intel®

iSBC® 486/12 Series Single Board Computers Hardware Reference Manual

Order Number: 507914-003

intel®

iSBC® 486/12 Series Single Board Computers Hardware Reference Manual

Order Number: 507914-003

In the United States, additional copies of this manual or other Intel literature may be obtained by writing:

Literature Distribution Center
Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

Or you can call the following toll-free number:

1-800-548-4725

In locations outside the United States, obtain additional copies of Intel documentation by contacting your local Intel sales office. For your convenience, international sales office addresses are located directly after the reader reply card in the back of this manual.

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

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 Agreement, 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.

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

i386	int$_e$l®	Intel387	iLBX	iSBC®
i486	Intel386	Intel486	iRMX®	iSBX
Intel®				

WARNING

RS-232-C driver devices and other serial communication devices can fail when subjected to transient, high-voltage surges. This type of failure can also cause irreparable damage to the board in the process.

To minimize the possibility of a failure, Intel recommends following these established, industry-standard guidelines:

- High-voltage surge protection is not provided at the board level; you must supply it at the system level. For example, you should always plug your system AC power into a surge suppressor device.

  You should also follow all specifications associated with the type of serial driver circuitry being used (such as the RS-232-C interface), when installing equipment on a serial communication circuit. This includes, but is not limited to, cable shielding and proper grounding practices.

- Do not route serial communications cabling near high-voltage lines, large electrical motors, or any other potential sources of induced energy.
- Never insert or remove a board from a system when the system power is on.
- Do not connect or disconnect serial communication cabling when the system power is on. This includes the equipment that you are connecting to, as well as any equipment already connected to the cabling.
- During lightning storms, when practical, disconnect AC power and serial communication cabling to all attached equipment.
- At all times, follow proper procedures to protect serial communication circuitry from electrostatic discharge (ESD).
- If you suspect that the serial driver circuitry on a system board has been damaged, turn off the power to the system, remove the board, and have it checked by qualified repair personnel.

REV.	REVISION	DATE
-001	Original Issue.	01/91
-002	Revised to correct minor errors, add MM3008, MM3016, and MM3032 memory modules, and update default jumper list.	09/91
-003	Revised to correct minor errors and to add the board versions with the Intel486™ DX2 CPU.	09/92

CAUTION

This board has been verified to be within energy emission limits for Class B computing devices defined in the FCC Rules, Part 15, Subpart J.

This board generates and uses energy of about the same frequency as radio and TV signals. Installed correctly, it will probably not interfere with your radio and TV. However, we do not guarantee that it will not interfere.

Only peripherals (computer input/output devices, terminals, printers, etc.) that are certified to comply with Class B limits may be attached to this board. Use shielded, grounded cables to connect them.

If the board is installed incorrectly or if it is operated with non-certified peripherals, it may interfere with the reception of radio and TV broadcasts.

For more information about interference, you can order the following booklet from the U. S. Government Printing Office, Washington, DC 20402. Ask for stock number 004-000-00345-4:

"How to Identify and Resolve Radio-TV Interference Problems"

CAUTION

Any changes or modifications to this device, which are not indicated in this manual, could cause harmful interference and void the user's authority to operate this device.

Contents

Chapter 1. Introduction to the iSBC® 486/12 Series Products

Introduction 1-2
System Considerations 1-3
Specifications 1-3
About This Manual 1-7
What's in This Manual 1-7
Notational Conventions 1-10
Active-Low Signal Name Conventions 1-10
Component Identifiers 1-10
Hexadecimal Numbers 1-10
Bit and Byte Order 1-11
Ranges of Bits or Signal Lines 1-11
Undefined Bits and Software Compatibility 1-11
Special Symbols 1-12
Related Publications 1-13
How To Report Errors in This Manual 1-14
About Intel 1-15
Intel Real-Time Operating Systems 1-15
Intel Customer Services 1-15
Intel Multibus Products 1-16

