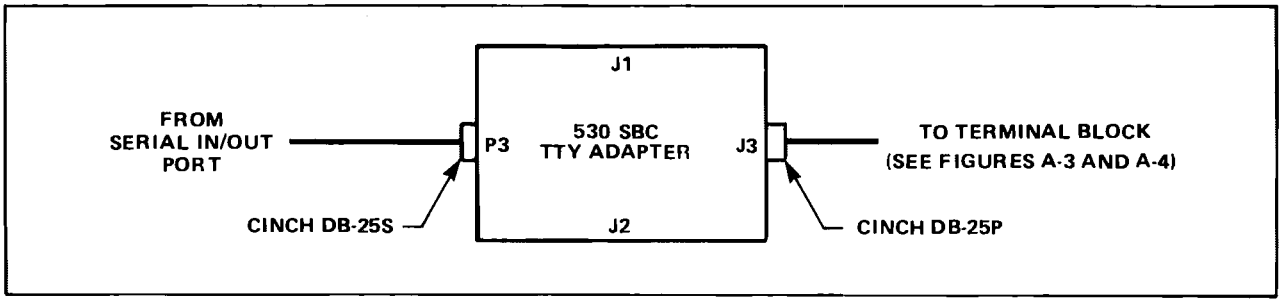


Figure B-8. TTY Adapter Cabling

APPENDIX C
ADDRESS DECODE PROM (U54)

C-1. INTRODUCTION

The address decode PROM (U54) is used for generating chip select signals for all on-board memory and I/O devices, and to identify off-board requests. This is accomplished by pattern recognition within the decode PROM and with other circuitry (shown on figure 5-3, sheet 4).

This appendix describes iSBC 80/10B board addressing and provides a memory map of each page in the decode PROM.

C-2. DESCRIPTION

The iSBC 80/10B board can address up to 4K of RAM on-board. Jumper 96-97 indicates if 1K or 4K of RAM exists. The base address of the RAM is shown below:

ROM TYPE	RAM AREA SIZE	
	1K	4K
2708 or 2716	3000	3000
2732	4000	4000

The PROM used for chip select decode is the Intel 3625-2 (1K x 4, bipolar) PROM. The 3625-2 utilizes ten inputs and produces four outputs. These are summarized as follows:

ISBC 80/10B SIGNAL	3625-2 INPUTS
ADRA-ADRF	A0-A5
ROMSEL0	A6
ROMSEL1	A7
RAMSIZ	A8
MMP/	A9
	3625-2 OUTPUTS
device select	D1-D3
ACK/	D4

The inputs MMP/ and RAMSIZ indicate the presence of a multimodule and the amount of on-board RAM. These signals divide the decode PROM into four pages; the page number is labeled A.

MMP/	RAMSIZ	PAGE (A)
0 (multimodule present)	0 (4K)	0
	1 (1K)	1
1 (multimodule not present)	0 (4K)	2
	1 (1K)	3

The inputs ROMSL1 and ROMSL0 determine the presence and type of EPROM being used.

ROMSL1	ROMSL0	EPROM TYPE
0	0	2708
0	1	2716
1	0	2732
1	1	none

The inputs ADRA-ADRF break the 64K memory addresses into 64 address blocks (1K addresses per block) or break the 256 I/O addresses into 64 address blocks (4 addresses per block). ADRF is also used to indicate whether the specified address is a memory or an I/O address. ADRF=0 for memory addresses, and ADRF=1 for I/O addresses. This restricts on-board memory addresses to the lower 32K (0000-7FFF) and on-board I/O addresses to the upper 128 (80-FF).

The inputs ROMSL1, ROMSL0, ADRF, and ADRE are grouped and labeled (B). The inputs ADRD, ADRC, ADRB, and ADRA are grouped and labeled (C). The combination (A), (B), (C) is the hexadecimal location of any 3625-2 entry. The output signals D1-D4 are decoded to become:

D4,D3,D2,D1	ADRF+0	ADRF+1
0	RAMCS0/	IOCS0/
1	RAMCS1/	IOCS1/
2	RAMCS2/	IOCS2/
3	RAMCS3/	MCS0/
4	ROMCS0/	not used
5	ROMCS1/	not used
6	ROMCS2/	not used
7	ROMCS3/	MCS1/
8-F	— OFF BOARD —	

The following table presents the I/O or memory address which corresponds to each entry position.

