iSBC® 80/24A
SINGLE BOARD COMPUTER
HARDWARE REFERENCE MANUAL

Order Number: 148437-001

THE GREEN PAGES AT THE END OF THIS MANUAL CONTAIN ERRATA. IF NO GREEN PAGES ARE PRESENT, THIS MANUAL HAD NO ERRATA AT PRESS TIME.

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

REV.	REVISION HISTORY	DATE
-001	Original Issue.	11/85

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.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without prior written consent of Intel Corporation.

Intel Corporation makes no warranty for the use of its products and assumes no responsibility for any errors which may appear in this document nor does it make a commitment to update the information contained herein.

Intel retains the right to make changes to these specifications at any time, without notice.

Contact your local sales office to obtain the latest specifications before placing your order.

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

Above	iLBX	iPDS	ONCE
BITBUS	im	iPSC	OpenNET
COMMputer	iMDDX	iRMX	Plug-A-Bubble
CREDIT	iMMX	iSBC	PROMPT
Data Pipeline	Insite	iSBX	Promware
GENIUS	Intel	iSDM	QueX
$\overset{\Delta}{i}$	intel	iSXM	QUEST
i	intelBOS	Library Manager	Ripplemode
I^2ICE	Intelevision	MCS	RMX/80
ICE	inteligent Identifier	Megachassis	RUPI
iCEL	inteligent Programming	MICROMAINFRAME	Seamless
iCS	Intellec	MULTIBUS	SLD
iDBP	Intellink	MULTICHANNEL	UPI
iDIS	iOSP	MULTIMODULE	VLSiCEL

MDS is an ordering code only and is not used as a product name or trademark. MDS® is a registered trademark of Mohawk Data Sciences Corporation.

*MULTIBUS is a patented Intel bus.

PREFACE

This manual provides general information, installation and setup instructions, programming guidelines for the on-board devices, board level principles of operation, and repair and replacement assistance information for the iSBC 80/24A Single Board Computer. Related information is provided in the following documents:

- *Intel MULTIBUS Specification,* Order Number 9800683.
  - *Intel MULTIBUS Interfacing,* Application Note AP-28A.
- *Intel Microsystems Component Handbook,* Order Number 230843.
  - *Using the 8259A Programmable Interrupt Controller,* Application Note AP-59.
- *Intel Memory Components Handbook,* Order Number 210830.
  - *Intel's 5V EPROM/ROM Family,* Application Note AP-30.
- *Intel MCS-80/85 Family User's Manual,* Order Number 205775.
- *Intel RMX/80 User's Guide,* Order Number 9800522.
- *Intel iSBC 604/614 Cardcage Hardware Reference Manual,* Order Number 9800708.
- *Intel iSBC 655 System Chassis Hardware Reference Manual,* Order Number 9800709.
- *Intel iSBC 660 System Chassis Hardware Reference Manual,* Order Number 9800505.
- *Intel iSBX 331 Fixed/Floating Point Math MULTIMODULE Board Hardware Reference Manual,* Order Number 142668.
- *Intel iSBX 344 Intelligent BITBUS™ Interface Board User's Guide,* Order Number 148009.
- *Intel iSBX 350 Parallel I/O MULTIMODULE Board Hardware Reference Manual,* Order Number 9803191.
- *Intel iSBX 351 Serial I/O MULTIMODULE Board Hardware Reference Manual,* Order Number 9803190.

NOTE TO READERS:

This hardware reference manual utilizes a *visual* scheme to denote section levels, rather than a *numerical* scheme used in many technical documents. This visual scheme allows you to more readily identify which section headings are subsections. Therefore, each section will have the same numbering convention throughout the manual (i.e., section 1-3, section 2-20, section 4-32). The visual distinction among the different sizes and the different fonts used for section headings indicate what level or order a particular section occupies. The following example illustrates how this system is used in this manual:

3-27. 8259A PIC PROGRAMMING	1st Order Heading
3-28. INTERRUPT PRIORITY MODES	2nd Order Heading
3-29. FULLY NESTED MODE. In this . . .	3rd Order Heading

By glancing through this manual before you start reading, the visual method of section ordering will become apparent. You may also refer to the Table of Contents on page vii, to see how the sections compare to each other.

CONTENTS

CONTENTS (Continued)

TABLES

TABLES (Continued)

FIGURES

CHAPTER 1
GENERAL INFORMATION

1-1. INTRODUCTION

The iSBC 80/24A Single Board Computer is a Multibus and Multimodule compatible computer system on a single printed circuit assembly (figure 1-1). The iSBC 80/24A board includes an Intel 8085A-2 microprocessor, 8K bytes of on-board random access memory (RAM), sockets for up to 32K bytes of on-board read-only memory (ROM), six programmable 8-bit I/O ports, one programmable serial communications channel, a programmable interval timer, a programmable interrupt controller, and advanced bus controller circuitry.

This manual provides the information you will need to promptly install and operate the iSBC 80/24A board. To optimize your application of this flexible board, we suggest reading the entire manual before attempting installation and operation.

1-2. DESCRIPTION

The iSBC 80/24A board is controlled by an Intel 8085A-2 microprocessor operating at 4.84 MHz. System access is provided by the Multibus connector and an auxiliary connector. Off-board peripheral I/O operations are handled through 48 parallel lines, a serial communications channel connector and two iSBX Multimodule connectors.

The iSBC 80/24A board can directly access up to 64K bytes of memory. A single SRAM device supplies 8K bytes of on-board RAM. Also, the board can accept up to 32K bytes of user-installed ROM, PROM, or EPROM devices (either 24- or 28-pin devices as defined in Chapter 2).

The on-board 8254 Programmable Interval Timer (PIT) provides three independent counter outputs which may be configured to a variety of applications, including frequency output, rate generator, interval timer and real-time interrupts. One of these counters serves as the baud rate clock for the on-board 8251A Programmable Communications Interface (PCI) device.

Serial I/O operation is handled by an Intel 8251A Programmable Communications Interface (PCI) device. The board is configured to the RS232C structure; however, it may be adapted to a TTY interface using optional equipment. Baud rates are software programmable via the on-board timer.

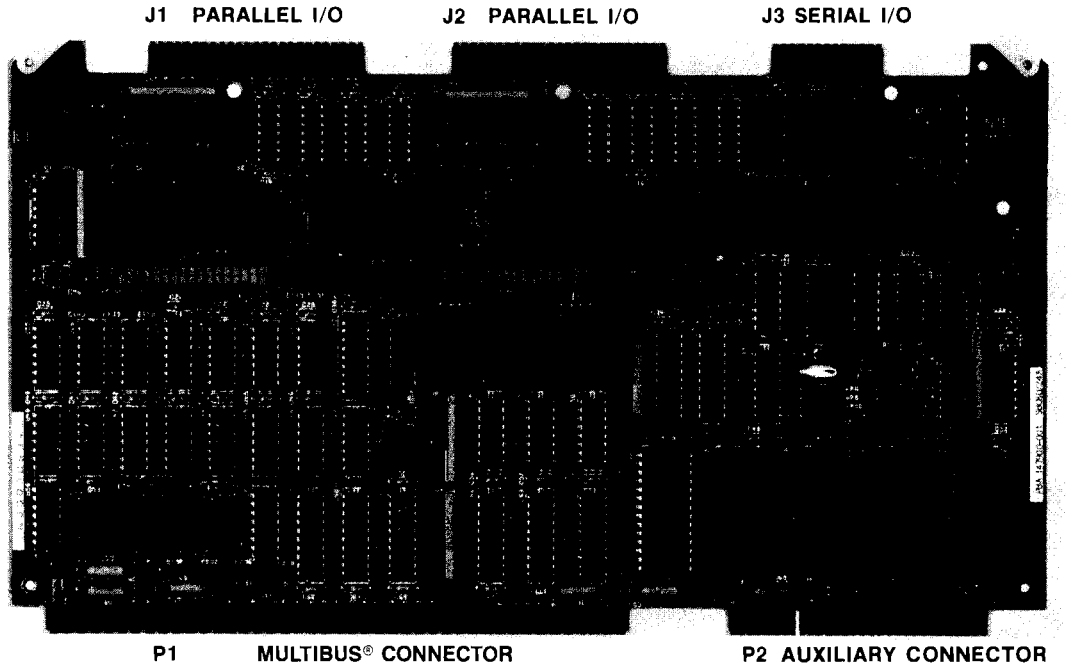


Figure 1-1. iSBC® 80/24A Single Board Computer

The iSBC 80/24A 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. Two Intel 8287 Bus Transceiver devices are supplied for two of the ports and sockets for line driver or terminator devices are provided for the other four ports.

Up to eight interrupts are controlled by the 8259A Programmable Interrupt Controller (PIC), while four additional interrupts are handled directly by the 8085A-2 CPU. An interrupt jumper matrix allows the interrupt structure to be easily configured to your application.

Two iSBX bus connectors are provided on the iSBC 80/24A board. These connectors are designed to expand the board's I/O functions, using special purpose add-on Multimodule boards, such as the iSBX 344 Intelligent BITBUS™ Interface Board. One or two iSBX Multimodule boards may be added, as required by your application.

The iSBC 80/24A board is designed to operate as a full master in any Intel Multibus compatible 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 80/24A board is shipped with a current set of schematic diagrams. Refer to Chapter 5 for related information.

1-4. ADDITIONAL EQUIPMENT REQUIRED

The iSBC 80/24A 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 80/24A board are provided in table 1-1.

Table 1-1. Specifications

CPU	Intel 8085A-2
Operating Rate:	4.84 MHz (default) 2.42 MHz (optional)
Single Clock Cycle:	206 ns (at 4.84 MHz)
Basic Instruction Cycle (four clock cycles):	824 ns
WORD SIZE	
Instruction:	8, 16, or 24 bits
Data:	8 bits
Address:	16 bits
SYSTEM CLOCK:	9.68 MHz
RAM ACCESS TIME:	70 ns maximum (Valid data out from READ command)
MEMORY ADDRESSING (factory configuration)	
On-Board ROM/EPROM:	0-7FFFH
On-Board RAM:	E000-FFFFH

Table 1-1. Specifications (Continued)

MEMORY CAPACITY

On-Board ROM/EPROM: Up to 32K bytes (user-installed)

On-Board RAM: 8K bytes

Off-Board Expansion: Up to 64K bytes in combinations of RAM, ROM, and EPROM.

Note: ROM/EPROM may be added in 1K, 2K, 4K, or 8K byte increments.

I/O ADDRESSING

On-Board Programmable I/O:

Port	8255 No. 1			8255 No. 2			8255 No. 1 Control	8255 No. 2 Control	8251 Data	8251 Control
	1	2	3	4	5	6				
Address	E4	E5	E6	E8	E9	EA	E7	EB	EC	ED

Optional:

J5 Multimodule	J6 Multimodule
C0-CF	F0-FF

ON-BOARD I/O CAPACITY

Parallel: 48 programmable lines.

Serial: 1 Transmit; 1 Receive; 1 SID; 1 SOD

Note: Expandable with Optional Multimodule boards

SERIAL COMMUNICATIONS 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

SERIAL BAUD RATES:

Output Frequency in kHz	Baud Rate (Hz)			8254 PIT Baud Rate Factor (Hex Notation)	
	Synchronous	Asynchronous			
		÷16	÷64	MSB	LSB
153.6	—	9600	2400	00	07
76.8	—	4800	1200	00	0E
38.4	38400	2400	600	00	1C
19.2	19200	1200	300	00	38
9.6	9600	600	150	00	70
4.8	4800	300	75	00	E0
2.4	2400	150	—	01	C0
1.76	1760	110	—	02	63

Baud Rate Register	DE

Note: Baud Rate Factor (16 bits) is loaded as two sequential output operations to same address (DE).

Table 1-1. Specifications (Continued)

INTERRUPTS

Register Address (Hex notation, I/O address space):

Interrupt Request Register	DA or D8
In-Service Register	DA or D8
Mask Register	DB or D9
Command Register	DA or D8
Block Address Register	DB or D9
Status (Polling Register)	DA or D8

TIMERS

Register Address (Hex notation, I/O address space):

Control Register	DF
Timer 0	DC
Timer 1	DD
Timer 2	DE

Input Frequencies:

Reference: 1.0752 MHz ±0.1% (0.930 μsec period, nominal)

Event Rate: 1.1 MHz max

Output Frequencies/ Timing Intervals:

Function	Single Timer/Counter		Dual Timer/Counter (Two Timers Cascaded)	
	Min.	Max.	Min.	Max.
Real-Time Interrupt	1.86 μsec	60.948 msec	3.72 μsec	1.109 hrs
Programmable One-Shot	1.86 μsec	60.948 msec	3.72 μsec	1.109 hrs
Rate Generator	16.407 Hz	537.61 kHz	0.00025 Hz	268.81 kHz
Square-Wave Rate Generator	16.407 Hz	537.61 kHz	0.00025 Hz	268.81 kHz
Software Triggered Strobe	1.86 μsec	60.948 msec	3.72 μsec	1.109 hrs
Hardware Triggered Strobe	1.86 μsec	60.948 msec	3.72 μsec	1.109 hrs

INTERFACES

Multibus: All signals TTL compatible

Parallel I/O: All signals TTL compatible

Interrupt Requests: All signals TTL compatible

Timer: All signals TTL compatible

Serial I/O: RS232C compatible, data set configuration

iSBX Bus: All signals TTL compatible

PHYSICAL CHARACTERISTICS

Width: 12 in (30.48 cm)

Length: 6.75 in (17.15 cm)

Height: 0.5 in (1.27 cm)

Weight: 13.50 oz (383 g)

Table 1-1. Specifications (Continued)

ENVIRONMENTAL REQUIREMENTS

Operating Temperature:	0°C to 55°C (32°F to 130°F)
Storage Temperature:	−40°C to 70°C (−40°F to 158°F)
Operating Humidity:	Up to 90% relative humidity without condensation at 55°C
Storage Humidity:	Up to 95% relative humidity without condensation at 55°C
Airflow Requirements:	Minimum of 200 linear feet per minute of airflow at an ambient temperature of 0°C to 55°C

ELECTRICAL CHARACTERISTICS

Input Power:

	Configuration Description	Current Required			
		I_{CC}	I_{DD}	I_{BB}	I_{AA}
1	Assumes the following items are installed: four 2708 EPROM devices, eight 220/330 ohm I/O termination packs in the parallel I/O interface (driven low).	3.28 A	300 ma	180 ma	20 ma
2	Assumes the 2708 EPROM devices and the iSBC 901 termination networks are *not* installed.	2.66 A	40 ma	0 ma	20 ma
3	Same as note 1, except four 2758 EPROM devices are installed instead of 2708 devices.	3.44 A	40 ma	0 ma	20 ma
4	Same as note 1, except four 2716 EPROM devices are installed instead of 2708 devices.	3.44 A	40 ma	0 ma	20 ma
5	Same as note 1, except four 2732A EPROM devices are installed instead of 2708 devices.	3.46 A	40 ma	0 ma	20 ma
6	Same as note 1, except four 2764A EPROM devices are installed instead of 2708 devices.	3.42 A	40 ma	0 ma	20 ma
7	Same as note 1, with the addition of an iSBC 530 TTY adaptor module connected.	3.28 A	400 ma	180 ma	120 ma
8	Battery backup current requirements for 8K RAM (board not operating).	10 ma	—	—	—
9	Current requirements for 8K RAM (board operating).	88 ma	—	—	—
	Voltage:	V_{CC} = +5v ±5%	V_{DD} = +12v ±5%	V_{BB} = −5v ±5%	V_{AA} = −12v ±5%

Table 1-1. Specifications (Continued)

LINE DRIVERS AND TERMINATORS

I/O Drivers:

The following line drivers are all compatible with the I/O driver sockets.

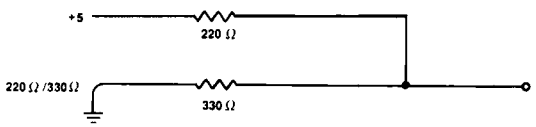
Driver	Characteristic	Sink Current (mA)
7438	I, OC	48
7437	I	48
7432	NI	16
7426	I, OC	16
7409	NI, OC	16
7408	NI	16
7403	I, OC	16
7400	I	16

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

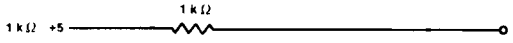
Ports E4 and E8 have 25 mA totem-pole dividers and 1 kΩ terminators.

I/O Terminators:

220/330 ohm Divider
Pull Up/Pull Down
Terminator Pack



1k ohm Pull Up
Terminator Pack



* * *

CHAPTER 2
PREPARATION FOR USE

2-1. INTRODUCTION

This chapter provides specific information enabling you to install the iSBC 80/24A board into your own 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 80/24A 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 water-stained, request that the carrier's agent be present when the carton is opened. If the carrier's agent is not present when the carton is opened and the contents are damaged, keep the carton and packing material for the agent's inspection.

For repairs to a product damaged in shipment contact Intel Customer Support Service (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 environmental 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 80/24A board default configuration is described in Chapter 1. In order to operate the board you may 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 2-16).

c. 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 80/24A board in most configurations: +5 Vdc, −5 Vdc, +12 Vdc, and −12 Vdc. All must be within ±5.0% of absolute. However, some configurations do not require all voltages (e.g., if 2708 EPROM devices are not used, the −5 Vdc requirement is eliminated). Power requirements for the various board configurations are listed in table 1-1. The table does not include power required by any optional Multimodule boards which may be installed on the iSBC 80/24A board. Refer to the specific Multimodule board hardware reference manual for its power requirements.

2-6. COOLING REQUIREMENTS

Operating temperature range for the iSBC 80/24A 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 by providing a minimum of 200 linear feet per minute of airflow at a ambient temperature of 0° to 55°C.

2-7. COMPONENT INSTALLATION

Instructions for installing optional ROM/EPROM, and line drivers or terminators are given in the following sections. Multimodule boards are discussed in section 2-26.

CAUTION

MOS-type devices are extremely sensitive to transient voltages, especially static electricity discharges. Caution should be exercised in low humidity environments during device installation, to prevent static discharge. Always ground yourself before handling MOS devices to ensure any static charge which may have accumulated is discharged. After picking up the device, do not walk on carpeted floors; install the device immediately following the grounding.

2-8. ROM/PROM/EPROM INSTALLATION

Sockets U52 through U55 are reserved for optional ROM/PROM/EPROM devices. A maximum of 32K bytes may be installed. A summary of compatible device types, capacity, and addressing is provided in table 2-1. Device types may not be mixed; however, empty sockets are allowed (provided they are not addressed).

After selecting the ROM type which best suits your application, carefully insert each device into its socket.

CAUTION

Never install any device into a board when power is applied. Damage to the board, device, and power supply could result.

CAUTION

ROM sockets U52-55 are 28-pin sockets. If you are inserting 24-pin devices, ensure they are positioned as shown in figure 2-1 to avoid damage to the devices or to the iSBC 80/24A board.

When using EPROMs with access times less than 300ns, no EPROM wait states are necessary at either operating speed. Insert jumper E98-107 for the EPROM acknowledge to by-pass the wait state generating circuit. Note that inserting E96-105 causes both the EPROM and on-board I/O to by-pass the wait state generation circuit. This should only be done when operating the iSBC 80/24A in its slower mode (2.43 MHz). Refer to table 2-2 for the wait state jumper options.

EPROMs with access times greater than 500ns cannot be used with the iSBC 80/24A board in the 4.84 MHz operating mode.

Table 2-1. EPROM/ROM Configurations

Device Type/Size	Socket and Address Ranges				Total ROM Address Range (Hex)
	U52	U53	U54	U55	
2708/2758 2608 (1K x 8)	0-03FF	0400-07FF	0800-0BFF	0C00-0FFF	0000-0FFF (4K)
2716 2316 E (2K x 8)	0-07FF	0800-0FFF	1000-17FF	1800-1FFF	0000-1FFF (8K)
2732, 2732A (4K x 8)	0-0FFF	1000-1FFF	2000-2FFF	3000-3FFF	0000-3FFF (16K)
2764, 2764A* (8K x 8)	0-1FFF	2000-3FFF	4000-5FFF	6000-7FFF	0000-7FFF (32K)

NOTE: EPROM/ROM types cannot be mixed.
* = Default configuration

2-9. EPROM DEVICE TYPE CONFIGURATION

Once your optional EPROM devices are installed, you must configure the EPROM device type jumper post array for your device type (see figure 2-2 for possible configurations). The default configuration is for 2764 devices (8K x 8).

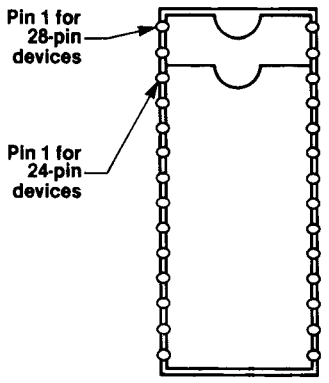


Figure 2-1. EPROM/ROM Device Positioning Guide

Table 2-2. Wait State Jumpers

EPROM MAXIMUM ACCESS TIME (tacc, in ns)	4.84 MHz Operating Speed		2.42 MHz Operating Speed	
	Wait State Needed	Jumpers Needed	Wait State Needed	Jumpers Needed
tac > 300ns	0	E98-107	0	E96-105
300 > tacc > 500	1*	none	0	E96-105
500 > tacc > 775	—	—	0	E96-105

* = Default configuration

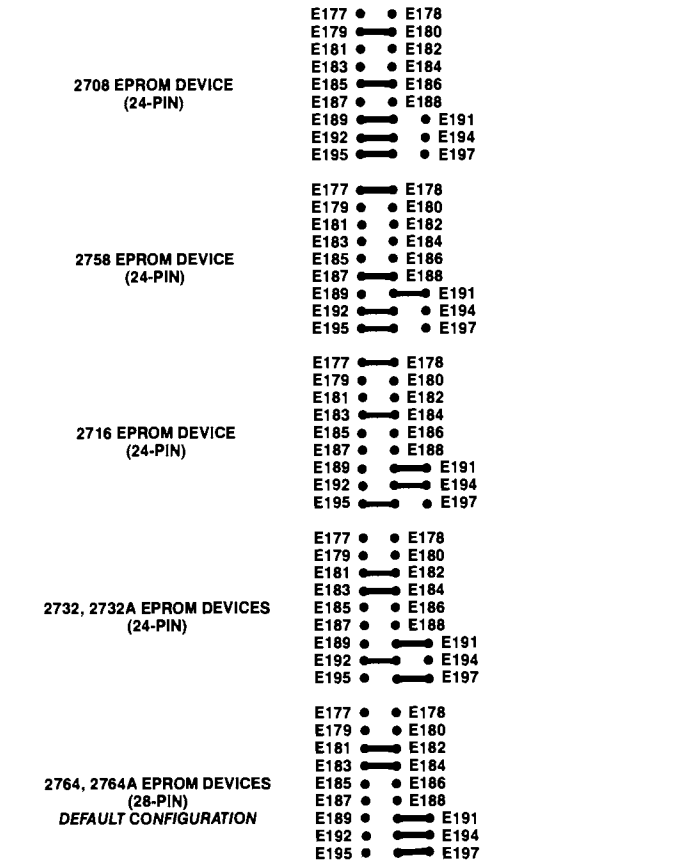


Figure 2-2. EPROM Device Type Jumper Configuration

CAUTION

Incorrectly configuring the EPROM device type jumpers can result in damage to the EPROM devices and to the iSBC 80/24A board itself.

2-10. LINE DRIVERS AND I/O TERMINATORS

Either line driver and/or I/O terminator devices can be installed (as required) in sockets U2-U5 and U7-U10. These sockets correspond to the 32 parallel I/O lines (without drivers or terminators). Ports E4 and E8 have Intel 8287 transceiver devices installed at the factory. Refer to table 2-3 for recommended I/O terminators and to table 1-1 (Chapter 1) for recommended line drivers for the iSBC 80/24A board.

Table 2-3. Recommended I/O Terminators

I/O Terminator Pack	Vendors/Part Number
220/330 ohm Pull Up/Pull Down Terminator Pack	iSBC 901 Beckman Instrument (Part #1899-746-1)
1k ohm Pull Up Terminator Pack	iSBC 902 Beckman Instrument (Part #1899-747-1) Dale Electronics (Part #MDP14-00-593)

2-11. JUMPER CONFIGURATIONS

Much of the flexibility of your iSBC 80/24A board is due to the use of jumper connections which may easily be altered from their default configurations to suit your particular application. Table 2-4 summarizes the jumper connections and their uses. Table 2-4A lists the jumper connections in numerical order, and indicates the factory default configurations. Physical locations of jumper posts on the board are shown in figure 5-3. Jumper connections are also shown schematically in figure 5-4.

NOTE

Jumper posts are shown on the schematic diagrams with an E prefix (e.g., E86).

2-12. RAM CAPACITY AND ADDRESSING

The on-board RAM capacity and addressing is jumper configurable. The default configuration is 8K bytes of RAM from E000-FFFFH. To alter these configurations, refer to table 2-4 and section 4-10.

Table 2-4. Jumper Connections

Function	Fig. 5-3 Ref.	Jumper Pair	Description
MULTIBUS SIGNALS			
AACK/	Sh. 2-D6	E202-203	Connects AACK/ signal from P1-25.
BPRO/	Sh. 2-D3	E200-201*	Connects BPRO/ to P1-16.
BCLK/	Sh. 2-A6	E198-199*	Connects BCLK/ to P1-13.
CCLK/	Sh. 2-A6	E204-205*	Connects CCLK/ to P1-31.
BTMO	Sh. 3-A2	E220-221*	Connects BTMO to P2-34.
INTA/	Sh. 3-B2	E218-219*	Connects INTA/ to P2-36.
PFSR/	Sh. 5-D4	E31-XX	Connects 8224 OSC output to U20 Counter.
TIMING & CPU			
4.84 MHz Operation	Sh. 3-D5	E166-168*	Uses 9.68 MHz clock input
2.42 MHz Operation	Sh. 3-D5	E163-166	Uses 4.84 MHz clock input
	Sh. 3-C5	E96-105	
8224 OSC Output	Sh. 3-D6	E171-174*	Connects 8224 OSC output to U20 Counter.
RAM: AMOUNT ON-BOARD	Sh. 4-C7	E109-100	Indicates 4K RAM on-board.
	Sh. 4-C7	E109-100 E101-110	Both jumpers in indicates 2K RAM; or Both jumpers out indicates 8K RAM. (default configuration)

Table 2-4. Jumper Connections (Continued)

Function	Fig. 5-3 Ref.	Jumper Pair	Description
RAM ADDRESSING Selects maximum (hexadecimal)	Sh. 4-C5	E112-113 E111-112 E103-104 E102-103*	Note: To disable all on-board RAM, remove all jumpers from posts E102, 103, 104, 111, 112 and 113. 2K: E112-113 3800-3FFF; E111-112 7800-7FFF; E103-104 B800-BFFF; E102-103* F800-FFFF 4K: E112-113 3000-3FFF; E111-112 7000-7FFF; E103-104 B000-BFFF; E102-103* F000-FFFF 8K: E112-113 2000-3FFF; E111-112 6000-7FFF; E103-104 A000-BFFF; E102-103* E000-FFFF NOTE: Install one jumper only.
8254 PIT			
Inputs/Outputs	Sh. 7-C6	E167-170	Connects Timer 0 output to Timer 1 input.
		E167-176	Connects Timer 0 output to Parallel port E6 jumper matrix.
		E169-170* E172-173*	Provides 1.075 MHz to CLK0 and CLK1 inputs.
		E170-175 E173-175	Provides 134.4 KHz to CLK0 and CLK1 inputs.
		E170-176 E173-176	Connects parallel port E6 jumper matrix to one or more timer inputs.
		E172-176	Connects 1.075 MHz to parallel port E6 jumper matrix.
		E175-176	Connects 134.4 KHz to parallel port E6 jumper matrix.
Gate Inputs	Sh. 5-D4	E13-28* E14-29*	Provides +5 volts to gate 0 and gate 1 inputs of 8254 PIT.
SERIAL PORT			Refer to section 2-13.
Internal Clocks			
RxC TxC	Sh. 7-C6 Sh. 7-C6	E87-88* E89-90*	Connects PCI device to PIT Timer 2 output. (Baud rate clock)
External Clocks			
RxC	Sh. 7-C6	E86-88	Connects RxC input to J3-7.
TxC	Sh. 7-C6	E89-91	Connects TxC input to J3-3.
Secondary RTS	Sh. 7-D7	E83-84	Connects J3-11 input to AUX1/ parallel port EA matrix.
Secondary TxD	Sh. 7-D7	E84-85	Connects J3-1 input to AUX1/ parallel port EA matrix.
Secondary CTS	Sh. 7-D2	E94-95	Connects J3-26 output to AUX0/ parallel port EA matrix.
Secondary RxD	Sh. 7-D2	E93-95	Connects J3-5 output to AUX0/ parallel port EA matrix.
Secondary TxC	Sh. 7-D2	E92-95	Connects J3-21 output to AUX0/ parallel port EA matrix.
Auxiliary Output 0 (AUX0/)	Sh. 6-D4	E47-XX	Connects selected bit to jumper post E95. Connect to selected port EA bit; remove factory jumper for that bit. Use for Serial Channel protocol. See Sh. 7-D2.
Auxiliary Input 1 (AUX1/)	Sh. 6-D4	E62-XX	Connects J3-11 or J3-1 to selected EA bit via jumper matrix E83-84-85. See Sh. 7-D7. Connect to selected port EA bit; remove factory jumper for that bit. Select one or both as required for Serial Channel protocol.
Ring Indicator (input)	Sh. 6-D4	E46-XX	Connects J3-17 input to selected port EA bit. Connect to selected port EA bit; remove factory jumper for that bit.
Received Line Signal Detector (input)	Sh. 6-D4	E45-XX	Connects J3-16 input to selected port EA bit. Connect to selected port EA bit; remove factory jumper for that bit.
PARALLEL PORTS			
Operating Modes	Sh. 5-D4 Sh. 6-D4	See table 2-7	Various configurations depending on 8255A mode and bit restrictions. Refer to section 2-14 and table 2-7.

* = Default jumper

Table 2-4. Jumper Connections (Continued)

Function	Fig. 5-3 Ref.	Jumper Pair	Description
Interrupt Out/ Bank Select	Sh. 5-D4 and Sh. 9-D5	E16-XX and E206-211	Output only. Allows the selected E6 bit (posts E24-27 or E19-22) to be used as output flag on P1-28. Port C must be in output mode.
Serial Input Data (SID) line	Sh. 6-D4	E61-XX	Data will be routed directly from J2 pin to 8085A-2 SID input. Connect to selected port EA bit (posts E41-44 or E36-39); remove factory jumper for that bit.
Serial Output Data (SOD) line	Sh. 6-D4	E60-XX	Data from 8085A-2 SOD output will be routed directly to J2 pin. Connect to selected port EA bit (posts E41-44 or E36-39); remove factory jumper.
PROM Enable	Sh. 5-D4	E32-XX	Allows software selection of PROM ACK/ signal. Connect to selected port E6 bit; remove factory jumper for that bit. This signal is usually generated by address decode PROM. Refer to Sh. 8-C5.
PFSR/ Signal	Sh. 5-D4	E31-XX	Connects Power Fail Sense Reset signal to P2-13. Refer to section 2-22. Connect to selected port E6 bit; remove factory jumper for that bit.
INTERRUPTS			Refer to section 2-15.
Priority	Sh. 9-B5	Matrix	Various configurations allowed. Refer to section 2-15 and table 2-7.
FAILSAFE TIMER	Sh. 3-B7	E97-106*	Provides READY signal to CPU after 10 ms if response is not provided by addressed I/O or memory. CPU will be in wait state until READY signal is true.
POWER CONNECTION			
5 volt Aux	Sh. 1-D6	E216-217*	Connects +5 volt Main to +5v Aux.
5 volt Aux	Sh. 5-B2	E1-2	Connects +5 volts to J1-50 output. See Note 1.
5 volt Aux	Sh. 6-B2	E33-48	Connects +5 volts to J2-50 output. See Note 1.
5 volt Aux	Sh. 7-A2	E76-77	Connects +5 volts to J3-23 output. See Note 1.
+12 volt Aux	Sh. 7-A2	E78-79	Connects +12 volts to J3-22 output. See Note 1.
−12 volt Aux	Sh. 7-A2	E75-80	Connects −12 volts to J3-19 output. See Note 1.
GND	Sh. 7-C2	E81-82	Connects GND to J3-2 output.

* = Default jumper

Note 1: Use caution with power output connections. Misconnection could cause damage to the board and power supply.

Table 2-4A. Numerical Listing of Jumpers

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Text Ref.
1-2	5-B2	Connects +5 volts to J1-50	None
3-4*	5-C4	Configures port E4 bus transceiver to input mode	2-24
4-18	5-C4	Configures port E4 bus transceiver to output mode	2-14
5-19*	5-C4	Connects port E6, bit 0 to driver/terminator socket	2-14
6-20*	5-C4	Connects port E6, bit 1 to driver/terminator socket	2-14
7-21*	5-C4	Connects port E6, bit 2 to driver/terminator socket	2-14
8-22*	5-C4	Connects port E6, bit 3 to driver/terminator socket	2-14
9-24*	5-C4	Connects port E6, bit 4 to driver/terminator socket	2-14
10-25*	5-C4	Connects port E6, bit 5 to driver/terminator socket	2-14
11-26*	5-C4	Connects port E6, bit 6 to driver/terminator socket	2-14
12-27*	5-C4	Connects port E6, bit 7 to driver/terminator socket	2-14
13-28*	5-C4	Applies +5 volts to timer gate 0 (ITG0)	None
14-29*	5-C4	Applies +5 volts to timer gate 1 (ITG1)	None
15-30	5-C4	Connects timer 1 output to timer input matrix	2-11
15-XX	5-C4	Connects selected J1 bit to timer input matrix	2-11
16-XX	5-C4	Connects INTROUT/BANKSEL to selected E6 bit	2-11
17-19	5-C4	Connects port E6, bit 0 to PIA1	2-14
22-23	5-C4	Connects port E6, bit 3 to PIB1	2-14
31-XX	5-C4	Connects PFSR/ to selected E6 bit	2-22
32-XX	5-C4	Connects PROM ENABLE to selected E6 bit	2-11
33-48	6-B2	Connects +5 volts to J2-50	None
34-35*	6-C4	Configures port E8 bus transceiver to input mode	2-14
35-50	6-C4	Configures port E8 bus transceiver to output mode	2-14
36-51*	6-C4	Connects port EA, bit 0 to driver/terminator socket	2-14
37-52*	6-C4	Connects port EA, bit 1 to driver/terminator socket	2-14
38-53*	6-C4	Connects port EA, bit 2 to driver/terminator socket	2-14
39-54*	6-C4	Connects port EA, bit 3 to driver/terminator socket	2-14
40-XX	6-C4	Ground post	2-14
41-56*	6-C4	Connects port EA, bit 4 to driver/terminator socket	2-14
42-57*	6-C4	Connects port EA, bit 5 to driver/terminator socket	2-14
43-58*	6-C4	Connects port EA, bit 6 to driver/terminator socket	2-14
44-59*	6-C4	Connects port EA, bit 7 to driver/terminator socket	2-14
45-XX	6-C4	Connects BDET/ to selected EA bit	2-14
46-XX	6-C4	Connects BRI/ to selected EA bit	2-14
47-XX	6-C4	Connects AUX0/ to selected EA bit	2-14
49-51	6-C4	Connects port EA, bit 0 to PIA2/	2-14
54-55	6-C4	Connects port EA, bit 3 to PIA1/	2-14
60-XX	6-C4	Connects SOD to selected EA bit	2-14
61-XX	6-C4	Connects SID to selected EA bit	2-14
62-XX	6-C4	Connects AUX1/ to selected EA bit	2-14
63-64*	7-C3	Connects transmit data to J3-4	2-13
63-74	7-C3	J3 serial jumper matrix	2-13
65-66*	7-C3	Connects receive data to J3-6	2-13
67-68*	7-C3	Connects RTS to J3-8	2-13
69-70*	7-C3	Connects CTS to J3-10	2-13
71-72*	7-C3	Connects DSR to J3-12	2-13
73-74*	7-C3	Connect DTR to J3-13	2-13
75-80	7-A2	Connects −12 volts to J3-19	2-25

Table 2-4A. Numerical Listing of Jumpers (Continued)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Text Ref.
76-77	7-A2	Connects +5 volts to J3-23	None
78-79	7-A2	Connects +12 volts to J3-22 (Used for TTY serial interface)	2-25
81-82	7-C2	RS232C Protective Ground J3-2	2-13
83-84	7-D7	Connects Secondary RST (J3-11) input to AUX1/	2-13
84-85	7-D7	Connects Secondary TxD (J3-11) input to AUX1/	2-13
86-88	7-D6	Connects external Receive Clock (RxC)	2-13
87-88*	7-D6	Connects internal Receive Clock (RxC)	2-13
89-90*	7-D6	Connects internal Transmit Clock (TxC)	2-13
89-91	7-D6	Connects external Transmit Clock (TxC)	2-13
92-95	7-D2	Connects AUX0/ output to J3-21	2-13
93-95	7-D2	Connects AUX0/ output to J3-5	2-13
94-95	7D2	Connects AUX0/ output to J3-26	2-13
96-105	3-C5	Connects for 4.84 MHz clock only	4-3
97-106*	3-B6	Enables failsafe timer	4-35
98-107	3-B7	Reserved	None
99-108*	3-B7	Connects RAMACK/ to READY gate	4-29
100-109	4-C7	RAM size jumper. See table 2-4	2-12
101-110	4-C7	RAM size jumper. See table 2-4	2-12
102-103*	4-C5	Indicates RAM area: E000-FFFFH	2-12
102-104	4-C5	RAM address matrix. See table 2-4	2-12
111-113	4-C5	RAM address matrix. See table 2-4	2-12
112-113	4-C5	Indicates RAM area: 2000-3FFFH	2-12
114-115*	9-B4	Connects ground to IR7 on PIC	None
114-162	9-C4	Interrrupt Jumper Matrix. See table 2-7	2-15
133-134*	9-C4	Connects INT1/ to IR1 on PIC	2-15
136-137	10-D5	Multimodule board option (reserved); J5	None
147-148	10-C3	Multimodule board option (reserved); J6	None
153-154*	9-C4	Connects OITO to IR2 on PIC	2-15
155-159*	9-C4	Connects ground to RST5.5 on CPU	2-15
156-160*	9-C4	Connects ground to RST6.5 on CPU	2-15
157-161*	9-C4	Connects ground to RST7.5 on CPU	2-15
158-162*	9-C4	Connects ground to TRAP on CPU	2-15
163-166	3-D5	4.84 MHz Clock input to 8085A-2 CPU	4-3
164-165*	7-C7	Connects 2.15 MHz to PIT Clock divider	None
166-168*	3-D5	9.68 MHz Clock input to 8085A-2 CPU	4-3
167-170	7-B6	Connects Timer 0 output to Timer 1 input	None
167-176	7-B6	Connects Timer 0 output to port E6 bit via CLK OUT line	None
169-170*	7-B6	Connects 1.075 MHz to CLK1 on PIT	None
171-174*	7-B6	Connects 8224 OSC output to divider network	None
172-173*	7-B6	Connects 1.075 MHz to CLK0 on PIT	None
175-XX	7-B6	Connects 134.4 KHz: See table 2-4	None
177-197	4-B6	EPROM Jumper Matrix	2-9
179-180	4-B6	Connects −5 volts to EPROM socket pin 23 Vbb/A11	2-9
181-182*	4-B6	Connects address AB to EPROM socket pin 23 Vbb/A11	2-9
183-184*	4-B6	Connects address AA to EPROM socket pin 21 Vdd/A10	2-9
185-186	4-B6	Connects +12 volts to EPROM socket pin 21 Vdd/A10	2-9
189-190	4-B6	Connects ground to EPROM socket pin 20 PGM/CE\	2-9
190-191*	4-B6	Connects PROMACK\ to EPROM socket pin 20 PGM/CE\	2-9

Table 2-4A. Numerical Listing of Jumpers (Continued)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Text Ref.
192-193	4-B6	Connects ground to PTYPE0	2-9
193-194*	4-B6	Connects +5 volts t PTYPE0	2-9
195-196	4-B6	Connects ground to PTYPE1	2-9
196-197*	4-B6	Connects +5 volts to PTYPE1	2-9
198-199*	2-A6	Connects 9.68 MHz BCLK\ to P1-13 (output)	4-3
200-201*	2-D3	Connects BPRO\ to P1-16 (output)	2-19
202-203	2-D6	Connects AACK\ to iSBC 80/24A board via P1-25 (input)	None
204-205*	2-A6	Connects 9.68 MHz CCLK\ to P1-31 (output)	4-3
206-211	9-C5	Connects INTROUT/BANKSEL to P1-28 (ADR10\) output	None
216-217*	1-D6	Used for +5 volt Battery Backup	2-22
218-219*	3-B2	Connects INTA\ to P2-36 (output)	4-23
220-221*	3-A2	Connects BTMO to P2-34 (output)	4-34

*Indicates default jumper configuration

2-13. SERIAL PORT

The iSBC 80/24A board serial port is default configured as a RS232C interface. The serial port is default jumpered as a data set (modem), but can easily be reconfigured as a data terminal. Table 2-5 lists the serial port jumper options and also provides pin identification. If your application requires external clocks or other serial interface modifications, refer to the Serial Interface section of table 2-4. For serial port cabling information, refer to section 2-24.

Table 2-5. Serial Port Jumper Options

8251A Function	Jumper #	Serial Port Connector	Signal Direction
TxD	E65-66	J3-6 REC DATA	OUTPUT
RxD	E63-64	J3-4 TRANS DATA	INPUT
DTR/	E71-72	J3-12 DSR	OUTPUT
DSR/	E73-74	J3-13 DTR	INPUT
RTS/	E69-70	J3-10 CTS	OUTPUT
CTS/	E67-68	J3-8 RTS	INPUT

2-14. PARALLEL PORTS

Parallel ports E6 and EA each have a jumper matrix between the 8255A PPI device and the driver/terminator sockets. This configuration allows a greater amount of flexibility when using either port.

All other parallel ports do not have jumper matrices, and their operation is determined by software programming. Refer to table 3-18 for a list of operating modes which are allowed for each parallel port.

Before configuring the parallel ports for your application, refer to section 3-22 for 8255A PPI programming information. Jumper information for all parallel ports, including bit restrictions, is provided in table 2-6.