Chapter 2. Board Operation

Introduction 2-2
CPU and Memory 2-4
ADMA, DAG, and SPC 2-5
I/O Subsystem 2-5
Multibus/Dual-Port Memory 2-6

Chapter 3. Installation

Introduction 3-2
Resistor Pack and Fuse Data 3-2
Electrostatic Discharge Damage 3-3
Unpacking and Inspection 3-4

Chapter 3. Installation (continued)

Component and Module Installation 3-4
MM3 Memory Module Installation 3-5
Parts Required for Installing MM3 Memory Modules 3-7
Memory Module Sizes 3-7
MM3 Installation Procedure A 3-8
MM3 Installation Procedure B 3-12
EPROM Installation 3-16
SCSI Termination Resistors and Fuse 3-18
Multibus Termination Resistors 3-19
RS-422A/449 Termination Resistors 3-20
3487 RS-422A/449 Line Driver 3-21
SBX Board Installation 3-21
Single-Wide Module Installation 3-23
Double-Wide SBX Module Installation 3-25
Board Installation Procedure 3-27

Chapter 4. Real Mode Memory Configuration

Introduction 4-2
Related Information 4-2
New Terms in this Chapter 4-2
Default Real Mode Memory Map 4-3
Configuration Priority 4-3
Switching Between Real and Protected Modes 4-4
Configuration Procedure 4-5
1st Priority: Multibus Window Configuration 4-5
Window Size 4-6
Local Address to Multibus Window Address Mapping 4-6
Window Jumper Configuration 4-6
2nd Priority: DRAM Configuration (Real Mode) 4-8
DRAM Size 4-8
DRAM Starting and Ending Addresses 4-8
DRAM Access from the Multibus Interface 4-8
DRAM Jumper Configuration 4-8
3rd Priority: EPROM Configuration 4-10
EPROM Size 4-10
EPROM Access Time 4-11
EPROM Address Range (Real Mode) 4-11
EPROM Access from the Multibus Interface 4-11
EPROM Jumper Configuration 4-12
4th Priority: Multibus Memory Space Configuration 4-13
Sample Worksheet 4-15
Configuration Questions and Answers 4-17

Chapter 5. Protected Mode Memory Configuration

Introduction 5-2
Related Information 5-2
New Terms in This Chapter 5-2
Configuration Procedure 5-2
Default Protected Mode Memory Map 5-3
Configuration Priority 5-5
Switching Between Real and Protected Modes 5-6
1st Priority: DRAM Configuration 5-6
DRAM Size 5-6
DRAM Starting and Ending Addresses (First 16 Mbyte Page) 5-6
DRAM Access from the Multibus Interface 5-7
DRAM Jumper Configuration 5-7
2nd Priority: EPROM Configuration 5-8
EPROM Size and Access Time 5-8
EPROM Address Range (PVAM) 5-9
EPROM Access from the Multibus Interface 5-9
EPROM Jumper Configuration 5-10
3rd Priority: Multibus Memory Space Configuration 5-10
Multibus Memory Space in the Third Gbyte Page 5-11
Multibus Memory Space in the First 16 Mbyte Page 5-12
Sample Worksheet 5-13
Configuration Questions and Answers 5-15

Chapter 6. System Memory Configuration

Introduction 6-2
Related Information 6-2
New Terms in this Chapter 6-2
Configuration Procedure 6-3
Default System Memory Map 6-4
Configuration Priority 6-5
Dual-Port Configuration 6-5
Overview 6-5
Dual-Port 16 Mbyte Page Register 6-5
Dual-Port Starting and Ending Addresses 6-7
Dual-Port Alias 6-8
Starting and Ending Addresses Configuration 6-11
Dual-Port Alias Configuration 6-15
Dual-Port Alias Example 6-16
Dual-Port Alias Jumpers 6-19
Dual-Port 16 Mbyte Page Register 6-19
Dual-Port Lock 6-20

Chapter 6. System Memory Configuration (continued)