(B)		(C) ADRD, ADRC, ADRB, ADRA																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
2708	0	0000	0400	0800	0C00	1000	1400	1800	1C00	2000	2400	2800	2C00	3000	3400	3800	3C00	MEM
	1	4000	4400	4800	4C00	5000	5400	5800	5C00	6000	6400	6800	6C00	7000	7400	7800	7C00	MEM
	2	80	84	88	8C	90	94	98	9C	A0	A4	A8	AC	B0	B4	B8	BC	I/O
	3	C0	C4	C8	CC	D0	D4	D8	DC	E0	E4	E8	EC	F0	F4	F8	FC	I/O
2716	4	0000	0400	0800	0C00	1000	1400	1800	1C00	2000	2400	2800	2C00	3000	3400	3800	3C00	MEM
	5	4000	4400	4800	4C00	5000	5400	5800	5C00	6000	6400	6800	6C00	7000	7400	7800	7C00	MEM
	6	80	84	88	8C	90	94	98	9C	A0	A4	A8	AC	B0	B4	B8	BC	I/O
	7	C0	C4	C8	CC	D0	D4	D8	DC	E0	E4	E8	EC	F0	F4	F8	FC	I/O
2732	8	0000	0400	0800	0C00	1000	1400	1800	1C00	2000	2400	2800	2C00	3000	3400	3800	3C00	MEM
	9	4000	4400	4800	4C00	5000	5400	5800	5C00	6000	6400	6800	6C00	7000	7400	7800	7C00	MEM
	A	80	84	88	8C	90	94	98	9C	A0	A4	A8	AC	B0	B4	B8	BC	I/O
	B	C0	C4	C8	CC	D0	D4	D8	DC	E0	E4	E8	EC	F0	F4	F8	FC	I/O
NONE	C	0000	0400	0800	0C00	1000	1400	1800	1C00	2000	2400	2800	2C00	3000	3400	3800	3C00	MEM
	D	4000	4400	4800	4C00	5000	5400	5800	5C00	6000	6400	6800	6C00	7000	7400	7800	7C00	MEM
	E	80	84	88	8C	90	94	98	9C	A0	A4	A8	AC	B0	B4	B8	BC	I/O
	F	C0	C4	C8	CC	D0	D4	D8	DC	E0	E4	E8	EC	F0	F4	F8	FC	I/O

NOTE

All on-board memory addresses must be less than 8000H. All on-board I/O addresses must be greater than or equal to 80H.

APPLICATION EXAMPLE

For example, suppose one designs a Multimodule™ board. The designer would like to access this Multimodule board through I/O ports A0-A3. The designer would also like his Multimodule board to respond to these ports irrespective of the amount of on-board RAM or the type of on-board PROM. Further, if the customer Multimodule board is not present, the iSBC 80/10B board should consider the port addresses A0-A3 to be off-board. The customer Multimodule board is designed to respond to the multimodule bus signal MCS0.

From Table 1, one finds address block A0-A3 in column 8, rows 2, 6, A, and E. Therefore the value for (C) is 8, and the values for (B) are 2, 6, A, and E. From the page description chart one sees that the MCS0/ entry belongs in pages 0 and 1 in which the Multimodule board is present, and an off-board entry belongs in pages 2 and 3. The output chart indicates that a MCS0/ signal is generated from a 3625-2 EPROM output of 3H, and an off-board signal is generated from a 3625-2 output of 8H. Therefore, for (A) = 0 or 1; (B) = 2, 6, A, or E; and (C) = 8 the 3625-2 should contain a 3H. And for (A) = 2 or 3; (B) = 2, 6, A, or E; and (C) = 8 the 3625-2 should contain an 8H. The designer programs a new 3625-2 making addresses 028H, 068H, 0A8H, 0E8H, 128H, 168H, 1A8H, and 1E8H contain a 3H; and addresses 228H, 268H, 2A8H, 2E8H, 328H, 368H, 3A8H, and 3E8 contain an 8H.

Tables C-1 through C-4 show the contents of the factory programmed 3625-2.

Table C-1. PAGE 0 (A=0)

(B) ADRE-F, ROMSL0-1		(C) CPU ADRA-D															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2708	0	4	5	6	7	*C	*D	*E	*F	*8	*9	*A	*B	0	1	2	3
	1	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	2	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	3	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7
2716	4	4	4	5	5	6	6	7	7	*8	*9	*A	*B	0	1	2	3
	5	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	6	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	7	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7
2732	8	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
	9	0	1	2	3	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	B	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7
NONE	C	*C	*D	*E	*F	*C	*D	*E	*F	*8	*9	*A	*B	0	1	2	3
	D	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	E	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	F	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7

Table C-2. PAGE 1 (A=1)

(B) ADRE-F, ROMSL0-1		(C) CPU ADRA-D															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2708	0	4	5	6	7	*C	*D	*E	*F	*8	*9	*A	*B	*8	*9	*A	*B
	1	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	2	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	3	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7
2716	4	4	4	5	5	6	6	7	7	*8	*9	*A	*B	*8	*9	*A	3
	5	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	6	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	7	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7
2732	8	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
	9	*8	*9	*A	*3	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	B	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7
NONE	C	*C	*D	*E	*F	*C	*D	*E	*F	*8	*9	*A	*B	*8	*9	*A	3
	D	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	E	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	F	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	3	3	7	7

*These entries indicate off-board locations and may be any value between 8 and F (Hex).