2-15. INTERRUPT MATRIX

The iSBC 80/24A board can resolve 12 levels of interrupt priority. Each of these levels are controlled by the 8259A PIC and the remaining four are direct inputs to the 8085A-2 CPU. All interrupts are routed through the interrupt matrix, shown on figure 5-4, sheet 9. Table 2-7 lists all the matrix jumper posts and their functions. The board is shipped from the factory with the following two jumpers installed:

E134-133 — INT2/ from Multibus Interface

E153-154 — Counter 0 output

Table 2-6. Configuration Jumpers for Parallel Ports E4 - EA

Port	Mode	Driver (D)/ Terminator (T)	Jumper Configuration				
			Remove	Add	Effect	Port	Restrictions
E4	0 Input	8287: U1	E3-4*	E4-18	8287 = input enabled.	E5	None; can be in Mode 0 or 1, input or output.
						E6	None; can be in Mode 0, input or output, unless Port E5 is in Mode 1.
E4	0 Output (latched)	8287: U1	None	E3-4*	8287 = output enabled.	E5	None; can be in Mode 0 or 1, input or output.
						E6	None; can be in Mode 0, input or output, unless Port E5 is in Mode 1.
E4	1 Input (strobed)	8287: U1 T: U2 D: U3	E3-4*	E4-18	8287 = output enabled.	E5	None; can be in Mode 0 or 1, input or output.
			—	E9-24*	Connects J1-26 to STB_A/ input.	E6	Port E6 bits perform the following: • Bits 0, 1, 2 — Control for Port E5 if in Mode 1. • Bit 3 — Port E4 Interrupt to interrupt jumper matrix. • Bit 4 — Port E4 Strobe (STB/) input. • Bit 5 — Port E4 Input Buffer Full (IBF) output. • Bits 6, 7 — Port E6 input or output (both must be in same direction).
			E10-25* and E8-22*	E8-25	Connects IBF_A output to J1-18.		
				—	Disconnect Port E6, bit 3 (INTR) from driver at U3.		
E4	1 Output (latched)	8287: U1 T: U2 D: U3	—	E3-4*	8287 = output enabled.	E5	None; can be in Mode 0 or 1, input or output.
			—	E11-26*	Connects J1-30 to ACK_A/ input.	E6	Port E6 bits perform the following: • Bits 0, 1, 2 — Control for Port E5 if in Mode 1. • Bit 3 — Port E4 Interrupt to interrupt jumper matrix. • Bits 4, 5 — Input or output (both must be in same direction). • Bit 6 — Port E4 Acknowledge (ACK/) input. • Bit 7 — Port E4 Output Buffer Full (OBF/) output.
			E12-27* and E8-22*	E8-27	Connects OBF_A/ output to J1-18.		
				—	Disconnect Port E6, bit 3 (INTR) from driver at U3.		
E4	2 (bidirectional)	8287: U1 T: U3	E3-4*	E4-26	Allows ACK_A/ output to control 8287 in/out direction.	E5	None.
			—	E9-24*	Connects J1-26 to STB_A input.	E6	Port E6 bits perform the following: • Bit 0 — Cannot be used. • Bits 1, 2 — Can be used for input or output if Port E5 is in Mode 0. • Bit 3 — Port E4 Interrupt to interrupt jumper matrix. • Bit 4 — Port E4 Strobe (STB/) input.
			E10-25* and E5-19*	E5-25	Connects IBF_A output to J1-24.		
			—	E11-26*	Connects J1-30 to ACK_A/ input.		
			E12-27*	E8-27	Connects OBF_A/ output to J1-18.		

* Default jumper connected at the factory.

Table 2-6. Configuration Jumpers for Parallel Ports E4 - EA (Continued)

Port	Mode	Driver (D)/ Terminator (T)	Jumper Configuration				
			Remove	Add	Effect	Port	Restrictions
			E8-22*	—	Disconnects Port E6, bit 3 (INTR) from driver at U3.		• Bit 5 — Port E4 Input Buffer Full (IBF) output. • Bit 6 — Port E4 Acknowledge (ACK/) input. • Bit 7 — Port E4 Output Buffer Full (OBF/) output.
E5	0 Input	T: U4, U5	None	None		E4	None
						E6	None; Port E6 can be in Mode 0, input or output, if Port E4 is also in Mode 0.
E5	0 Output (latched)	D: U4, U5	None	None		E4	None.
						E6	None; Port E6 can be in Mode 0, input or output, if Port E4 is also in Mode 0.
E5	1 Input (strobed)	T: U2, U4, U5 D: U3	— E12-27* and E7-21* E5-19*	E6-20* E12-21 —	Connects IBF_B output to J1-22. Connects J1-32 to STB_B/ input. Disconnects Port E6, bit 0 (INTR) from driver at U3.	E4	None.
						E6	Port E6 bits perform the following: • Bit 0 — Port E5 Interrupt to interrupt jumper matrix. • Bit 1 — Port E5 Input Buffer Full (IBF) output. • Bit 2 — Port E5 Strobe (STB/) input. • Bit 3 — If Port E4 is in Mode 0, bit 3 can be input or output. Otherwise, bit 3 is reserved. • Bits 4-7 — Depends on Port E4 mode; see table 3-21.
E5	1 Output (latched)	T: U2 D: U3, U4, U5	— E12-27* and E7-21* E5-19*	E6-20* E12-21 —	Connects OBF_B/ output to J1-22. Connects J1-32 to ACK_B/ input. Disconnects Port E6, bit 0 (INTR) from driver at U3.	E4	None.
						E6	Port E6 bits perform the following: • Bit 0 — Port E5 Interrupt to interrupt jumper matrix. Bit 1 — Port E5 Output Buffer Full (OBF/) output. • Bit 2 — Port E5 Acknowledge (ACK/) input. • Bit 3 — If Port E4 is in Mode 0, bit 3 can be input or output. Otherwise, bit 3 is reserved. • Bits 4-7 — Input or output (both must be in same direction); see table 3-21.
E6 (upper)	0 Input	T: U2	None	E9-24* E10-25* E11-26* E12-27*	Connects bit 4 to J1-26. Connects bit 5 to J1-28. Connects bit 6 to J1-30. Connects bit 7 to J1-32.	E4	Port E4 must be in Mode 0 for all four bits to be available.
						E5	Port E5 must be in Mode 0 for all four bits to be available.

* Default jumper connected at the factory.

Table 2-6. Configuration Jumpers for Parallel Ports E4 - EA (Continued)

Port	Mode	Driver (D)/ Terminator (T)	Jumper Configuration				
			Remove	Add	Effect	Port	Restrictions
E6 (lower)	0 Input	T: U3	None	E5-19*	Connects bit 0 to J1-24.	E4	Port E4 must be in Mode 0 for all four bits to be available.
				E6-20*	Connects bit 1 to J1-22.		
				E7-21*	Connects bit 2 to J1-20.	E5	Port E5 must be in Mode 0 for all four bits to be available.
				E8-22*	Connects bit 3 to J1-18.		
E6 (upper)	0 Output (latched)	D: U2	None	Same as for Port E6 (upper) Mode 0 Input.		E4 & E5	Same as for Port E6 (upper) Mode 0 Input.
E6 (lower)	0 Output (latched)	D: U3	None	Same as for Port E6 (lower) Mode 0 Input.		E4 & E5	Same as for Port E6 (lower) Mode 0 Input.
E8	0 Input	8287: U6	E34-35*	E35-50	8287 = input enabled.	E9	None; can be in Mode 0 or 1, input or output.
						EA	None; can be in Mode 0, input or output, unless Port E9 is in Mode 1.
E8	0 Output (latched)	8287: U6	None	E34-35*	8287 = output enabled.	E9	None; can be in Mode 0 or 1, input or output.
						EA	None; can be in Mode 0, input or output, unless Port E9 is in Mode 1.
E8	1 Input (strobed)	8287: U6 T: U7 D: U8	E34-35*	E35-50	8287 = output enabled.	E9	None; can be in Mode 0 or 1, input or output.
			—	E41-56*	Connects J2-26 to STB_A/ input.	EA	Port EA bits perform the following: • Bits 0, 1, 2 — Control for Port E9 if in Mode 1. • Bit 3 — Port E8 Interrupt to interrupt jumper matrix. • Bit 4 — Port E8 Strobe (STB/) input. • Bit 5 — Port E8 Input Buffer Full (IBF) output. • Bits 6, 7 — Port EA input or output (both must be in same direction).
			E42-57*	E39-57	Connects IBF_A output to J2-18.		
			E39-54*	—	Disconnects Port EA, bit 3 (INTR) from driver at U8.		
E8	1 Output (latched)	8287: U6 T: U7 D: U8	—	E34-35*	8287 = output enabled.	E9	None; can be in Mode 0 or 1, input or output.
			—	E43-58*	Connects J2-30 to ACK_A/ input.	EA	Port EA bits perform the following: • Bits 0, 1, 2 — Control for Port E9 if in Mode 1. • Bit 3 — Port E8 Interrupt to interrupt jumper matrix. • Bits 4, 5 — Input or output (both must be in same direction). • Bit 6 — Port E8 Acknowledge (ACK/) input. • Bit 7 — Port E8 Output Buffer Full (OBF/) output.
			E44-59*	E39-59	Connects OBF_A/ output to J2-18.		
			E39-54*	—	Disconnects Port EA, bit 3 (INTR) from driver at U8.		

* Default jumper connected at the factory.

Table 2-6. Configuration Jumpers for Parallel Ports E4 - EA (Continued)

Port	Mode	Driver (D)/ Terminator (T)	Jumper Configuration				
			Remove	Add	Effect	Port	Restrictions
E8	2 (bidirectional)	8287: U6 T: U7 D: U8	E34-E35*	E35-58	Allows ACK_A/ output to control 8287 in/out direction.	E9	None.
			—	E41-56*	Connects J2-26 to STB_A input.	EA	Port EA bits perform the following: • Bit 0 — Can only be used for serial interface. • Bits 1, 2 — Can be used for input or output if Port E9 is in Mode 0. • Bit 3 — Port E8 Interrupt to interrupt jumper matrix. • Bit 4 — Port E8 Strobe (STB/) input. • Bit 5 — Port E8 Input Buffer Full (IBF) output. • Bit 6 — Port E8 Acknowledge (ACK/) input. • Bit 7 — Port E8 Output Buffer Full (OBF/) output.
			E42-57* and E36-51*	E36-57	Connects IBF_A output to J2-24.		
			—	E43-58*	Connects J2-30 to ACK_A/ input.		
			E44-59*	E39-59	Connects OBF_A/ output to J2-18.		
			E39-54*	—	Disconnects Port EA, bit 3 (INTR) from driver at U8.		
E9	0 Input	T: U9, U10	None	None		E8	None
						EA	None; Port EA can be in Mode 0, input or output, if Port E8 is also in Mode 0.
E9	0 Output (latched)	D: U9, U10	None	None		E8	None.
						EA	None; Port EA can be in Mode 0, input or output, if Port E8 is also in Mode 0.
E9	1 Input (strobed)	T: U7, U9, U10 D: U8	—	E37-52*	Connects IBF_B output to J2-22.	E8	None.
			E38-53* and E44-59*	E43-53	Connects J2-32 to STB_B/ input.	EA	Port EA bits perform the following: • Bit 0 — Port E9 Interrupt to interrupt jumper matrix. • Bit 1 — Port E9 Input Buffer Full (IBF) output. • Bit 2 — Port E9 Strobe (STB/) input. • Bit 3 — If Port E8 is in Mode 0, bit 3 can be input or output. Otherwise, bit 3 is reserved. • Bits 4, 5 — Depends on Port E8 mode; see table 3-21. • Bits 6, 7 — Input or output (both must be in same direction). See table 3-21.
			E36-51*	—	Disconnects Port EA, bit 0 (INTR) from driver at U8.		
E9	1 Output (latched)	T: U7 D: U8, U9, U10	—	E37-52*	Connects OBF_B/ output to J2-22.	E8	None.
			E38-53* and U44-59*	E44-53	Connects J2-32 to ACK_B/ input.	EA	Port EA bits perform the following: • Bit 0 — Port E9 Interrupt to interrupt jumper matrix.

* Default jumper connected at the factory.

Table 2-6. Configuration Jumpers for Parallel Ports E4 - EA (Continued)

Port	Mode	Driver (D)/ Terminator (T)	Jumper Configuration				
			Remove	Add	Effect	Port	Restrictions
			E36-51*	—	Disconnects Port EA, bit 0 (INTR) from driver at U8.		Bit 1 — Port E9 Output Buffer Full (OBF/) output. • Bit 2 — Port E9 Acknowledge (ACK/) input. • Bit 3 — If Port E8 is in Mode 0, bit 3 can be input or output. Otherwise, bit 3 is reserved. • Bits 4-7 — Input or output (both must be in same direction); see table 3-21.
EA (upper)	0 Input	T: U7	None	E41-56*	Connects bit 4 to J2-26.	E8	Port E8 must be in Mode 0 for all four bits to be available.
				E42-57*	Connects bit 5 to J2-28.		
				E43-58*	Connects bit 6 to J2-30.	E9	Port E9 must be in Mode 0 for all four bits to be available.
				E44-59*	Connects bit 7 to J2-32.		
EA (lower)	0 Input	T: U8	None	E36-51*	Connects bit 0 to J2-24.	E8	Port E8 must be in Mode 0 for all four bits to be available.
				E37-52*	Connects bit 1 to J2-22.		
				E38-53*	Connects bit 2 to J2-20.	E9	Port E9 must be in Mode 0 for all four bits to be available.
				E39-54*	Connects bit 3 to J2-18.		
EA (upper)	0 Output (latched)	D: U7	None	Same as for Port EA (upper) Mode 0 Input.		E8 & E9	Same as for Port EA (upper) Mode 0 Input.
EA (lower)	0 Output (latched)	D: U8	None	Same as for Port EA (lower) Mode 0 Input.		E8 & E9	Same as for Port EA (lower) Mode 0 Input.

* Default jumper connected at the factory.

Refer to section 3-30 for 8259A PIC programming information, and section 3-37 for 8085A-2 CPU interrupt handling.

2-16. CONNECTOR INFORMATION

For systems applications, the iSBC 80/24A board is designed for installation into a standard Intel iSBC 604/614 Cardcage assembly. For OEM applications the board may be interfaced to other hardware by means of separately purchased connectors. Table 2-8 lists recommended suppliers for each iSBC 80/24A board connector.

2-17. MULTIBUS® SIGNAL CHARACTERISTICS

Connectors P1 and P2 interface your iSBC 80/24A board signals and power lines to the power supply and other boards in your system. Where applicable, these signals conform to the Intel Multibus Interface standard. Pin assignments for P1 and P2 are listed in tables 2-9 and 2-11 respectively. Brief descriptions of these signals are provided in tables 2-10 and 2-12.

AC characteristics, with the board operating at 4.84 MHz are provided in table 2-13. With the board operating at 2.42 MHz, these parameters are shown in table 2-14. A system timing diagram is provided in figure 2-3.

DC characteristics for P1 signals are provided in table 2-15; P2 signal characteristics are shown in table 2-16.

2-18. PARALLEL I/O DC CHARACTERISTICS

Parallel I/O DC characteristics for connectors J1 and J2 are provided in table 2-17.

Table 2-7. Interrupt Matrix Jumper Connections

8259A Priority Level Inputs	Jumper[1] Pin (Column A)	Interrupt Request Line	Jumper[1] Pin (Column B)	Source of Interrupt Request
IR0	E135	Transmitter Empty (TXE) Transmitter Ready (TXR) Receiver Ready (RXR)	E149 E151 E152	Serial I/O Interface
IR1	E134†	Parallel I/O Port 1 (PIA1)	E125	Parallel I/O Interface
IR2	E153†	Parallel I/O Port 2 (PIB1)	E126	
IR3	E132	Parallel I/O Port 4 (PIA2)	E127	
IR4	E131	Parallel I/O Port 5 (PIB2)	E128	
IR5	E130	INT7/	E146	(P1-36)
IR6	E129	INT6/	E145	(P1-35)
		INT5/	E144	(P1-38)
		INT4/	E143	(P1-37)
IR7	E114* or E116 E117 E118 E119	INT3/	E140	Multibus Edge Connector P1 (P1-40)
		INT2/	E133	(P1-39)
		INT1/	E141	(P1-42)
		INT0/	E142	(P1-41)
8085A Inputs		INTR/	E139	(P1-33)
TRAP	E158[2]	Power-Fail Interrupt	E150	External Power-Fail Logic (P2-19)
RST 7.5	E157[2]	Output From Counter 0 (OIT0)	E154	8254 Interval Timer
RST 6.5	E156[2]	Output From Counter 1 (OIT1)	E138	
RST 5.5	E155[2]	Ground	E115 E159 E160 E161 E162	—
		MIOINT 0 (J5) MIOINT 1 (J5) MIOINT 2 (J6) MIOINT 3 (J6)	E120 E121 E122 E123	Multimodule Board(s) J5/J6

[1]Jumper appropriate pins from Column A to those in Column B to connect the desired interrupt.

[2]These interrupts are grounded in default configuration.

†IR1 connected to INT2/ at factory.
IR2 connected to OIT0 at factory.

Table 2-8. User Furnished Connector Details

Function	Pins	Centers (Inches)	Connector Type	Vendor	Vendor Part No.
Multibus Connector P1	43/86	0.156	Solder PCB (no mounting ears)	ELFAB VIKING	BS1562A43PBF 2VT43/2AMK12
			Wire Wrap (no mounting ears)	EDAC ELFAB	337-086-540-201 BW1562A-43PBF
			Wire Wrap (with 0.128 mounting holes)	EDAC ELFAB	337-086-540-202 BW1562A-43PBF
Auxiliary Connector P2	30/60	0.100	Wire Wrap	EDAC ELFAB	345-060-524-202 BW1020A-30PBF
			Solder PCB (with 0.128 mounting holes)	TI VIKING	H421121-30 3VT30/2JNK1
			Wire Wrap (no mounting ears)	EDAC ELFAB	345-060-540-201 BW1020E-30PBF
Parallel Port	25/50	0.100	Flat Crimp	3M 3M ANSLEY SAE	3415-0001 (w/o ears) 3415-0000 (w/ears) 609-5015 (w/o ears) S06750 Series
			Soldered	GTE VIKING	6AA10-25-1A5 3VT25/2JNK5 (0.125 hole)
			Wire Wrap	VIKING TI	3VT25/JND5 (0.125 hole) H421011-25
Serial Port	13/26	0.100	PCB Soldered mounting holes	AMP EDAC	1-583715-1 345-026-520-202
			Flat Crimp	3M	3462-001
			Soldered, pierced tail (no mounting ears)	EDAC	345-026-500-201
			Wire Wrap (no mounting ears)	EDAC	345-026-540-201

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

Table 2-9. Connector P1 Pin Assignments

	Pin	(Component Side)		Pin	(Circuit Side)	
		Mnemonic	Description		Mnemonic	Description
POWER SUPPLIES	1	GND	Signal GND	2	GND	Signal GND
	3	+5V	+5Vdc	4	+5V	+5Vdc
	5	+5V	+5Vdc	6	+5V	+5Vdc
	7	+12V	+12Vdc	8	+12V	+12Vdc
	9	−5V	−5Vdc	10	−5V	−5Vdc
	11	GND	Signal GND	12	GND	Signal GND
BUS CONTROLS	13	BCLK/	Bus Clock	14	INIT/	Initialize
	15	BPRN/	Bus Priority In	16	BPRO/	Bus Priority Out
	17	BUSY/	Bus Busy	18	BREQ/	Bus Request
	19	MRDC/	Mem Read Cmd	20	MWTC/	Mem Write Cmd
	21	IORC/	I/O Read Cmd	22	IOWC/	I/O Write Cmd
	23	XACK/	XFER Acknowledge	24		Reserved
BUS CONTROLS AND ADDRESS	25		Reserved	26		Reserved
	27		Reserved	28	ADR10/	Address 10
	29	CBRQ/	Common Bus Request	30		Reserved
	31	CCLK/	Constant Clock	32		Reserved
	33		Reserved	34		Reserved
INTERRUPTS	35	INT6/	Interrupt Requests	36	INT7/	Interrupt Requests
	37	INT4/		38	INT5/	
	39	INT2/		40	INT3/	
	41	INT0/		42	INT1/	
ADDRESS	43	ADRE/	Address Bus	44	ADRF/	Address Bus
	45	ADRC/		46	ADRD/	
	47	ADRA/		48	ADRB/	
	49	ADR8/		50	ADR9/	
	51	ADR6/		52	ADR7/	
	53	ADR4/		54	ADR5/	
	55	ADR2/		56	ADR3/	
	57	ADR0/		58	ADR1/	
DATA	59		Reserved	60		Reserved
	61		Reserved	62		Reserved
	63		Reserved	64		Reserved
	65		Reserved	66		Reserved
	67	DAT6/	Data Bus	68	DAT7/	Data Bus
	69	DAT4/	Data Bus	70	DAT5/	Data Bus
	71	DAT2/	Data Bus	72	DAT3/	Data Bus
	73	DAT0/	Data Bus	74	DAT1/	Data Bus
POWER SUPPLIES	75	GND	Signal GND	76	GND	Signal GND
	77		Reserved	78		Reserved
	79	−12V	−12Vdc	80	−12V	−12Vdc
	81	+5V	+5Vdc	82	+5V	+5Vdc
	83	+5V	−5Vdc	84	+5V	+5Vdc
	85	GND	Signal GND	86	GND	Signal GND

NOTE: 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 2-10. Connector P1 Signal Definitions

BCLK/	*Bus clock;* used to synchronize bus control circuits on all master modules. BCLK/ has a period of 103.3 nanoseconds (9.677 MHz frequency), 50% duty cycle. BCLK/ may be slowed, stopped or single stepped if desired.
INIT/	*Initialization signal;* resets the entire system to a known internal state.
BPRN/	*Bus priority input signal;* indicates to the iSBC 80/24A board that no higher priority master module is requesting use of the system bus. BPRN/ must be synchronized with BCLK/ in multi-master system; BPRN/ must be connected to signal ground in a single-master system.
BPRO/	*Bus priority out signal;* used with serial (daisy chain) bus priority resolution schemes. BPRO/ is passed to the BPRN/ input of the master module with next lower bus priority.
BREQ/	*Bus request signal;* used with a parallel bus priority network to indicate that a particular master module requires use of the bus for one or more data transfers. BREQ/ is synchronized with BCLK/.
MRDC/	*Memory read command;* indicates that the address of a memory location has been placed on the system address lines and specifies that the contents of the addressed location are to be read and placed on the system data bus.
MWTC/	*Memory write command;* indicates that the address of a memory location has been placed on the system address lines and that a data word has been placed on the system data bus. MWTC/ specifies that the data word is to be written into the addressed memory location.
IORC/	*I/O read command;* indicates that the address of an input port has been placed on the system address bus and that the data at that input port is to be read and placed on the system data bus.
IOWC/	*I/O write command;* indicates that the address of an output port has been placed on the system address bus and that the contents of the system data bus are to be output to the addressed port.
XACK/	*Transfer acknowledge signal;* the required response of an external memory location or I/O port which indicates that the specified read/write operation has been completed. That is, data has been placed on, or accepted from the system data bus lines.
CCLK/	*Constant clock;* provides a clock signal of constant frequency (9.677 MHz) for use by optional memory and I/O expansion boards. CCLK/ has a period of 103.3 nanoseconds, 50% duty cycle.
INT0/ - INT7/	Externally generated *interrupt requests.*
ADR0/ - ADRF/	*16 Address lines;* used to transmit the address of the memory location or I/O port to be accessed. ADRF/ is the most significant bit.
DAT0/ - DAT7/	Bi-directional *data lines;* used to transmit/receive information to/from a memory location or I/O port. DAT7/ is the most significant bit.
CBRQ/	*Common Bus Request;* indicates that a bus master wishes control of the bus but does not presently have control. As soon as control of the bus is obtained, the requesting bus controller raises the CBRQ/ signal.

Table 2-11. Connector P2 Pin Assignments

Pin Assignment	Signal Mnemonic	Description
P2-1, 2 P2-21, 22	Signal GND	Battery Ground
P2-3 P2-4	+5V AUX	Battery +5V Power Input
P2-13	PFSR/	Power Fail Sense Reset
P2-17	PFSN/	Power Fail Sense
P2-19	PFIN/	Power Fail Interrupt
P2-20	MPRO/	Memory Protect
P2-28	HALT/	Halt Indicator
P2-32	ALE	Bus Master ALE
P2-34	BTMO	Bus Timeout
P2-36	INTA/	Interrupt Acknowledge
P2-38	AUX RESET/	System Reset Switch Input

Table 2-12. Connector P2 Signal Definitions

PFIN/	*Power Fail Interrupt.* This input from the power supply interrupts the CPU when a power failure occurs. See section 2-22.
PFSN/	*Power Fail Sense.* This line is the output of a latch which indicates a power failure has occurred. It is reset by PFSR/ and must be powered by the standby power source. See section 2-22.
PFSR/	*Power Fail Sense Reset.* This line is used to reset the power-fail sense latch. See section 2-22.
MPRO/	*Memory Protect.* When true, this externally generated signal prevents access to the on-board RAM during periods of uncertain DC power. See section 2-22.
HALT/	*Halt.* Indicates the 8085A-2 CPU has halted. Typically, this signal is used to drive a front panel HALT indicator.
ALE	*Address Latch Enable.* Indicates the 8085A-2 CPU is operating. Typically, this signal is used to drive a front panel RUN indicator.
BTMO	*Bus Timeout.* This signal is asserted when the on-board failsafe timer provides a READY signal after the specified period. See section 4-25.
INTA/	*Interrupt Acknowledge.* This signal is issued by the CPU in response to an interrupt request.
AUX RESET/	*Auxiliary Reset.* Typically this RESET signal is generated by a front panel switch. The signal is functionally equivalent to INIT/.

Table 2-13. AC Characteristics at 4.84 MHz

Parameter	Overall		Read		Write		Description	Remarks
	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)		
t_{AS}	50		50		50		Address setup time to command	
t_{AH}	50		50		50		Address hold time from command	
t_{DS}	50				50		Data setup to command	
t_{DHW} (t_{DH})	50				50		Data hold time from command	
t_{CY}	206	208					CPU cycle time	
t_{CMDR1}			240				Read command width	With 0 wait state
t_{CMDR1}					240		Write command width	With 0 wait state
t_{CMDR2}			446				Read command width	With 1 wait state
t_{CMDR2}					446		Write command width	With 1 wait state
t_{CSWR}	251						Read-to-write command separation	In override mode
t_{CSRR}	251						Read-to-read command separation	In override mode
t_{CSWW}	251						Write-to-write command separation	In override mode
t_{CSRW}	251						Write-to-read command separation	In override mode
t_{XACKR}			−304				Read command to XACK 1st sample point	In override mode
t_{XACKW}					−304		Write command to XACK 1st sample point	In override mode
t_{SAM}	206	208					Time between XACK samples	In override mode
t_{DHR}			0				Read data hold time	
t_{DXL}			− 97				Read data setup to XACK	
t_{XKH}	0		0	65	0	65	XACK hold time	
t_{CPM} (tppd)		35					Parallel Priority Resolution	
t_{BWS}	35	∞					Bus clock low or high interval	Supplied by system
t_{BS}	23						BPRN to BCLK setup time	
t_{DBY}		55					BCLK to BUSY delay	
t_{NOD} (t_{PNO})		30					BPRN to BPRO delay	
t_{BCY}	102	104					Bus clock period (BCLK)	From iSBC 80/24A when terminated
t_{BW}	35	74					Bus clock low or high interval	From iSBC 80/24A when terminated
t_{INIT} (t_{INT})	10 ms						Initialization width	After all voltages have stabilized
t_{CBRQ}		50					BCLK to CBRQ delay	

NOTE: ( ) indicates former iSBC 80/20 board terminology.

Table 2-14. AC Characteristics at 2.42 MHz

Parameter	Overall		Read		Write		Description	Remarks
	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)	Min. (ns)	Max. (ns)		
t_{AS}	50		50		50		Address setup time to command	
t_{AH}	50		50		50		Address hold time from command	
t_{DS}	50				50		Data setup to command	
t_{DHW} (t_{DH})	50				50		Data hold time from command	
t_{CY}	412	414					CPU cycle time	
t_{CMDR0}			549				Read command width	With 0 wait states
t_{CMDR0}					549		Write command width	With 0 wait states
t_{CMDR1}			963				Read command width	With 1 wait state
t_{CMDR1}					963		Write command width	With 1 wait state
t_{CSWR}	561						Read-to-write command separation	In override mode
t_{CSRR}	561						Read-to-read command separation	In override mode
t_{CSWW}	561						Write-to-write command separation	In override mode
t_{CSRW}	561						Write-to-read command separation	In override mode
t_{XACKR}			−200				Read command to XACK 1st sample point	In override mode
t_{XACKW}					−200		Write command to XACK 1st sample point	In override mode
t_{SAM}	412	414					Time between XACK samples	In override mode
t_{DHR}			0				Read data hold time	
t_{DXL}			−303				Read data setup to XACK	
t_{XKH}	0		0	65	0	65	XACK hold time	
t_{CPM} (tppd)		35					Parallel Priority Resolution	
t_{BWS}	35	∞					Bus clock low or high interval	Supplied by system
t_{BS}	23						BPRN to BCLK setup time	
t_{DBY}		55					BCLK to BUSY delay	
t_{NOD} (t_{PNO})		30					BPRN to BPRO delay	
t_{BCY}	102	104					Bus clock period (BCLK)	From iSBC 80/24A when terminated
t_{BW}	35	74					Bus clock low or high interval	From iSBC 80/24A when terminated
t_{INIT} (t_{INT})	10 ms						Initialization width	After all voltages have stabilized
t_{CBRQ}		50					BCLK to CBRQ delay	

NOTE: ( ) indicates former iSBC 80/20 board terminology.

Table 2-15. DC Characteristics (P1 Signals)

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
ADR0/-ADRF/	V_{OL}	Output Low Voltage	I_{OL} = 24 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = −2.6 mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 2.7V		20	μA
	I_{LL}	Output Leakage Low	V_O = 0.4		−20	μA
	C_L[1]	Capacitive Load			18	pF
MRDC/, MWTC/ IORC/, IOWC/	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH} = −2 mA	2.4		V
	I_{LH}	Output Leakage High	V_O = 5.25V		100	μA
	I_{LL}	Output Leakage Low	V_O = 0.45V		100	μA
	C_L[1]	Capacitive Load			15	pF
DAT0/-DAT7/	V_{OL}	Output Low Voltage	I_{OL} = 23 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = −2.55 mA	2.4		V
	V_{IL}	Input Low Voltage			0.80	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		−0.1	mA
	I_{LH}	Output Leakage High	V_O = 2.7V		70	μA
	I_{LL}	Output Leakage Low	V_O = 0.4V		−0.22	mA
	C_L[1]	Capacitive Load			25	pF
XACK/, INTR/	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		−1.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		40	μA
	C_L[1]	Capacitive Load			18	pF
CCLK/	V_{OL}	Output Low Voltage	I_{OL} = 60 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = −3 mA	2.7		V
	C_L[1]	Capacitive Load			15	pF
BPRO/	V_{OL}	Output Low Voltage	I_{OL} = 5.0 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH} = −0.4 mA	2.4		V
	C_L[1]	Capacitive Load			10	pF
BREQ/	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.45	V
	V_{OH}	Output High Voltage	I_{OH} = −0.4 mA	2.4		V
	C_L[1]	Capacitive Load			10	pF
CBRQ/ (O.C.)	V_{OL}	Output Low Voltage	I_{OL} = 60 mA		0.5	V
	C_L[1]	Capacitive Load			15	pF

[1] Capacitive Load values are approximations.

Table 2-15. DC Characteristics (P1 Signals) (Continued)

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
INT0/-INT7/	V_{IL}	Input Low Voltage			.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		−.2	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		30	μA
	C_L[1]	Capacitive Load			18	pF
BPRN/	V_{IL}	Input Low Voltage			.8	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		−0.5	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		100	μA
	C_L[1]	Capacitive Load			15	pF
AACK/	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		−2.0	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		−1.0	mA
	C_L[1]	Capacitive Load			18	pF
BUSY/ (OPEN COLLECTOR)	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		0.5	V
	C_L[1]	Capacitive Load			20	pF
INIT/ (SYSTEM RESET)	V_{OL}	Output Low Voltage	I_{OL} = 56 mA		0.5	V
	V_{OH}	Output High Voltage	OPEN COLLECTOR			
	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} = .4V		−3.0	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		−1.0	mA
	C_L[1]	Capacitive Load			18	pF
BCLK/	V_{OL}	Output Low Voltage	I_{OL} = 59.5 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = −3 mA	2.7		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} = 0.45V		−0.5	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		100	μA
	C_L[1]	Capacitive Load			15	pF

[1] Capacitive Load values are approximations.

Table 2-16. DC Characteristics (P2 Signals)

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
PFSN/	V_{IL}	Input Low Voltage			0.8	V
PFIN/	V_{IH}	Input High Voltage		2.4		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.4V		−1.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 2.4V		40	μA
	C_L[1]	Capacitive Load			10	pF
AUX RESET/	V_{IL}	Input Low Voltage			0.8	V
	V_{IH}	Input High Voltage		2.6		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		−.25	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		10	μA
	C_L[1]	Capacitive Load			10	pF
BTMO	V_{OL}	Output Low Voltage	I_{OL} = 16 mA		0.4	V
	V_{OH}	Output High Voltage	I_{OH} = −400 μA	2.4		V
	C_L[1]	Capacitive Load			20	pF
	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.5	V
ALE	V_{OH}	Output High Voltage	I_{OH} = −1 mA	2.7		V
HALT/	C_L[1]	Capacitive Load			20	pF
MPRO/	V_{IL}	Input Low Voltage			0.80	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45V		−.6	mA
	I_{IH}	Input Current at High V	V_{IN} = 5.25V		100	μA
	C_L[1]	Capacitive Load			60	pF
INTA/	V_{OL}	Output Low Voltage	I_{OL} = 20 mA		0.5	V
	V_{OH}	Output High Voltage	I_{OH} = −1 mA	2.7		V
	C_L[1]	Capacitive Load			20	pF

[1] Capacitive Load values are approximations.

Table 2-17. DC Characteristics (J1 & J2 Parallel I/O Ports)

Signals	Symbol	Parameter Description	Test Conditions	Min.	Max.	Units
PORTS E4 AND E8	V_{OL}	Output Low Voltage	I_{OL} = 32 mA		.5	V
BIDIRECTIONAL	V_{OH}	Output High Voltage	I_{OH} = −5 mA	2.4		V
DRIVERS	V_{IL}	Input Low Voltage			.90	V
	V_{IH}	Input High Voltage		2.0		V
	I_{IL}	Input Current at Low V	V_{IN} = 0.45		−.2	mA
	C_L[1]	Capacitive Load			18	pF
8255A	V_{OL}	Output Low Voltage	I_{OL} = 1.7 mA		.45	V
DRIVER/RECEIVER	V_{OH}	Output High Voltage	I_{OH} = −200 μA	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[1]	Capacitive Load			10	pF

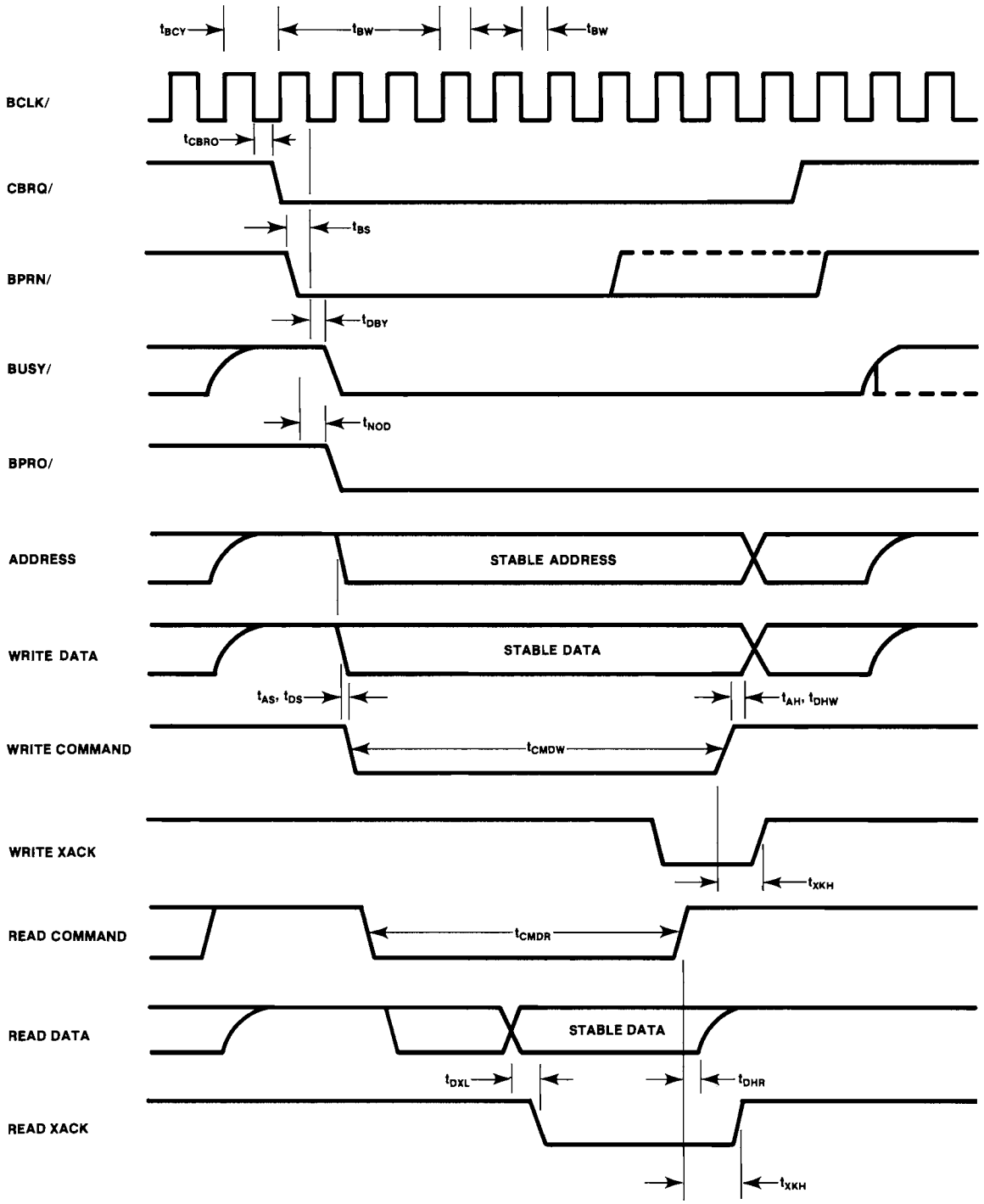


Figure 2-3. System Timing Diagram

2-19. BOARD PRIORITY RESOLUTION

Your iSBC 80/24A board has been designated as a "Full Master" board. This means the board is equipped with bus-arbitration logic and can acquire and relinquish control of the common system Multibus lines (see section 4-33). In order for this system to be effective, a board priority scheme needs to be established in your system.

If your iSBC 80/24A board is the only master in the system, it should be placed in the top slot (J2) of the iSBC 604/614 Cardcage. Normally, pin 15 (BPRN/) of the iSBC 604/614 backplane connector J2 is grounded with jumper B-N. However, this connection should be verified. This will allow the other three slots to be used for any expansion boards or for future master boards.

When using the iSBC 80/24A board in a multimaster system, a priority resolution scheme *must* be established. Two methods of priority are possible: a serial (daisy chain) method; or a parallel method.

Another important consideration in setting up a multi-master system is the Multibus clock signal source. You must ensure only one master board is supplying the BCLK/ & CCLK/ signals to the Multibus lines. All master boards have provisions for disabling the output of these two signals (i.e., preventing the signals from going off-board). Refer to the schematic diagrams of the particular master board, in your hardware reference manual.

2-20. SERIAL PRIORITY

Serial priority is implemented by board placement. If your iSBC board resides in slot J2, as previously described, the next lower priority will be assigned to slot J3. Slot J4 will have the lowest priority in this scheme. Due to the propagation delay of the BPRO/ signal path, this scheme is limited to a maximum of three master boards. In the configuration shown in figure 2-4, the master board installed in slot J2 has the highest priority and is able to acquire control of the Multibus lines any time the bus is not busy, since the BPRN/ input is always true (tied to ground via jumper connections B-N on the iSBC 604/614 Backplane).

If the master board in slot J2 desires control of the Multibus lines, it drives its BPRO/ output high (false)

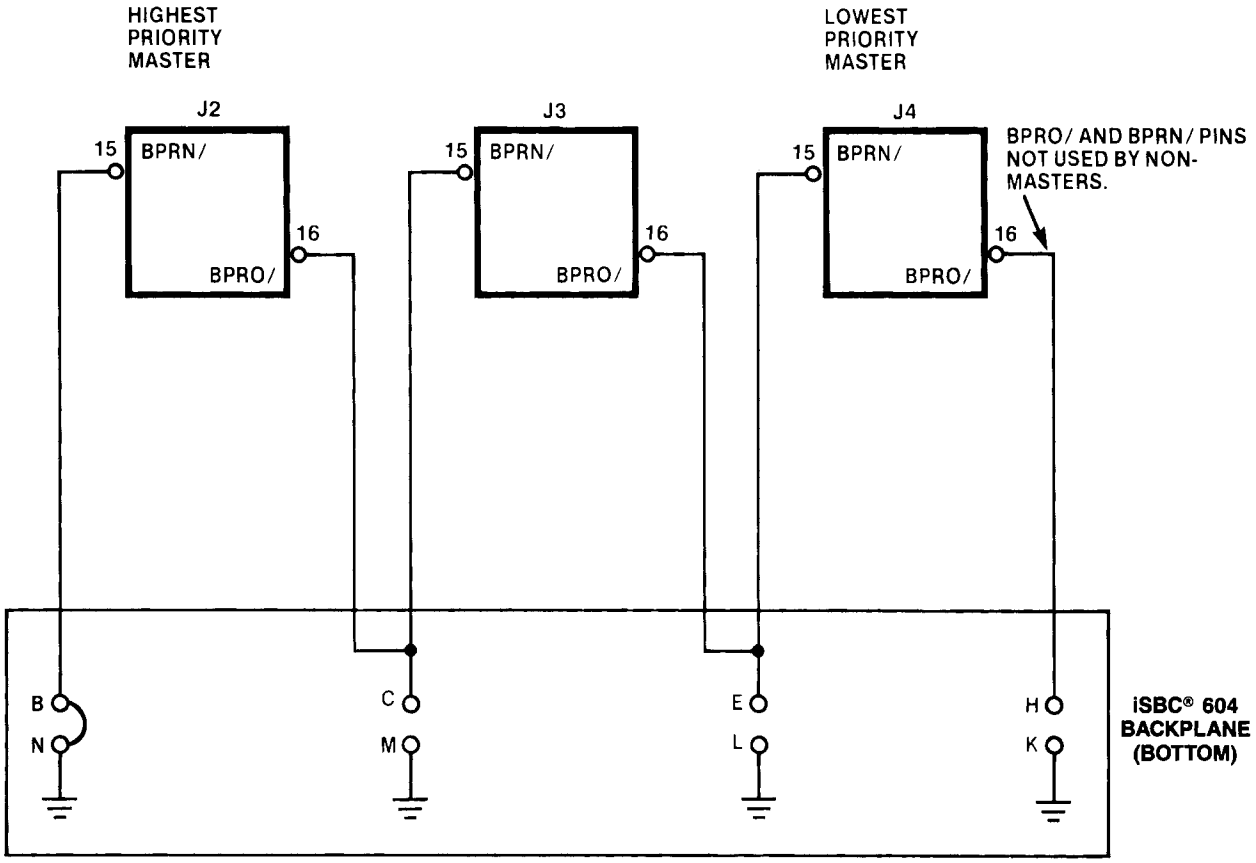


Figure 2-4. Serial Priority Resolution Scheme

and inhibits the BPRN/ inputs to the remaining lower priority master boards. The master board then takes control of the Multibus lines when the current bus cycle is completed. When finished using the Multibus lines, the J2 master board pulls its BPRO/ output low (true) and gives the J3 master board the opportunity to acquire Multibus line control. If the J3 master board does not want the Multibus lines, it pulls its BPRO/ output low (true) and gives the J4 master board the opportunity to assume control of the Multibus lines.