Real Mode Multiprocessing Configuration 6-21
Real Mode Multiprocessing Configuration Options 6-21
Real Mode Multiprocessing Examples 6-21
Protected Mode Multiprocessing Configuration 6-25
Protected Mode Multiprocessing Configuration Options 6-25
Protected Mode Multiprocessing Example 6-25
Sample Worksheets 6-28
Configuration Question and Answers 6-30

Chapter 7. I/O Subsystem Configuration

Introduction 7-4
Related Information 7-4
New Terms in this Chapter 7-4
Configuration Procedure 7-5
Default Configuration 7-5
Default I/O Addresses 7-6
Special I/O Addresses 7-9
I/O Configuration 7-10
82C54 Programmable Interval Timer (PIT) 7-10
82C54 PIT Default Configuration 7-11
82C54 PIT Jumpers 7-12
82C54 PIT Programming 7-13
82C54 PIT Modes of Operation 7-13
82C54 PIT Initialization 7-13
82C54 I/O Port Addresses 7-13
82C54 Programming Example 7-14
82C54 PIT Count Values 7-16
82C55A Programmable Peripheral Interface (PPI) 7-17
82C55A PPI Overview 7-17
82C55A PPI Default Configuration 7-19
Port A Configuration 7-20
Port A Configuration for Output 7-20
Port A Configuration for Input 7-20
Port A Configuration for Bidirectional I/O 7-21
Port B Configuration 7-23
PB<2:0> User-Defined Code 7-23
PB3 LPT Interrupt Input 7-24
PB4 LPT Acknowledge* Input 7-25
PB5 Power-Fail Sense (PFSN*) Signal 7-25
PB6 LPT Error Input 7-26
PB7 Used To Control Direction of 74ALS245 (Optional) 7-26
PB7 LPT Busy Input 7-26

Chapter 7. I/O Subsystem Configuration (continued)

Port C Configuration 7-27
PC0 LPT Strobe* Output 7-27
PC1 Dual-Port Lock Output 7-27
PC2 NMI Mask Output 7-28
PC3 Multibus Time-Out Clear Output 7-28
PC4 CLREDGE1* Output (Optional) 7-29
PC<6:5> Multibus Interrupt Drivers 7-30
PC7 LPT Interrupt Latch Clear Output 7-31
82C55A PPI Stake Pins 7-33
82C55A Programming 7-34
82C55A PPI I/O Addresses 7-34
Supported Modes of Operation 7-34
82C55A PPI Initialization 7-36
Bit Set/Reset Control of Port C 7-38
82C55A PPI Applications 7-39
Centronics-Compatible Printer 7-39
NMI Mask Control 7-42
Switching the Board Between Real and Protected Modes 7-42
MB87033B SCSI Peripheral Controller (SPC) 7-44
SCSI Interface Overview 7-44
iSBC 486/12S SCSI Block Diagram 7-44
SCSI Signals 7-46
SCSI ID Jumpers and GP3 Bit Jumper 7-48
SCSI Termination Resistors 7-48
SCSI Termination Power 7-49
MB87033B SPC Registers 7-49
MB87033B SPC Register I/O Addresses. 7-50
MB87033B SPC Programming 7-51
SCSI Control Registers 7-51
SCZID Register (1A1H - Read) 7-52
SCZINTDIS Register (1A1H - Write) 7-53
BLAST Register (1A3H) 7-53
HIN Register (1A5H) 7-54
BYTXFER Register (1A7H) 7-54
TURBO Register (1A9H) 7-54
SCZRST Register (1ABH - Write Only) 7-54
FIFO Data (1AEH) 7-54
8274 Multiple-Protocol Serial Controller (MPSC) 7-55
Serial Channel Configurations 7-57
Serial Connectors 7-62
MPSC I/O Ports 7-62
MPSC Configuration 7-62
Default Serial Configuration 7-62
Channel A RS-232C DCE Configuration 7-64
Channel A RS-232C DTE Configuration 7-64
Channel A RS-422A/449 DCE Configuration 7-66
Channel A RS-422A Multidrop Configuration 7-68