Table C-3. PAGE 2 (A=2)

(B) ADRE-F, ROMSL0-1		(C) CPU ADRA-D															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2708	0	4	5	6	7	*C	*D	*E	*F	*8	*9	*A	*B	0	1	2	3
	1	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	2	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	3	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F
2716	4	4	4	5	5	6	6	7	7	*8	*9	*A	*B	0	1	2	3
	5	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	6	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	7	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F
2732	8	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
	9	0	1	2	3	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	B	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F
NONE	C	*C	*D	*E	*F	*C	*D	*E	*F	*8	*9	*A	*B	0	1	2	3
	D	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	E	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	F	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F

Table C-4. PAGE 3 (A=3)

(B) ADRE-F, ROMSL0-1		(C) CPU ADRA-D															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2708	0	4	5	6	7	*C	*D	*E	*F	*8	*9	*A	*B	*8	*9	*A	3
	1	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	2	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	3	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F
2716	4	4	4	5	5	6	6	7	7	*8	*9	*A	*B	*8	*9	*A	3
	5	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	6	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	7	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F
2732	8	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
	9	*8	*9	*A	*3	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	B	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F
NONE	C	*C	*D	*E	*F	*C	*D	*E	*F	*8	*9	*A	*B	*8	*9	*A	3
	D	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B
	E	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A	*B	*8	*9	*A
	F	*B	*8	*9	*A	*B	*8	*9	*A	*B	0	1	2	*B	*B	*F	*F

*These entries indicate off-board locations and may be any value between 8 and F (Hex).

APPENDIX D
iSBC 80/10A BOARD AND iSBC 80/10B BOARD FUNCTIONAL DIFFERENCES

The new iSBC 80/10B board has been designed to be completely downward compatible with the existing iSBC 80/10A board. There are, however, several minor revisions implemented on the iSBC 80/10B board which may affect your particular application. These revisions can be placed into the following categories:

1. Connector Pin Signal Assignments.
2. Jumper Post Numbering and Board Jumper Configurations.
3. Operational Differences.

Table D-1 provides a comparison of connector pin differences between the two boards, in the default configuration. Table D-2 gives a jumper post (stake pin) cross reference between the two boards. Default strapping of the two boards is functionally identical. Table D-3 lists all default and optional jumper connections used on the iSBC 80/10B board, and provides a brief description of the function. Table D-3 is grouped according to iSBC 80/10B board schematic diagram sheet number. Grid locations are provided for reference purposes.

There are three operation differences between the two boards. These differences are enhancements which should provide greater flexibility to your system. The following operational differences exist:

1. The iSBC 80/10B board command and data drivers will be turned off for *on-board* read and write operations. (The iSBC 80/10A board has drivers *on* for both on-board and off-board operations).
2. The iSBC 80/10B board will acknowledge an on-board write-to-ROM, but will not functionally support such an operation. (The iSBC 80/10A board does not acknowledge this operation).
3. On-board ROM/PROM space for the iSBC 80/10B board may extend to 3FFF depending on the type of devices installed. (The iSBC 80/10A board is limited to 1FFF on-board).

Table D-1. Connector Pin Assignment Differences

DIFFERENCES:

Pin Number	ISBC 80/10A	ISBC 80/10B Default	ISBC 80/10B Optional
P1-13	BCLK/	BCLK/ (Inverted)	—
P2-28	ϕ TTL	HALT	ϕ02 TTL
P2-30	RAM RDY IN	WAIT	RAM 3C00 EN
P2-32	NOT USED	(SYNC')ALE	RAM 3D00 EN
Deletions			
P2-48	INIT/	DELETED	DELETED
P2-55	B & C CLK SET/	DELETED	DELETED

Additions	80/10A	80/10B
P2-1	NOT USED	GND
P2-2	NOT USED	GND
P2-3	NOT USED	+5 VDC AUX
P2-4	NOT USED	+5 VDC AUX
P2-13	NOT USED	PFSR
P2-17	NOT USED	PFSM
P2-19	NOT USED	PFIN
P2-20	NOT USED	MPRO/
P2-21	NOT USED	GND
P2-22	NOT USED	GND
P2-34	NOT USED	RAM CS2 (RAM 3E00 EN)
P2-36	NOT USED	RAM CS3 (RAM 3F00 EN)