To implement the serial priority scheme in your own custom chassis, wire the BPRO/ and BPRN/ signals as shown in figure 2-4. Slots between masters may be left empty by installing a jumper between J1-15 and J1-16 on the empty slot.

2-21. PARALLEL PRIORITY RESOLUTION

A parallel priority resolution scheme allows up to 16 bus masters to acquire and control the Multibus lines. Figure 2-5 illustrates one method of implementing such a scheme for resolving bus contention in a system containing eight bus masters installed in two iSBC 604/614 Cardcages. Notice that the two highest and two lowest priority bus masters are shown installed in the iSBC 604 Cardcage.

In the scheme shown in figure 2-5, the priority encoder is a Texas Instruments 74148 and the priority decoder is an Intel 8205. Input connections to the priority encoder determine the bus priority, with input 7 having the highest priority and input 0 having the lowest priority. Here, the J3 bus master has the highest priority and the J5 bus master has the lowest priority.

IMPORTANT: In a parallel priority resolution scheme, the BPRO/ output must be disabled on all bus masters. On the iSBC 80/24A board, disable the BPRO/ output signals by removing jumper E200-201. If a similar jumper cannot be removed on the other master boards, either clip the IC pin that supplies the BPRO/ output signal to the Multibus connector or cut the signal trace.

2-22. POWER-FAIL/BATTERY BACKUP PROVISIONS

The iSBC 80/24A board has provisions for interrupting the CPU in the event of a system power

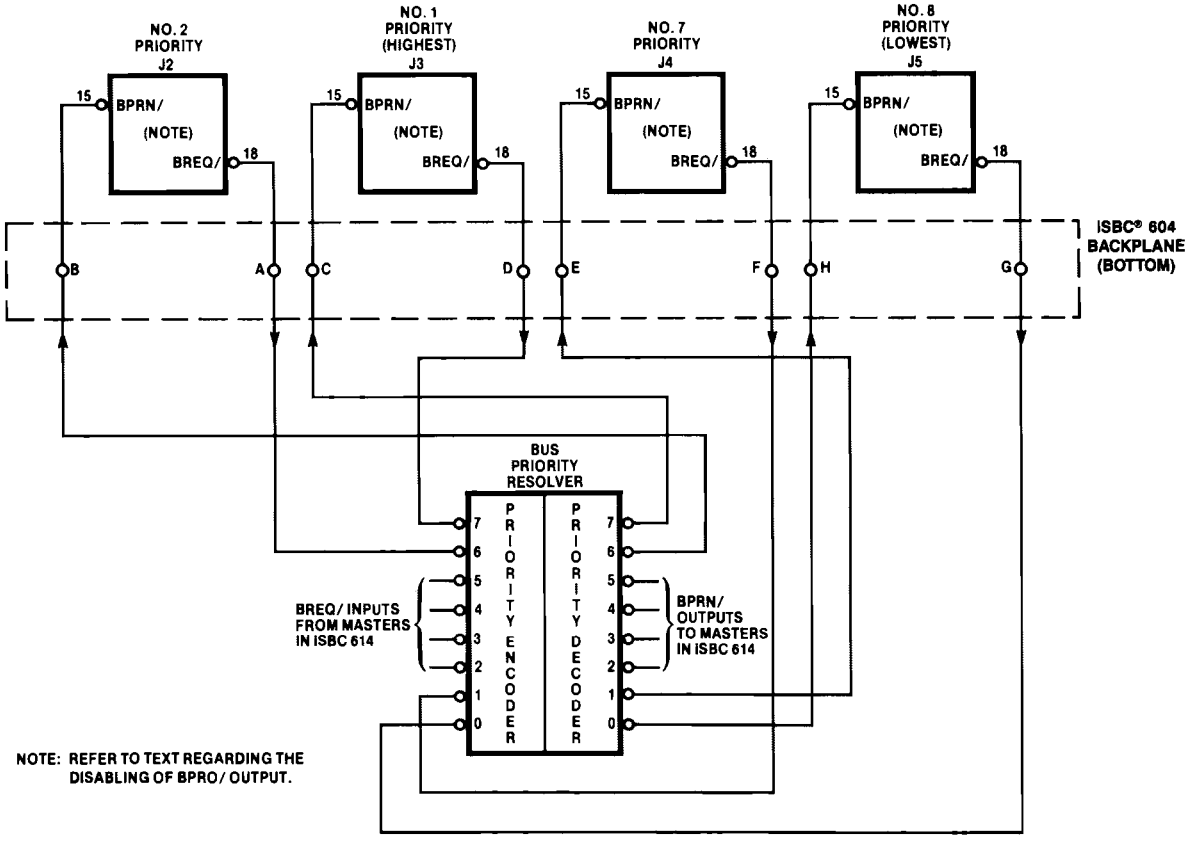


Figure 2-5. Parallel Priority Resolution Scheme

failure, maintaining power-fail status, and providing user-supplied battery backup over the P2 connector for the RAM in the event of a drop in power.

The Power-Fail Interrupt signal (PFIN/) is received at connector pin P2-19 and clocks the Power-Fail Interrupt latch at U15. The output from this latch can be connected to the PIC, via the interrupt jumper matrix. When an output instruction to port D4 is executed, the Power-Fail Interrupt latch is reset.

A Power-Fail Status signal (PFSN/) is received at connector pin P2-17. When an input instruction to port D4 is executed, the level on PFSN/ is input on the least significant data bus line (DB0). The Power-Fail Status signal is generated by your own off-board power-fail circuitry. This alows your main program to differentiate between power coming back up from a power failure or power being initially turned on. In other words, the program can tell whether to restart to where it left off or to initialize the system.

Connector pins P2-3 and 4 can be connected to a +5 Vdc back-up battery that will provide power to your iSBC 80/24A RAM in the event of a power failure. Refer to table 1-1 for battery current requirements. Jumper pair E216-217 must be disconnected when the battery back-up feature is used. Refer to figures 5-3 and 5-4, sheet 1.

The Memory Protect (MPRO/) signal is received at connector pin P2-20 and is ANDed with PROM ACK/ through U38 and then applied to the CE2 input (pin 26) of RAM device U51. This signal is used to prevent she spurious signals, which may be generatd on the memory command lines during power going down, from affecting the contents of memory. When MPRO/ is true (low; 0 volts) the RAM devices will be powered down and disabled.

The PFSR/ signal (Power-Fail Sense Reset) is used to reset the ACLO flip-flop, which is an external circuit used to trigger the PFIN/ (power-fail interrupt) signal. Typically, the ACLO flip-flop is triggered when AC line voltage falls below 90% of its nominal value. Refer to table 2-4 for instructions on implementing the PFSR/ jumper, which is enabled by parallel port circuity.

The following is a typical example of how the various power-fail signals could be used:

a. A power failure occurs. When the AC line voltage reaches some predetermined level (−10% typically) the power-fail circuitry generates a Power-Fail Interrupt (PFIN/) signal.

b. The CPU is interrupted and goes into a program status saving routine. The routine should end with a HALT instruction.

c. After a user-determined length of time (5 ms for example) the power-fail circuitry should generate Memory Protect (MPRO/) signal.

d. DC power goes down.

e. AC power returns. The power-fail circuitry should generate a SYSTEM RESET signal and hold it until the power supplies are stable. Reset should also be held until the MPRO/ signal is cleared.

f. The system program can now read the Power-Fail Status signal and determine what type of restart procedure is to be executed.

For additional information on power-fail schemes for Intel Single Board Computers, refer to the *Intel Multibus Specification.*

2-23. PARALLEL I/O CABLING

Parallel I/O ports E4, E5 and E6 are controlled by the PPI device in socket U15, and are interfaced via edge connector J1. Similarly, parallel I/O ports E8, E9 and EA are controlled by the PPI device in socket U17 and are interfaced via edge connector J2.

Pin assignments for J1 and J2, in the factory default configuration, are listed in tables 2-18 and 2-19 respectively. Bit order for ports E6 and EA may be altered by jumper connection. Refer to section 2-14 for instructions. DC characteristics of the parallel I/O signals are listed in table 2-17. Connector information for J1 and J2 is provided in table 2-9.

The transmission path from the I/O source to the iSBC 80/24A board should be limited to 3 meters (10 feet).

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 footnotes in tables 2-18 and 2-19.)

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

a. 50-conductor flat cable, 3M 3365-50.

b. 50-conductor flat cable (with ground plane), 3M 3469-50

c. 25-pair flat-twisted cable, Belden 9V28050.

Table 2-18. Connector J1 Pin Assignments

Pin	Signal	Pin	Signal
1	GND	2	PORT E5 - BIT 7
3	GND	4	PORT E5 - BIT 6
5	GND	6	PORT E5 - BIT 5
7	GND	8	PORT E5 - BIT 4
9	GND	10	PORT E5 - BIT 3
11	GND	12	PORT E5 - BIT 2
13	GND	14	PORT E5 - BIT 1
15	GND	16	PORT E5 - BIT 0
17	GND	18	PORT E6 - BIT 3
19	GND	20	PORT E6 - BIT 2
21	GND	22	PORT E6 - BIT 1
23	GND	24	PORT E6 - BIT 0
25	GND	26	PORT E6 - BIT 4
27	GND	28	PORT E6 - BIT 5
29	GND	30	PORT E6 - BIT 6
31	GND	32	PORT E6 - BIT 7
33	GND	34	PORT E4 - BIT 7
35	GND	36	PORT E4 - BIT 6
37	GND	38	PORT E4 - BIT 5
39	GND	40	PORT E4 - BIT 4
41	GND	42	PORT E4 - BIT 3
43	GND	44	PORT E4 - BIT 2
45	GND	46	PORT E4 - BIT 1
47	GND	48	PORT E4 - BIT 0
49	GND	50	EXTERNAL INTERRUPT OR + 5V OR OPEN

NOTES:
1. All odd-numbered pins are on the component side of the board. Pin 1 is rightmost when board is viewed with I/O connectors on top.
2. Cable connector numbering convention may not agree with board edge connector numbering.

Table 2-19. Connector J2 Pin Assignments

Pin	Signal	Pin	Signal
1	GND	2	PORT E9 - BIT 7
3	GND	4	PORT E9 - BIT 6
5	GND	6	PORT E9 - BIT 5
7	GND	8	PORT E9 - BIT 4
9	GND	10	PORT E9 - BIT 3
11	GND	12	PORT E9 - BIT 2
13	GND	14	PORT E9 - BIT 1
15	GND	16	PORT E9 - BIT 0
17	GND	18	PORT EA - BIT 3
19	GND	20	PORT EA - BIT 2
21	GND	22	PORT EA - BIT 1
23	GND	24	PORT EA - BIT 0
25	GND	26	PORT EA - BIT 4
27	GND	28	PORT EA - BIT 5
29	GND	30	PORT EA - BIT 6
31	GND	32	PORT EA - BIT 7
33	GND	34	PORT E8 - BIT 7
35	GND	36	PORT E8 - BIT 6
37	GND	38	PORT E8 - BIT 5
39	GND	40	PORT E8 - BIT 4
41	GND	42	PORT E8 - BIT 3
43	GND	44	PORT E8 - BIT 2
45	GND	46	PORT E8 - BIT 1
47	GND	48	PORT E8 - BIT 0
49	GND	50	EXTERNAL INTERRUPT OR + 5V OR OPEN

NOTES:
1. All odd-numbered pins are on the component side of the board. Pin 1 is rightmost when board is viewed with I/O connectors on top.
2. Cable connector numbering convention may not agree with board edge connector numbering.

2-24. SERIAL I/O CABLING

Pin assignments and signal names for connector J3 (serial I/O interface) are listed in table 2-20. An Intel iSBC 955 Cable Set is recommended for RS232C interfacing. One cable assembly (in the set) 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 other 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.

For OEM applications where cables will be made for the iSBC 80/24A 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 2-21 provides pin correspondence between connector J3 and an 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 2-20.)

If your application requires a 20 mA current loop (TTY) interface refer to section 2-25.

Table 2-20. Connector J3 Pin Assignments

Pin	Signal	Pin	Signal
1	SECONDARY TRANSMITTED DATA	2	PROTECTIVE GROUND
3	TRANSMITTER SIGNAL ELEMENT TIMING	4	TRANSMITTED DATA
5	SECONDARY RECEIVED DATA	6	RECEIVED DATA
7	RECEIVER SIGNAL ELEMENT TIMING	8	REQUEST TO SEND
9	NOT USED BY RS232	10	CLEAR TO SEND
11	SECONDARY REQUEST TO SEND	12	DATA SET READY
13	DATA TERMINAL READY	14	SIGNAL GROUND
15	NOT USED BY RS232	16	RECEIVED LINE SIGNAL DETECT
17	RING INDICATOR	18	NOT USED BY RS232
19	−12V (TTY ADAPTER POWER)	20	NOT USED BY RS232
21	TRANSMITTER SIGNAL ELEMENT TIMING	22	+12V (TTY ADAPTER POWER)
23	+5V	24	NOT USED BY RS232
25	SIGNAL GROUND	26	SECONDARY CLEAR TO SEND

NOTES:
1. All odd-numbered pins are on the component side of the board. Pin 1 is rightmost when board is viewed with I/O connectors on top.
2. Cable connector numbering convention may not agree with board connector numbering.

Table 2-21. Connector Pin Correspondence (25-Pin to 26-Pin)

Signal Name Pin	J3	25-Pin Connector
Protective Ground	2	1
Secondary Transmitted Data	1	14
Transmitted Data	4	2
Transmitter Signal Element Timing	3	15
Received Data	6	3
Secondary Received Data	5	16
Request to Send	8	4
Receiver Signal Element Timing	7	17
Clear to Send	10	5
(No connection)	9	18
Data Set Ready	12	6
Secondary Request to Send	11	19
Signal Ground	14	7
Data Terminal Ready	13	20
Received Line Signal Detect	16	8
(No connection)	15	21
(No connection)	18	9
Ring Indicator	17	22
(No connection)	20	10
TTY Adapter Power (−12V)	19	23
TTY Adapter Power (+12V)	22	11
Transmitter Signal Element Timing	21	24
(No connection)	24	12
(+5V)	23	25
Secondary Clear to Send	26	13
Signal Ground	25	No Connection

2-25. CURRENT LOOP (TTY) INTERFACE

To adapt your iSBC 80/24A board to a 20 mA current loop (TTY) interface, optional jumper connections are required. In addition, the iSBC 530 Teletypewriter Adaptor module must be installed between the board and the serial peripheral. The necessary jumper connection are as follows:

Install jumper E75-80 (provides −12V to module)
Install jumper E78-79 (provides +12V to module)

CAUTION

Current limiting is not provided for these outputs. If improperly connected, damage to the board and power supply could result.

Refer to figure 5-3 for jumper post locations, and figure 5-4, sheet 7 for schematic drawing. Connector J3 pin assignments are listed in table 2-21.

The optional TTY interface module (iSBC 530 Teletypewriter Adapter) converts iSBC 80/24A RS232C signal levels to an optically isolated 20 mA current loop interface and provides signal translation for transmitted data, received data and a teletypewriter paper tape reader relay.

2-26. MULTIMODULE™ BOARDS AND THE iSBX™ BUS

The iSBC 80/24A board is equipped with two iSBX (single board expansion) bus connectors (J5 and J6). This bus allows on-board I/O expansion, using optional iSBX Multimodule boards. Connectors J5 and J6 may be used *only* for iSBX Multimodule boards. Table 2-22 provides the iSBX bus connector pin assignments, and table 2-22A provides iSBX signal descriptions. Each of the two connectors has identical pin assignments, and physical layout.

For installation instructions, refer to the specific iSBX Multimodule board hardware reference manual. When a Multimodule board is installed, the iSBC 80/24A board's power requirement will increase by the amount specified in the Multimodule board manual.

2-27. FINAL INSTALLATION

CAUTION

Always turn off the computer system power supply before installing or removing the iSBC 80/24A 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 80/24A board in any slot that has not been wired for a dedicated function. In an Intellec 800 System, install the iSBC 80/24A board in any slot except slot 1 or 2 (slots 1 and 2 are reserved). Ensure auxiliary edge connector P2 (if used) is correctly inserted. Attach the appropriate cable assembles to connectors J1 through J3, and any Multimodule board connectors.

Table 2-22. iSBX™ Bus Connector Pin Assignments

Pin	Mnemonic	Description	Pin	Mnemonic	Description
35	GND	SIGNAL GROUND	36	+5V	+5 Volts
33	MD0	MDATA BIT 0	34	—	RESERVED
31	MD1	MDATA BIT 1	32	—	RESERVED
29	MD2	MDATA BIT 2	30	OPT0	OPTION 0
27	MD3	MDATA BIT 3	28	OPT1	OPTION 1
25	MD4	MDATA BIT 4	26	—	RESERVED
23	MD5	MDATA BIT 5	24	—	RESERVED
21	MD6	MDATA BIT 6	22	MCS0/	M CHIP SELECT 0
19	MD7	MDATA BIT 7	20	MCS1/	M CHIP SELECT 1
17	GND	SIGNAL GROUND	18	+5V	+5 Volts
15	IORD/	IO READ COMMAND	16	MWAIT/	M WAIT
13	IOWRT/	IO WRITE COMMAND	14	MINTR0	M INTERRUPT 0
11	MA0	M ADDRESS 0	12	MINTR1	M INTERRUPT 1
9	MA1	M ADDRESS 1	10	—	RESERVED
7	MA2	M ADDRESS 2	8	MPST/	M PRESENT
5	MRESET	M RESET	6	MCLK	M CLOCK
3	GND	SIGNAL GROUND	4	+5V	+5 Volts
1	+12V	+12 Volts	2	−12V	−12 Volts

Table 2-22A. iSBX™ Bus Signal Descriptions

IORD/	Commands the Multimodule board to perform the read operation.
IOWRT/	Commands the Multimodule board to perform the write operation.
MRESET/	Initializes the Multimodule board to a known internal state.
MCS0/	Chip select. Selects I/O addresses C0-C7 on the J5 Multimodule board and addresses F0-F7 on the J6 Multimodule board.
MCS1/	Chip select. Selects I/O addresses C8-CF on the J5 Multimodule board and addresses F8-FF on the J6 Multimodule board.
MA0-2	Least three bits of the I/O address. Used in conjunction with the chip select and command lines.
MPST/	Multimodule present indicator. Informs iSBC 80/24A board that a Multimodule board(s) is installed.
MINTR0-1	Interrupt request lines from the Multimodule board to the iSBC 80/24A board interrupt matrix.
MWAIT/	Causes iSBC 80/24A board to execute wait states until Multimodule board is ready to respond.
MCLK/	9.68 MHz Multimodule board timing reference from iSBC 80/24A board.
OPT0-1	Optional use lines. May be used for additional interrupt request lines.
MD0-7	Bidirectional data lines.

* * *

CHAPTER 3
PROGRAMMING INFORMATION

3-1. INTRODUCTION

Several Intel programmable devices reside on the iSBC 80/24A board. This chapter provides programming information for these devices and gives typical examples for most applications. Memory and I/O addressing are provided in table form for quick reference.

3-2. MEMORY ADDRESSING

The iSBC 80/24A board can accommodate up to 32K bytes of ROM/EPROM on-board. Four sockets are provided for these devices. Table 3-1 lists the address ranges for the type of ROM/EPROM installed (1K, 2K, 4K, or 8K). The factory configuration for 2764-type devices (8K bytes each) indicates an on-board ROM/EPROM address range from 0000-7FFF (hexadecimal).

In the factory configuration, 8K bytes of RAM reside on-board. RAM addressing in this configuration is assigned from E000 to FFFF (hexadecimal). RAM addressing in optional configurations will depend on the amount of ROM/EPROM installed on-board. Table 3-2 summarizes these RAM address options.

Table 3-1. EPROM/ROM Memory Size

Device Type	Quantity of EPROMs/ROMs Installed[1]				Address Range (Hex)
	1	2	3	4	
2708, 2758 2608	1K	2K	3K	4K	0000-0FFF
2716 2316E	2K	4K	6K	8K	0000-1FFF
2732, 2732A	4K	8K	12K	16K	0000-3FFF
2764, 2764A	8K	16K	24K	32K	0000-7FFF

[1]EPROM/ROM types cannot be mixed.

If non-existent 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 failsafe timer jumper information refer to section 4-35, and table 2-4.

Table 3-2. On-Board RAM Address Space (Hexadecimal)

Max. RAM Jumper	2K	4K	8K
E112-113	3800-3FFF	3000-3FFF	2000-3FFF
E111-112	7800-7FFF	7000-7FFF	6000-7FFF
E103-104	B800-BFFF	B000-BFFF	A000-BFFF
E102-103*	F800-FFFF	F000-FFFF	E000-FFFF*

*Default configuration.

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 8085A-2 CPU communicates with the programmable devices through a sequence of I/O read and I/O write commands. Each device has a specific, fixed (dedicated) address, or group of addresses, through which commands and data are issued and accepted. All of these fixed on-board I/O addresses are listed in table 3-3. In addition to the board's programmable I/O devices, certain other functions have specific addresses assigned to them. These addresses are also included in the table.

3-4 SYSTEM INITIALIZATION

When power is initially applied to the system, the initialize (INIT/) signal is automatically generated by the 8224 Clock Generator. This clears the internal program counter, instruction register, and the interrupt enable flip-flop circuit of the CPU. The INIT/ signal also resets the 8251A Programmable Communication Interface (PCI) device and the two 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.

Table 3-3. I/O Port Addressing

I/O Port Address (hexadecimal)	I/O Device	Input Function	Output Function
D4	Power Fail	Read Power Fail Status	Reset Power Fail Latch
D5	System Bus Override	Not used	Set or reset override control (using data bus line 0).
D6	LED Diagnostic Indicator	Not used	Flash LED for 1 millisecond
D7	Reserved	—	—
D8 or DA D9 or DB	8259A Interrupt Controller	Various Read Functions Refer to section 3-36	Various Write Functions Refer to section 3-36
DC DD DE DF	8254 Interval Timer	Read counter 0 Read counter 1 Read counter 2 (Baud Rate Generator) Read Status	Load counter 0 Load counter 1 Load counter 2 Mode control
E4 E5 E6 E7	8255A at U15 J1	Read port A Read port B Read port C —	Write port A Write port B Write port C Control
E8 E9 EA EB	8255A at U17 J2	Read port A Read port B Read port C —	Write port A Write port B Write port C Control
EC or EE ED or EF	8251A J3	Data in Read Status	Data out Control
C0-C7 C8-CF F0-F7 F8-FF	Multimodule 1 MCS0/ } J5 Multimodule 1 MCS1/ } Multimodule 2 MCS0/ } J6 Multimodule 2 MCS1/ }	Various Read Functions	Various Write Functions
Note: When either Multimodule is present, the I/O ports C0-FF are dedicated.			

b. All three ports of each 8255A PPI device are set to Mode 0, input.

The INIT/ signal is also gated to other boards in the system, via the Multibus lines, to set all components to a known internal condition. The INIT/ signal can also be generated by an auxiliary reset switch, such as on the iSBC 655 System Chassis front panel. Pressing and releasing the switch produces the same effect previously described.

3-5. 8254 PROGRAMMABLE INTERVAL TIMER (PIT) PROGRAMMING

Each of the three counters (Counter 0, Counter 1, and Counter 2) can use either the on-board 2.15 MHz oscillator, or an external clock input (D.C. to 10 MHz) from Port E6. The factory set jumpers select the on-board 2.15 MHz reference; refer to table for these jumper settings, and the jumpers required to select the external references.

Default jumpers connect the O_2 output of the counter to the on-board PCI device. The other two outputs are routed to the interrupt matrix. The O_1 output is also routed to the port E6 jumper matrix. Refer to table 2-5.

3-6. MODE CONTROL WORD AND COUNT

All three counters must be initialized separately prior to their use. The initialization for each counter consists of two steps:

a. A mode control word (figure 3-1) is written to the control register for each individual counter.

b. A down-count number is loaded into each counter. One or two bytes must be sent to the PIT, as determined by the mode control word.

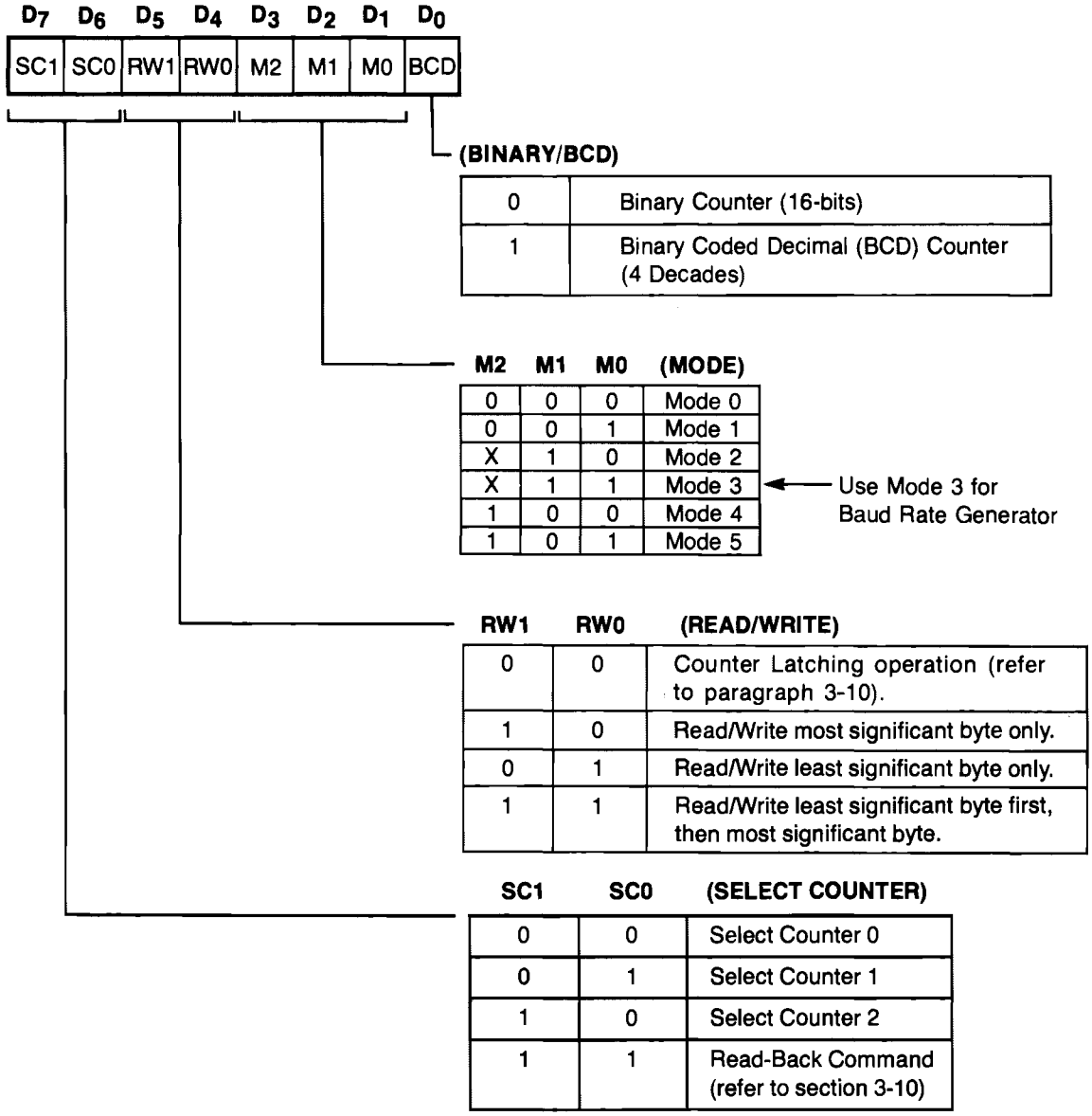


Figure 3-1. PIT Mode Control Word Format

The mode control word (figure 3-1) does the following:

a. Selects counter to be loaded.

b. Selects counter operating mode.

c. Selects one of the following four counter read/load functions:

   (1) Counter latch (for stable read operation).

   (2) Read or load most-significant byte only.

   (3) Read or load least-significant byte only.

   (4) Read or load least-significant byte first, then most-significant byte.

d. Sets counter for either binary or BCD count.

Before you can use a counter, it must be programmed for mode and count. The mode can be programmed at any time with the Mode Control Word (figure 3-1). This word specifies which counters are selected, what their mode is, and the type of count byte which will subsequently be sent.

Notice that the Read/Write sequence specified by the Mode Control Word with the RW bits (figure 3-1) *must* be followed when these bytes are sent to the PIT. These bytes do not necessarily have to follow the associated Mode Control Word. For example, if you select RW1 = 0 and RW0 = 1, indicating Read/Write *least* significant byte only, your desired

count must be placed in the least significant count byte. This is especially important when using counts which require two bytes.

If this procedure is followed for each counter, the PIT can be programmed in any convenient sequence. For example, Mode control words for each counter can be loaded first followed by the count bytes. Figure 3-2 illustrates a typical PIT programming sequence.

Since all PIT counters are downcounters, the values loaded into the count registers are decremented. Loading all zeros into a count register results in the maximum count possible: 2^{16} for binary numbers and 10^4 for BCD numbers.

The count mode selected in the control word controls the counter output. As shown in figure 3-1, the PIT device can operate in any of six modes:

a. Mode 0: Interrupt on terminal count. In this mode, Counters 1 and 2 can be used for auxiliary functions such as generating real-time interrupt intervals. After the count value is loaded into the count register, the counter output goes low and remains low until the terminal count is reached. The output then goes high until either the count register or the mode control register is reloaded.

b. Mode 1: Programmable one-shot. In this mode, the output of Counter 1 and/or Counter 2 will go low on the count following the rising edge of the GATE input from Port E6. The output will go high on terminal count. If a new count value is loaded while the output is low, it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse. The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

c. Mode 2: Rate generator. In this mode, the output of Counter 1 and/or Counter 2 will be low for one period of the clock input. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses, the present period will not be affected but the subsequent period will reflect the new value. The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter. When Mode 2 is set, the output will remain high until after the count register is loaded; thus, the count can be synchronized by software.

PROGRAMMING FORMAT

Step		
1		Mode Control Word Counter n
2	LSB	Count Register Byte Counter n
3	MSB	Count Register Byte Counter n

ALTERNATE PROGRAMMING FORMAT

Step		
1		Mode Control Word Counter 0
2		Mode Control Word Counter 1
3		Mode Control Word Counter 2
4	LSB	Counter Register Byte Counter 1
5	MSB	Count Register Byte Counter 1
6	LSB	Count Register Byte Counter 2
7	MSB	Count Register Byte Counter 2
8	LSB	Count Register Byte Counter 0
9	MSB	Count Register Byte Counter 0

Figure 3-2. PIT Programming Sequence Examples

d. Mode 3: Square wave generator. Mode 3 is the primary operating mode for all three counters on the 8254. In this mode, the counter output remains high until one half of the count value in the count register has been decremented (for even numbers). The output then goes low for the other half of the count. If the value in the count register is odd, the counter output is high for (N + 1)/2 counts, and low for (N – 1)/2 counts.

e. Mode 4: Software triggered strobe. After this mode is set, the output will be high. When the count is loaded, the counter begins counting. On terminal count, the output will go low for one input clock period and then go high again. If the count register is reloaded between output pulses, the present count will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is low. Reloading the count register will restart the counting for the new value.

f. Mode 5: Hardware triggered strobe. The counter will start counting on the rising edge of the gate input and the output will go low for one clock period when the terminal count is reached. The counter is retriggerable; the output will not go low until the full count after the rising edge of the gate input.

Table 3-4 provides a summary of the counter operation versus the gate inputs.

Table 3-4. PIT Counter Operation Vs. Gate Inputs

Modes \ Signal Status	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output high immediately	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output high immediately	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

3-7. ADDRESSING

As listed in Table 3-5 the PIT uses four consecutive I/O addresses: DC - DF. Addresses DC, DD, and DE respectively, are used in loading and reading the count in Counters 0, 1, and 2. Address DF is used in writing the mode control word to the desired counter.

Table 3-5. PIT Register Addresses

CS	RD	WR	A_1	A_0	Activity	I/O Address (hex)
0	1	0	0	0	Load Counter No. 0	DC
0	1	0	0	1	Load Counter No. 1	DD
0	1	0	1	0	Load Counter No. 2	DE
0	1	0	1	1	Write Mode Word	DF
0	0	1	0	0	Read Counter No. 0	DC
0	0	1	0	1	Read Counter No. 1	DD
0	0	1	1	0	Read Counter No. 2	DE
0	0	1	1	1	No-Operation 3-State	DF
1	X	X	X	X	Disable 3-STate	—
0	1	1	X	X	No-Operation 3-State	—

NOTE: X = Irrelevant Bit

3-8. INITIALIZATION

To initialize the PIT chips, perform the following:

a. Write mode control word for Counter 0 to 0DFH. Note that all mode control words are written to 0DFH, since the mode control word must specify which counter is being programmed. (Refer to figure 3-1). Table 3-6 provides a sample subroutine for writing a mode control word for Counter 0.

b. Assuming the mode control word has selected a 2-byte load, load least-significant byte of count into Counter 0 at 0DCH. Table 3-6 provides a sample subroutine for loading a 2-byte count value.

c. Load most-significant byte of count into Counter 0 at 0DCH.

NOTE

Be sure to enter the downcount in two bytes if the counter was programmed for a two-byte entry in the mode control word. Similarly, enter the downcount value in BCD if the counter was so programmed.

d. Repeat steps a, b, and c for Counters 1 and 2.

Table 3-7 shows another example of a subroutine for writing a mode control word to Counter 0.

Table 3-6. Typical PIT Control Word

```
;'PT0M0'          PROGRAMS TIMER 0 FOR MODE 0
;                 (INTERRUPT ON TERMINAL COUNT)
;                 DOWN COUNTER
;
;                 REG. BC CONTAINS TERMINAL COUNT
;                 REG. A DESTROYED

T0M0 EQU 30H
T MODE EQU 0DFH
T0 EQU 0DCH

PT0M0:    MVI     A, T0M0     ;SETUP TIMER 0 FOR MODE 0
          OUT     T MODE
          MOV     A, C        ;LOAD LSB
          OUT     T0
          MOV     A, B        ;LOAD MSB
          OUT     T0
          RET
```

Table 3-7. PIT Control Word & Count Loading

```
;'PT0M3'          PROGRAMS TIMER 0 FOR MODE 3 (SQUARE WAVE)
;                 REG. BC CONTAINS COUNT FOR OUTPUT PERIOD
;                 REG. A DESTROYED

T0M3 EQU 36H
T MODE EQU 0DFH
T0 EQU 0DCH

PT0M3:    MVI     A, T0M3     ;SETUP TIMER 0 FOR MODE 3
          OUT     T MODE
          MOV     A, C        ;LOAD LSB
          OUT     T0
          MOV     A, B        ;LOAD MSB
          OUT     T0
          RET
```

3-9. OPERATION

The following paragraphs describe operating procedures for a counter read, clock frequency divider/ ratio selection, and interrupt timer count selection.

3-10. COUNTER READ There are three methods that can be used to read the contents of a particular counter. The first method involves a simple read or the desired counter. The only requirement with this method is that, in order to ensure a stable count reading, the desired counter must be *inhibited* by controlling its gate input. Only Counter 0 and Counter 1 can be read using this method because the gate input to Counter 2 is not controllable.

The second method allows the counter to be read "on-the-fly." The recommended procedure is to use a mode control word to latch the contents of the count register; this ensures that the count reading is accurate and stable. The latched value of the count can be read.

NOTE

If a counter is read during count, it is mandatory to complete the read procedure; that is, if two bytes were programmed to the counter, then two bytes *must* be read before any other operations are performed with that counter.

To read the count of a particular counter, proceed as follows (a typical counter read subroutine is given in table 3-8):

a. Write counter register latch control word (figure 3-3) to DF. Control word specifies desired counter and selects counter latching operation. Bits D0-D3 are irrelevant.

b. Perform a read operation of desired counter; refer to table 3-5 for counter addresses.

NOTE

Be sure to read one or two bytes, whichever was specified in the initialization mode control word. For two bytes, read in the order specified.

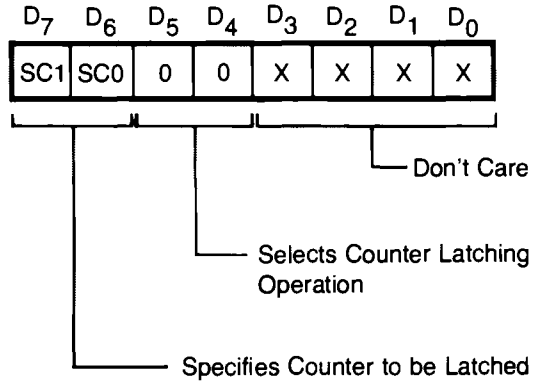


Figure 3-3. PIT Counter Register Latch Control Word Format

The third method uses the Read-Back Command. This command allows the user to check the count value, programmed Mode, and current status of the OUT pin and Null Count flag of the selected counter(s).

The command is written into the Control Word Register and has the format shown in figure 3-4. The command applies to the counters selected by setting their corresponding bits D3,D2,D1 = 1.

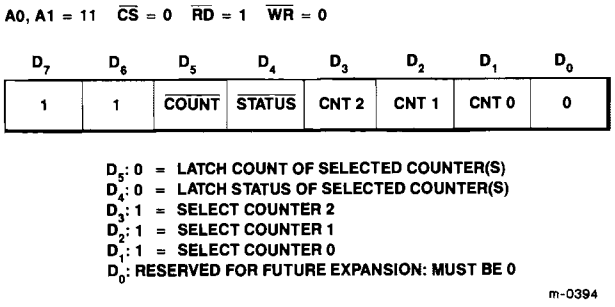


Figure 3-4. Read-Back Command Format

The read-back command may be used to latch multiple counter output latchs (OL) by setting the $\overline{\text{COUNT}}$ bit D5 = 0 and selecting the desired counter(s). This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read (or the counter is reprogrammed.) That counter is automatically unlatched when read, but other counters remain latched until they are read. If multiple count readback commands are issued to the same counter

Table 3-8. Typical PIT Counter Read Subroutine

```
;'RT0'          ALLOWS THE USER TO READ THE COUNT FOR T0
;               ON THE FLY WITHOUT DISTURBING THE COUNT OPERATION
;               REG. A DESTROYED
;               REG. HL WILL RETURN CURRENT COUNT

LT EQU 0H
T MODE EQU 0DFH
T0 EQU 0DCH

RT0:            MVI     A, LT       ;LATCH TIMER 0 COUNT
                OUT     T MODE
                IN      T0          ;READ TIMER 0 COUNT LSBYTE
                MOV     L, A
                IN      T0          ;READ TIMER 0 COUNT MSBYTE
                MOV     H, A
                RET
```

without reading the count, all but the first are ignored; i.e., the count which will be read is the count at the time the first read-back command was issued.

The read-back command may also be used to latch status information of the selected counter(s) by setting $\overline{\text{STATUS}}$ bit D4 = 0. Status must be latched to be read; status of a counter is accessed by a read from that counter.

The counter status format is shown in figure 3-5. Bits D5 through D0 contain the counter's programmed Mode exactly as written in the last Mode Control Word. OUTPUT bit D7 contains the current state of the OUT pin. This allows the user to monitor the counter's output via software, possibly eliminating some hardware from a system.

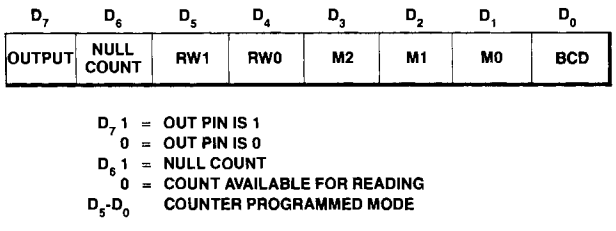


Figure 3-5. Status Byte

NULL COUNT bit D6 indicates when the last count written to the counter register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the MODE of the counter and is described in the Mode Definitions, but until the count is loaded in the counting element (CE), it can't be read from the counter. If the count is latched or read before this time, the count value will not reflect the new count just written.

If multiple status latch operations of the counter(s) are performed without reading the status, all but the first are ignored; i.e., the status that will be read is that status of the counter at the time the first status read-back command was issued.

Both count and status of the selected counter(s) may be latched simultaneously by setting both $\overline{\text{COUNT}}$ and $\overline{\text{STATUS}}$ bits D5,D4 = 0. This is functionally the same as issuing two separate read-back commands at once, and the above discussions apply here also. Specifically, if multiple count and/or status read-back commands are issued to the same counter(s) without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation of that counter will return latched status, regardless of which was latched first. The next one or two reads (depending on whether the counter is programmed for one or two type counts) return latched count. Subsequent reads return unlatched count.

For more information on the 8254 PIT device refer to the Intel Microsystem Components Handbook.

3-11. CLOCK FREQUENCY/DIVIDE RATIO SELECTION Table 2-5 lists the default and optional timer input frequencies to Counters 0 through 2. The timer input frequencies are divided by the counters to generate the OIT0 Interrupt Clock (Counter 0), OIT1 Clock (Counter 1), and the 8251A Baud Rate Clock (Counter 2).

Each counter must be programmed with a downcount number, or count value N. When count value N is loaded into a counter, it becomes the clock divisor. To derive N for either synchronous or asynchronous RS232C operation, use the procedures described in the following paragraphs.

SYNCHRONOUS MODE. In the synchronous mode, the TXC and/or RXC rates equal the Baud rate. Therefore, the count value is determined by

$$N = C/B$$

where N is the count value,
B is the desired Baud rate, and
C is 1.075 MHz, the input clock frequency.

Thus, for a 4800 Baud rate, the required count value (N) is:

$$N = \frac{1.075 \times 10^6}{4800} = 224.$$

If the binary equivalent of count value N = 224 is loaded into Counter 2, then the output frequency is 4800 Hz, which is the desired clock rate for synchronous mode operation.

ASYNCHRONOUS MODE. In the asynchronous mode, the TXC and/or RXC rates equal the Baud rate times one of the following multipliers: 16 or 64. Therefore, the count value is determined by:

$$N = C/BM$$

where N is the count value,
B is the desired Baud rate,
M is the Baud rate multiplier (16 or 64), and
C is 1.075 MHz, the input clock frequency.

Thus, for a 4800 Baud rate, the required count value (N) is:

$$N = \frac{1.075 \times 10^6}{4800 \times 16} = 14.$$

If the binary equivalent of count value N = 14 is loaded into Counter 2, then the output frequency is 4800 × 16 Hz, which is the desired clock rate for asynchronous mode operation. Count values (N) versus rate multiplier (M) for each Baud rate are listed in table 3-9. A simplified version of this table, with hex notations is provided in table 3-10.

NOTE

During initialization, be sure to load the count value (N) into the appropriate counter and the Baud rate multiplier (M) into the 8251A PCI.

ISOSYNCHRONOUS MODE. In some cases it is desirable to operate in a hybrid mode which involves transmitting data with the asynchronous format using a synchronous modem. This occurs when an increase in operating speed is required without a change in the basic protocol of the system. This hybrid technique is known as isosynchronous and involves the generation of the start and stop bits associated with the asynchronous format, while still using the modem clock for bit synchronization.

In the asynchronous mode the PCI can be programmed to accept clock rates of 16 or 64 times the required baud rate. Isosynchronous operation is a special case of asynchronous with the multiplier rate programmed as one instead of 16 or 64. Note that ×1 operation is only valid if the clocks of the receiver and transmitter are synchronized.

Table 3-9. PIT Count Values & Rate Multipliers

Baud Rate (B)	[1]Count Value (N) For		
	M = 1	M = 16	M = 64
75	14319	895	224
110	9763	610	153
150	7160	447	112
300	3580	224	56
600	1790	112	28
1200	895	56	14
2400	447	28	7
4800	224	14	—
9600	112	7	—
19200	56	—	—
38400	28	—	—

[1]Count Values (N) assume clock is 1.075 MHz. (N) and Rate Multiplier (M) are in decimal.

Table 3-10. PIT Baud Rate Factors

Output Frequency in kHz	Baud Rate (Hz)			8254 PIT Baud Rate Factor (Hex Notation)	
	Synchronous	Asynchronous			
		÷16	÷64	MSB	LSB
153.6	—	9600	2400	00	07
76.8	—	4800	1200	00	0E
38.4	38400	2400	600	00	1C
19.2	19200	1200	300	00	38
9.6	9600	600	150	00	70
4.8	4800	300	75	00	E0
2.4	2400	150	—	01	C0
1.76	1760	110	—	02	63

3-12. RATE GENERATOR/INTERVAL TIMER. Table 3-11 shows the maximum and minimum rate generator frequencies and timer intervals for Counters 0 and 1 when these counters, respectively, have 1.075 MHz and 152.6 KHz clock inputs. The table also provides the maximum and minimum generator frequencies and time intervals and may be obtained by connecting Counters 0 and 1 in series.

3-13. INTERRUPT TIMER. To program an interval timer for an interruption terminal count, program the appropriate timer for the correct operating mode (Mode 0) in the control word. Then load the count value (N), which is derived by

$$N = TC$$

where N is the count value for Counter 0,
T is the desired interrupt time interval in seconds, and
C is the internal clock frequency (Hz).

Table 3-12 shows the count value (N) required for several time intervals (T) that can be generated for Counters 0 and 1.

3-14. 8251A PROGRAMMABLE COMMUNICATION INTERFACE (PCI) 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 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.

Table 3-11. PIT Rate Generator Frequencies & Timer Intervals

	Single Timer[1] (Counter 0)		Single Timer[2] (Counter 1)		Dual Timer[3] (0 and 1 in Series)	
	Minimum	Maximum	Minimum	Maximum	Minimum	Maximum
Rate Generator (Frequency)	16.407 Hz	537.6 kHz	2.05 Hz	67.19 kHz	0.00025 Hz	268.81 kHz
Real-Time Interrupt Interval	1.859 μsec	60.95 msec	14.88 μsec	487.8 msec	3.719 μsec	66.66 minutes

NOTES:
1. Assuming a 1.075 MHz clock input.
2. Assuming a 134.38 KHz.
3. Assuming Counter 0 has 1.075 MHz. clock input.

Table 3-12. PIT Timer Intervals & Timer Counts

T	N[1]
10 μsec	11
100 μsec	107
1 msec	1075
10 msec	10750
50 msec	53750

[1]Count Values (N) assume clock is 1.075 MHz. Count Values (N) are in decimal.

3-15. 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
   (5) Single or double character sync

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

Instruction word and data transmission formats for synchronous and asynchronous modes are shown in figure 3-6 through 3-10.

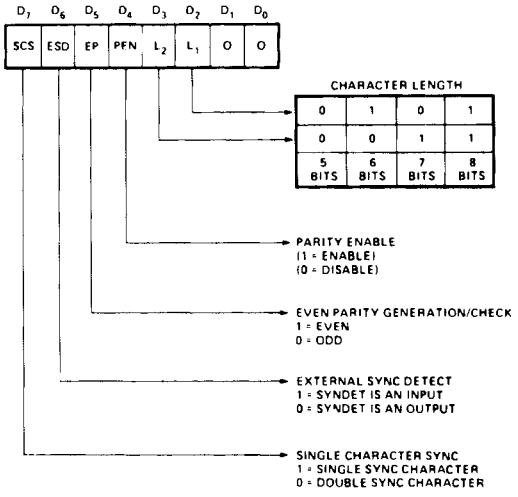


Figure 3-6. PCI Synchronous Mode Instruction Word Format

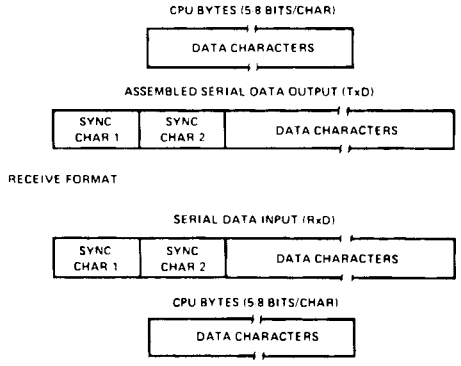


Figure 3-7. PCI Synchronous Mode Transmission Format

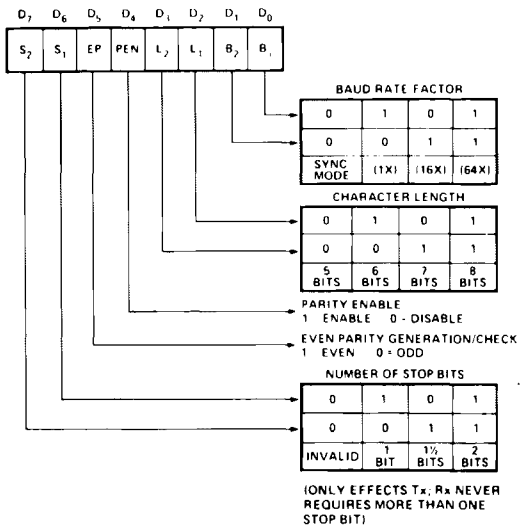


Figure 3-8. PCI Asynchronous Mode Instruction Word Format

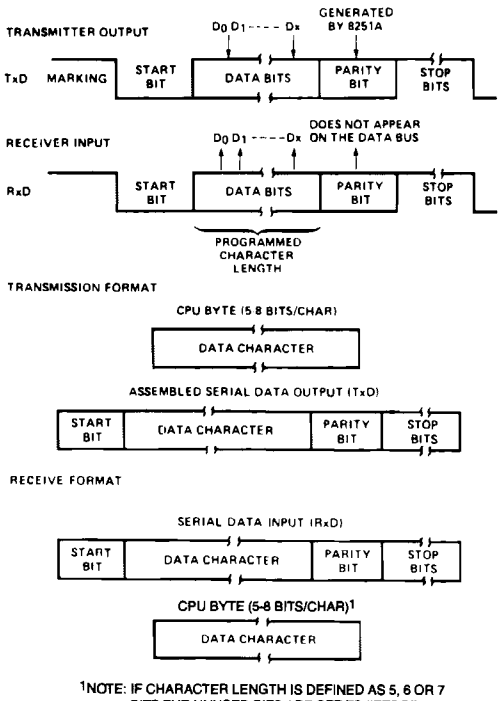


Figure 3-9. PCI Asynchronous Mode Transmission Format

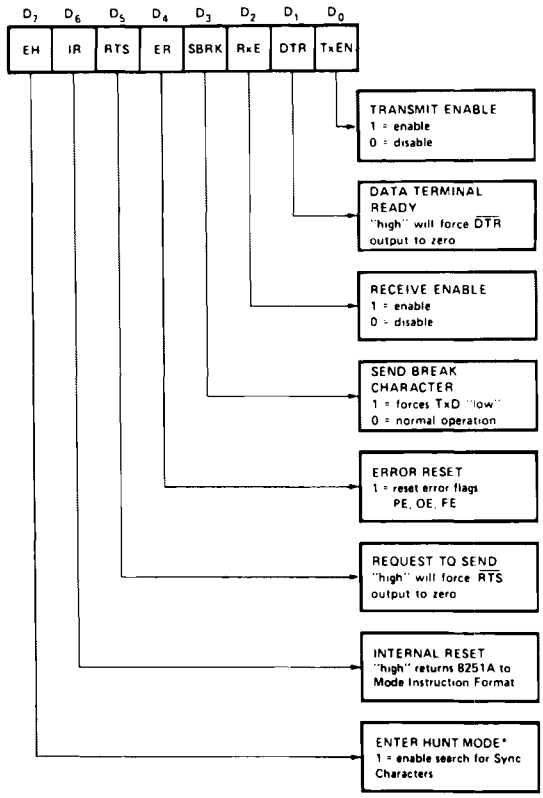


Figure 3-10. PCI Command Instruction Word Format

3-16. 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.

3-17. COMMAND INSTRUCTION FORMAT

The Command instruction word shown in figure 3-10 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.

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 PCI 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.

3-18. RESET

To change the Mode instruction word, the PCI must receive a Reset command. This can be either a hardware reset or a reset generated by bit 6 of the Command Instruction. 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.

3-19. 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 3-13).

Table 3-13. PCI Address Assignments

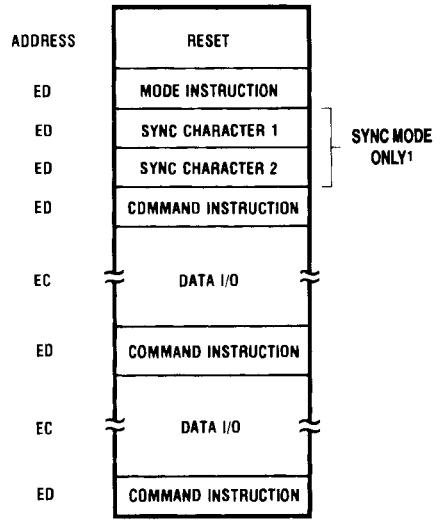
I/O Address (hexadecimal)	Command	Function	Direction
ED or EF	OUTPUT	CONTROL	CPU → PCI
EC or EE	OUTPUT	DATA	CPU → PCI
ED or EF	INPUT	STATUS	PCI → CPU
EC or EE	INPUT	DATA	PCI → CPU

3-20. INITIALIZATION

A typical PCI initialization and I/O data sequence is presented in figure 3-11. 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.
c. If synchronous mode is selected, write one or two synch 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.



[1]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 3-11. Typical PCI Initialization & Data I/O Sequence

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.

If the initialization procedure is interrupted, ensure correct reset and programming by writing a series of four command words, with the contents of each word = 00. Then repeat the PCI reset and initialization procedure.

Next, write a Mode instruction word to the PCI. (See figures 3-6 through 3-9). A typical subroutine for writing both Mode and Command instructions is given in table 3-14.

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 3-10 and table 3-14.

Table 3-14. 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:             IN      0EDH        ;GET PCI STATUS
                ANI     1           ;CHECK TXRDY
                JZ      LP          ;TXRDY MUST BE TRUE
                POP     PSW         ;RESTORE DATA AND CPU STATUS:
                OUT     0EDH        ;SEND COMMAND/MODE WORD TO PCI
                RET

                END
```

3-21. 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.

Prior to any operating change, a new command word must be written with command bits changed as appropriate. (Refer to figure 3-10 and table 3-14).

For data receive or transmit operations perform a read or write, respectively, to the PCI. Tables 3-15 and 3-16 provide examples of typical character read and write subroutines.

During normal transmit operation, the PCI generates a Transmit Ready (TxRDY) signal that indicates 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 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 2-13 for instructions.

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 3-12. A typical status read subroutine is given in table 3-17.

Table 3-15. Typical PCI Data Character Read Subroutine

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

                EXTRN   STAT

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

                END
```

Table 3-16. 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:           IN      0EDH        ;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
```

Table 3-17. Typical PCI Status Read Subroutine

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

;STAT:          IN      0EDH        ;GET PCI STATUS
                RET

                END
```

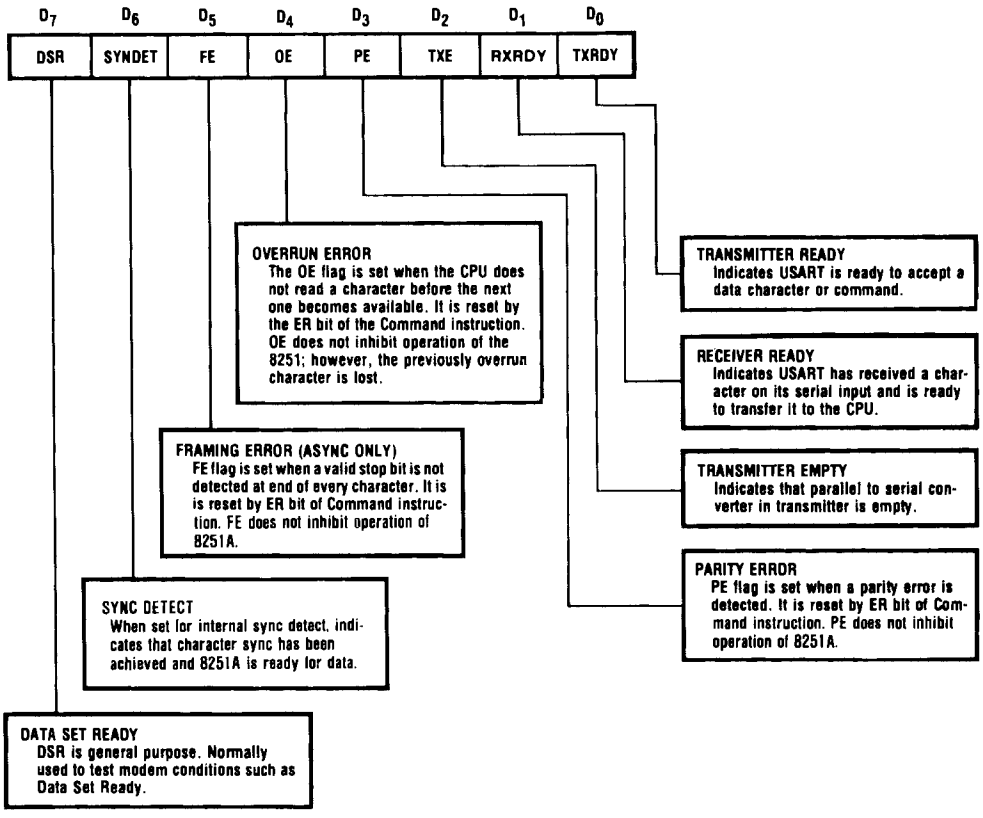


Figure 3-12. PCI Status Read Format

3-22. 8255A PROGRAMMABLE PERIPHERAL INTERFACE (PPI) PROGRAMMING

The iSBC 80/24A 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 tables 2-18 and 2-19.

Each of the six ports (three on each 8255A device) may be independently programmed for a different I/O configuration. All the configurations are summarized in table 3-18.

Default jumpers set the port E4 and E8 bus transceivers to the output mode only. Optional jumper connections allow the bus transceivers for these ports to be set to the input only mode, or to either the input or the output mode. Refer to section 2-14 for instructions on these jumper conversions. A description of 8255A operation is provided in the *Intel Microsystem Components Handbook.*

Table 3-18. Parallel Port Configurations

Ports E4 & E8
Mode 0, input
Mode 0, output (latched)
Mode 1, input (strobed)
Mode 1, output (latched)
Mode 2, bidirectional
Ports E5 & E9
Mode 0, input
Mode 0, output (latched)
Mode 1, input (strobed)
Mode 1, ouput (latched)
Port E6 & EA[1]
Mode 0, 8-bit input
Mode 0, 8-bit output (latched)
Mode 0, split (4-bit input, 4-bit output)
[1]Control mode may depend on mode of other ports; see table 2-6.

3-23. CONTROL WORD FORMAT

The control word format shown in figure 3-13 is used to initialize each PPI port. 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. (Refer to table 3-19 for port identification). Bit 7 of the control word controls the mode set flag. Control words are sent to port E7 for the J1 PPI device, and to port EB for the J2 PPI device (table 3-19). There are restrictions associated with the use of certain ports in modes 1 and 2. Refer to table 2-6 for restrictions.

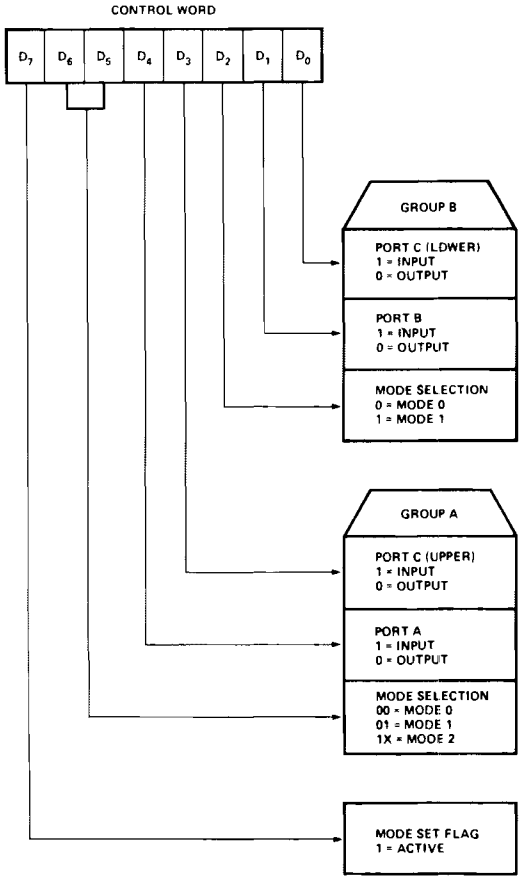


Figure 3-13. PPI Control Word Format

Table 3-19. Parallel Port I/O Addresses

8255A Device Location	Eight-Bit Address (hexadecimal)
8255A #1 Port (A)	E4
8255A #1 Port (B)	E5
8255A #1 Port (C)	E6
8255A #1 Control	E7 For I/O write only
8255A #2 Port (A)	E8
8255A #2 Port (B)	E9
8255A #2 Port (C)	EA
8255A #2 Control	EB For I/O write only

3-24. ADDRESSING

The J1 PPI (U15) uses three consecutive data addresses: E4, E5 an E6, plus command port E7. The J2 PPI (U17) also uses three consecutive data addresses: E8, E9, and EA, plus command port EB. Refer to table 3-19.

3-25. INITIALIZATION

To initialize a PPI, write a control word to its control port (E7 or EB). Refer to figure 3-13 and table 3-20 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

3-26. 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 3-18),
b. direction of data flow (input, output or bidirectional),
c. choice of driver/terminator networks.

Single Bit Set/Reset Feature

Any of the eight bits of Port C (E6 or EA) can be Set or Reset using a single output instruction (see figure 3-14). This feature reduces software requirements in Control-based applications.

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.

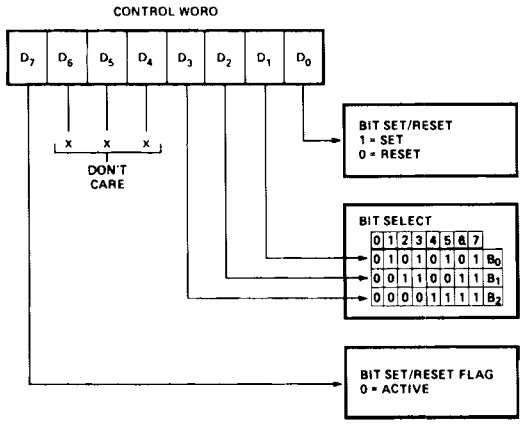


Figure 3-14. PPI Port C Bit Set/Reset Control Word Format

Mode Combinations

Table 3-21 summarizes the various mode combinations possible with ports A and B of each PPI, and indicates how each port C bit can be used. This table can serve as a useful starting point for selecting your particular configuration. Once you select the desired mode combination and the port C bit assignments are made, refer to the jumper configuration table (2-6) for implementation details.

Table 3-20. Typical PPI Intialization Subroutine

```
;INTPAR INTIALIZES 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
```

Table 3-21. Parallel I/O Interface Configurations

Configuration Number	PPI Port A (E4 or E8)	PPI Port B (E5 or E9)	PPI Port C (E6 or EA) Lower				PPI Port C (E6 or EA) Upper			
			C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
1	MODE 0-IN	MODE 0-I/O	— I/O —				— I/O —			
2	MODE 0-OUT	MODE 0-I/O	— I/O —				— I/O —			
3	MODE 0-IN	MODE 1-I/O	R	R	R	I	O	O	O	U
4	MODE 0-IN	MODE 1-I/O	R	R	R	O	I	I	I	U
5	MODE 0-OUT	MODE 1-I/O	R	R	R	I	O	O	O	U
6	MODE 0-OUT	MODE 1-I/O	R	R	R	O	I	I	I	U
7	MODE 1-IN	MODE 0-I/O	I	I	I	R	R	R	O	O
8	MODE 1-IN	MODE 0-I/O	O	O	O	R	R	R	I	I
9	MODE 1-OUT	MODE 0-I/O	I	I	I	R	O	O	R	R
10	MODE 1-OUT	MODE 0-I/O	O	O	O	R	I	I	R	R
11	MODE 1-IN	MODE 1-I/O	R	R	R	R	R	R	I	I
12	MODE 1-IN	MODE 1-I/O	R	R	R	R	R	R	O	O
13	MODE 1-OUT	MODE 1-I/O	R	R	R	R	I	I	R	R
14	MODE 1-OUT	MODE 1-I/O	R	R	R	R	O	O	R	R
15	MODE 2-B	MODE 0-I/O	U	I	I	R	R	R	R	R
16	MODE 2-B	MODE 0-I/O	U	O	O	R	R	R	R	R
17	MODE 2-B	MODE 1-I/O	R	R	R	R	R	R	R	R

NOTES:
I - INPUT
O - OUTPUT
I/O - INPUT OR OUTPUT
B - BIDIRECTIONAL
R - RESERVED
U - No unused drivers/terminators available. These bits may be used, however, to connect to the serial I/O interface or the Interval Timer.

3-27. 8259A PROGRAMMABLE INTERRUPT CONTROLLER (PIC) PROGRAMMING

The 8259A Programmable Interrupt Controller (PIC) is used to monitor and control all on-board and off-board interrupt requests, except the four special 8085A-2 CPU inputs. These special inputs are discussed in section 3-41.

Interrupt priority is determined by hardware jumper connection *and* 8259A device programming. All interrupt priority jumper connections are implemented at the interrupt jumper matrix, shown in figure 5-4, sheet 9. Possible jumper connections are listed in table 2-7. If your board is in the factory default configuration, the following jumpers are installed:

a. *On-board* interrupt: output 0 of the PIT is wired to IR2 on the PIC.

b. *Off-board* interrupt: pin P1-39 (INT2/) is wired to IR1 on the PIC.

Programming modes and initialization instructions for the 8259A device are provided in sections 3-28 through 3-37.

3-28. INTERRUPT PRIORITY MODES

The PIC has three modes for resolving the priority of interrupt inputs: (1) fully nested mode (2) rotating mode and (3) polled mode. The rotating mode has two variations: auto-rotating and specific rotating.

3-29. FULLY NESTED MODE. In this mode the PIC input signals are assigned priority from 0 through 7. The PIC operates in this mode unless specifically programmed otherwise. Interrupt IR0 has the highest priority and IR7 has the lowest priority. When an interrupt is acknowledged, the highest priority request is available to the CPU. Lower priority interrupts are inhibited; higher priority interrupts will be able to generate an interrupt that will be acknowledged if the CPU has enabled its own interrupt input through software. An End-Of-Interrupt (EOI) command is required to reset the PIC for the next interrupt.

3-30. AUTO-ROTATING MODE. In this mode the interrupt priority rotates. Once an interrupt on a given input is serviced, that interrupt assumes the

lowest priority. Thus, if there are a number of simultaneous interrupts, the priority will rotate among the interrupts in numerical order. For example, if interrupts IR4 and IR6 request service simultaneously, IR4 will receive the highest priority. After service, the priority level rotates so that IR4 has the lowest priority and IR5 assumes the highest priority. In the worst case, seven other interrupts are serviced before IR4 again has the highest priority. Of course, if IR4 is the only request, it is serviced promptly. In the Auto-Rotating Mode, priority shifts when the PIC chip receives an End-Of-Interrupt (EOI or AEOI) command.

3-31. SPECIFIC ROTATING MODE. In this mode the software can change interrupt priority by specifying the bottom priority, which automatically sets the highest priority. For example, if IR5 is assigned the bottom priority, IR6 assumes the highest priority. In the specific rotating mode, the priority can be rotated by writing a Specific Rotate at EOI (SEOI) command to the PIC chip. This command contains the BCD code of the interrupt being serviced; that interrupt is reset as the bottom priority. In addition, the bottom priority interrupt can be fixed at any time by writing a command word to the PIC chip.

3-32. POLLED MODE. In this mode, the 8085A-2 interrupt input is disabled. Service to the devices is achieved by programmer initiative using a Poll command. This mode is useful if there is a routine command common to several levels so that the INTA sequence is not needed.

3-33. INTERRUPT MASK

Each interrupt request input can be individually masked (disabled) by the Interrupt Mask Register (IMR). The register is programmed by OCW1 (figure 3-17) Each bit in the IMR corresponds to one interrupt input. Setting the bit (bit = 1) will mask its input. The input will remain masked until its IMR register bit is reset (bit = 0). Masking one input does not affect any of the other inputs.

Note that if an interrupt is already acknowledged by the PIC (an INTA/ pulse has been sent to the CPU) then the interrupting level will inhibit the lower priorities. There are two ways to avoid this condition:

a. Write an End of Interrupt (EOI) command (OCW2) to reset the ISR bit, or;

b. Set the Special Mask Mode using OCW3 (refer to section 3-34).

3-34. SPECIAL MASK MODE

This mode is useful when some bits of the IMR are set, masking the corresponding interrupt inputs (refer to section 3-33). Since lower priority interrupts are disabled while the current interrupt is being serviced, you may wish to override this mechanism. If you want to enable lower priority interrupts when a routine is in service, you must select the Special Mask Mode (SMM). In the Special Mask Mode, the lower priority interrupts are enabled until the SMM is reset (cancelled). The higher priorities are not affected. The Special Mask Mode is selected with OCW3 (figure 3-17). For additional information on the Special Mask Mode and the operation of the Intel 8259A PIC, refer to Application Note AP-59, *Using the Intel 8259A Programmable Interrupt Controller.*

3-35. STATUS READ

Interrupt request inputs are handled by the following two internal PIC registers:

a. Interrupt Request Register (IRR), which stores all interrupt levels that are requesting service.

b. In-Service Register (ISR), which stores all interrupt levels that are being serviced.

Either register can be read by writing a suitable command word and then performing a read operation.

3-36. INITIALIZATION COMMAND WORDS

The eight devices serviced by the PIC have eight addresses equally spaced in memory that can be programmed at intervals of four or eight bytes. Interrupt service routines for these eight devices thus occupy a 32- or 64-byte block, respectively, of memory. The address format for device interrupt service routine is shown in figure 3-15.

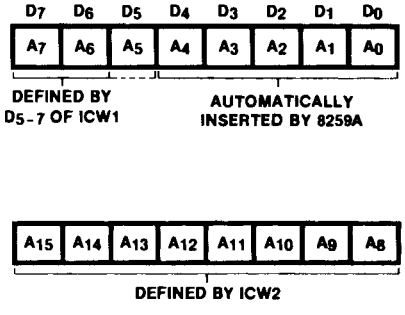


Figure 3-15. PIC Device Interrupt Addresses

Bits 0-4 are automatically inserted by the PIC whereas bits 6-15 are programmed by the Initialization Command Words ICW1 and ICW2. If the address interval is eight bytes, bit 5 is automatically inserted by the PIC; if the address interval is four bytes, bit 5 is programmed in ICW1. Thus, the 32-byte or 64-byte block of addresses reserved for interrupt service routines can be located anywhere in the available memory space. Table 3-22 shows the address format inserted by the PIC for each device.

Initialization of the PIC consists of writing two or three 8-bit Initialization Command words as shown in figure 3-16. Since the iSBC 80/24A board does not support slave PICs, the initialization for the one PIC consists of writing two Initialization Command Words as follows:

a. The first Initialization Command Word (ICW1) consists of the following:

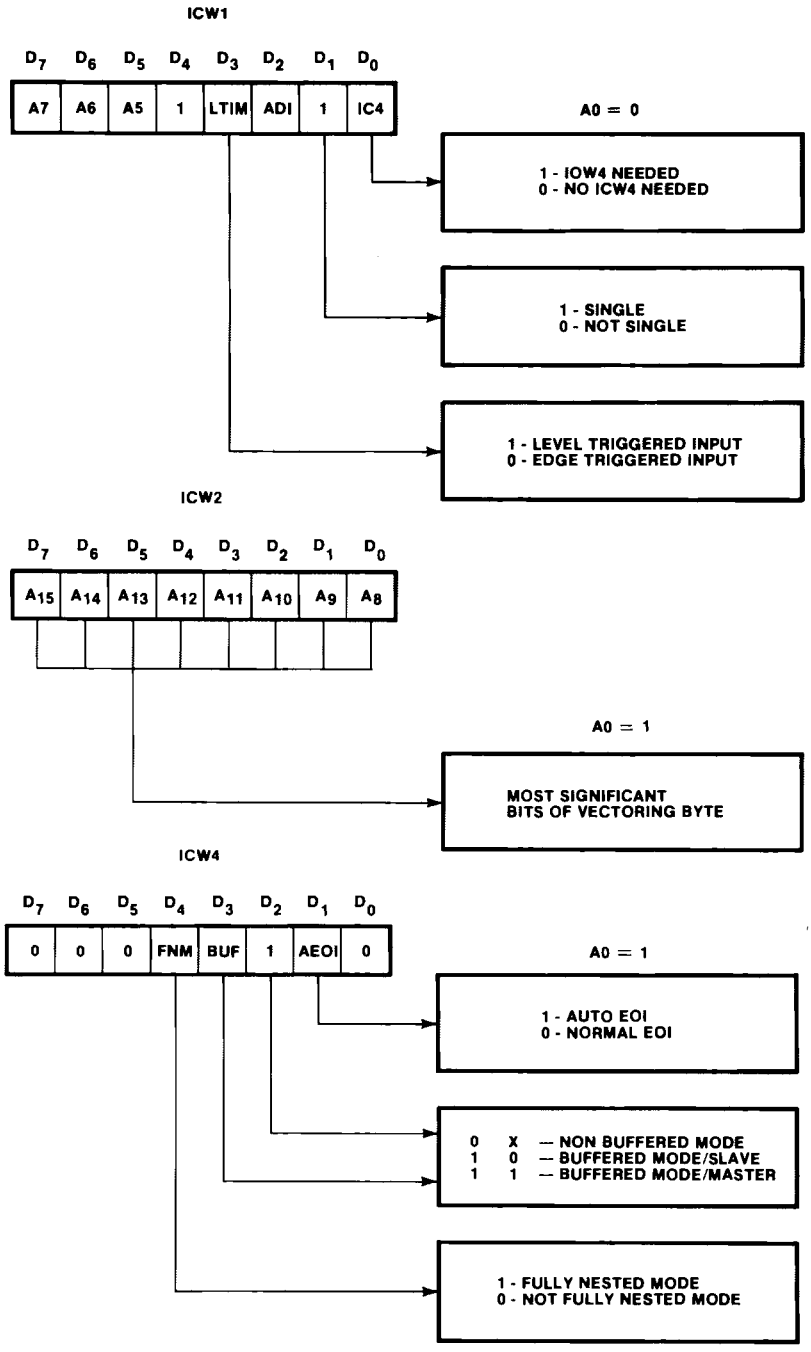


Figure 3-16. PIC Initialization Command Word Formats

Table 3-22. PIC Device Address Insertion

Device	Lower Routine Address Byte															
	Interval = 4								Interval = 8							
	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
IR7	A7	A6	A5	1	1	1	0	0	A7	A6	1	1	1	0	0	0
IR6	A7	A6	A5	1	1	0	0	0	A7	A6	1	1	0	0	0	0
IR5	A7	A6	A5	1	0	1	0	0	A7	A6	1	0	1	0	0	0
IR4	A7	A6	A5	1	0	0	0	0	A7	A6	1	0	0	0	0	0
IR3	A7	A6	A5	0	1	1	0	0	A7	A6	0	1	1	0	0	0
IR2	A7	A6	A5	0	1	0	0	0	A7	A6	0	1	0	0	0	0
IR1	A7	A6	A5	0	0	1	0	0	A7	A6	0	0	1	0	0	0
IR0	A7	A6	A5	0	0	0	0	0	A7	A6	0	0	0	0	0	0

(1) Bits 5-7 specify the most-significant bits of the lower address byte of the interrupt service routine.

(2) Bit 3 establishes whether the interrupts are requested by a positive-true or requested by a low-to-high transition input. This applies to all input requests handled by the PIC. In other words, if bit 3 = 0, a low-to-high transition is required to request an interrupt on any of the eight levels handled by the PIC.

(3) Bit 2 specifies a 4-byte or 8-byte address interval.

(4) Bit 1 specifies whether or not there are slave (cascaded) PIC's. Since there are no slave PIC's, set bit 1 = 1.

(5) Bits 0 and 4 identify the word as ICW1.

b. The second word (ICW2) specifies the upper byte (bits 8-15) of the interrupt service routine.

3-37. OPERATION COMMAND WORDS

After being initialized, the PIC can be programmed at any time for various interrupt modes. The Operation Command Word (OCW/) formats are shown in figure 3-17.

3-38. ADDRESSING

The PIC uses two consecutive addresses for writing to and reading internal registers. Address functions pertinent to programming are identified in table 3-23.

3-39. INITIALIZATION

To initialize the PIC, proceed as follows (table 3-24 provides a typical initialization subroutine):

a. Disable system interrupts by executing a DI (Disable Interrupts) instruction.

b. Write ICW1 to Port D8.

c. Write ICW2 to Port D9.

d. Write ICW4 if required by your application.

e. Enable system interrupts by executing an EI (Enable Interrupts) instruction.

NOTE

Table 3-25 should be referenced when studying the programming examples. All programming procedures are PL/M calling conventions.

NOTE

The PIC operates in the fully nested mode after the initialization sequence without requiring any Operation Control Word (OCW).

Table 3-23. PIC Addressing

I/O Port Address (hex)	Data Bits		Input Operation
	D4	D3	
D8 or DA	—	—	Read IRR, ISR or current interrupt level (depending on contents of OCW3)
D9 or DB	—	—	Read Interrupt Mask Register (IMR)
			Output Operation
D8 or DA	0	0	Load OCW2
D8 or DA	0	1	Load OCW3
D8 or DA	1	X	Load ICW1
D9 or DB	X	X	Load OCW1, ICW2, or ICW4, (depending on sequence)

3-40. OPERATION

After initialization, the PIC chips can be programmed at any time for the following operations:

a. Auto-rotating priority.

b. Specific rotating priority.

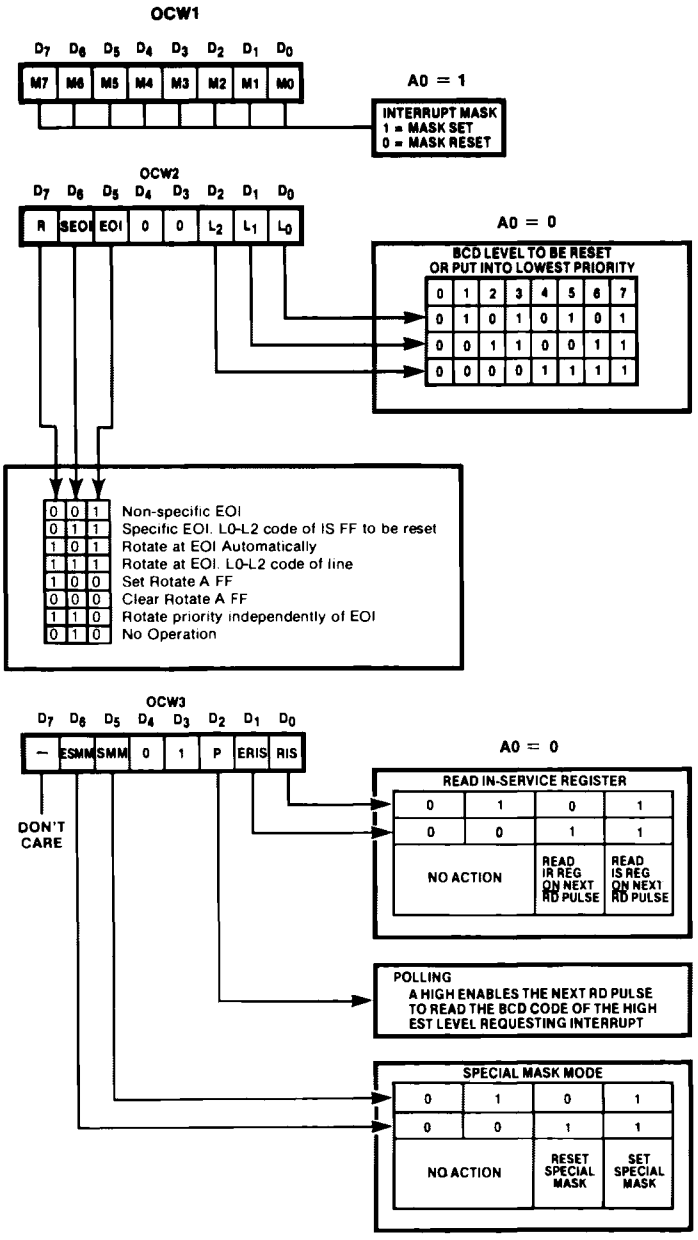


Figure 3-17. PIC Operation Control Word Formats

Table 3-24. Typical PIC Initialization Subroutine

```
;'SETVEC'        WILL INITIALIZE THE 8259A POINTER FOR THE
;                INTERRUPT JUMP VECTORS WITH 8 BYTE SPACING.
;                REG. BC CONTAINS ADDRESS OF FIRST JUMP VECTOR
;                NOTE: IT MUST BE EVENLY DIVISIBLE BY 64
;                REG. A DESTROYED

SETVEC:         MOV     A, C
                ANI     MASK8
                ORI     ICW1C
                OUT     ICW1
                MOV     A, B
                OUT     ICW2
                RET
```

c. Status read of Interrupt Request Register (IRR).

d. Status read of In-Service Register (ISR).

e. Interrupt mask bits set, reset, or read.

f. Special mask mode set or reset.

Table 3-26 lists details of the PIC operation. Note that an End-Of-Interrupt (EOI) or a Special End-Of-Interrupt (SEOI) command is required at the end of each interrupt service routine to reset the ISR. The EOI command is used in the fully nested and auto-rotating priority modes and the SEOI command, which specifies the bit to be reset, is used in the specific rotating priority mode. Tables 3-27 through 3-31 provide typical subroutines for the following:

a. Read IRR.

b. Read ISR.

c. Set mask register.

d. Read mask register.

e. Issue EOI command.

Table 3-25. PIC Equates

```
PIC     EQU     0D8H
ICW1    EQU     PIC + 0
ICW2    EQU     PIC + 1
OCW2    EQU     PIC + 0
OCW3    EQU     PIC + 0
IMR     EQU     PIC + 1
ICW1A   EQU     16H
ICW1C   EQU     12H
MASK4   EQU     0E0H
MASK8   EQU     0C0H
RIS     EQU     0BH
RR      EQU     0AH
SM      EQU     68H
RSM     EQU     48H
EOI     EQU     20H
```

Table 3-26. PIC Operation Procedures

Operation	Procedure
Auto-Rotating Priority Mode	To set: In OCW2, write a Rotate Priority at EOI command (A0H) to Port D8H. Terminate interrupt and rotate priority: In OCW2, write EOI command (20H) to D8H.
Specific Rotating Priority Mode	To set: In OCW2, write a Rotate Priority at SEOI command in the following format to Port D8H. D7 D6 D5 D4 D3 D2 D1 D0 1 1 1 0 0 L2 L1 L0 BCD OF IR LINE TO BE RESET AND/OR PUT INTO LOWEST PRIORITY. To terminate interrupt and rotate priority: In OCW2, write an SEOI command in the following format to Port D8H. D7 D6 D5 D4 D3 D2 D1 D0 0 1 1 0 0 L2 L1 L0 BCD OF ISR FLIP-FLOP TO BE RESET. To rotate priority without EOI: In OCW2, write a command word in the following format to Port D8H. D7 D6 D5 D4 D3 D2 D1 D0 1 1 0 0 0 L2 L1 L0 BCD OF BOTTOM PRIORITY IR LINE.

Table 3-26. PIC Operation Procedures (Continued)

Operation	Procedure
Interrupt Request Register (IRR) Status	The IRR stores a "1" in the associated bit for each IR input line that is requesting an interrupt. To read the IRR (refer to footnote): (1) Write OAH to Port D8H. (2) Read Port D8. Status format is as follows: D | D6 | D5 | D4 | D3 | D2 | D1 | D0 IR LINE: 7 6 5 4 3 2 1 0
In-Service Register (ISR) Status	The ISR stores a "1" in the associated bit for priority inputs that are being serviced. The ISR is updated when an EOI command is issued. To read the ISR (refer to footnote): (1) Write OBH to Port D8. (2) Read Port D8. Status format is as follows: D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 IR LINE: 7 6 5 4 3 2 1 0 Be sure to reset ISR bit at end-of-interrupt when in the following modes: Auto-Rotating (both types) and Special Mask. To reset ISR in OCW2, write: D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 0 1 1 0 0 L2 L1 L0 BCD IDENTIFIES BIT TO BE RESET.
Interrupt Mask Register	To set mask bits in OCW1, write the following mask byte to Port D9H. D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 IR BIT MASK: M7 M6 M5 M4 M3 M2 M1 M0 1 = MASK SET, 0 = MASK RESET To read mask bits, read Port D9H.
Special Mask Mode	The Special Mask Mode enables desired bits that have been previously masked; lower priority bits are also enabled. To set, write 68H to D9H. To reset, write 48H to D9H.
NOTE: If previous operation was addressed to same register, it is not necessary to rewrite the OCW.	

Table 3-27. Typical PIC Interrupt Request Register Read Subroutine

```
;'RDIRR'          READS 8259A INTERRUPT REQUEST REGISTER
;                 REG. A RETURNS RESULT

RDIRR:           MVI     A, RR
                 OUT     OCW3
                 IN      OCW3
                 RET
```

Table 3-28. Typical PIC In-Service Register Read Subroutine

```
;'RDISR'        READS 8259A IN-SERVICE REGISTER
;               REG. A RETURNS RESULT

RDISR:          MVI     A, RIS
                OUT     OCW3
                IN      OCW3
                RET
```

Table 3-29. Typical PIC Set Mask Register Subroutine

```
;'WRIMR'        SETS THE 8259A INTERRUPT MASK REGISTER
;               NOTE: A '0' WILL ENABLE THE CORRESPONDING IR LEVEL
;               REG. C CONTAINS NEW IR MASK
;               REG. A DESTROYED

WRIMR:          MOV     A, C
                OUT     IMR
                RET
```

Table 3-30. Typical PIC Mask Register Read Subroutine

```
;'RDIMR'        READS THE 8259A INTERRUPT MASK REGISTER
;               REG. A RETURNS RESULT

RDIMR:          IN      IMR
                RET
```

Table 3-31. Typical PIC End Of Interrupt Command Subroutine

```
;'NSEOI'        SENDS A NON-SPECIFIC END-OF-INTERRUPT TO THE 8259
;               IT CAN BE USED TO EXIT AN INTERRUPT SERVICE ROUTINE AS FOLLOWS:
;               ISRO:           ;USER SERVICE ROUTINE
;
;
;                               CALL NSEOI
;                               EI
;                               RET

NSEOI:          MVI     A, EOI
                OUT     OCW2
                RET
```

3-41. 8085A-2 INTERRUPT HANDLING

In addition to the interrupts handled by the 8259A PIC device, the 8085A-2 CPU will accept four other interrupts, without PIC intervention. In the factory default configuration, all four of these iSBC 80/24A board inputs are grounded by jumper connection at the interrupt matrix, and therefore disabled. Refer to table 2-7 and figure 5-4, sheet 9.