Table D-2. Jumper Post Cross Reference

ISBC 80/10A	ISBC 80/10B	ISBC 80/10A	ISBC 80/10B
1	36	37	45
2	35	38	46
3	34	39	41
4	50	40	59
5	51	41	60
6	52	42	(No Equiv.-
7	53		Remove Strap)
8	54	43	61
9	58	44	4
10	57	45	9
11	56	46	10
12	55	47	5
13	16	48	25
14	21	49	20
15	75	50	19
16	74	51	24
17	73	52	90
18	77	53	91
19	78	54	83
20	76	55	82
21	79	56	81
22	37	57	80
23	38	58 (Not Used)	—
24	39	59 (Not Used)	—
25	44	60 (Not Used)	—
26	49	61	92
27	30	62	94
28	29	63	93
29	31	64	95
30	32	65-78	(Replaced by
31	33		Jumper Block
32	49		See Schematic)
33	48	79-80	W2
34	43		
35	47		
36	42		

Table D-3. iSBC 80/10B™ Board Jumper Configuration

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
1 thru 4	Configure Port E6 bits	7 B5	2-17
5-10*	Disable E6 interrupt	7 B5	2-17
6 thru 9	Configure Port E6 bits	7 B5	2-17
11-X	Enable CTI for millisecond timer	7 C4	2-20
12-X	Enable PFSN/	7 C4	2-25
14-X	Enable PFSR/	7 C4	2-25
15-X	Enable millisecond timer (MST)	7 B4	2-20
16 thru 19	Configure Port E6 bits	7 B5	2-17
20-25*	Disable E6 interrupt	7 B5	2-17
21 thru 24	Configure Port E6 bits	7 B5	2-17
26-27*	GND to J2-1	8 B2	None
27-28	+5V to J2-1	8 B2	2-28
30-31*	Connects RTS/ to CTS/	6 A4	None
32-33*	Sets CTS driver to +12 volts	6 A4	None
34-35	Connects TxD to RS232C driver	6 B4	2-15

Table D-3. iSBC 80/10B™ Board Jumper Configuration (Cont'd.)

Jumper Pair	Function	Schematic Sheet/ Grid Loc.	Text Reference
35-36*	Connects TxD to TTY driver	6 B4	None
37-38	Connects DTR to RS232C driver	6 B4	2-15
38-39*	Connects DTR to TTY driver	6 B4	None
41-42	Connects external clock to RxC	6 B5	2-16
41-46*	Connects TTY return to RxD	6 B5	None
42-47*	Connects internal clock to RxC	6 B5	None
43-48*	Connects internal clock to TxC	6 B5	None
44-49*	Connects DTR to DSR input	6 B5	None
45-46	Connects RS232C data to RxD	6 B5	2-15
48-49	Connects external clock to TxC	6 B5	2-16
50-54*	Selects 110 baud for PCI device (see table 2-9)	6. D4	2-13
59-60*	Sets Port E4 to output mode	7 C5	2-17
60-61	Port E4 mode programmed by Port E6 Bit 6	7 C5	2-17
63-64*	Connects HALT/ to P2 Connector	4 A4	2-21
67-68	Enables Multimodule interrupt 0 (MINTR0)	8 A6	2-24
68-69*	Disables Multimodule interrupt 0 (MINTR0)	8 A6	2-24
70-71	Enables Multimodule interrupt 1 (MINTR1)	8 A6	2-24
71-72*	Disables Multimodule interrupt 1 (MINTR1)	8 A6	2-24
73-74	Enable RxRDY interrupt	6 C4	2-14
74-75*	Disable RxRDY interrupt	6 C4	2-14
76-78*	Disable TxRDY interrupt	6 B4	2-14
76-79	Enable TxRDY interrupt	6 B4	2-14
77-78	Enable TxE interrupt	6 B4	2-14
80-81*	Selects 110 baud for PCI device (see table 2-9)	6 D4	2-13
82-83*	Connects RESET to Multibus	2 C6	2-21
84-85*	Connects BPRN to board	2 C6	2-23
85-86	Implements BPRN/ (Multibus compatible)	2 C6	2-23
87-88*	Connects INTR0 to CPU INT	2 D7	2-21
88-89	Connects PFIN/ to CPU INT	2 D7	2-25
90-91	Connects AACK/ to board	4 B7	None
92-93*	Connects BCLK/ to Multibus	3 A3	2-21
94-95*	Connects CCLK/ to Multibus	3 A3	2-21
96-97	Specifies amount of on-board RAM	4 C7	2-9
98-99*	Reserved	5 B6	None
100-101*	Connects WAIT/ to P2 Connector	4 A2	2-21
103-104*	Connects SYNC to P2 Connector	4 C3	None
106-107*	Enables failsafe timer	4 B4	2-19
108-109	Wait State Jumper	4 A7	2-8
W1	-12V to J3-19	6 B2	2-27
W2	GND to J3-1	6 A7	None
W3	+12V to J3-22	6 C6	2-28

Note: * indicates default connection