Both on-board and off-board sources may be used as inputs to these CPU interrupt inputs. Table 3-32 provides the priority value of each input. Vector locations are fixed within the 8085A-2 CPU for these interrupts, and cannot be altered. Priority is thus established by hardware jumper connection, as shown in jumber table 2-7. Interrupt operation of these four inputs is described in sections 3-42 thorugh 3-44.

The 8085A-2 CPU has five interrupt inputs: TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR. The three restart interrupts (RST 7.5, RST 6.5, and RST 5.5) are maskable. INTR and the three restart interrupts can be enabled and disabled under software control. TRAP is non-maskable and cannot be disabled.

The RST 7.5, RST 6.5, and RST 5.5 interrupts cause the internal execution of an RST (RESTART) instruction if the interrupts are enabled and if the interrupt mask has not been set by a previously executed SIM instruction. The non-maskable TRAP interrupt causes the execution of an RST instruction independent of the state of the interrupt enable or interrupt mask. The priority and vector locations of each of the restart interrupts are given in table 3-32.

Table 3-32. Interrupt Vector Locations & Priority

Interrupt	Hex Vector Location	Priority
TRAP	24	Highest
RST 7.5	3C	2nd
RST 6.5	34	3rd
RST 5.5	2C	4th
INTR	*	5th - 12th

* Controlled by 8259A PIC programming.

3-42. TRAP INTERRUPT

There are special considerations that must be made when the TRAP interrupt is used. The fact that the TRAP interrupt is non-maskable can present problems in at least two areas.

Interrupt driven systems often contain parameters that must be modified only within critical regions. A critical region can be roughly defined as a section of code that once begun must complete execution before it or another critical region that corresponds to the same system parameter(s) can be executed. A TRAP interrupt handler cannot safely alter such parameters either directly or indirectly by causing the execution of procedures or tasks that may alter such parameters.

If the hardware generates a TRAP interrupt on power up or power fail, the system must be able to process the TRAP interrupt before it is completely initialized. It should also take into account that an interrupt routine that runs with interrupts disabled can still be interrupted by a TRAP.

Because of these considerations, it is recommended that the TRAP interrupt only be used for system startup and/or catastrophic error handling such as a power failure.

It should be noted that TRAP does not destroy a previously established interrupt enable status. Executing the first RIM instruction following a TRAP interrupt yields the interrupt enable status as it was before the TRAP occurred. Following this first *mandatory* RIM instruction, subsequently executed RIM instructions provide current interrupt enable status.

3-43. RST 7.5, 6.5, AND 5.5 INPUTS

These interrupts can be individually masked by a SIM instruction and can thus be prevented from interrupting the processor. The priorities shown in table 3-32 do not take into account the priority of a routine that was started by a higher priority interrupt. An RST 6.5 interrupt can interrupt an RST 7.5 routine if the interrupts are re-enabled before the end of the RST 7.5 routine.

The RST 7.5 interrupt is rising-edge-sensitive and can be set by a pulse; once the request is set by a pulse, it will be remembered until the request is serviced or reset by a SIM instruction. The RST 6.5 and 5.5 interrupts are high-level-sensitive and must remain high to be recognized. Table 3-33 shows a typical restart routine.

3-44. INTR INTERRUPT

The INTR interrupt from the 8259A PIC has the lowest priority and is sampled only during the last clock cycle of a given instruction. When INTR is active, the Program Counter (PC) is pushed onto the stack and three bytes, timed by INTA/ pulses, are transferred from the PIC to the 8085A CPU.

Table 3-33. Typical RST 5.5 Interrupt Routine

```
                ORG     2CH         ;START ADDRESS FOR RST 5.5
                JMP     RST55
RST55:
                PUSH    H           ;
                —       —           ;
                —       —           ; USER ROUTINE
                —       —           ;
                —       —           ;
                EI
                RET
```

a. Byte 1 = Call instruction (0CDH).

b. Byte 2 = Lower byte of routine address.

c. Byte 3 = Upper byte of routine address.

The 16-bit routine address must be on a 4-byte boundary or an 8-byte boundary as determined by the Initialization Command Words (ICW) previously written to the PIC. (Refer to section 3-36).

The INTR signal can be enabled or disabled by the program. It is disabled when a "RESET" occurs, and immediately after an interrupt is accepted by the 8085A-2 CPU.

3-45. RMX/80™ SOFTWARE

The iSBC 80/24A board is compatible with Intel's RMX/80 Real time Multitasking Executive. User program (tasks) can take advantage of RMX/80 to do all necessary scheduling, intertask communication, and memory space allocation. RMX/80 also provides standard I/O support software such as the Disk File handler, the Intel Analog Board handler, and the terminal handler. (Use of the RMX/80 Analog Handler may require the iSBC 80/24A board to be configured for 2.42 MHz operation.

3-46. SYSTEM PROGRAM DEVELOPMENT

The development cycle of iSBC 80/24A board based products may be significantly reduced using the Intellec Microcomputer Development System. The resident macroassembler, text editor, and system monitor greatly simplify the design, development, and debug of the system software. Optional Diskette Operating Software for the Development System programs to be loaded, assembled, edited, and executed faster than using conventional paper tape, card, or cassette peripherals. A unique In-Circuit Emulator (ICE-85) option provides the capability to use the Development System to develop and debug software directly on your system.

Intel's high-level resident programming language, PL/M-80, provides the capability to program in a natural, algorithmic language and eliminates the need to manage register usage or allocate memory. PL/M-80 programs usually can be written in a much shorter time than assembly language programs.

* * *

CHAPTER 4
PRINCIPLES OF OPERATION

4-1. INTRODUCTION

This chapter is divided into two basic subject areas: a description of each major functional block and more detailed circuit analysis of each block. The first area, sections 4-2 through 4-12, provides a basic functional description of the major iSBC 80/24A board elements. The second area, sections 4-13 through 4-35, provides an in-depth look at the internal workings of each functional block.

4-2. FUNCTIONAL DESCRIPTION

A brief description of each iSBC 80/24A board functional block is provided in sections 4-3 through 4-12. Figure 4-1 provides a simplified block diagram of the iSBC 80/24A board.

4-3. TIMING

On-board timing is generated by the 8224 Clock Generator (U21) in conjunction with a step-down counter (U20). A 19.3536 MHz crystal (Y1) is used as a reference. These circuits produce three basic timing rates: a 9.68 MHz signal used for CPU input, a 2.15 MHz signal for PIT input, and an optional 4.84 MHz input signal for slower CPU operation. The 9.68 MHz signal is also gated off-board as BCLK/ and CCLK/.

4-4. CENTRAL PROCESSING UNIT (CPU)

The 8085A-2 Microprocessor (CPU) performs system processing functions and generates the address and control signals required to access memory and I/O devices. The AD0-AD7 pins are used to multiplex the 8-bit input/output data and the lower eight

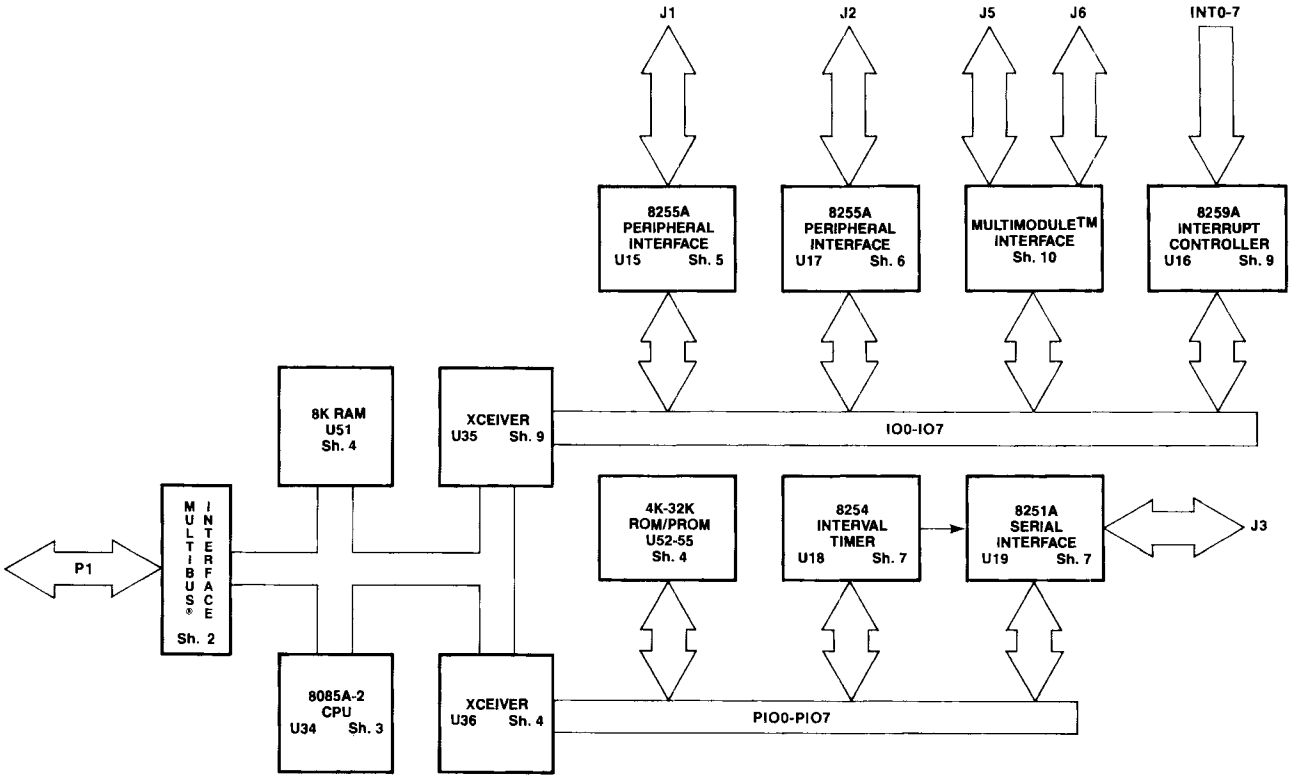


Figure 4-1. iSBC® 80/24A Board Simplified Block Diagram

bits of address are strobed into Latch U64 by the Address Latch Enable (ALE) signal; the outputs of U64 are combined with the upper eight bits of the address to form the 16-bit address bus. During the remainder of the machine cycle, AD0-AD7 pins of the CPU are used for data input/output.

4-5. INTERVAL TIMER

The 8254 Programmable Interval Timer (PIT) includes three independently controlled counters which are used for on-board I/O and CPU interrupts. Each counter has its own input and output. Counter 0 is used as the CPU interrupt interval, connected to IR2 on the interrupt controller. It can also be used as an input to counter 1. Counter 1 can also be routed to the interrupt matrix or off-board via the parallel interface. Counter 2 is used exclusively for the baud rate timer, which is used to clock the serial interface.

4-6. SERIAL I/O

The 8251A PCI provides RS232C compatibility and is configured as a data set. Synchronous or asynchronous mode, character size, parity bits, stop bits, and baud rates are all programmable. Data, clocks, and control lines to and from connector J3 are buffered with drivers and receivers.

A second serial I/O channel may be derived by jumper connection, in conjunction with port EA. (Port EA is controlled by 8255A PPI device.)

4-7. PARALLEL I/O

The two 8255A Programmable Peripheral Interface devices provide 48 programmable I/O lines. Two bus transceivers are included to interface 16 of the I/O lines to connector J1 and J2. Four IC sockets are provided so that, depending on the application, TTL drivers or I/O terminators may be installed to complete the interface to connectors J1 and J2. The 48 lines are grouped into six ports of eight lines each. These ports can be programmed to be simple I/O ports or strobed I/O ports with handshaking. Two ports can be programmed as bidirectional ports with control lines. The iSBC 80/24A board includes various features such as RS232C interface lines, timer gates, and interrupts that can be controlled by the parallel I/O lines.

4-8. INTERRUPT CONTROL

The interrupt section provides 12 priority interrupts. Eight interrupts connect to the inputs of the 8259A Programmable Interrupt Controller (PIC); the remaining four interrupts connect directly to the 8085A-2 CPU. All interrupts, except TRAP, are maskable; the TRAP interrupt may be used to handle a power-fail (PFIN/) signal input via auxiliary connector P2.

There are 23 jumper-selectable interrupt sources: PPI (4), PCI (3), PIT (2), external via J1 and J2 (2), Multibus lines (8), and Multimodule boards (4).

4-9. ROM/EPROM CONFIGURATION

Four sockets (U52-55) are provided for ROM/EPROM devices. The sockets are 28-pin DIP receptacles, and may be used for both 24-pin and 28-pin devices. (Refer to section 2-8 for installation instructions.) Either 1K, 2K, 4K or 8K ROM/EPROM devices may be used.

In the default configuration, ROM/PROM addressing is set for 8K × 8 devices from 0-7FFFH (device types cannot be mixed). If you do not have an on-board ROM/PROM, this address space can be assigned off-board by using the PROM ENABLE signal from port E6 of the 8255A device.

The PROM enable signal enables or disables the on-board PROM. If the PROM is disabled (PROM ENABLE = 0), then all memory addresses not recognized as on-board RAM addresses are assumed to be off-board addresses. The PROM ENABLE signal originates at jumper post E32, which may be connected to an 8255A data line or a ground jumper post, or it may be left disconnected. PROM ENABLE jumpered to an 8255A output allows software control of on-board PROM access. If PROM ENABLE is connected to ground, no on-board PROM accesses are made. If PROM ENABLE is left disconnected, a 10K-ohm pull-up resistor holds PROM ENABLE = 1.

PROM Enable Example:

To shadow on-board PROM over off-board memory by using bit 5 of port E6 (port C of 8255A at location U15) to control PROM ENABLE:

Install jumper: E25-E32 (PC5-PROM ENABLE)

Design software to:

- Initialize port E6, bit 5 into the output mode
- Set port E6, bit 5 to a one to access on-board PROM
- Reset port E6, bit 5 to a zero to disable on-board PROM

NOTE

After power-up or system reset, all the 8255A ports are in the input mode, and since the PROM ENABLE signal is pulled high by a 10K-ohm resistor, the on-board PROM is enabled.

4-10. RAM CONFIGURATION

The iSBC 80/24A board has three possible RAM configurations: 8K bytes (default configuration), 4K bytes (accomplished by disabling the lower 4k bytes of the default configuration), and 2K bytes (accomplished by disabling the lower 6K bytes of the default configuration). Addressing for the default 8K bytes configuration is set by jumpers to E000-FFFFH. Refer to Table 2-4 for RAM jumper connections.

In an application where the entire 64K address space must be accessed as system memory, you can disable the on-board RAM by removing all jumpers from posts E102, 103, 111, 112 and 113.

The INTROUT/BANKSEL signal can be jumpered to Multibus line ADR10/ to increase addressing capability to 128K bytes. INTROUT/BANKSEL can originate from one of several possible signals including the 8225A's port C bits and the 8085A-2's SOD signal. The INTROUT/BANKSEL line includes a 74S38 open-collector driver which either drives low or presents a high impedance.

Bank Select Example:

To address two 64K-byte banks of memory by using bit 7 of port E6 (port C of the 8255A at location U15) to control the Multibus line ADR10/:

Install jumpers:
E16-E27 (INTROUT/BANKSEL from PC7)
E206-E211 (INTROUT/BANKSEL to ADR10/)

Design software to:

- Initialize port E6, bit 7 into the output mode
- Address memory addresses 00000H through 0FFFFH by setting port E6, bit 7 to a one, then performing a normal memory read or write using address lines ADR0-ADRF
- Address memory addresses 10000H through 1FFFFH by resetting port E6, bit 7 to a zero, then performing a normal memory read or write using address lines ADR0-ADRF

NOTE

The iSBC 80/24A's on-board RAM and PROM (if enabled) are addresses in both banks. To use the bank select feature in a multimaster configuration, be sure to lock the Multibus whenever ADR10/ is driven.

4-11. MULTIMODULE™ BOARDS

Optional Multimodule boards are interfaced to the iSBC 80/24A board via iSBX bus connectors J5 and J6. If present, the Multimodule board at J6 will utilize addresses F0-FF. Likewise, if present, the J5 Multimodule board will utilize addresses CO-CF. If neither of the boards are present, these addresses may be used for other applications.

4-12. MULTIBUS™ INTERFACE CONTROL

The Multibus interface includes an Intel Bus Controller, bidirectional address an data bus drivers, and the bus interrupt driver/receivers. The bus controller allows the iSBC 80/24A board to operate as a bus master in a serial or parallel priority arrangement with other bus masters in the system in which the 8085A-2 CPU can request the Multibus lines only when needed.

4-13. CIRCUIT ANALYSIS

The schematic diagram for the iSBC 80/24A board is provided in figure 5-4, sheets 1 through 10. Signals which traverse from one sheet to another are identified with an alpha character(s) in a box (e.g., G or AK). Each boxed character(s) will represent the same signal throughout figure 5-4. To follow a particular signal from one sheet to another, merely turn to the sheet indicated next to the signal name. After you turn to the indicated sheet, look for the signal's boxed character. Signals which enter or exit the board do not have boxed characters.

Both active-low and active-high signals are used on the iSBC 80/24A board. A signal which ends with a slash (e.g., READ/) is active-low, meaning it will be true at $\leq$ 0.4 volts. Conversely, a signal without the slash (e.g., ALE) is active-high, meaning it will be true at $\geq$ 2.4 volts.

The following sections make reference to the signals in figure 5-4.

4-14. INITIALIZATION

When power is applied in a start-up sequence, the contents of the 8085A-2 CPU program counter, instruction register, and interrupt enable flip-flop are subject to random factors and cannot be predicted. For this reason, a power-up sequence is used to set the CPU, bus controller, and I/O ports to a known internal state.

The 8224 clock generator produces the RESET signal at power-up time (figure 5-4, sheet 3). This signal is routed to the CPU ad throughout the board. If power is already on, the RESET signal may be generated by the AUX RESET/signal (front panel switch closure). The RESET signal is routed off-board via P1-14, as INIT/. Notice that INIT/ may also be an input from P1-14.

The RESET signal clears the CPU program counter, instruction register, and interrupt enable flip-flop; sets the parallel I/O ports to the input mode; resets the serial I/O port to the "idle" mode; resets the bus controller; and, via the INIT/signal sets the system to a known internal state.

4-15. CLOCK CIRCUITS

The 8224 Clock Generator (U21) OSC output is divide-by-2 (at U20) to produce the 9.68 MHz reference signal (figure 5-4, sheet 3). A divide-by-4 signal (4.84 MHz) is available by jumper connection. The reference signal is routed to three areas:

a. 8085A-2 CPU (sheet 3)

b. iSBX Bus connectors J5 and J6 (sheet 10)

c. BCLK/and CCLK/ gates (sheet 2)

The 2.15 MHz 02TTL signal, produced by the 8224 Clock Generator, is used as a reference for the 8254 PIT (sheet 7). This 2.15 MHz signal is reduced to 1.075 MHz by counter U20, and serves as the CKL0, CLK1 and CLK2 inputs for the PIT.

4-16. 8085A-2 CPU TIMING

The 8085A-2 CPU internally divides the 9.68 or 4.84 MHz clock input by two to develop the timing requirements for the various time-dependent functions. These functions are described in the following paragraphs.

4-17. INSTRUCTION TIMING

The execution of any program consists of read and write operations, where each operation transfers one byte of data between the CPU and memory or between the CPU and an I/O device.

An instruction cycle is the time required to fetch and execute an instruction. During the fetch phase, the selected instruction (consisting of up to three bytes) is read from memory and stored in the operating registers of the CPU. During the execution phase, the instruction is decoded by the CPU and translated into specific processing activities.

Each instruction cycle consists of up to five machine cycles. A 'machine cycle' is required each time the CPU accesses memory or an I/O device. The fetch phase requires one machine cycle for each byte to be fetched. Some instructions do not require any machine cycles other than those necessary to fetch the instructions from memory; other instructions, however, require an additional machine cycle(s) to write or read data to or from memory or I/O devices.

Every instruction cycle has at least one reference to memory during which time the instruction is fetched. An instruction cycle must always have a fetch, even if the execution of that instruction requires no reference to memory. The first machine cycle in every instruction cycle is therefore a fetch, and beyond that there are no specific rules. For instance, the IN (input) and OUT (output) instructions each require three machine cycles: fetch (to obtain the instruction), memory read (to obtain the I/O address of the device), and an input or output machine cycle (to complete the transfer).

Each machine cycle consists of a minimum of three and a maximum of six states (excluding any WAIT states) designated T_1 through T_6. A 'state' is the smallest unit of processing activity and is defined as the interval between two successive falling edges of the CPU clock. Each state (or CPU clock cycle) has a duration of 206 nanoseconds (assuming 9.68 MHz input clock).

Every machine cycle normally consists of three T-states with the exception of an opcode fetch, which consists of either four or six T-states (plus WAIT states). The actual number of states required to execute any instruction depends on the instruction being executed, the particular machine cycle within the instruction cycle, and the number of 'wait' states inserted into the machine cycle. The wait state is generated by circuitry external to the CPU.

At the 4.84 MHz operating speed there are no wait states imposed when the CPU is addressing on-board RAM. There will be wait states imposed, however, in the following operations (at 4.84 MHz):

a. When addressing on-board ROM/EPROM with access time greater than 300 ns;

b. For interrupt acknowledge cycles; or

c. When addressing off-board RAM, PROM, or I/O devices via the Multibus lines, or iSBX Bus.

At the 2.42 MHz operating speed, no wait states are required for any on-board requests but are required for iSBX Multimodule operation, and other off-board requests. Figure 4-2 is presented to show the relationship between an instruction cycle, machine cycle, and T-state. This example shows the execution of a Store Accumulator Direct (STA) instruction involving on-board memory. Notice that for this instruction the opcode fetch (machine cycle M_1) requires four T-states and the remaining three cycles each require three T-states.

The opcode fetch is the only machine cycle that requires more than three T-states. This is because the CPU must interpret the requirements of the opcode fetched during T_1 through T_3 before it can decide what must be done in the remaining T-state(s).

There are seven types of machine cycles, each of which can be differentiated by the states of three CPU status lines (IO/M, S0, and S1) and three CPU control lines (RD, WR, and INTA). Table 4-1 lists the states of the CPU status and control lines during each of the seven machine cycles and during a CPU halt.

Table 4-1. CPU Status & Control Lines

Machine Cycle	Status			Control		
	IO/$\overline{M}$	S0	S1	$\overline{RD}$	$\overline{WR}$	$\overline{INTA}$
Opcode Fetch	0	1	1	0	1	1
Memory Read	0	0	1	0	1	1
Memory Write	0	1	0	1	0	1
I/O Read	1	0	1	0	1	1
I/O Write	1	1	0	1	0	1
INTR Acknowledge	1	1	1	1	1	0
Bus Idle	X	X	X	1	1	1
Halt	TS	0	0	TS	TS	1

0 = Logic "0" TS = High Impedance
1 = Logic "1" X = Unspecified

4-18. OPCODE FETCH TIMING. Figure 4-3 shows the timing relationship of a typical opcode fetch machine cycle. At the beginning of T_1 of *every* machine cycle, the CPU performs the following:

a. Pulls IO/M low to signify that the machine cycle is a memory reference operation.

b. Drives status lines S0 and S1 high to identify the machine cycle as an opcode fetch.

c. Places high-order bits (PCH) of program counter onto address lines A8-A15. These address bits will remain true until at least T_4.

d. Places low-order bits (PCL) of program counter onto address/data lines AD0-AD7. These address bits will remain true for only one clock cycle, after which AD0-AD7 go to their high-impedance state as indicated by the dashed line in figure 4-3.

e. Activates the Address Latch Enable (ALE) signal.

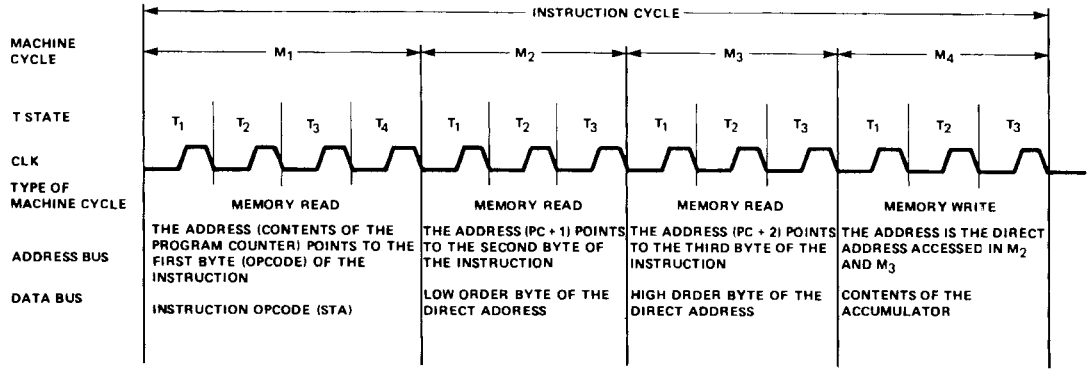


Figure 4-2. Typical CPU Instruction Cycle

At the beginning of T_2, the CPU pulls the RD/ line low to enable the addressed memory device. The device will then drive the AD0-AD7 lines. After a period of time, as determined by the access time of the addressed memory device, valid data (the DCX instruction in this example) will be present on the D0-D7 lines. During T_3 the CPU loads the data on the D0-D7 lines into its instruction register and drives RD/ high, disabling the addressed memory device. During T_4 the CPU decodes the opcode and decides whether or not to enter T_5 on the next clock cycle or start a new machine cycle and enter T_5 and then T_6 before beginning a new machine cycle.

Figure 4-4 is identical to figure 4-3 with one exception, which is the use of the READY input to the CPU. As shown in figure 4-4 the CPU examines the state of the READY input during T_2. If the READY input is high, the CPU will proceed to T_3 as shown in figure 4-3. If the READY input is low, however, the CPU will enter the T_{wait} state and stay there until READY goes high. When READY goes high, the CPU will exit the T_{wait} state and enter T_3. The external effect of using the READY input is to preserve the exact state of the CPU signals at the end of T_3 for an integral number of clock periods before finishing the machine cycle. This 'stretching' of the system timing, in effect,

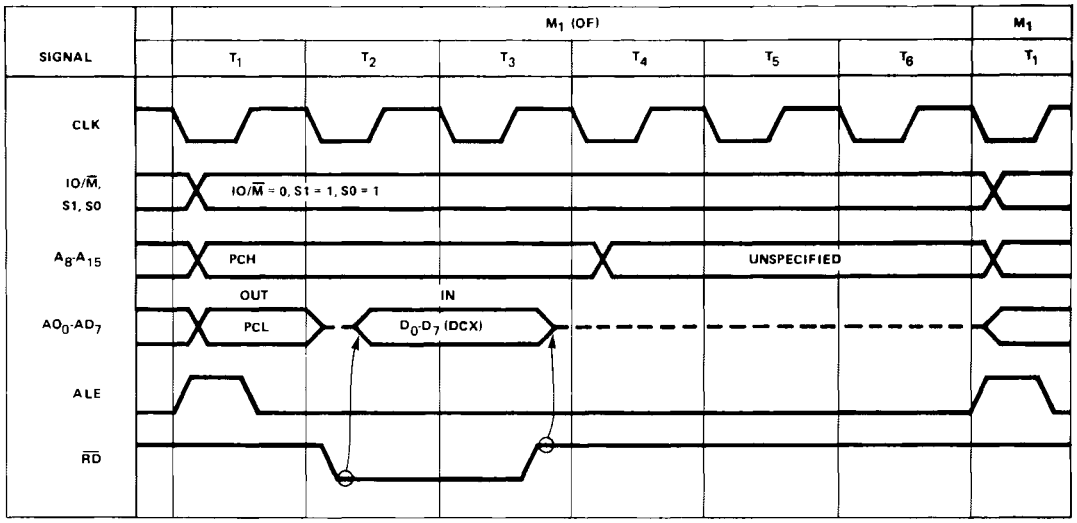


Figure 4-3. Opcode Fetch Machine Cycle

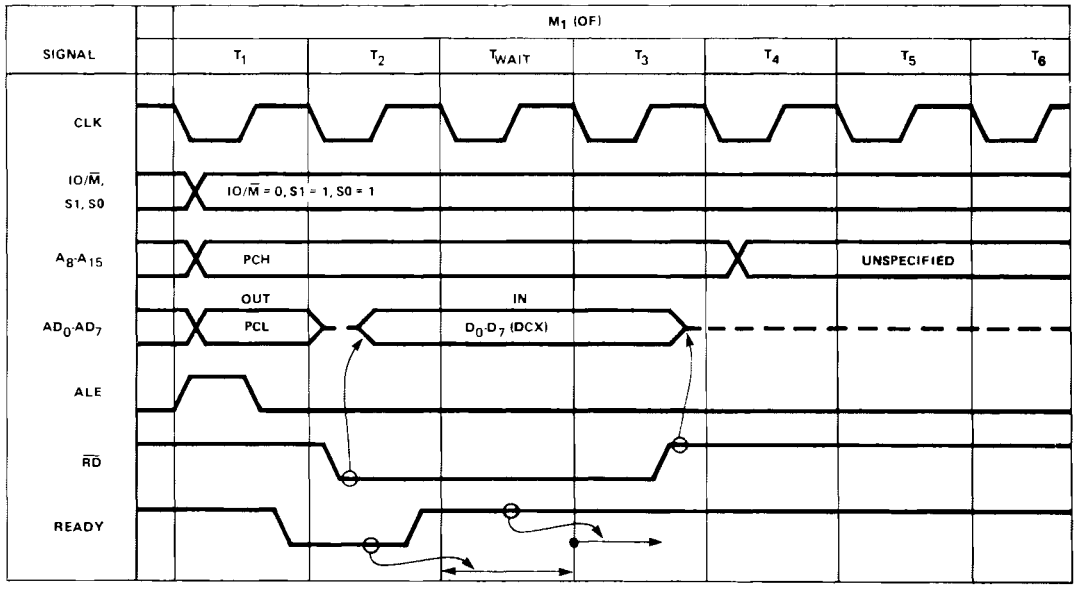


Figure 4-4. Opcode Fetch Machine Cycle With Wait State

increases the allowable access time for memory or I/O devices. By inserting T_{wait} states, the CPU can accommodate slower memory or slower I/O devices. It should be noted, however, that access to the on-board ROM/PROM and I/O ports usually imposes a T_{wait} state only when operating at 4.84 MHz. T_{wait} states are always imposed when accessing system memory or I/O devices via the Multibus lines.

4-19. MEMORY READ TIMING. Figure 4-5 shows the timing of two successive memory read machine cycles, the first without a T_{wait} state and the second with one T_{wait} state. Disregarding the states of the S0 and S1 lines, the timing during T_1 through T_3 is identical with the opcode fetch machine cycle shown in figure 4-3. The major difference between the opcode fetch and memory read cycles is that an opcode fetch machine cycle requires four or six T-states whereas the memory read machine cycle requires only three T-states. One minor difference between the two cycles is that the memory address used for the opcode fetch cycle is always the contents of the program counter (PC), which points to the current instruction; the address used for a memory read cycle can be one of several origins. Also, the data read from memory is placed into the appropriate register instead of the instruction register. Note that T_{wait} state is not imposed during a read of on-board ROM/PROM.

4-20. I/O READ TIMING. Figure 4-5 also illustrates the timing of two successive I/O read machine cycles, the first without a T_{wait} state and the second with one T_{wait} state. With the exception of the IO/M status signal, the timing of a memory read cycle and an I/O read cycle is identical. For an I/O read, IO/M is driven high to identify that the current machine cycle is referencing an I/O port. One other minor exception is that the address used for an I/O read cycle is derived from the second byte of an IN instruction; this address is duplicated onto both the A8-A15 and AD0-AD7 lines. The data read from the I/O port is always placed in the accumulator specified by the IN instruction. Note that in the 2.42 MHz mode a T_{wait} is not imposed during the access of on-board I/O devices; T_{wait} states are imposed during the access of system I/O devices via the Multibus lines.

4-21. MEMORY WRITE TIMING. Figure 4-6 shows the timing of two successive memory write machine cycles, the first without a T_{wait} state. Again, disregarding the states of the S0 and S1 lines, the timing during T_1 is identical to the timing of an opcode fetch, memory read, and I/O read cycles. The difference occurs, however, at the end of T_1. For instance, in a memory read cycle the AD0-AD7 lines are disabled (high impedance) at the beginning of T_2 in anticipation of the returned data. In a memory write cycle, the AD0-AD7 lines are not disabled and

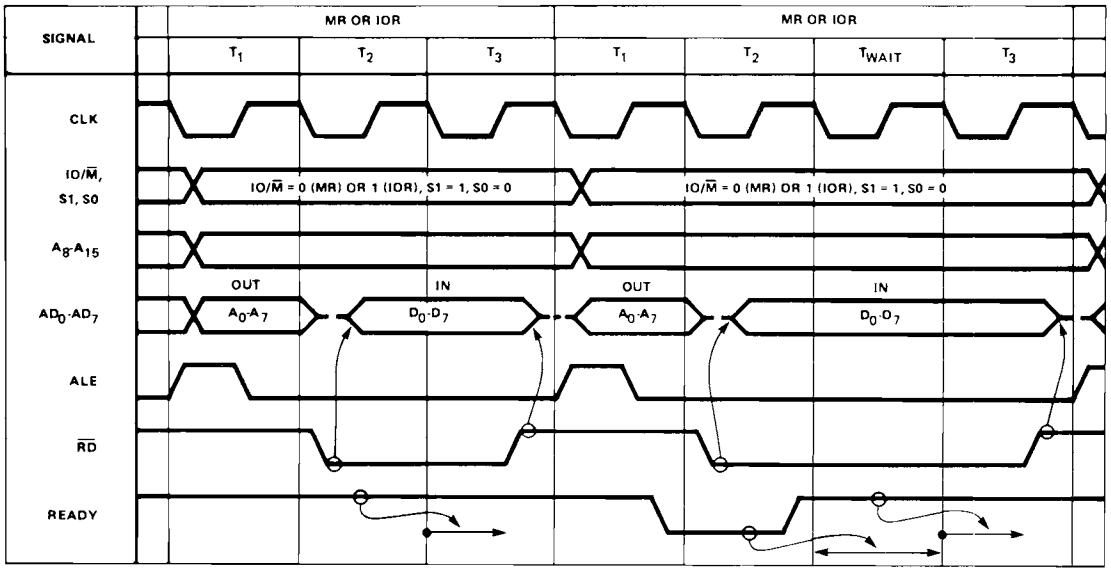


Figure 4-5. Memory or I/O Read Machine Cycle

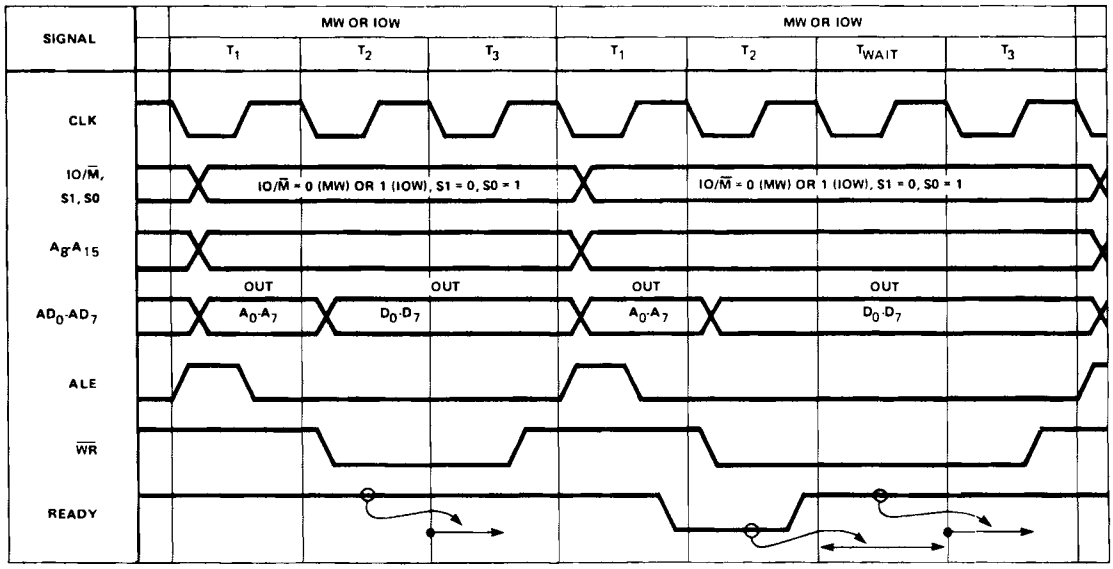


Figure 4-6. Memory or I/O Write Machine Cycle

the data to be written into memory is placed on these lines at the beginning of T_2. The Write (WR/) line is driven low at this time to enable the addressed memory device. During T_2 the READY input is checked to determine if a T_{wait} state is required. If the READY input is low, T_{wait} states are inserted until READY goes high. During T_3, the WR/ line is driven high to disable the addressed memory device and terminate the memory write operation. Note that the contents on the address and data lines do not change until the next T_1 state.

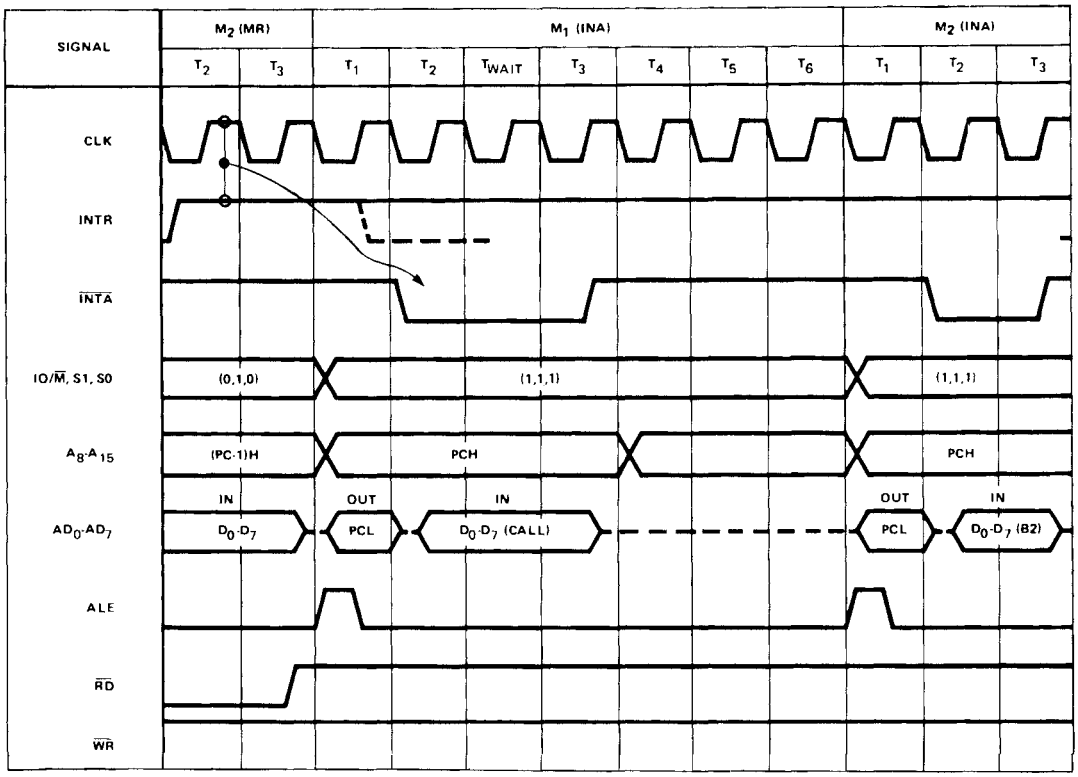
4-22. I/O WRITE TIMING. Figure 4-6 also illustrates the timing of two successive I/O write machine cycles, the first without a T_{wait} state and the second with one T_{wait} state. With the exception of the IO/M status signal, the timing of a memory write cycle and an I/O cycle are identical.

4-23. INTERRUPT ACKNOWLEDGE TIMING. Figure 4-7 shows the CPU timing in response to the INTR input being driven high by PIC U17 (assuming the CPU interrupt enable flip-flop has been set by a previously executed Enable Interrupt instruction). The status of the TRAP, RST 7.5, RST 6.5, RST 5.5, and INTR inputs are sampled during CLK of the T-state immediately preceding T_1 of M_1. If INTR was the only valid interrupt, the CPU clears its interrupt enable flip-flop and enters the Interrupt Acknowledge (INA) machine cycle. With two exceptions, the INA machine cycle is identical with the Opcode Fetch (OF) machine cycle. The first exception is that IO/M = 1, which signifies that the opcode fetched will be from an I/O device. The second exception is that INTA is asserted instead of RD. Although the contents of CPU program counter are sent out on the address lines, the address lines are ignored.

When INTA is asserted, the PIC provides a CALL instruction which causes the CPU to push the contents of the program counter onto the stack before jumping to a new location. After receiving the CALL opcode, the CPU performs a second INA machine cycle (M_2) to access the second byte of the CALL instruction from the PIC. The timing of M_2 is identical with M_1 except that M_2 has three T-states. M_2 is followed by M_3 to access the third byte of the CALL instruction. When all three bytes have been received, the CPU executes the instruction. The CPU inhibits the incrementing of the program counter during the three INA cycles so that the correct program counter value can be pushed onto the stack during M_4 and M_5.

During M_4 and M_5, the CPU performs Memory Write (MW) machine cycles to write (push) the contents of the program counter onto the top of the stack. The CPU then places the two bytes access during M_2 and M_3 into the upper and lower bytes of the program counter. This has the same effect as jumping the execution of the program to the location specified by the CALL instruction.

After the interrupt service routine is executed, the CPU pops the stack and loads it into the program counter, and resumes system operation at the point of the interrupt. (It is the programmer's responsibility to ensure that the interrupt enable flip-flop is set before returning from the service routine.)



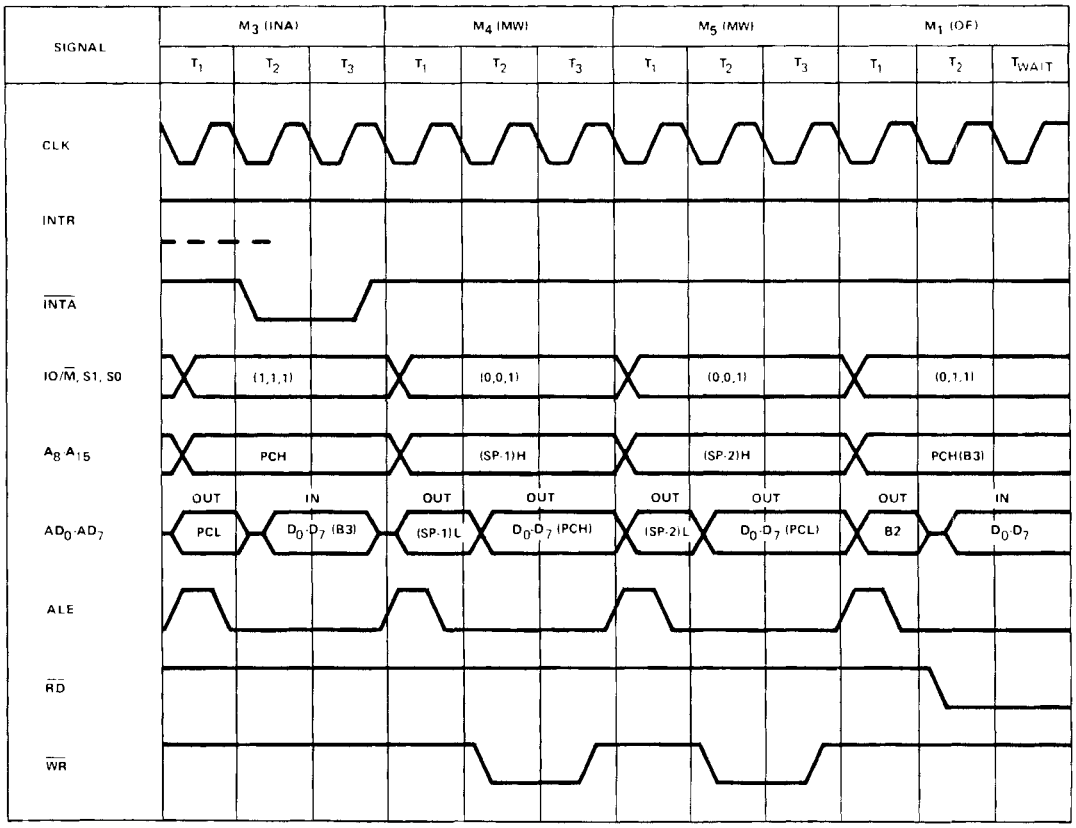


Figure 4-7. Interrupt Acknowledge Machine Cycles

4-24. ADDRESS BUS

The lower eight bits (AD0-AD7) of the memory address or I/O address (depending on whether a memory reference machine cycle or an I/O reference machine cycle is in progress) are output by the CPU during the first clock cycle (T_1). The CPU AD0-AD7 pins become the source to or input from the data bus during the second and third cycles (T_2 and T_3). The trailing edge of the Address Latch Enable (ALE) signal issued by the CPU during T_1 strobes these eight address bits into Latch U64 (sheet 3). The lower eight address bits (A0-A7) from U64 are placed on the address bus together with the high-order address bits (A8-AF). These address bits are routed throughout the board and off-board via the Multibus address drivers (U60, 61, sheet 2).

Address bits AA through AF are routed to the Decode PROM (U32) on sheet 8. This PROM is a factory programmed deice designed to generate most of the key I/O signals used on the board. Decode PROM operation is described in section 4-30, "Board I/O Operation". Appendix B provides a Decode PROM map and structure description.

4-25. BUS TIME-OUT

Bus Time-Out one-shot U39-12 (sheet 3) is retriggered by the leading edge of the ALE signal, which is asserted by the CPU during T_1 of every machine cycle. If the CPU is halted, or if the CPU is in a wait-state for approximately 10 milliseconds, U39 times out and asserts the BTMO signal on pin 34 of the auxiliary but (P2). If jumper E97-106 is installed, the BUS TIME OUT/ signal (U39-5) drives the CPU READY line high through U31 and allows the CPU to exit the wait state.

4-26. DATA BUS

The CPU ADO-AD7 pins become the source or destination of the data but DBO-DB7 during T_2 and T_3 clock cycles. Data can be sourced to or input from the following:

a. Data Latch/Driver U62 (sheet 2).

b. Data Bus Transceiver U36 (sheet 4).

c. RAM device U51 (sheet 4).

d. Data Latch/Driver U35 (sheet 9).

Latch/Driver U35 (sheet 9) also serves as the data bus latch/drive for each PPI device (U15, U17) and the Multimodule board devices (sheet 10). Bus Transceiver U36 serves as the data bus receiver/driver for the PIC (U19) and the PIT (U18).

4-27. READ/WRITE SIGNAL GENERATION

The COMMAND/ signal, which is used in conjunction with the various on-board read/write operations, is generated when the CPU RD/ or WR/ signal is true. The following paragraphs describe how the various I/O and memory control signals are generated.

4-28. I/O CONTROL SIGNALS

The I/O control signals are derived simply by gating the status of the CPU IO/$\overline{M}$, RD/, and WR/ pins. For example, the IO/ signal is the buffered IO/$\overline{M}$ pins status; the I/O WRT/ signal is the logical AND of the IO/ and WR/ pin status.

4-29. MEMORY CONTROL SIGNALS

The MRD/ and MWRT/ signals (sheet 3) are also derived simply by gating the status of the CPU IO/$\overline{M}$, RD/, and WR/ pins. For example, the MWRT/ signal is the logical AND of the WR/ and inverted IO/$\overline{M}$ pin status.

The CPU signals $\overline{RD}$ and $\overline{WR}$ are routed to the RAM (sheet 4) as 85RD/ and 85WRT/. These signals are used by the RAM device to indicate Read or Write mode.

The Bus Controller (U57, sheet 2) uses the MRD/ and MWRT/ signals to control off-board memory transactions.

The signal READ/ is used to gate the U63 receiver device (sheet 2). This device is used for all off-board memory read transactions.

The CPU signal S1 (sheet 3) is used to establish the mode of the U36 bus transceiver. All on-board PROM data is routed through this device during the read cycle.

If no on-board RAM address is decoded by the 74LS138 (U28, sheet 4), the RAMACK/ is false and the bus controller will assume an off-board read/write request has been made. The off-board request will be acknowledged by XACK/(P1-23, sheet 2) which subsequently produces the MULTIBUS RDY/ signal. This in turn, provides the CPU ready signal.

For an on-board ROM/EPROM read, address bits A0-AF are routed to PROM jumper matrix E177-E197 and the ROM/EPROM devices. bits A0-A9 are used for addressing 4K and smaller devices; bits

A0-A9 and AC are used for 8K devices. The signals PTYPE0 and PTYPE1 indicate the amount of on-board PROM. The following must be true for an on-board PROM read to occur:

a. Address must be recognized by Decode PROM U32 (sheet 8) to produce either DCDPROM1/ or DCDPROM2/.

b. The CPU IO/$\overline{M}$ signal must be false (IO/ = 1).

c. PROMACK/ must be true.

d. PROM ENABLE must be true.

When these conditions occur, PROMACK/ will disable the RAM chip select decoder. The DCDPROM signals will decode one-of-four on-board ROM chip selects to enable one of the ROM/EPROM devices. This decoder (U49, sheet 4) is gated by PROMACK/ and 85RD/.

If the ROM read is an off-board request, there will be no on-board acknowledge signal and wait states will be inserted by U31 (sheet 3) until XACK is received to produce MULTIBUS RDY/ (sheet 2). Wait states will be inserted for 10ms maximum (refer to section 4-35).

4-30. I/O OPERATION

The iSBC 80/24A board must distinguish between on-board and off-board I/O operations. The Decode PROM (U32, sheet 8) examines address bits AA-AF to determine if the current address is valid. Several other qualifiers are used by the Decode PROM to establish whether the address is on or off-board. The signal IO/ must be true to indicate an I/O request rather than memory. The signal MMPRESENT/ will be true if one or both iSBX Bus connectors (sheet 10) have board installed.

4-31. ON-BOARD I/O OPERATION

If the address is determined to be an on-board location, the signal ONBDIO (sheet 8) will be true (ONBDIO = 1). This signal is routed to the bus controller, preventing a bus cycle and to the Multibus data receivers (U63, sheet 2), turning them off. At the same time, the Decode PROM will enable one of the I/O chip select signals, via decoder U48 (sheet 8).

Each of these I/O chip select signals indicates its destination (e.g., 59CS/ enables the 8259A PIC). Port addresses are also shown.

Multimodule chip select signals (MMCSO/-MMCS3/) are selected by further decoding address bits AB and AC. This decoder (U49, sheet 8) is gated by bit 3 of the U48 chip select decoder.

Two signals can be used to furnish the required CPU READY signal. At 4.84 MHz operation, if the addressed, on-board I/O device is not a Multimodule board, the signal IOSYNC/ (sheet 8) will furnish the READY signal, via the wait-state circuitry (sheet 3). If the CPU is operating at the slower speed (2.42 MHz), IOSYNC/ will provide a READY signal without a wait state, via jumper E96-105 (sheet 3).

If the addressed, on-board I/O device is on a Multimodule board, the signal IOASYNC/ will supply the required READY signal (via wait-state circuitry). A minimum of one wait-state is required when accessing a Multimodule board.

4-32. OFF-BOARD I/O OPERATION

An off-board I/O operation is characterized by the signal ONBDIO being false (ONBDIO = 0), and the absence of any on-board acknowledge (ACK, sheet 3) signal. These two conditions will allow the bus controller (U57, sheet 2) to institute a bus cycle, provided the command signal (CMD, sheet 2) is true. At the same time, the Multibus data receivers (U63, sheet 2) will be turned-on. If an XACK/ signal is not received within 10 ms of command time, the failsafe timer will produce the required READY signal (U39, sheet 3).

4-33. MULTIBUS® INTERFACE

The Multibus Interface allows the iSBC 80/24A board to use a common system bus with other master devices, thus sharing memory and I/O resources. The Multibus Interface (sheet 2) consists of a 3218 bus controller (U57), a data latch/driver (U62), a data receiver (U63), and address drivers (U60, 61).

The bus controller arbitrates all iSBC 80/24A board requests for use of the Multibus lines, synchronously with respect to the Bus Clock (BCLK/). When the iSBC 80/24A board acquires control of the bus, the bus controller generates the appropriate memory or I/O command signal, gates data on/off the bus, at the appropriate times. An external RC network connected the bus controller's Delay-Adjust (DLYADJ/) input guarantees the required setup and hold time relationship between the address/data lines and the control signals.

The negative-going edge of the Bus Clock (BCLK/) signal provides a timing reference for the controller's bus arbitration logic.

Bus arbitration activity begins when an off-board memory or an I/O request is generated and applied to the Bus Request (BCR1) and Transfer Start Request (XSTR) inputs. The request is strobed in by RSTB/ (which is always enabled). Following the next rising and falling edge of BLCK/ the bus controller generates Bus Request, BREQ/ (connector pin P1-18) and forces Bus Priority Out, BPRO/ (connector pin P1-16) inactive (high). BREQ/ is used to request the system bus when priority is decided by a parallel priority resolution circuit (section 2-21). BPRO/ is used to allow lower priority masters to gain control of the bus when a serial priority resolution structure is used (section 2-20). BPRO/ would go to the Bus Priority In (BPRN/) input of the next lower priority master.

When an off-board request activates the Transfer Start Request input (XSTR) and ADEN/ is activated, the bus controller's timing logic starts the internal sequence which ultimately (depending on the RC network at DLYADJ/) generates the appropriate memory/IO read/write control output (MRDC/, MWTC/, IORC/, or IOWC/) based on the active command request input (MRD/, MWRT/, IORD/ or IOWRT/) from the CPU.

When control of the bus is granted to the iSBC 80/24A board, a low level appears on its Bus Priority In (BPRN/) input. The bus controller activates its BUSY/ (connector pin P1-17) and Address and Data Enable (ADEN/) outputs. BUSY/ "locks" the iSBC 80/24A board onto the bus by prohibiting any other master from acquiring control of the bus. ADEN/ enables the system address drivers (U60, 61) and data latch/drivers (U62). When the external acknowledge signal (XACK/ or AACK/) is received it will be grated (as MULTIBUS RDY/) the CPU RDY input via the wait circuity (sheet 3).

The iSBC 80/24A board can only lose control of the Multibus lines if its BPRN/ input goes high or the bus request inputs (ACK • ONBDIO • CMD) go inactive causing the Transfer Complete input (XCP/) to be activated. In either case, the iSBC 80/24A board will only lose the buse if it is not in the middle of a transfer and its override flip-flop is not set (see section 4-34).

A low level on the bus controller's INIT/ input will initialize (reset) the device.

4-34. MULTIBUS® OVERRIDE FEATURE

If it is necessary to guarantee that the iSBC 80/24A board not lose control of the Multibus lines, the override function can be invoked by executing an OUTPUT instruction to port D5 (hexadecimal). If data bit 0 is a "1" when this output instruction is executed, the Override flip-flop (U31, sheet 2) is set. While this flip-flop is set, the board cannot relinquish control of the Multibus lines. If data bit 0 is "0" when the output instruction to port D5 is executed, or if the Reset signal (RST/) is activated, the Override flip-flop is reset. The iSBC 08/24A board must clear the Override capability when it is finished with exclusive use of the Multibus lines.

4-35. FAILSAFE TIMER

If the CPU tries to access a memory or I/O device but that device, for whatever reason, does not return an acknowledgment (READY) indication, the 8085A-2 remains in a WAIT state until READY is received. The failsafe timer is designed to prevent holding the system in the WAIT condition. A 74LS123 one-shot device (U39, sheet 3) is triggered by ALE at the beginning of each machine cycle. If the one-shot is not re-triggered (i.e., if another cycle does not begin) with approximately 10 ms, the 74LS123 times-out and its output is gated through to the RDY pin on the 8085A-2, thus ending the WAIT state. The iSBC 80/24A board is shipped with the failsafe timer enabled. To disable the timer, remove jumper E97-106.

4-36. INTERRUPT OPERATION

Interrupt circuity is shown on sheet 9. The major components include the jumper matrix, the 8259A PIC, and the 8286 Bus Transceiver.

The jumper matrix is shown in its factory configuration. External interrupt requests (INT0/-INT7/, INTR/, PFIN/) enter the board via connectors P1 and P2. These interrupts may be assigned priority by jumper connection to the appropriate posts. Similarly, internal interrupt requests may be jumper connected to establish priority. Internal requests include:

OIT0	—	Timer 0
OIT1	—	Timer 1
RxR	—	Receiver Ready
TxR	—	Transmitter Ready
TxE	—	Transmitter Empty

MIOINT0 —
MIOINT1 —
MIOINT2 —
MIOINT3 — } Multimodule Board Interrupts

PIA1
PIB1
PIA2
PIB2 } Parallel Port Interrupts

Jumper connections are shown in table 2-7.

In addition to the interrupt requests which are handled by the 8259A PIC, four lines are direct inputs to the 8085A-2 restart functions. These functions are described in section 3-37.

The iSBC 80/24A board can generate external interrupts (via P1-28) by jumpering the INTROUT/BANKSEL signal to any of the Multibus interrupt lines (INT0-INT7). INTROUT/BANKSEL can originate from one of several possible signals including the 8255A's port C bits and the 8085A-2's SOD signal. The INTROUT/BANKSEL line includes a 74S38 open-collector driver which either drives low or presents a high impedance.

Interrupt Out Example:

To generate an off-board interrupt by using bit 6 of port E6 (port C of the 8255A – location U15) to control the Multibus signal INT6/:

Install jumpers:
E16-E26 (INTROUT/BANKSEL from PC6)
E206-E212 (INTROUT/BANKSEL to INT6)

Design software to:

- Initialize port E6, bit 6 to the output mode
- Set port E6, bit 6 to a one
- Generate an interrupt by resetting port E6, bit 6 to a zero, then setting port E6, bit 6 to a one

NOTE

This method produces an interrupt pulse approximately 1 microsecond in duration on the Multibus INT6/ line.

Port addressing and operation of the 8259A PIC are described in sections 3-27 through 3-36.

4-37. CONCLUSION

This chapter presents a basic description of each major iSBC 80/24A board functional block, as shown in figure 4-1. The Preface of this manual (page iii) provides a list of additional Intel publications with information on related topics.

* * *

CHAPTER 5
SERVICE AND REPAIR INFORMATION

5.1 INTRODUCTION

This chapter provides service and repair information, component layout, jumper location, and schematic diagrams for the iSBC 80/24A Single Board Computer.

5.2 SERVICE AND REPAIR ASSISTANCE

Customer Support Service Engineering provides two separate services for product replacement and repair: a Return Replacement Authorization (RRA) for direct replacement, and a Direct Return Authorization (DRA) for repair.

The RRA service replaces a defective product. Return the defective product to Intel, freight prepaid. Intel will replace the product (bearing a new serial number). This service is not offered on all products, is subject to availability, and is available only to customers in nonserviceable areas. Typically, Intel ships the replacement product within 48 hours of receiving the defective product.

The DRA service provides repair work. Return the defective product to Intel, freight prepaid. Intel will repair, test, and update the product with all mandatory Engineering Change Orders. The serial number will not change. Normal turnaround time is four to six weeks.

Determine which service you need, RRA or DRA. Before calling Customer Support Service (refer to figure 5-1 for the United States regional and international service numbers), have the necessary information ready:

a. Part and serial number of the product.

b. Purchase order number, for repair and shipping charges.

c. If it is a warranty repair, the proof of purchase showing the product was received within 90 days of the service request date. Without proof, services will be billed at the current rate.

d. Your shipping and billing addresses.

e. Your telephone number.

In all correspondence with Customer Support Service, reference the authorization number on the packing slip, the purchase order, and any other related documents.

Before shipping, remove all user modifications. Protect the equipment from damage in transit:

a. Place boards in antistatic bags and then in padded shipping bags. Wrap power supplies and other large items in antistatic material.

b. Use a proper sized box which allows room for protective padding (i.e., flow pack, foam, etc.).

c. Write the return authorization number on the outside of the box and label the box "FRAGILE."

NOTE

Damage due to lack of compliance with the shipping guidelines could result in the customer incurring extra repair charges.

d. In the U.S. forward the product and all correspondence to:

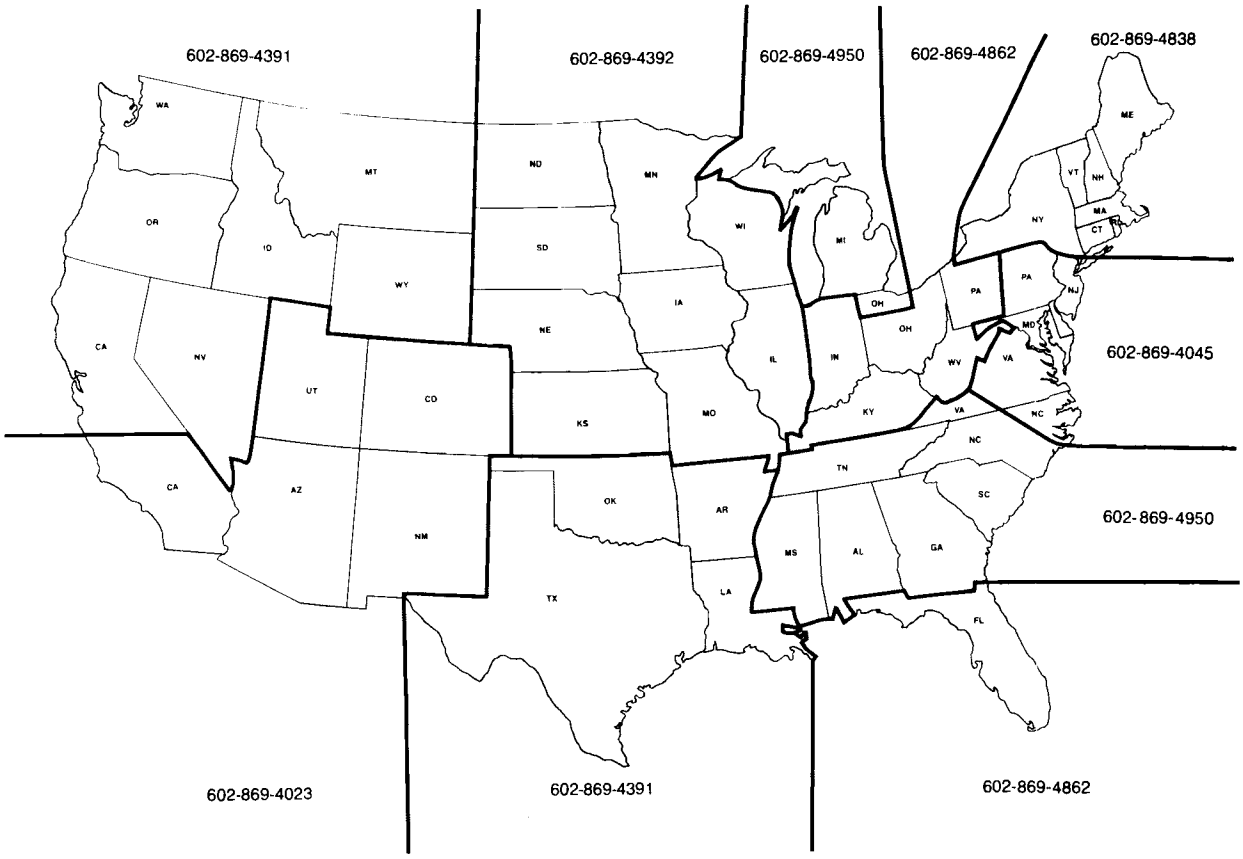
Intel Corporation
Customer Support Marketing Administration
Billing Department — DV-1-704
2402 W. Beardsley Road
Phoenix, Arizona 85027

Authorization #_____

5-3. SERVICE DIAGRAMS

Schematic diagrams of the iSBC 80/24A board are provided in figure 5-4, sheets 1 through 10. Notice that a functional description of each jumper connection on a particular schematic sheet is referenced on the previous page.

The schematic diagrams 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.



International Service Numbers:

Canada	416-675-2105	Japan	03-427-7561
France	(01)-687-22-21	West Germany	(89)-53891
Italy	(02)-824-00-06	Netherlands	011-31-10-212377
Sweden	(08)-98-53-85	Switzerland	(01)-55-45-02
United Kingdom	(0793)-488388	Israel	4/524-261

Other International: 602-869-4862

Figure 5-1. U.S. and International Service Telephone Numbers

5-4. INTERNAL SIGNALS

The internal board signal mnemonics are listed and defined in table 5-1. The signals are listed according to boxed code alphabetical order.

Internal board signals which traverse from one schematic sheet to another in figure 5-4 are identified by a single or double alpha character within a box (e.g., G 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. or example, if you are going to trace the path of READ/ when it exits sheet 2, the first step would be to turn to the indicated sheet. Since READ/ will be entering sheet 2, as indicated on sheet 3, look for the G symbol on the left side of the sheet. Notice that the inputs also list the source sheet number (sheet 3 in this example).

Each signal will keep the same boxed character through figure 5-4. This will enable you to trace the signal to any sheet with minimal effort.

Table 5-1. Glossary of Internal Signal Mnemonics

Code	Mnemonic	Description
A	IORD/	I/O Read
B	IOWRT/	I/O Write
C	MRD/	Memory Read
D	MWRT/	Memory Write
E	RESET/	Reset (Initialize)
F	PORT D5/	Port D5 (Bus Override)
G	READ/	CPU Read Command
H	ACK	CPU Acknowledge
I	Not Used	
J	ONBDIO	On-Board I/O
K	AD0-AD7	Address/Data Lines 0-7
L	CMD	CPU Read or Write Command
M	A0-AF	CPU Address Lines
N	AB1, AC1	Buffered Address Lines B, C
O	Not Used	
P	9.68 MHz	Board Clock
Q	Not Used	
R	MULTIBUSRDY/	Multibus Ready
S	SID	CPU Serial Input Data
T	INTR 5.5	Restart Interrupt 5.5
U	INTR 6.5	Restart Interrupt 6.5
V	INTR 7.5	Restart Interrupt 7.5
W	TRAP	Non-Maskable TRAP Interrupt
X	INTR	8259A Interrupt Request
Y	RAMACK/	On-Board RAM Acknowledge

Table 5-1. Glossary of Internal Signal Mnemonics (Continued)

Code	Mnemonic	Description
Z	PROMACK/	On-Board PROM Acknowledge
AA	IOSYNC/	On-Board I/O Synchronous Device Acknowledge
AB	2.15 MHz	PCI & PIT Clock
AC	RESET	Complement of RESET/ (Code E)
AD	SOD	CPU Serial Output Data
AE	ALE	Address Latch Enable
AF	Not Used	
AG	85WRT/	RAM Write Command
AH	85RD/	RAM Read Command
AI	Not Used	
AJ	IO/	I/O Command
AK	INTA/	Interrupt Acknowledge
AL	S1	CPU Data Bus Status Pin
AM	DCDPROM2	Decode PROM Bit 2
AN	DCDPROM1	Decode PROM Bit 1
AO	Not Used	
AP	IOPROMEN/	Enables PIO Transceiver U36
AQ	Not Used	
AR	OIT1	Interval Timer Output 1
AS	PIO0-PIO7	PROM/IO Bus
AT	PTYPE1	PROM Type 1
AU	PTYPE0	PROM Type 0
AV	PROMENA/	PROM Enable
AW	55CS0/	8255A PPI Chip Select 0
AX	IO0-IO7	I/O Device Bus
AY	PROM ENABLE	Programmable PROM Enable
AZ	INTROUT	Programmable Interrupt Output
AZ	BANK SEL	Programmable Off-Board Memory Select
BA	CLOCK IN/OUT	Bidirectional Clock Line
BB	ITG1	Interval Timer Gate 1
BC	ITG0	Interval Timer Gate 0
BD	PIA1	Parallel Port E6 Interrupt A
BE	PIB1	Parallel Port E6 Interrupt B
BF	55CS1/	8255A PPI Chip Select 1
BG	AUX1/	Auxiliary Serial Channel Input
BH	AUX0/	Auxiliary Serial Channel Output
BI	Not Used	
BJ	BDET/	Serial Channel Receive Line Detect
BK	BRI/	Serial Channel Ring Indicator
BL	PIA2	Parallel Port EA Interrupt A
BM	PIB2	Parallel Port EA Interrupt B
BN	51CS/	8251A PCI Chip Select
BO	Not Used	
BP	54CS/	8254 PIT Chip Select
BQ	Not Used	
BR	MISCS/	Miscellaneous Chip Select
BS	RXR	Receiver Ready
BT	TXR	Transmitter Ready
BU	TXE	Transmitter Empty
BV	Not Used	
BW	OIT0	Interval Timer Counter 0 Output

Table 5-1. Glossary of Internal Signal Mnemonics (Continued)

Code	Mnemonic	Description
BX	PORT D4/	Clears PFIN Flip-Flop (U47, Sh. 9)
BY	59CS/	8259A PIC Chip Select
BZ	MMCS0/	Multimodule Chip Select 0
CA	MMCS1/	Multimodule Chip Select 1
CB	MMCS2/	Multimodule Chip Select 2
CC	MMCS3/	Multimodule Chip Select 3
CD	IOP2EN/	Enables Bus Transceiver U35
CE	MMWAIT/	Multimodule WAIT Command
CF	MMPRESENT/	Multimodule Present
CG	MIOINT 1	Multimodule Interrupt 1
CH	MIOINT 0	Multimodule Interrupt 0
CI	Not Used	
CJ	MIOINT 3	Multimodule Interrupt 3
CK	MIOINT 2	Multimodule Interrupt 2
CL	IOASYNC/	Multimodule Acknowledge
CM	CLR	Clear Counter U20
CN	U31PU	U31 Pull-Up Resistor
CO	Not Used	
CP	A1	Address Line 1
CQ	Not Used	
CR	A2	Address Line 2
CS	A3-A9	Address Lines 3-9
CT	AA, AD-AF	Address Lines A, D-F
CU	AB, AC	Address Lines B, C

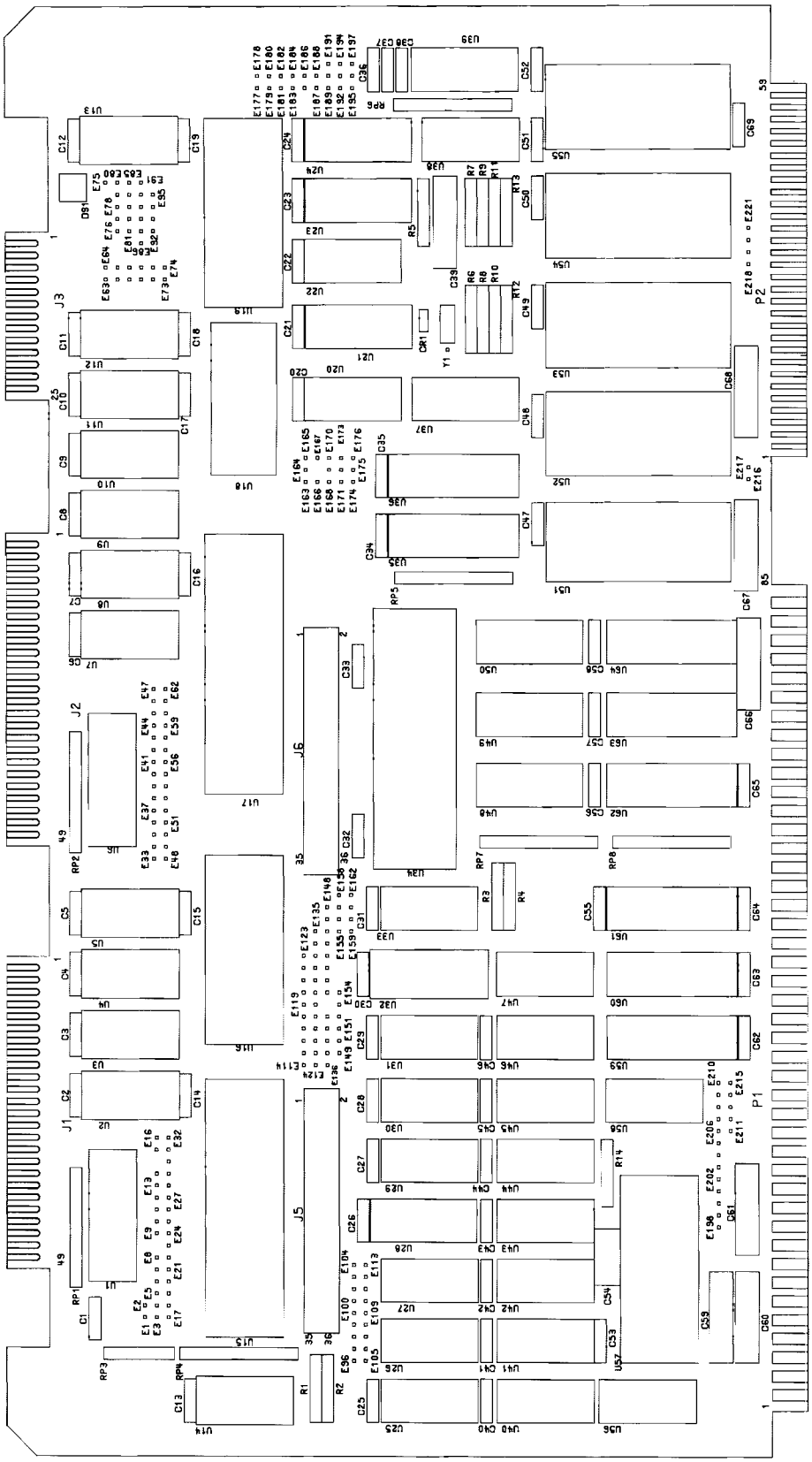
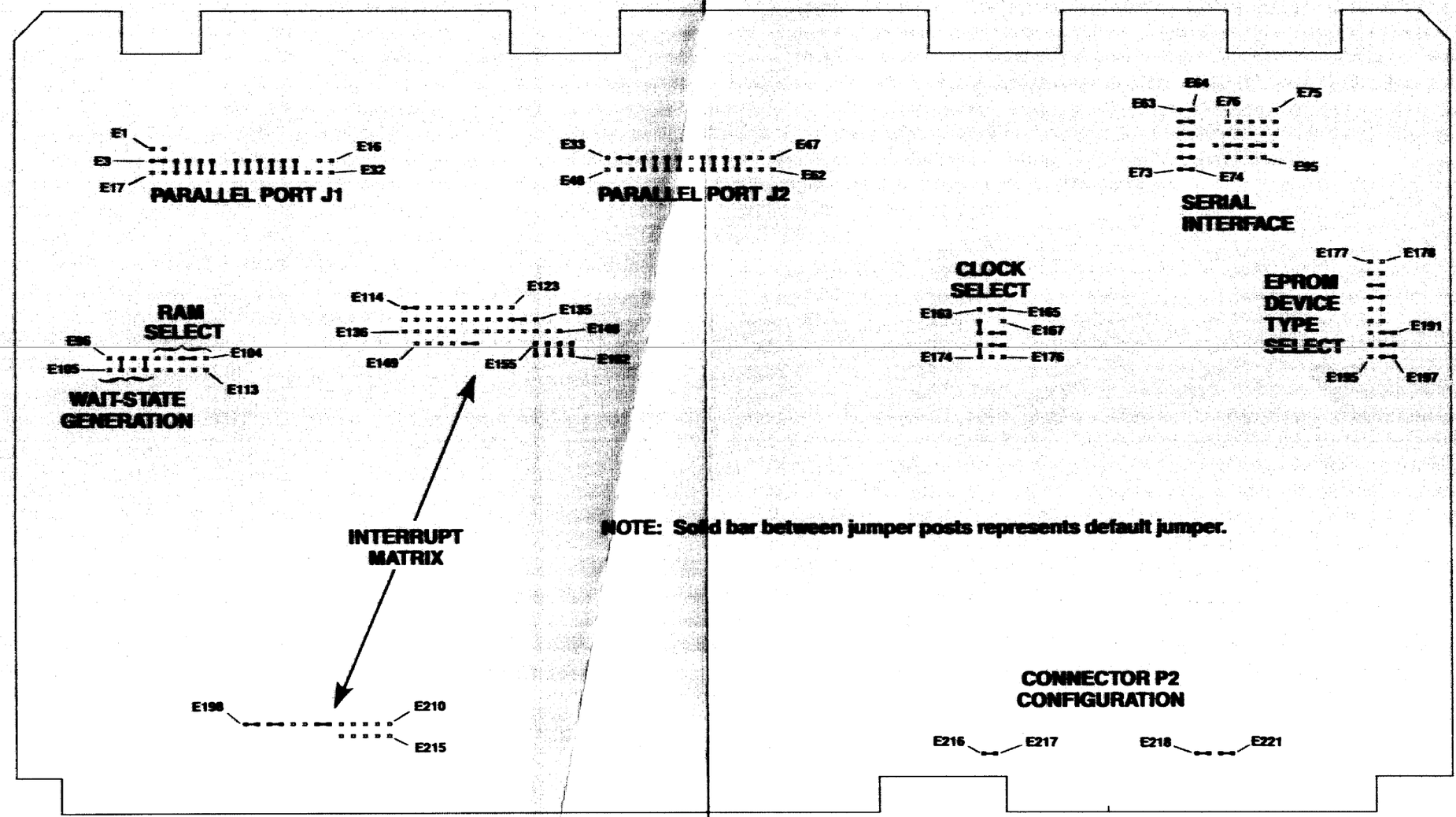


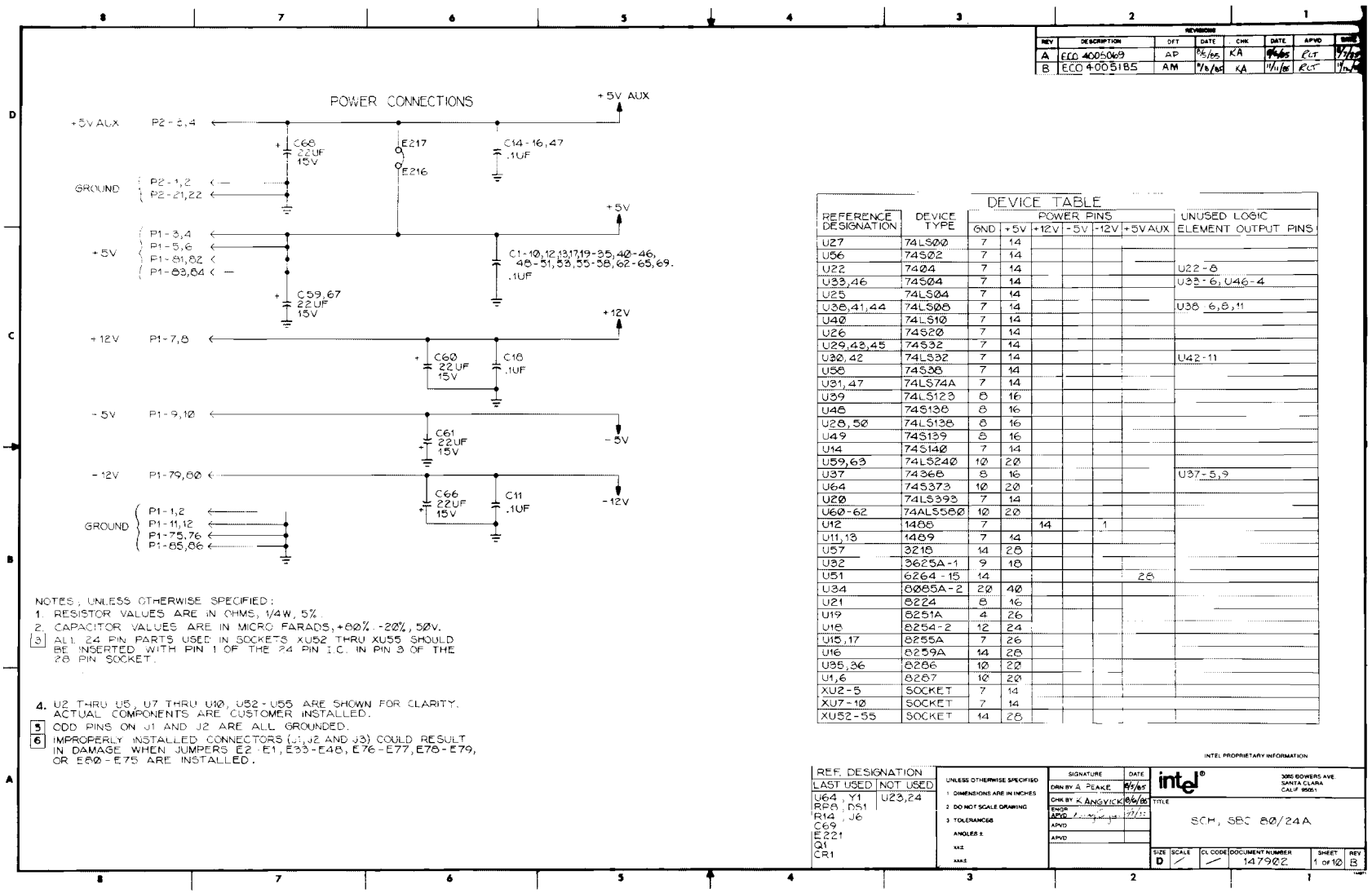
Figure 5-2. iSBC® 80/24A Board Component Location Diagram



NOTE: Solid bar between jumper posts represents default jumper.

Figure 5-3. iSBC® 80/24A Board Jumper Location Diagram

EEBC 69264A Jumper Pair	EEBC 69264A Sheet Grid Reference	Function	Section Ref.
E216-217*	1-D6	Used for +5 volt battery backup	2-22



NOTES; UNLESS OTHERWISE SPECIFIED:
1. RESISTOR VALUES ARE IN OHMS, 1/4W, 5%.
2. CAPACITOR VALUES ARE IN MICRO FARADS, +80% -20%, 50V.
3. ALL 24 PIN PARTS USED IN SOCKETS XU52 THRU XU55 SHOULD BE INSERTED WITH PIN 1 OF THE 24 PIN I.C. IN PIN 3 OF THE 28 PIN SOCKET.
4. U2 THRU U5, U7 THRU U10, U52-U55 ARE SHOWN FOR CLARITY. ACTUAL COMPONENTS ARE CUSTOMER INSTALLED.
5. ODD PINS ON J1 AND J2 ARE ALL GROUNDED.
6. IMPROPERLY INSTALLED CONNECTORS (J1, J2 AND J3) COULD RESULT IN DAMAGE WHEN JUMPERS E2-E1, E33-E48, E76-E77, E78-E79, OR E80-E75 ARE INSTALLED.

REV	DESCRIPTION	DFT	DATE	CHK	DATE	APVD	DATE
A	ECO 4005069	AP	8/5/85	KA	8/6/85	RLT	8/7/85
B	ECO 4005185	AM	9/5/85	KA	11/1/85	RLT	11/1/85

DEVICE TABLE

REFERENCE DESIGNATION	DEVICE TYPE	POWER PINS GND	+5V	+12V	-5V	-12V	+5V AUX	UNUSED LOGIC ELEMENT OUTPUT PINS
U27	74LS00	7	14					
U56	74S02	7	14					
U22	7404	7	14					U22-8
U33,46	74S04	7	14					U33-6, U46-4
U25	74LS04	7	14					
U38,41,44	74LS08	7	14					U38 6,8,11
U40	74LS10	7	14					
U26	74S20	7	14					
U29,43,45	74S32	7	14					
U30,42	74LS32	7	14					U42-11
U55	74S38	7	14					
U31,47	74LS74A	7	14					
U39	74LS123	8	16					
U48	74S138	8	16					
U28,50	74LS138	8	16					
U49	74S139	8	16					
U14	74S140	7	14					
U59,63	74LS240	10	20					
U37	7436B	8	16					U37-5,9
U64	74S373	10	20					
U20	74LS393	7	14					
U60-62	74ALS580	10	20					
U12	1488	7		14		-1		
U11,13	1489	7	14					
U57	3218	14	28					
U32	3625A-1	9	18					
U51	6264-15	14					28	
U34	8085A-2	20	40					
U21	8224	8	16					
U19	8251A	4	26					
U18	8254-2	12	24					
U15,17	8255A	7	26					
U16	8259A	14	28					
U35,36	8286	10	20					
U1,6	8287	10	20					
XU2-5	SOCKET	7	14					
XU7-10	SOCKET	7	14					
XU52-55	SOCKET	14	28					

REF. DESIGNATION	
LAST USED	NOT USED
U64, Y1 RP8, DS1 R14, J6 C69 E221 Q1 CR1	U23,24

UNLESS OTHERWISE SPECIFIED
1. DIMENSIONS ARE IN INCHES
2. DO NOT SCALE DRAWING
3. TOLERANCES
ANGLES ±
XX±
XXX±

SIGNATURE	DATE
DRN BY A. PEAKE	8/5/85
CHK BY K. ANGVICK	8/6/85
ENGR	
APVD	11/1/85
APVD	
APVD	

INTEL PROPRIETARY INFORMATION

intel® 3065 BOWERS AVE. SANTA CLARA CALIF 95051

TITLE: SCH, SBC 80/24A

SIZE	SCALE	CL CODE	DOCUMENT NUMBER	SHEET	REV
D	/	/	147902	1 of 10	B

Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 1 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E198-199*	2-A6	Connects 9.68 MHz BCLK/ to P1-13 (output)	4-3
E200-201*	2-D3	Connects BPRO/ to P1-16 (output)	2-19
E202-203	2-D6	Connects AACK/ to iSBC 80/24A board via P1-25 (input)	None
E204-205*	2-A6	Connects 9.68 MHz CCLK/ to P1-31 (output)	4-3

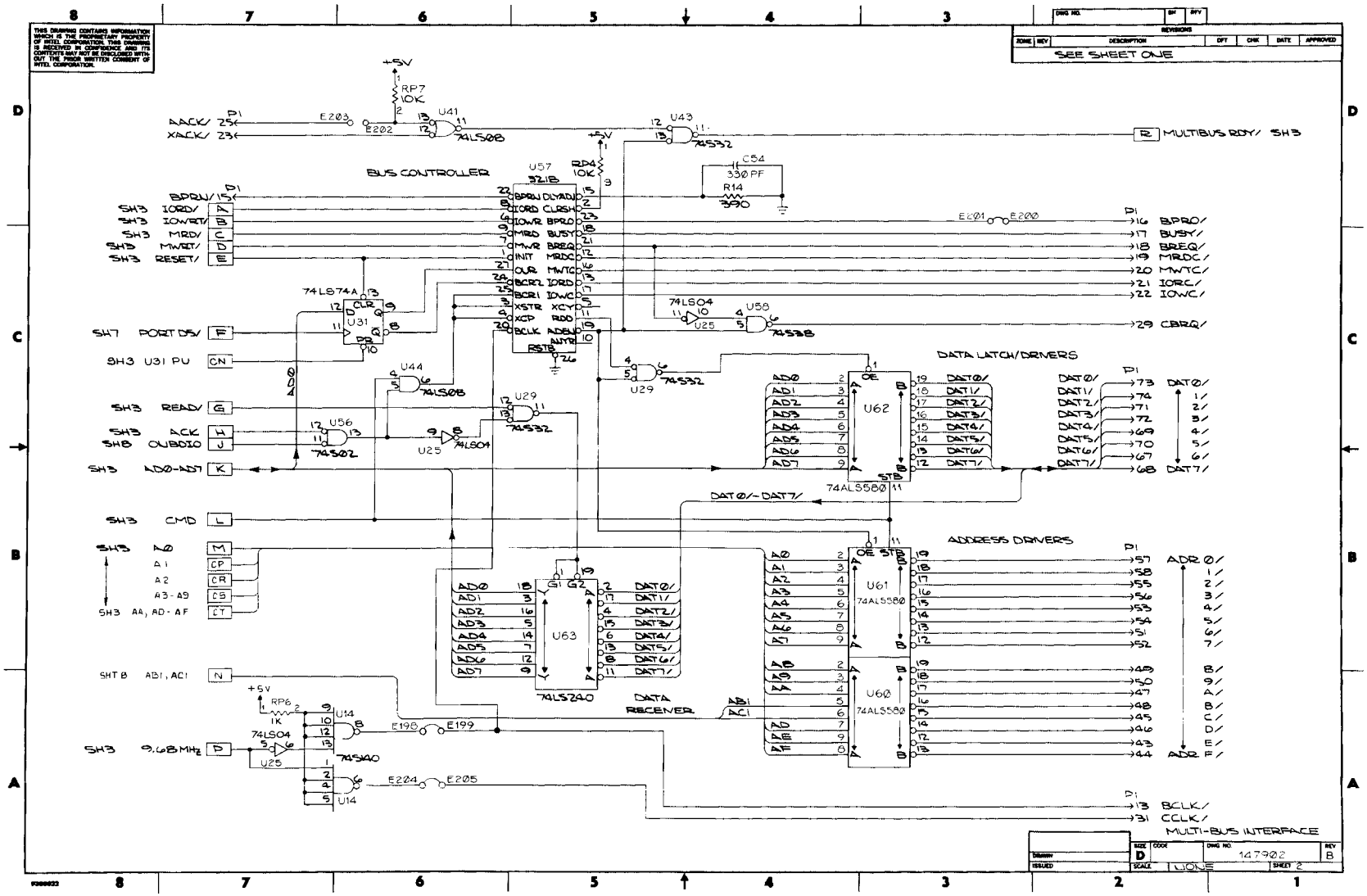


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 2 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E96-105	3-C5	Connect for 4.84 MHz clock only	4-3
E97-106*	3-B6	Enables failsafe timer	4-35
E98-107	3-B7	Reserved	None
E99-108*	3-B7	Connects RAMACK/ to READY gate	4-29
E163-166	3-D5	4.84 MHz Clock input to 8085A-2 CPU	4-3
E166-168*	3-D5	9.68 MHz Clock input to 8085A-2 CPU	4-3
E171-174*	3-D6	Connects 8224 output to divider network	None
E218-219*	3-B2	Connects INTA/ to P2-36 (output)	4-23
E220-221*	3-A2	Connects BMTO to P2-34 (output)	4-34

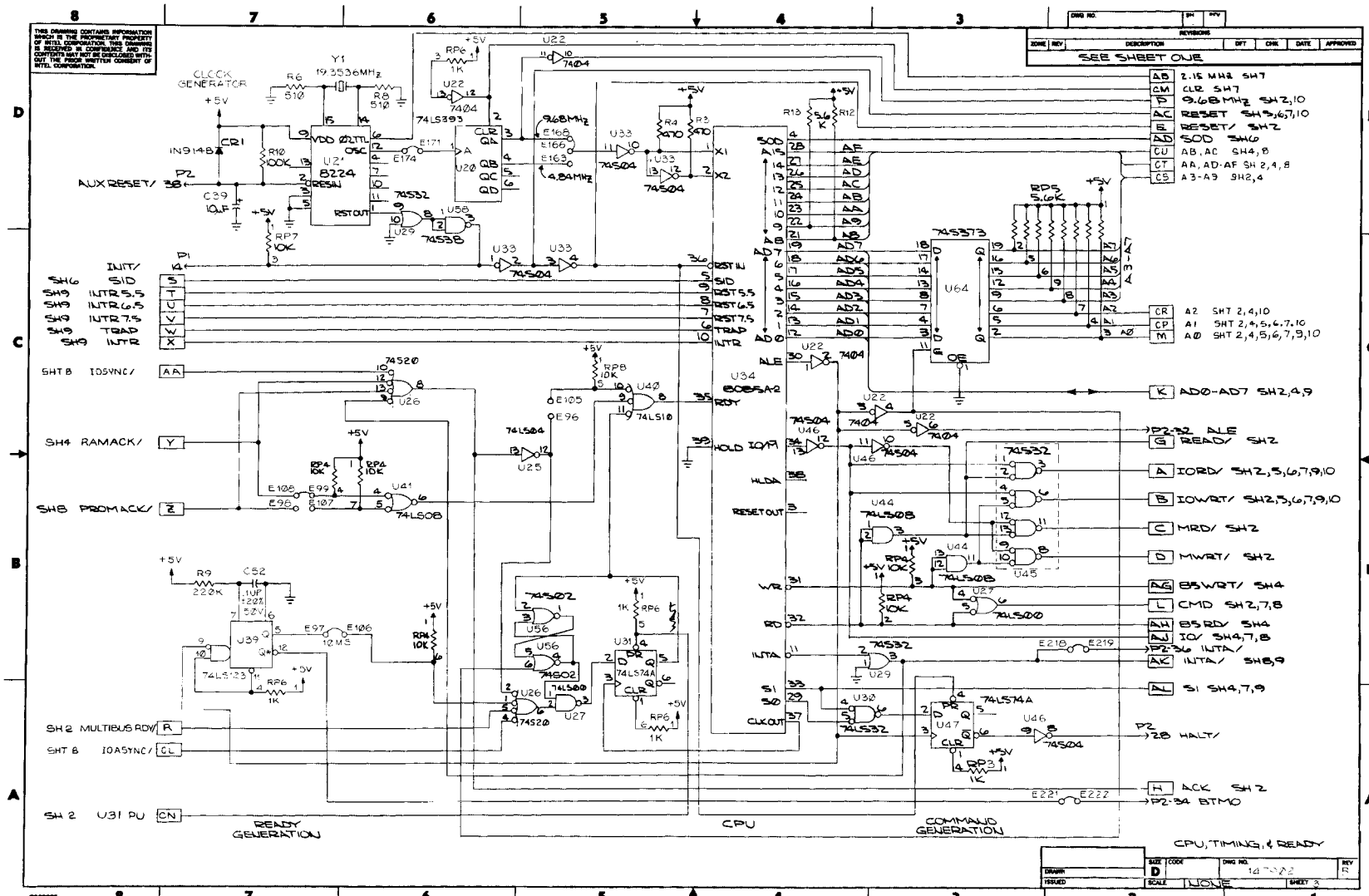


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 3 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E102-103*	4-C5	Indicates default RAM space: E000-FFFH (8K)	2-12
E102-103*	4-C5	RAM address matrix. See table 2-4	2-12
E103-104			
E111-112			
E112-113			
E100-109	4-C7	RAM size jumper. See table 2-4	2-12
E101-110	4-C7	RAM size jumper. See table 2-4	2-12
E177-197	4-B6	EPROM device type select matrix	2-8

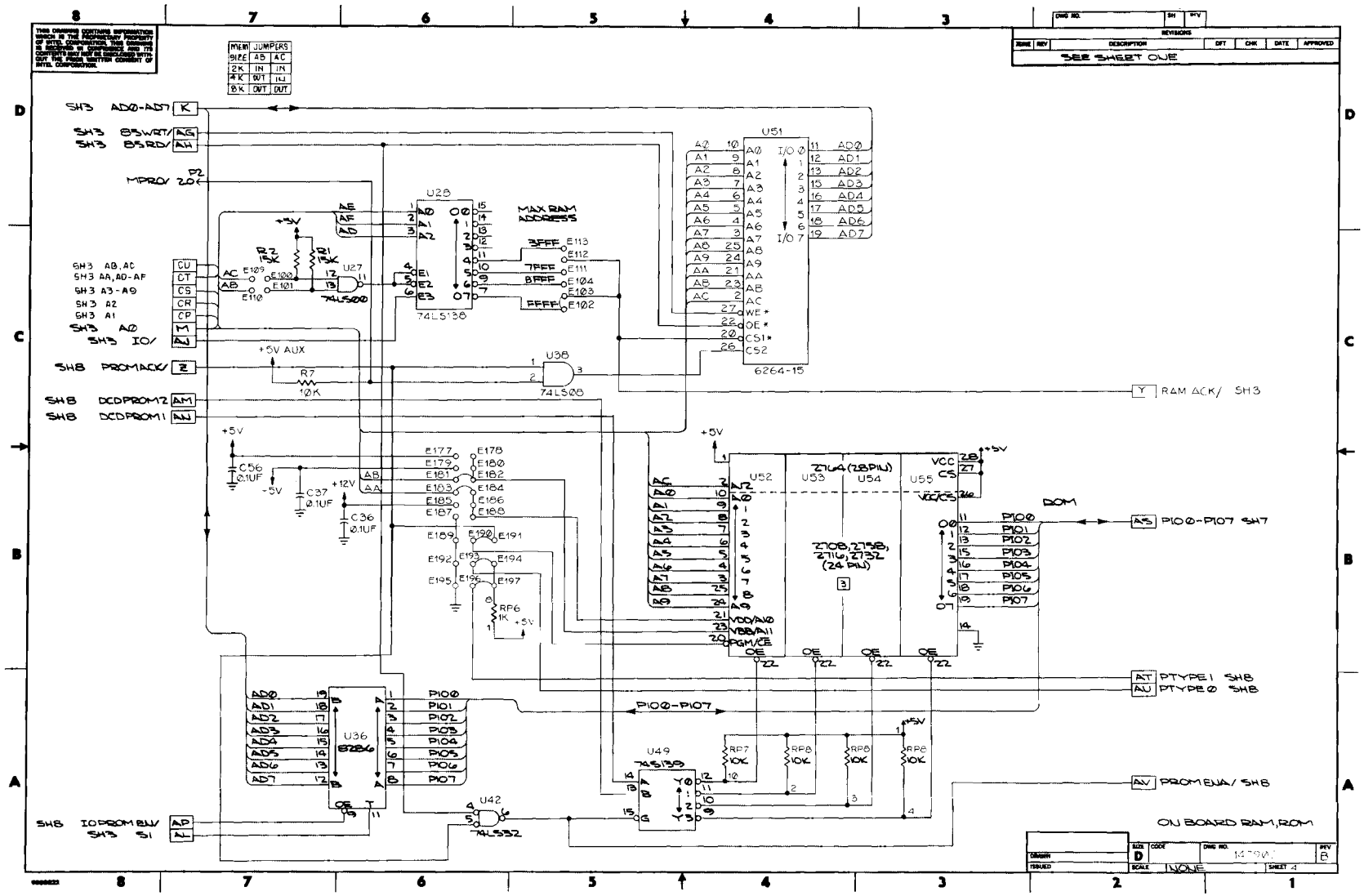


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 4 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E1-2	5-B2	Connects +5 volts to J1-50	None
E3-4*	5-C4	Configures port E4 bus transceiver to input mode	2-14
E4-18	5-C4	Configures port E4 bus transceiver to output mode	2-14
E5-19*	5-C4	Connects port E6, bit 0 to driver/terminator	2-14
E6-20*	5-C4	Connects port E6, bit 1 to driver/terminator	2-14
E7-21*	5-C4	Connects port E6, bit 2 to driver/terminator	2-14
E8-22*	5-C4	Connects port E6, bit 3 to driver/terminator	2-14
E9-24*	5-C4	Connects port E6, bit 4 to driver/terminator	2-14
E10-25*	5-C4	Connects port E6, bit 5 to driver/terminator	2-14
E11-26*	5-C4	Connects port E6, bit 6 to driver/terminator	2-14
E12-27*	5-C4	Connects port E6, bit 7 to driver/terminator	2-14
E13-28*	5-C4	Applies +5 volts to timer gate 0 (ITG0)	None
E14-29*	5-C4	Applies +5 volts to timer gate 1 (ITG1)	None
E15-30	5-C4	Connects timer 1 output to timer input matrix	2-11
E15-XX	5-C4	Connects selected J1 bit to timer input matrix	2-11
E16-XX	5-C4	Connects INTROUT/BANKSEL to selected E6 bit	2-11
E17-19	5-C4	Connects port E6, bit 0 to PIA1	2-14
E22-23	5-C4	Connects port E6, bit 3 to PIB1	2-14
E31-XX	5-C4	Connects PFSR/ to selected E6 bit	2-22
E32-XX	5-C4	Connects PROM ENABLE to selected E6 bit	2-11

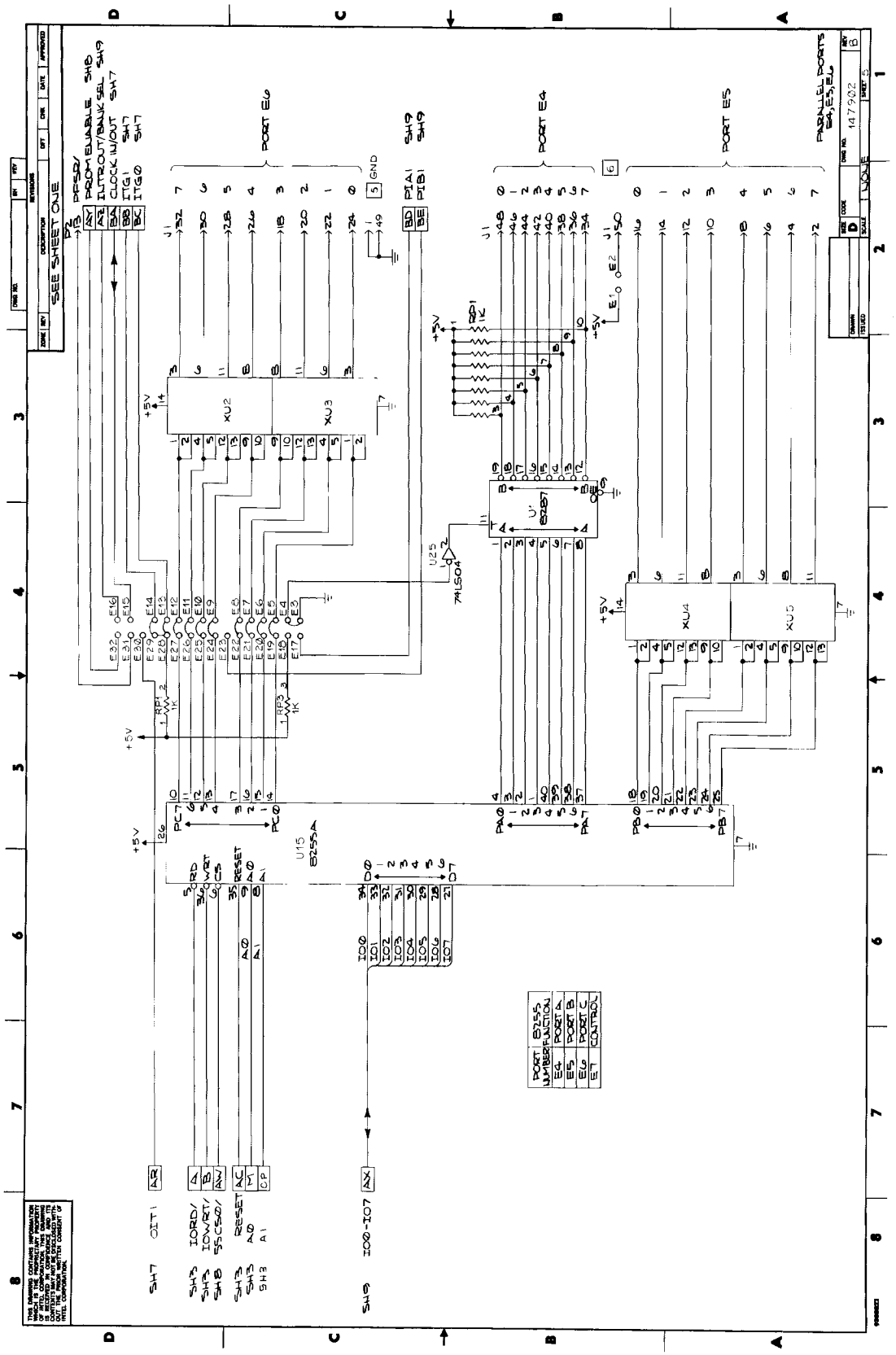


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 5 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E33-48	6-B2	Connects +5 volts to J2-50	None
E34-35*	6-C4	Configures port E8 bus transceiver to input mode	2-14
E35-50	6-C4	Configures port E8 bus transceiver to output mode	2-14
E36-51*	6-C4	Connects port EA, bit 0 to driver/terminator socket	2-14
E37-52*	6-C4	Connects port EA, bit 1 to driver/terminator socket	2-14
E38-53*	6-C4	Connects port EA, bit 2 to driver/terminator socket	2-14
E39-54*	6-C4	Connects port EA, bit 3 to driver/terminator socket	2-14
E40-XX	6-C4	Ground post	2-14
E41-56*	6-C4	Connects port EA, bit 4 to driver/terminator socket	2-14
E42-57*	6-C4	Connects port EA, bit 5 to driver/terminator socket	2-14
E43-58*	6-C4	Connects port EA, bit 6 to driver/terminator socket	2-14
E44-59*	6-C4	Connects port EA, bit 7 to driver/terminator socket	2-14
E45-XX	6-C4	Connects BDET/ to selected EA bit	2-14
E46-XX	6-C4	Connects BRI/ to selected EA bit	2-14
E47-XX	6-C4	Connects AUX0/ to selected EA bit	2-14
E49-51	6-C4	Connects port EA, bit 0 to PIA2	2-14
E54-55	6-C4	Connects port EA, bit 3 to PIB2	2-14
E60-XX	6-C4	Connects SOD to selected EA bit	2-14
E61-XX	6-C4	Connect SID to selected EA bit	2-14
E62-XX	6-C4	Connects AUX1/ to selected EA bit	2-14

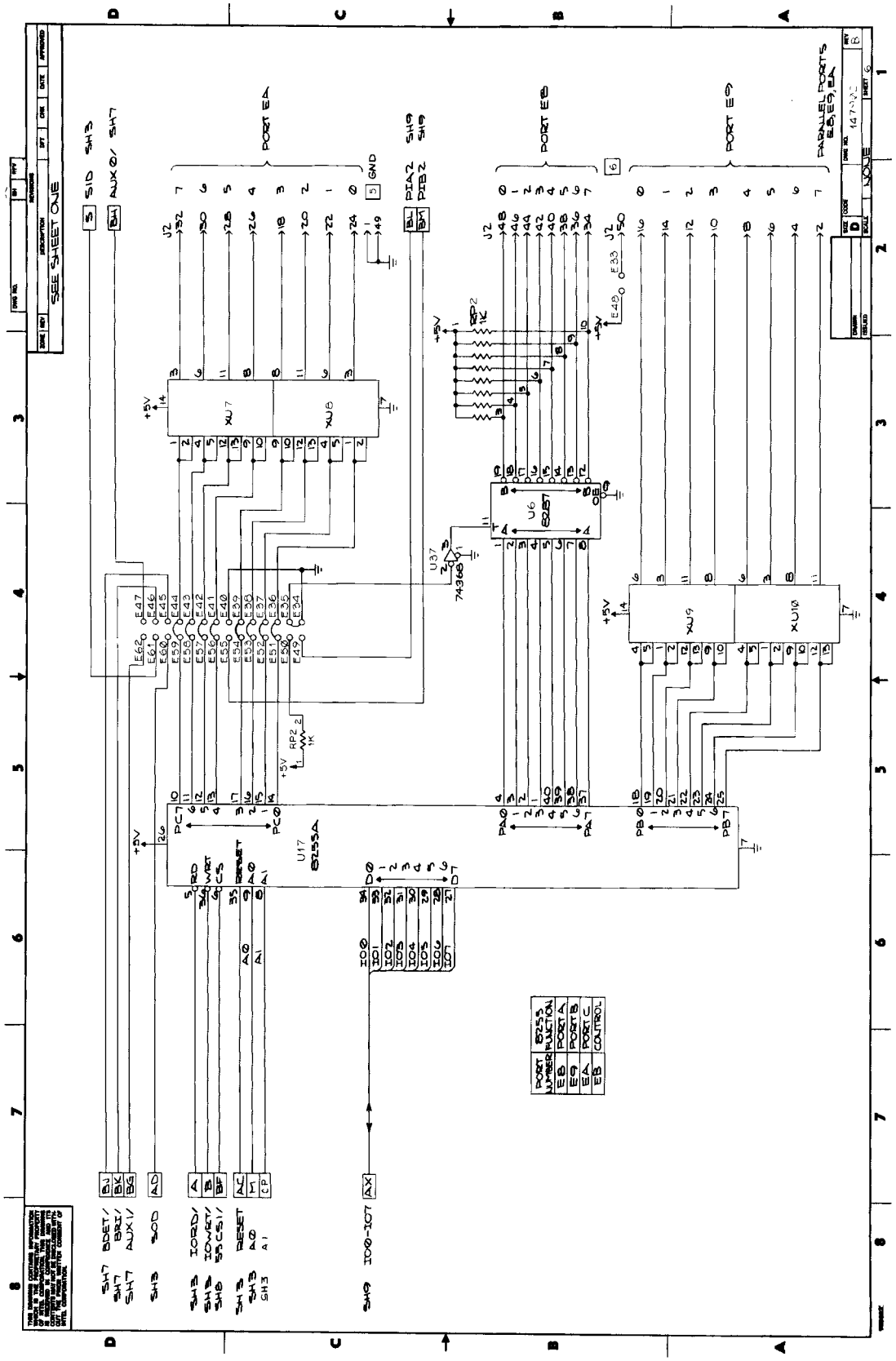


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 6 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E63-74	7-C3	RS232C jumper matrix	2-13
E75-80	7-A2	Connects −12 volts to J3-19	2-25
E76-77	7-A2	Connects +5 volts to J3-23	None
E78-79	7-A2	Connects +12 volts to J3-22. Use for TTY serial interface	2-25
E81-82	7-C2	RS232C Protective Ground J3-2	2-13
E83-84	7-D7	Connects Secondary RTS (J3-11) input to AUX1/	2-13
E84-85	7-D7	Connects Secondary TxD (J3-11) input to AUX1/	2-13
E86-88	7-D6	Connects external Receive Clock (RxC)	2-13
E87-88*	7-D6	Connects internal Receive Clock (RxC)	2-13
E89-90*	7-D6	Connects internal Transmit Clock (TxC)	2-13
E89-91*	7-D6	Connects external Transmit Clock (TxC)	2-13
E92-95	7-D2	Connects AUX0/ output to J3-21	2-13
E93-95	7-D2	Connects AUX0/ output to J3-5	2-13
E94-95	7-D2	Connects AUX0/ output to J3-26	2-13
E164-165*	7-B6	Connects 2.15 MHz output to divider network	None
E167-170	7-B6	Connects Timer 0 output to Timer 1 input	None
E167-176	7-B6	Connects Timer 0 output to port E6 bit via CLK OUT line	None
E169-170*	7-B6	Connects 1.075 MHz to CLK1 on PIT	None
E172-173*	7-B6	Connects 1.075 MHz to CLK0 on PIT	None
E175-XX	7-B6	Connects 134.4 KHz. See table 2-4	None

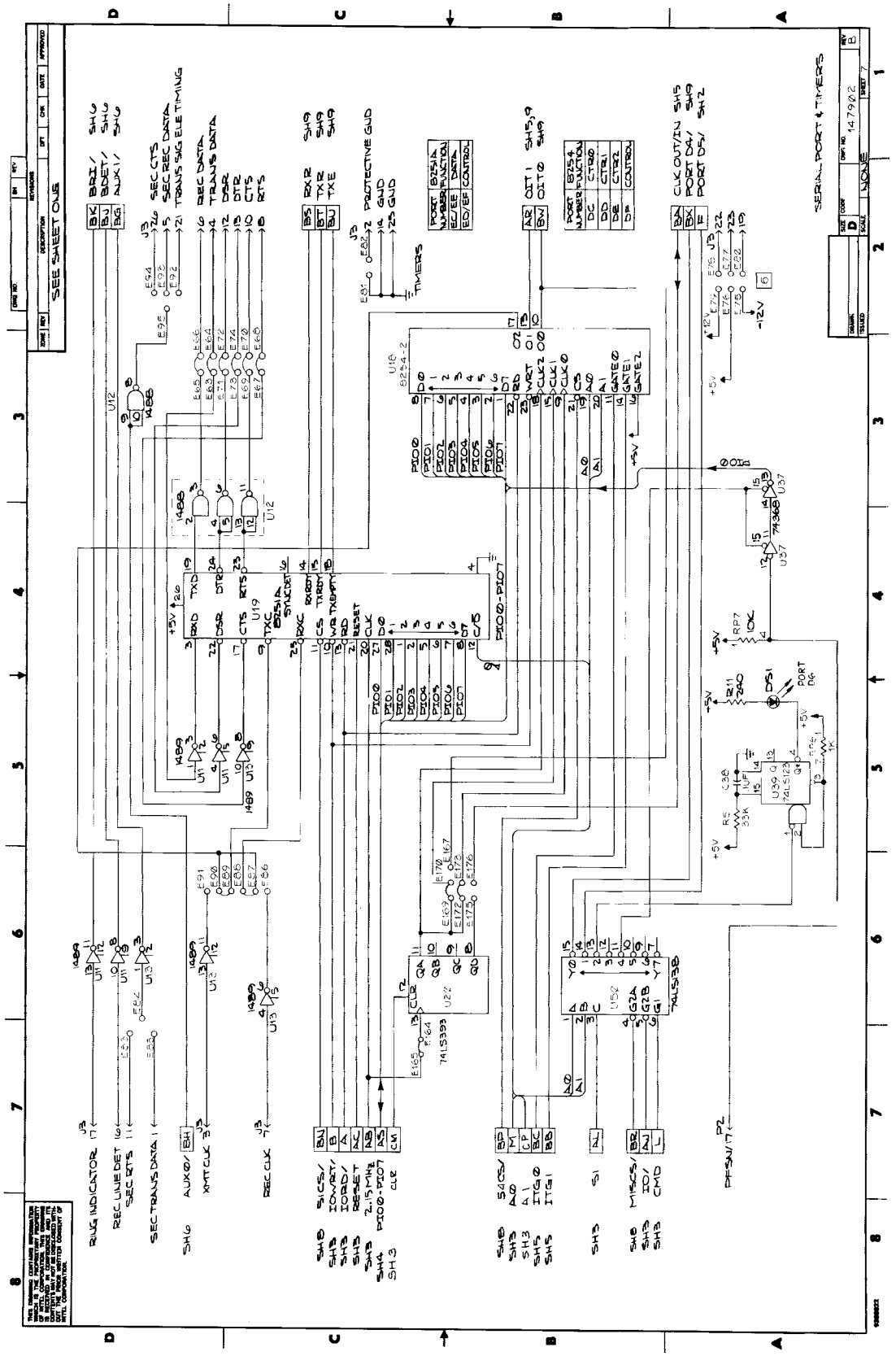


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 7 of 10)

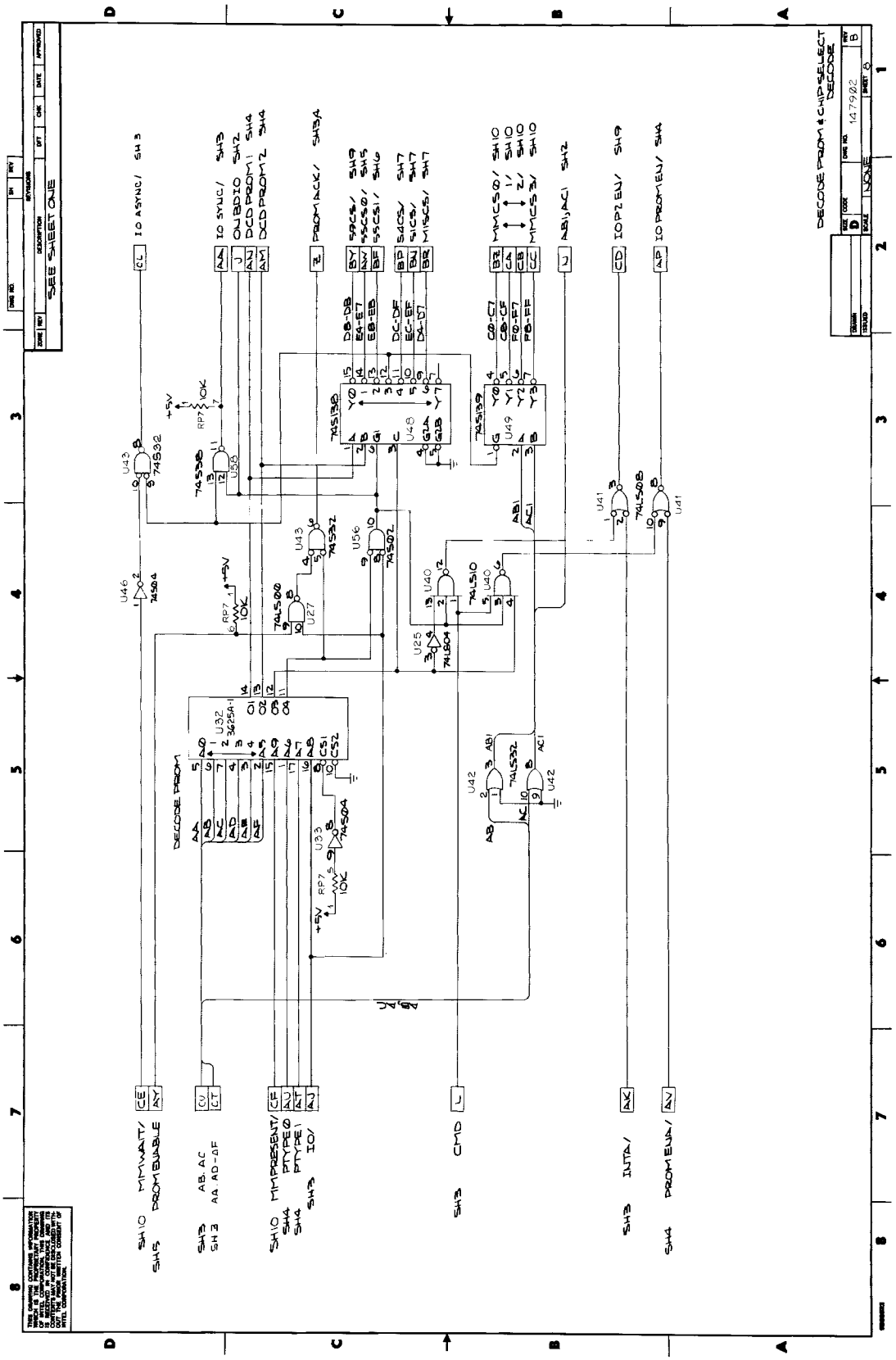


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 8 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E114-115*	9-B4	Connects IR7 to ground	None
E114-135,	9-C4	Interrupt Jumper Matrix. See table 2-7	None
E138-146,			
E149-162	9-C4		
E133-134*	9-C4	Connects INT1/ to IR1 on PIC	None
E153-154*	9-C4	Connects OIT0 to IR2 on PIC	None
E206-211	9-C5	Connects INTROUT/BANKSEL to P1-28 (ADR10/) output	4-7

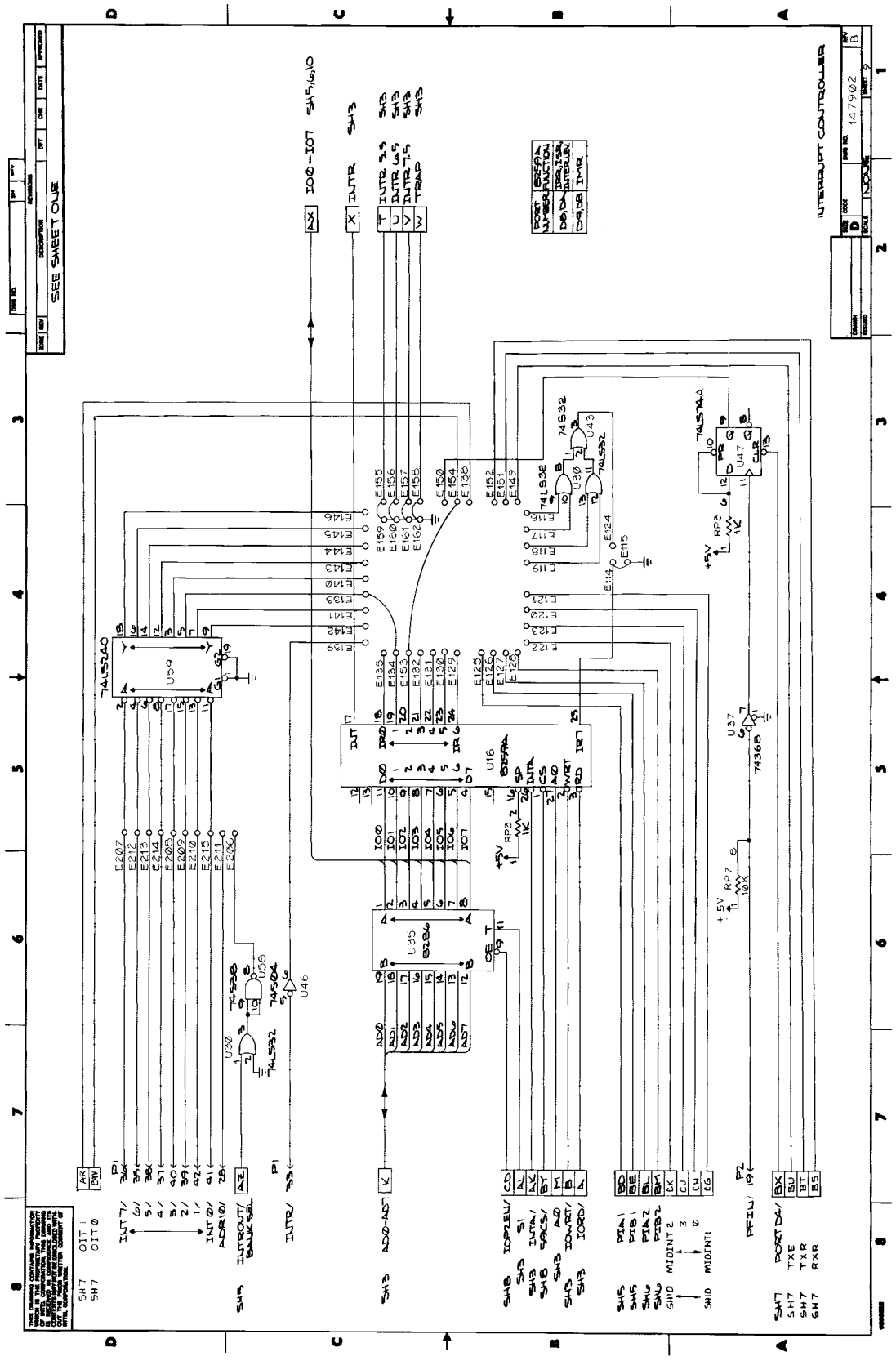


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 9 of 10)

iSBC 80/24A Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Section Ref.
E136-137	10-D5	Multimodule I/O board option (reserved); J5	None
E147-148	10-C3	Multimodule I/O board option (reserved); J6	None

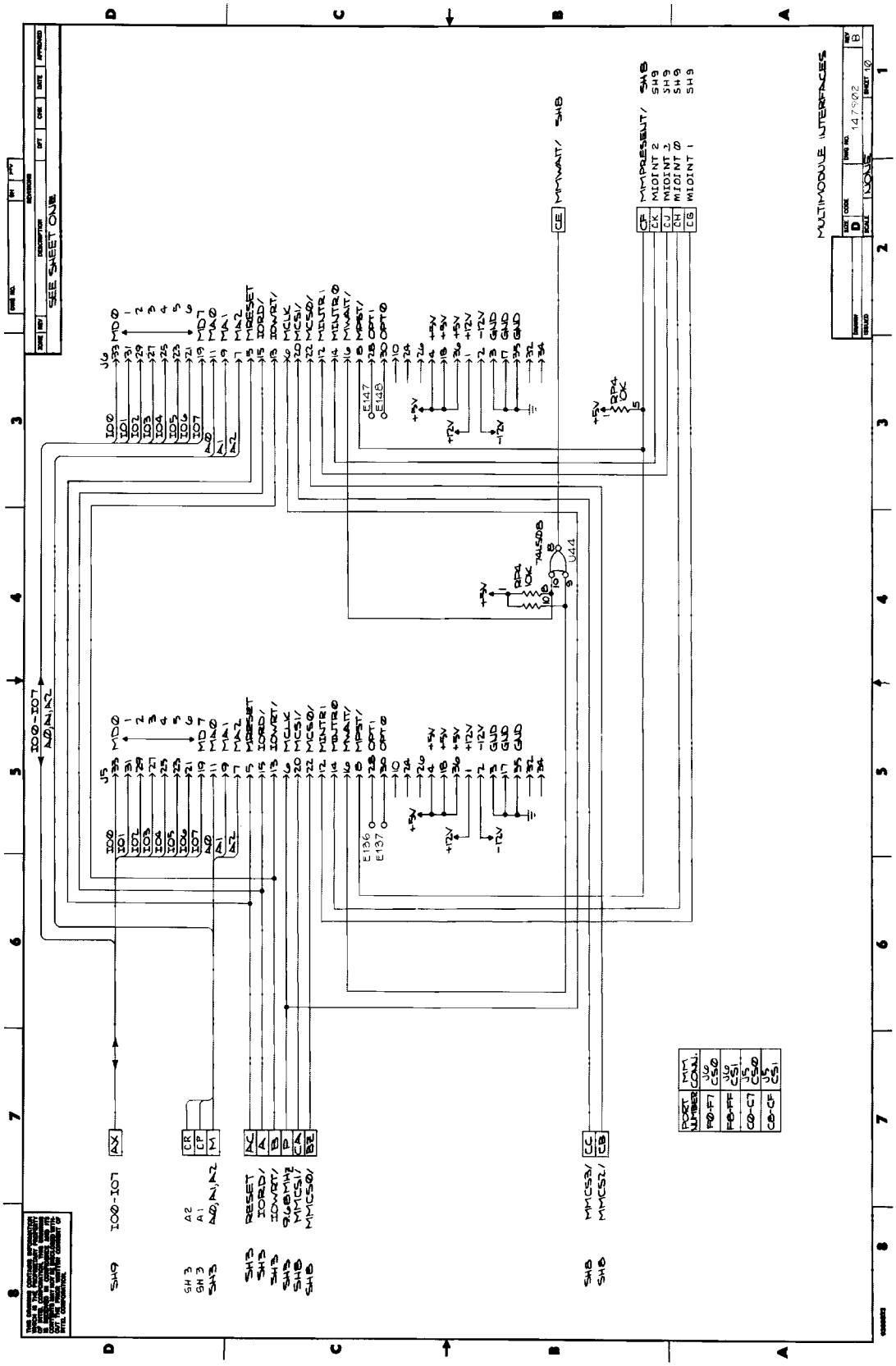


Figure 5-4. iSBC® 80/24A Board Schematic Diagram (Sheet 10 of 10)

* * *

APPENDIX A
8085A 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 computer's 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., 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 8085A INSTRUCTION SET

The 8085A instruction set includes five different types of instructions:

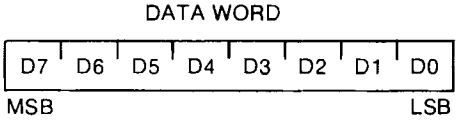
- **Data Transfer Group** — move data between registers or between memory and registers
- **Arithmetic Group** — add, subtract, increment or decrement data in registers or in memory
- **Logical Group** — AND, OR, EXCLUSIVE-OR, compare, rotate or complement data in registers or in memory
- **Branch Group** — 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 8085A is organized into 8-bit quantities, called **Bytes**. Each byte has a unique 16-bit binary address corresponding to its sequential position in memory.

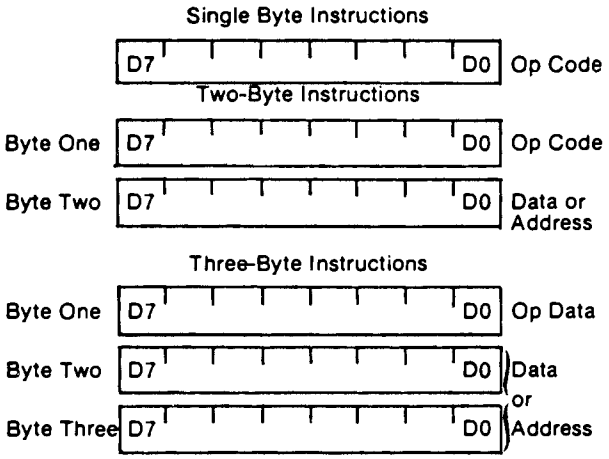
The 8085A can directly address up to 65,536 bytes of memory, which may consist of both read-only memory (ROM) elements and random-access memory (RAM) elements (read/write memory).

Data in the 8085A 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 8085A, BIT 0 is referred to as the **Least Significant Bit (LSB)**, and BIT 1 (of an 8 bit number) is referred to as the **Most Significant Bit (MSB)**.

The 8085A 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 8085A 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 and 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 the three-bit field.

Condition Flags:

There are five condition flags associated with the execution of instructions on the 8085A. 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.

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 module 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 8085A 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 NAME
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 of registers enclosed in the parentheses.

← — "Is transferred to"

∧ — Logical AND

⊻ — Exclusive OR

∨ — Inclusive OR

\+ — Addition

− — Two's complement subtraction

* — Multiplication

⟷ — "Is exchanged with"

— — The one's complement (e.g., (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 8085A. Each instruction is described in the following manner:

1. The MCS 85™ macro 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 A-2) are listed. The last line lists any of the five Flags that are affected by the execution of the instruction.

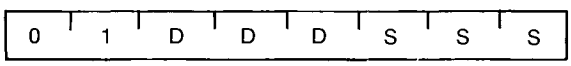
Data Transfer Group:

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.

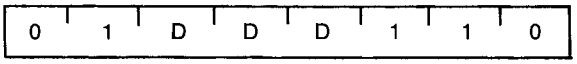


Cycles: 1
States: 4
Addressing: register
Flags: none

MOV r, M (Move from memory)

(r) ← ((H) (L))

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

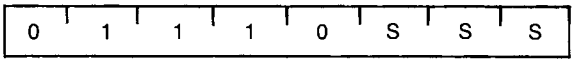


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

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.

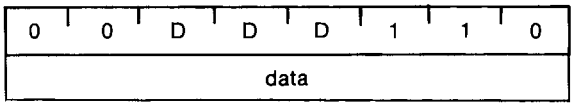


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move Immediate)

(r) ← (byte 2)

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

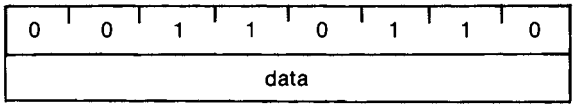


Cycles: 2
States: 7
Addressing: immediate
Flags: none

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.

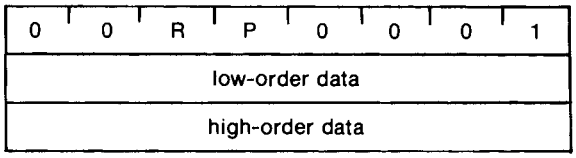


Cycles: 3
States: 10
Addressing: immed./reg. indirect
Flags: none

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.

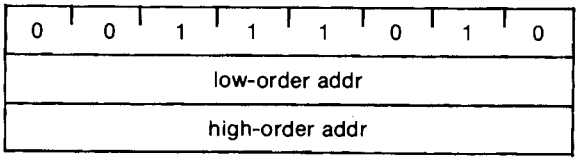


Cycles: 3
States: 10
Addressing: immediate
Flags: none

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.

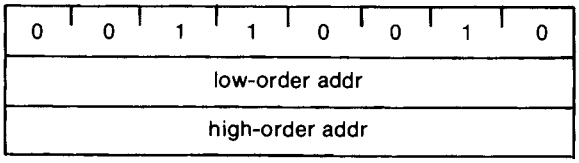


Cycles: 4
States: 13
Addressing: direct
Flags: none

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.



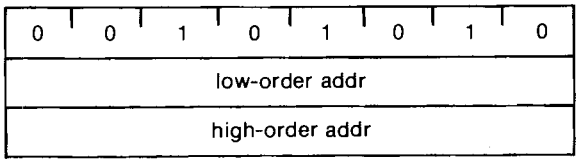
Cycles: 4
States: 13
Addressing: direct
Flags: none

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.



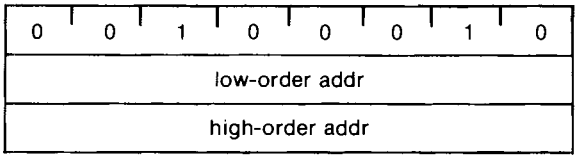
Cycles: 5
States: 16
Addressing: direct
Flags: none

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))

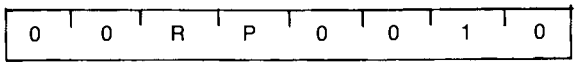
The content of the memory location, whose address is in the register pair rp, is moved to register A. Note: Only register pairs rp=B (registers B and C) or rp=D (registers D and E) may be specified.

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.

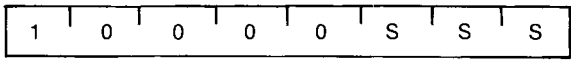
Unless indicated otherwise, all instructions in this group affect the Zero, Sign, Parity, Carry, and Auxiliary Carry flags according to the standard rules.

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.

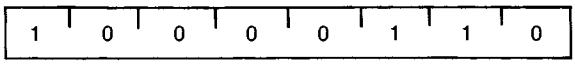


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

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.

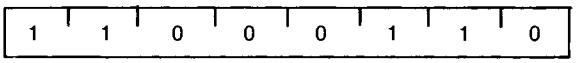


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

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.



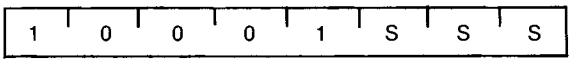
data

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.

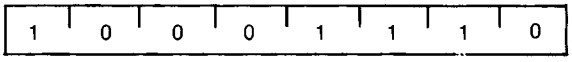


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

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 contents of the CY flag are added to the accumulator. The result is placed in the accumulator.

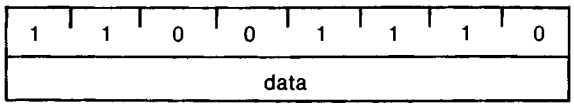


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

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.

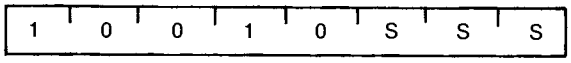


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

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))

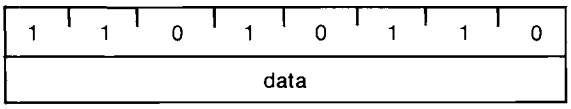
The content of the memory location whose address is contained in the H and L registers is subtracted from the content of the accumulator. The result is placed in the accumulator.

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.

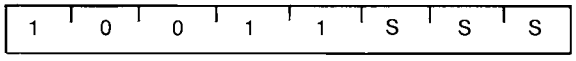


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

SBB r (Subtract Register with borrow)

(A) ← (A) - (r) - (CY)

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

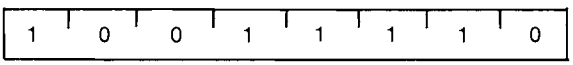


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

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.

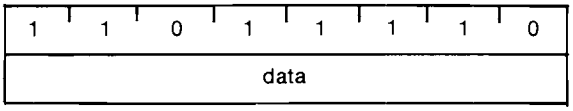


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

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.

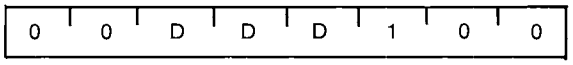


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

INR r (Increment Register)

(r) ← (r) + 1

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

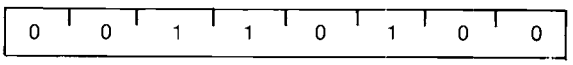


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,AC

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.

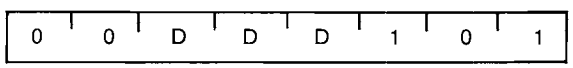


Cycles: 3
States: 10
Addressing: reg. indirect
Flags: Z,S,P,AC

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: 4
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.

0	0	1	1	0	1	0	1

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

INX rp (Increment register pair)

(rh) (rl) ← (rh) (rl) + 1

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

0	0	R	P	0	0	1	1

Cycles: 1
States: 6
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.**

0	0	R	P	1	0	1	1

Cycles: 1
States: 6
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.

0	0	R	P	1	0	0	1

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.

0	0	1	0	0	1	1	1

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 and AC is set.**

1	0	1	0	0	S	S	S

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 and AC is set.**

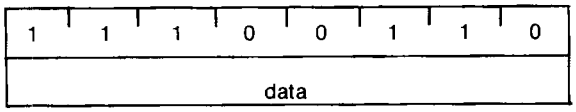
1	0	1	0	0	1	1	0

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 flag is cleared and AC is set.**

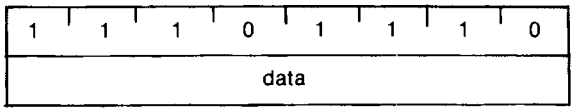


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

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.**

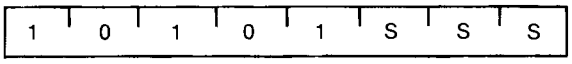


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

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.**

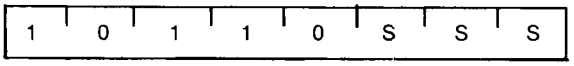


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ORA r (OR Register)

(A) ← (A) ∨ (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.**

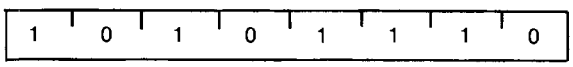


Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

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.**

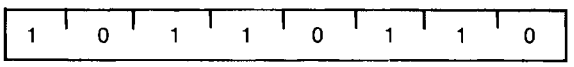


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORA M (OR memory)

(A) ← (A) ∨ ((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.**

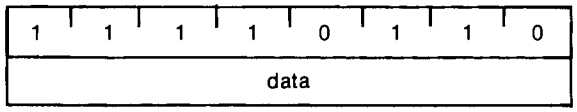


Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

ORI data (OR immediate)

(A) ← (A) ∨ (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.**

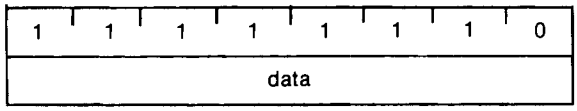


Cycles: 2
States: 7
Addressing: immediate
Flags: Z,S,P,CY,AC

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

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).**

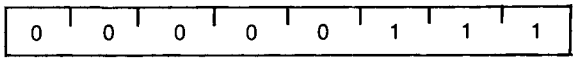
Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

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.**

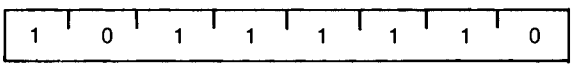


Cycles: 1
States: 4
Flags: CY

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)).**



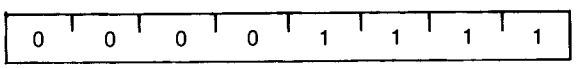
Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY,AC

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.**

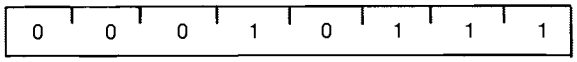


Cycles: 1
States: 4
Flags: CY

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.**

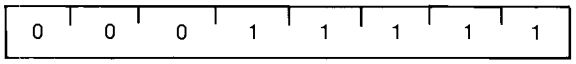


Cycles: 1
States: 4
Flags: CY

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.**

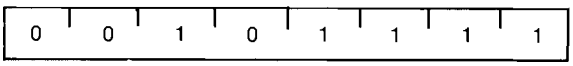


Cycles: 1
States: 4
Flags: CY

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.**

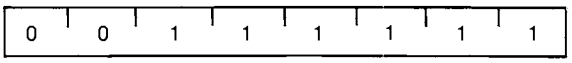


Cycles: 1
States: 4
Flags: none

CMC (Complement carry)

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

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

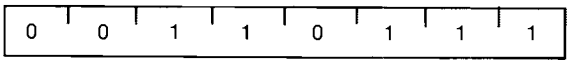


Cycles: 1
States: 4
Flags: CY

STC (Set carry)

$(CY) \leftarrow 1$

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



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.

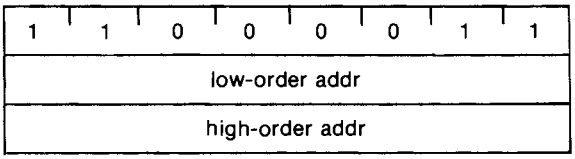
The two types of branch instructions are unconditional and conditional. Unconditional transfers simply perform the specified operation on register PC (the program counter). Conditional transfers examine the status of one of the four processor flags to determine if the specified branch is to be executed.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

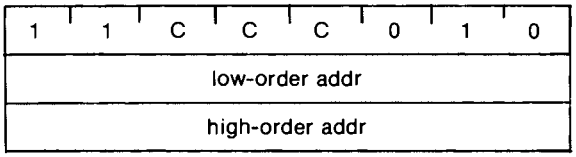


Cycles: 3
States: 10
Addressing: immediate
Flags: none

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.

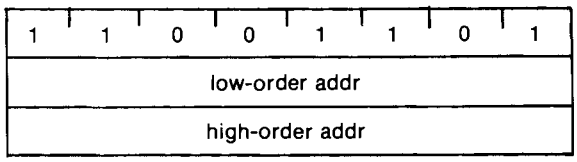


Cycles: 2/3
States: 7/10
Addressing: immediate
Flags: none

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.

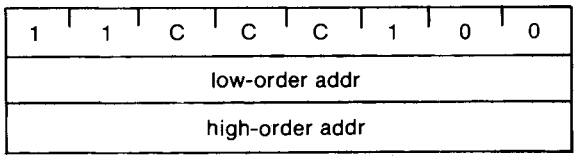


Cycles: 5
States: 18
Addressing: immediate/reg. indirect
Flags: none

Ccondition addr (Condition call)

If (CC),
((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.

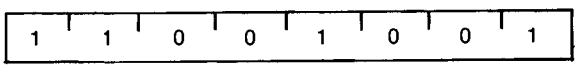


Cycles: 2/5
States: 9/18
Addressing: immediate/reg. indirect
Flags: none

RET (Return)

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

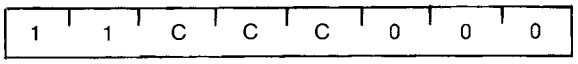
The content of the memory location whose address is specified in register SP is moved to the low-order eight bits of register PC. The content of the memory location whose address is one more than the content of register SP is moved to the high-order eight bits of register PC. The content of register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



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

Rcondition (Conditional return)

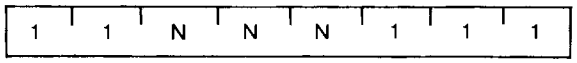
If (CCC),
(PCL) ← ((SP))
(PCH) ← ((SP) + 1)
(SP) ← (SP) + 2
If the specified condition is true, the actions specified in the RET instructions (see above) are performed; otherwise, control continues sequentially.



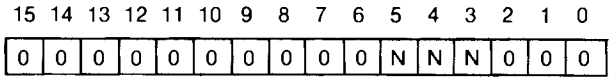
Cycles: 1/3
States: 6/12
Addressing: reg. indirect
Flags: none

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.



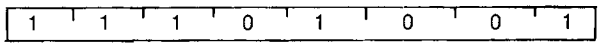
Cycles: 3
States: 12
Addressing: reg. indirect
Flags: none



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: 6
Addressing: register
Flags: none

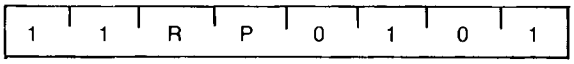
Stack, I/O, and Machine Control Group:

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: 12
Addressing: reg. indirect
Flags: none

PUSH PSW (Push processor status word)

((SP) - 1) ← (A)
((SP) - 2)0 ← (CY), ((SP) - 2)1 ← X
((SP) - 2)2 ← (P), ((SP) - 2)3 ← X
((SP) - 2)4 ← (AC), ((SP) - 2)5 ← X
((SP) - 2)6 ← (Z), ((SP) - 2)7 ← (S)
(SP) ← (SP) - 2 X: Undefined.
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.

1	1	1	1	0	1	0	1

Cycles: 3
States: 12
Addressing: reg. indirect
Flags: none

FLAG WORD

D7	D6	D5	D4	D3	D2	D1	D0
S	Z	X	AC	X	P	X	CY

X: undefined

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.**

1	1	R	P	0	0	0	1

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
(AP) ← ((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.

1	1	1	1	0	0	0	1

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.

1	1	1	0	0	0	1	1

Cycles: 5
States: 16
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.

1	1	1	1	1	0	0	1

Cycles: 1
States: 6
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.

1	1	0	1	1	0	1	1
port							

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.

1	1	0	1	0	0	1	1
port							

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

EI (Enable interrupts)

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

1	1	1	1	1	0	1	1

Cycles: 1
States: 4
Flags: none

DI (Disable interrupts)

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

1	1	1	1	0	0	1	1

Cycles: 1
States: 4
Flags: none

HLT (Halt)

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

0	1	1	1	0	1	1	0

Cycles: 1
States: 5
Flags: none

NOP (No op)

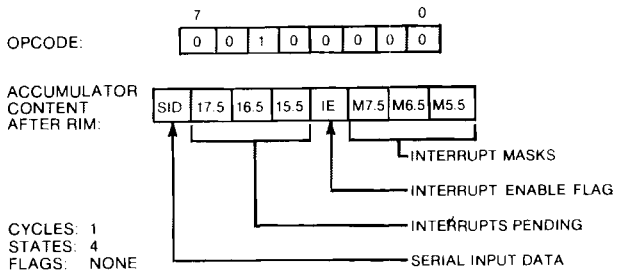
No operation is performed. The registers and flags are unaffected.

0	0	0	0	0	0	0	0

Cycles: 1
States: 4
Flags: none

RIM (Read Interrupt Mask)

After the execution of the RIM instruction, the accumulator is loaded with the restart interrupt masks, any pending interrupts, and the contents of the serial input data line (SID).



SIM (Set Interrupt Masks)

During execution of the SIM instruction, the contents of the accumulator will be used in programming the restart interrupt masks. Bits 0-2 will set/reset the mask bit for RST 5.5, 6.5, 7.5 of the interrupt mask register, if bit 3 is 1 ("set"). Bit 3 is a "Mask Set Enable" control.

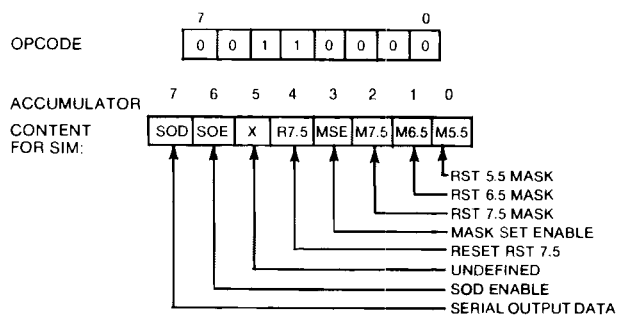
Setting the mask (i.e., masked bit = 1) disables the corresponding interrupt.

	Set	Reset
RST 5.5 MASK	if bit 0 = 1	if bit 0 = 0
RST 6.5 MASK	bit 1 = 1	bit 1 = 0
RST 7.5 MASK	bit 2 = 1	bit 2 = 0

RST 7.5 (edge trigger enable) internal request flip flop will be reset if bit 4 of the accumulator = 1; regardless of whether RST 7.5 is masked or not.

$\overline{\text{RESET IN}}$ input (pin 36) will set all RST MASKs, and reset/disable all interrupts.

SIM can, also load the SOD output latch. Accumulator bit 7 is loaded into the SOD latch if bit 6 is set. The latch is unaffected if bit 6 is a zero. $\overline{\text{RESET IN}}$ input sets the SOD latch to zero.



CYCLES: 1
STATES: 4
FLAGS: NONE

INSTRUCTION SET

Summary of Processor Instructions

Mnemonic	Description	Instruction Code [1] D7	D6	D5	D4	D3	D2	D1	D0	Clock[2] Cycles
MOV r1, 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
HLT	Halt	0	1	1	1	0	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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
RST	Restart	1	1	A	A	A	1	1	1	11
IN	Input	1	1	0	1	1	0	1	1	10
OUT	Output	1	1	0	1	0	0	1	1	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
LXI SP	Load immediate stack pointer	0	0	1	1	0	0	0	1	10
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
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
XCHG	Exchange D & E, H & L Registers	1	1	1	0	1	0	1	1	4
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
PCHL	H & L to program counter	1	1	1	0	1	0	0	1	5
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
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
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
INX SP	Increment stack pointer	0	0	1	1	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
DCX SP	Decrement stack pointer	0	0	1	1	1	0	1	1	5
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
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
EI	Enable Interrupts	1	1	1	1	1	0	1	1	4
DI	Disable Interrupts	1	1	1	1	0	0	1	1	4
NOP	No-operation	0	0	0	0	0	0	0	0	4
RIM	Read Interrupt Mask	0	0	1	0	0	0	0	0	4
SIM	Set Interrupt Mask	0	0	1	1	0	0	0	0	4

NOTES: 1. DDD or SSS — 000B — 001 C — 010 D — 011 E — 100 H — 101 L — 110 Memory — 111 A.
2. Two possible cycle times, (5/11) indicate instruction cycles dependent on condition flags.

* * *

APPENDIX B
DECODE PROM (U32)

B-1. INTRODUCTION

The iSBC 80/24A board utilizes a factory programmed decode PROM to produce certain memory and I/O related signals. The PROM (device U32) is shown in figure 5-4, sheet 8. This appendix provides a brief description of the decode PROM, including a memory map.

B-2. DESCRIPTION

An Intel 3625A-1 PROM (1K × 4) is used for this application. It is programmed at the factory, as shown in table B-1. There are ten active inputs and four active outputs, as outlined below:

INPUTS

Addresses AA-AF — From the 8085A-2.

MMPRESENT/ — Indicates that I/O addresses C0-CF and F0-FF are valid, on-board I/O addresses.

PTYPE 0 & PTYPE 1 — Indicate whether 1K, 2K, 4K or 8K PROMs are being utilized for on-board program storage.

IO/ — Indicates an I/O cycle when true. However, when IO/ is false, a memory access cycle is indicated.

SIGNAL	DECODE PROM INPUT
AA-AF	A0-A5
PTYPE 0-1	A6-A7
IO/	A8
MMPRESENT/	A9

OUTPUTS

DCDPROM 1 & DCDPROM 2 — These signals (01 & 02) are used in two ways depending on the state of the IO/ input. When the IO/ input is false, the two signals (DCDPROM 1 & DCDPROM 2) are decoded to select one out of four PROMs. When the IO/ is true, the two signals (DCDPROM 1 & DCD PROM 2), in conjunction with another PROM output (03), are decoded to select on-board I/O devices.

PROMACK/ — Enables the on-board PROM chip selects.

ONBDIO — Indicates that an I/O access is on-board.

OUTPUT 4 — Is included in both PROM ACK/ and ONBDIO and is considered a "valid access" bit.

MMPRESENT/	IO/	PTYPE 0	PTYPE 1	On-Board Address Space (Hex)	PROM Size
X	1	0	0	0000-0FFF	1K
X	1	1	0	0000-1FFF	2K
X	1	0	1	0000-3FFF	4K
X	1	1	1	0000-7FFF	8K

X = Irrelevant

MMPRESENT/	IO/	PTYPE 0	PTYPE 1	On-Board I/O Address Space
1	0	X	X	D4-EF
0	0	X	X	C0-CF, D4-FF

X = Irrelevant

Table B-1. Decode PROM Map

PROM ADDRESS	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
010	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
020	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
030	3	3	3	3	F	6	0	4	F	1	2	5	3	3	3	3
040	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
050	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
060	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
070	3	3	3	3	F	6	0	4	F	1	2	5	3	3	3	3
080	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
090	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0A0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0B0	3	3	3	3	F	6	0	4	F	1	2	5	3	3	3	3
0C0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0D0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0E0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
0F0	3	3	3	3	F	6	0	4	F	1	2	5	3	3	3	3
100	4	5	6	7	F	F	F	F	F	F	F	F	F	F	F	F
110	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
120	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
130	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
140	4	4	5	5	6	6	7	7	F	F	F	F	F	F	F	F
150	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
160	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
170	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
180	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
190	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
1A0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
1B0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
1C0	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
1D0	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
1E0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
1F0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
200	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
210	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
220	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
230	F	F	F	F	F	6	0	4	F	1	2	5	F	F	F	F
240	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
250	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
260	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
270	F	F	F	F	F	6	0	4	F	1	2	5	F	F	F	F
280	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
290	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
2A0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
2B0	F	F	F	F	F	6	0	4	F	1	2	5	F	F	F	F
2C0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
2D0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
2E0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
2F0	F	F	F	F	F	6	0	4	F	1	2	5	F	F	F	F
300	4	5	6	7	F	F	F	F	F	F	F	F	F	F	F	F
310	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
320	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
330	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
340	4	4	5	5	6	6	7	7	F	F	F	F	F	F	F	F
350	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
360	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
370	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
380	4	4	4	4	5	5	5	5	6	6	6	6	7	7	7	7
390	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3A0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3B0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3C0	4	4	4	4	4	4	4	4	5	5	5	5	5	5	5	5
3D0	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
3E0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
3F0	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	F

* * *

APPENDIX C
iSBC 80/20-4 BOARD;iSBC 80/24 BOARD; iSBC 80/24A BOARD FUNCTIONAL DIFFERENCES

C-1. INTRODUCTION

This appendix summarizes the differences you need to consider if using the iSBC 80/24A board as a direct replacement for the iSBC 80/24 or 80/20-4 boards. Although the iSBC 80/24A board is functionally compatible with both the iSBC 80/24 and 80/20-4 boards, differences do exist. The primary differences are jumper post numbering, default jumper configuration, enhancements, and added features. The iSBC 80/20-4 board also differs in its operating speed and microprocessor type.

C-2. FUNCTIONAL DIFFERENCES

The differences between the boards are divided into the following categories:

a. Power Requirements

b. Board Timing and CPU

c. Memory

d. I/O

e. Pin Assignments

f. Signal Definitions

g. Jumper Post Numbering and Default Jumpers

h. Physical Characteristics

These differences are described in Sections C-3 through C-10.

C-3. POWER REQUIREMENTS

Due to component upgrading, the iSBC 80/24A board requires less power (at maximum load) than the iSBC 80/24 and 80/20-4 boards. The 80/24A boards' +5V requirement is 0.8 Amp less than for the iSBC 80/24 board, and 1.6 Amps less than for the iSBC 80/20-4 board. In addition, both the iSBC 80/24A and 80/24 boards' + 12V requirement is 0.05 Amp less than for the iSBC 80/20-4 board.

C-4. BOARD TIMING AND CPU

The iSBC 80/24A and 80/24 boards operate at 4.84 MHz (default configuration). You can select 2.42 MHz operation by reconfiguring jumpers (refer to Table 2-4 in Chapter 2). The iSBC 80/20-4 board operates at 2.15 MHz.

Since the iSBC 80/24A and 80/24 boards are equipped with the 8085A-2 microprocessor rather than the 8080A microprocessor used on the iSBC 80/20-4 board, actual processing times will be less for similar operations. The 8080A instruction cycle time is 2 microseconds, as compared to 0.8 microsecond for the 8085A-2 instruction cycle.

All 8080A instructions will operate on the 8085A-2 CPU. The 8085A-2 uses two additional instructions (RIM & SIM), and requires additional states for certain instructions. This information is summarized in table C-1.

Due to the higher CPU operating speed the iSBC 80/24A and 80/24 boards may not execute some time-dependent code written for the iSBC 80/20-4 board. However, all other code will function as written for the iSBC 80/20-4 board.

Table C-1. Summary of 8085A Instructions

Instr.	States	Cycles[1]	Notes
AR r	4	1	
AR I	7	2	
AR M	7	2	
CALL	18	5	2
CCN	9/18	2/5	2, 4
CMC	4	1	
CMA	4	1	
DAA	4	1	
DAD	10	3	
DCR r	4	1	3
DCR M	10	3	
DCX	6	1	2
DI	4	1	
EI	4	1	
HLT	5	1	5
IN	10	3	
INR r	4	1	3
INR M	10	3	
INX	6	1	2
JMP	10	3	
JCN	7/10	2/3	4
LDS	13	4	
LDAX	7	2	
LHLD	16	5	
LXU	10	3	
MVI M	10	3	
MVI r	7	2	
MOV M,r	7	2	
MOV r,M	7	2	
MOV r,r	4	1	3

Table C-1. Summary of 8085A Instructions (Continued)

Instr.	States	Cycles[1]	Notes
NOP	4	1	
OUT	10	3	
PCHL	6	1	2
POP	10	3	
PUSH	12	3	2
ROT	4	1	
RET	10	3	
RCN	6/12	1/3	2
RIM[1]	4	1	
RST	12	3	2
SHLD	16	5	
SIM[1]	4	1	
STA	13	4	
STAX	7	2	
STC	4	1	
XCHG	4	1	
XTHL	16	5	
SPHL	6	1	2

[1]New Instruction

NOTES:

1. Two possible state-cycle times indicates dependence on condition flag.
2. Increase in states over 8080 due to necessity of a T_6 during M_1.
3. Decreased from 5 to 4 states during M_1.
4. Instruction branches over last address fetch if condition false.
5. 5 cycles to get a HLT state. 1 cycle necessary to get out a HLT.

The 8085A-2 CPU has four interrupt inputs in addition to the INTR input. These inputs are accessed through the interrupt matrix, via jumper connection (refer to table 2-7 in text).

C-5. MEMORY

The iSBC 80/24A and 80/24 boards offer more on-board ROM (up to 32K bytes) and RAM (up to 8K bytes) than the iSBC 80/20-4 board. In certain configurations, ROM and RAM addressing may overlap. Chapter 2 provides instructions for proper memory configurations and addressing.

All ROM/PROM/EPROM devices used on the former boards may also be used on the iSBC 80/24A board. Chapter 2 provides installation instructions.

However, the RAM devices used on the iSBC 80/24 board are not compatible with the iSBC 80/24A and 80/20-4 boards.

D-6. I/O FACILITIES

All I/O capabilities and addressing are identical among the three boards. However, the iSBC 80/24A and 80/24 boards have additional I/O capability through the two iSBX Bus connectors (J5 and J6). If Multimodule boards are present, they require the following addressing):

J5	J6
Ports C0-C7 for MCS0/	Ports F0-F7 for MCS0
Ports C8-CF for MCS1/	Ports F8-FF for MCS1/

If Multimodule boards are not present, these addresses may be used for other applications.

C-7. PIN ASSIGNMENTS

Table C-2 provides pin assignment differences among the three boards and the Multibus interface.

Table C-2. Pin Assignments

Pin #	80/24A, 80/24	80/20-4	Multibus
P1-25	Reserved[1]	AACK/	RESERVED
P1-28	ADDRESS 10	RESERVED	ADDRESS 10
P1-29	CBRQ/	RESERVED	CBRQ/
P1-33	RESERVED[1]	INTR/	INTA/
P2-13	PFSR/	RESERVED	PFSR/
P2-19	PFIN/	PFI/	PFIN/
P2-21, 22	GRD	RESERVED	GRD
P2-30	RESERVED	WAIT/	WAIT/
P2-32	RESERVED[1]	BTMO	RESERVED
P2-36	RESERVED[1]	INTA/	RESERVED
P2-38	AUX RESET/	RESET/	AUX RESET/
J1-50	EXTERNAL INT OR	NOT USED	N/A
J2-50	+5 VOLTS		

[1]Reserved in accordance with Multibus interface specification.

C-8. SIGNAL DEFINITIONS

The signals BCLK/ and CCLK/ are approximately 108° out of phase with each other, and the new cycle period is reduced from 108.5 nsec to 103.3 nsec.

There is no WAIT/ signal provided on the iSBC 80/24A and 80/24 boards. For front panel (iSBC 655 Chassis) use, the signal ALE, P2-32, should be used for the RUN indicator. This is the standard front panel use for the RUN indicator and does not require any other wiring.

INTA is supported only to the extent of being compatible with the iSBC 80/2-4 board. Because the iSBC 80/24A and 80/24 boards do not support bus vectored interrupts, INTA is not provided on P1.

C-9. JUMPER POST NUMBERING AND DEFAULT JUMPERS

There is no correlation between the jumper posts used on former boards and those used on the iSBC 80/24A board. In the factory default mode, the boards are jumpered for the same functions. These jumper connections are compared in table C-3.

To alter a particular factory configuration jumper, refer to Chapter 2.

C-10. PHYSICAL CHARACTERISTICS

All connector locations on the iSBC 80/24A and 80/24 boards comply with the Multibus Interface Specification.

In addition, the two iSBX Bus connectors (J5 and J6) require consideration for component heights and connector locations.

Table C-3. Jumper Pairs

iSBC 80/24A Jumper Pair	iSBC 80/24 Jumper Pair	iSBC 80/20-4 Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Text Ref.
1-2	1-2	None	5-B2	Connects +5 volts to J1-50	None
3-4*	26-27*	52-53*	5-C4	Configures port E4 bus transceiver to output mode	2-24
4-18	27-41	51-52	5-C4	Configures port E4 bus transceiver to input mode	2-14
5-19*	28-42*	62-63*	5-C4	Connects port E6, bit 0 to driver/terminator socket	2-14
6-20*	29-43*	64-65*	5-C4	Connects port E6, bit 1 to driver/terminator socket	2-14
7-21*	30-44*	66-67*	5-C4	Connects port E6, bit 2 to driver/terminator socket	2-14
8-22*	31-45*	68-69*	5-C4	Connects port E6, bit 3 to driver/terminator socket	2-14
9-24*	32-47*	60-61*	5-C4	Connects port E6, bit 4 to driver/terminator socket	2-14
10-25*	33-48*	58-59*	5-C4	Connects port E6, bit 5 to driver/terminator socket	2-14
11-26*	34-49*	56-57*	5-C4	Connects port E6, bit 6 to driver/terminator socket	2-14
12-27*	35-50*	54-55*	5-C4	Connects port E6, bit 7 to driver/terminator socket	2-14
13-28*	36-51*	93-94*	5-C4	Applies +5 volts to timer gate 0 (ITGO)	None
14-29*	37-51*	94-95*	5-C4	Applies +5 volts to timer gate 1 (ITG1)	None
15-30	38-52	None	5-C4	Connects timer 1 output to timer input matrix	2-11
15-XX	38-XX	None	5-C4	Connects selected J1 bit to timer input matrix	2-11
16-XX	39-XX	None	5-C4	Connects INTROUT/BANKSEL to selected E6 bit	2-11
17-19	40-42*	hardwired	5-C4	Connects port E6, bit 0 to PIA1	2-14
22-23	45-46*	hardwired	5-C4	Connects port E6, bit 3 to PIB1	2-14
31-XX	53-XX	None	5-C4	Connects PFSR/ to selected E6 bit	2-22
32-XX	54-XX	None	5-C4	Connects PROM ENABLE to selected E6 bit	2-11
33-48	3-4	None	6-B2	Connects +5 volts to J2-50	None
34-35*	55-56*	71-72*	6-C4	Configures port E8 bus transceiver to output mode	2-14
35-50	56-70	70-71	6-C4	Configures port E8 bus transceiver to input mode	2-14
36-51*	57-71*	81-82*	6-C4	Connects port EA, bit 0 to driver/terminator socket	2-14
37-52*	58-72*	83-84*	6-C4	Connects port EA, bit 1 to driver/terminator socket	2-14
38-53*	59-73*	85-86*	6-C4	Connects port EA, bit 2 to driver/terminator socket	2-14
39-54*	60-74*	87-88*	6-C4	Connects port EA, bit 3 to driver/terminator socket	2-14
40-XX	61-XX	72-XX	6-C4	Ground post	2-14
41-56*	62-76*	79-80*	6-C4	Connects port EA, bit 4 to driver/terminator socket	2-14
42-57*	63-77*	77-78*	6-C4	Connects port EA, bit 5 to driver/terminator socket	2-14
43-58*	64-78*	75-76*	6-C4	Connects port EA, bit 6 to driver/terminator socket	2-14
44-59*	65-79*	73-74*	6-C4	Connects port EA, bit 7 to driver/terminator socket	2-14
45-XX	66-XX	97-XX	6-C4	Connects BDET/ to selected EA bit	2-14
46-XX	67-XX	96-XX	6-C4	Connects BRI/ to selected EA bit	2-14
47-XX	68-XX	90-XX	6-C4	Connects AUX0/ to selected EA bit	2-14

Table C-3. Jumper Pairs (Continued)

iSBC 80/24A Jumper Pair	iSBC 80/24 Jumper Pair	iSBC 80/20-4 Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Text Ref.
49-51	69-71*	hardwired	6-C4	Connects port EA, bit 0 to PIA2/	2-14
54-55	74-75*	hardwired	6-C4	Connects port EA, bit 3 to PIA1/	2-14
60-XX	80-XX	None	6-C4	Connects SOD to selected EA bit	2-14
61-XX	81-XX	None	6-C4	Connects SID to selected EA bit	2-14
62-XX	82-XX	89-XX	6-C4	Connects AUX1/ to selected EA bit	2-14
63-64*	J4-1-J4-14*	hardwired	7-C3	Connects transmit data to J3-4	2-13
63-74	J4	None	7-C3	J3 serial jumper matrix	2-13
65-66*	J4-2-J4-13*	hardwired	7-C3	Connects receive data to J3-6	2-13
67-68*	J4-3-J4-12*	hardwired	7-C3	Connects RTS to J3-8	2-13
69-70*	J4-4-J4-11*	hardwired	7-C3	Connects CTS to J3-10	2-13
71-72*	J4-5-J4-10*	hardwired	7-C3	Connects DTR to J3-13	2-13
73-74*	J4-6-J4-9*	hardwired	7-C3	Connects −12 volts to J3-19	2-13
75-80	9-10	9-10	7-A2	Connects −12 volts to J3-19	2-25
76-77	5-6	1-2	7-A2	Connects +5 volts to J3-23	None
78-79	7-8	3-4	7-A2	Connects +12 volts to J3-22. Used for TTY serial interface	2-25
81-82	11-12	11-12	7-C2	RS232C Protective Ground J3-2	2-13
83-84	13-14	7-8	7-D7	Connects Secondary RST (J3-11) input to AUX1/	2-13
84-85	14-15	6-8	7-D7	Connects Secondary TxD (J3-11) input to AUX1/	2-13
86-88	16-17	17-18	7-D6	Connects external Receive Clock (RxC)	2-13
87-88*	17-18*	16-18*	7-D6	Connects internal Receive Clock (RxC)	2-13
89-90*	18-19*	19-21*	7-D6	Connects internal Transmit Clock (TxC)	2-13
89-91	19-20	20-21	7-D6	Connects external Transmit Clock (TxC)	2-13
92-95	21-24	5-14	7-D2	Connects AUX0/ output to J3-21	2-13
93-95	22-24	14-15	7-D2	Connects AUX0/ output to J3-5	2-13
94-95	23-24	13-14	7-D2	Connects AUX0/ output to J3-26	2-13
96-105	156-157	None	3-C5	Connects for 4.84 MHz clock only	4-3
97-106*	140-143*	137-138*	3-B6	Enables failsafe timer	4-35
98-107	141-144	None	3-B7	Reserved	None
99-108*	142-145*	None	3-B7	Connects RAMACK/ to READY gate	4-29
100-109	148-149*	W1	4-C7	RAM size jumper. See table 2-4	2-12
101-110	146-147	W1	4-C7	RAM size jumper. See table 2-4	2-12
102-103*	150-151	117-121	4-C5	Indicates RAM area: E000-FFFFH	2-12
102-104	150-155	117-121	4-C5	RAM address matrix. See table 2-4	2-12
111-113	150-155	117-121	4-C5	RAM address matrix. See table 2-4	2-12
112-113	154-155*	120-121*	4-C5	Indicates RAM area: 2000-3FFFH	2-12
114-115*	83-84*	36-37	9-B4	Connects ground to IR7 on PIC	None
	84-85*	37-38	9-B4	Connects ground to IR7 on PIC	None
	85-86*	38-39	9-B4	Connects ground to IR7 on PIC	None
	86-87*	39-31	9-B4	Connects ground to IR7 on PIC	None
114-162	83-123	24-50	9-C4	Interrupt Jumper Matrix. See table 2-7	2-15
133-134*	101-106*	26-35*	9-C4	Connects INT1/ to IR1 on PIC	2-15
136-137	182-183	None	10-D5	Multimodule board option (reserved); J5	None
147-148	184-185	None	10-C3	Multimodule board option (reserved); J6	None
153-154*	100-118*	25-45*	9-C4	Connects OIT0 to IR2 on PIC	2-15
155-159*	119-120*	None	9-C4	Connects ground to RST5.5 on CPU	2-15
156-160*	120-121*	None	9-C4	Connects ground to RST6.5 on CPU	2-15
157-161*	121-122*	None	9-C4	Connects ground to RST7.5 on CPU	2-15
158-162*	122-123*	None	9-C4	Connects ground to TRAP on CPU	2-15

Table C-3. Jumper Pairs (Continued)

iSBC 80/24A Jumper Pair	iSBC 80/24 Jumper Pair	iSBC 80/20-4 Jumper Pair	iSBC 80/24A Sheet Grid Reference	Function	Text Ref.
163-166	124-125	None	3-D5	4.84 MHz Clock input to 8085A-2 CPU	4-3
164-165*	131-136*	91-92*	7-C7	Connects 2.15 MHz to PIT Clock divider	None
166-168*	125-126*	None	3-D5	9.68 MHz Clock input to 8085A-2 CPU	4-3
167-170	134-135	141-143	7-B6	Connects Timer 0 output to Timer 1 input	None
167-176	132-135	None	7-B6	Connects Timer 0 output to port E6 bit via CLK OUT line	None
169-170*	130-134*	141-142*	7-B6	Connects 1.075 MHz to CLK1 on PIT	None
171-174*	138-139*	None	7-B6	Connects 8224 OSC output to divider network	None
172-173*	129-133*	hardwired	7-B6	Connects 1.075 MHz to CLK0 on PIT	None
175-XX	128-XX	None	7-B6	Connects 134.4 KHz. See table 2-4	None
177-197	J7 & J8	W7 & W8	4-B6	EPROM Jumper Matrix	2-9
179-180	J8-2-J8-13*	W7 A-B*	4-B6	Connects −5 volts to EPROM socket pin 23 Vbb/A11	2-9
181-182*	J7-4-J7-11	None	4-B6	Connects address AB to EROM socket pin 23 Vbb/A11	2-9
183-184*	J8-8-J8-18	W8 A-B	4-B6	Connects address AA to EROM socket pin 21 Vdd/A10	2-9
185-186	J8-1-J8-14*	WB A-C*	4-B6	Connects +12 volts to EPROM socket pin 21 Vdd/A10	2-9
189-190	J8-4-J7-8*	hardwired	4-B6	Connects ground to EPROM socket pin 20 PGM/CE\	2-9
190-191*	J7-8-J7-6	None	4-B6	Connects PROMACK\ to EPROM socket pin 20 PGM/CE\	2-9
192-193	J8-5-J8-15*	None	4-B6	Connects ground to PTYPE0	2-9
193-194*	J8-15-J8-9	None	4-B6	Connects +5 volts to PTYPE0	2-9
195-196	J8-7-J7-16*	None	4-B6	Connects ground to PTYPE1	2-9
196-197*	J8-16-J7-2	None	4-B6	Connects +5 volts to PTYPE1	2-9
198-199*	158-159*	110-111*	2-A6	Connects 9.68 MHz BCLK\ to P1-13 (output)	4-3
200-201*	160-161*	hardwired	2-D3	Connects BPRO\ to P1-16 (output)	2-19
202-203	162-163	135-136	2-D6	Connects AACK\ to iSBC 80/24A board via P1-25 (input)	None
204-205*	164-165*	112-113*	2-A6	Connects 9.68 MHz CCLK\ to P1-31 (output)	4-3
206-211	166-167	None	9-C5	Connects INTROUT/BANKSEL to P1-28 (ADR10\) output	None
216-217*	176-177*	W5: A-B*	1-D6	Used for +5 volt Battery Backup	2-22
218-219*	178-179*	hardwired	3-B2	Connects INTA\ to P2-36 (output)	4-23
220-221*	180-181*	hardwired	3-A2	Connects BTMO to P2-34 (output)	4-34

* Indicates default jumper or connection

* * *

REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readability, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

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

3. Is this the right type of publication for your needs? Is it the right level? What other types of publications are needed?

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

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating). _____

NAME _____ DATE _____

TITLE _____

COMPANY NAME/DEPARTMENT _____

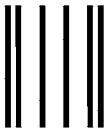
ADDRESS _____

CITY _____ STATE _____ ZIP CODE _____
(COUNTRY)

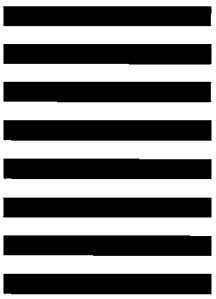
Please check here if you require a written reply. ☐

WE'D LIKE YOUR COMMENTS . . .

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



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 79 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Intel Corporation
OMS Technical Publications, MS: HF2-52
5200 N.E. Elam Young Parkway
Hillsboro, Oregon 97124-9987