Chapter 7. I/O Subsystem Configuration (continued)

Serial Interrupt Configuration 7-68
Serial Break Detect Circuit 7-68
8274 MPSC Programming 7-68
SBX Interface 7-71
SBX I/O Addresses 7-71
SBX I/O Command Recovery Time 7-77
SBX DMA Operations 7-77
SBX Option Lines 7-79
SBX Interrupts 7-79
Multibus Interface (Connector P1) 7-80
BCLK* Signal Configuration 7-80
CCLK* Signal Configuration 7-80
INIT* Signal Configuration 7-81
Software Reset Configuration 7-82
Multibus Address Line Termination Resistors 7-83
Multibus Interrupts 7-84
Multibus LOCK* Signal 7-85
Multibus Buffered I/O Write Option 7-85
Multibus Arbitration 7-85
Configuration Questions and Answers 7-88

Chapter 8. CPU Subsystem Configuration

Introduction 8-3
Related Topics 8-3
New Terminology in This Chapter 8-3
Configuration Procedure 8-3
Intel486™ Microprocessor 8-4
Switching Between Real and Protected Mode 8-4
Interrupts 8-4
Board ID Register 8-5
Resets 8-7
Multibus Time-Out Action 8-9
Multibus Request Error 8-10
CPU Self-Test 8-11
Cache 8-11
Floating Point Unit 8-12
82258 ADMA 8-12
ADMA Configuration 8-12
ADMA Default Configuration 8-13
ADMA Configuration Options 8-14
ADMA Programming 8-14
General Mode Register Initialization 8-15
One- and Two-Cycle Modes 8-16
DMA Start/Termination Conditions 8-17

Chapter 8. CPU Subsystem Configuration (continued)

DMA Address Generator (DAG) 8-17
DAG Block Diagram 8-17
DAG Registers 8-18
DAG Register I/O Addresses 8-20
DAG Register Programming 8-22
DAG Control Register (Channel 0) 8-23
One-Cycle Page Register 8-24
Memory Address Page Registers 8-24
Test Register 8-25
Maximum DMA Transfer Rates 8-26
DMA Transfer Specifications 8-27
One-Cycle Transfer Modes 8-28
Paged/Non-Paged Mode 8-29
Burst/Non-Burst Mode 8-29
Compatibility/Extended Mode 8-29
One-Cycle Transfers 8-30
Paged, Non-Burst Transfer 8-30
Paged, Burst Transfer 8-30
Non-Paged, Non-Burst Transfer 8-31
Non-Paged, Burst, Compatibility Mode Transfer 8-31
Non-Paged, Burst, Extended Mode Transfer 8-32
Two-Cycle Transfers 8-33
Memory-to-Memory Transfers 8-34
Memory-to-I/O Transfers 8-34
I/O-to-Memory Transfers 8-35
I/O-to-I/O Transfers 8-35
82C59A Programmable Interrupt Controller (PIC) 8-36
Interrupt Configuration 8-37
Interrupt Jumper Matrix 8-37
Default PIC Configuration 8-40
Default NMI Assignments 8-40
Interrupt Destinations 8-41
Master PIC 8-41
Slave PIC 8-42
Non-Maskable Interrupt Request (NMIR) 8-42
Spare Gates 8-43
Multibus Interrupt Destinations 8-43
Interrupt Sources 8-44
82C54A PIT Counter 0, Counter 1 Interrupts 8-45
Multibus Interrupts 8-45
Multibus Time-Out Interrupt 8-47
MB87033B SPC Interrupt (SCZINT) 8-48
External Interrupt (EXTINT) 8-48
Serial Interrupt 8-48
DMA Interrupt 8-48
SBX Interrupts 8-48
Multibus Request Error Interrupt 8-49

Chapter 8. CPU Subsystem Configuration (continued)

Line Printer Interrupt 8-49
Break Detect Interrupt 8-50
Parity Error Interrupt 8-50
Spare Inverter 8-50
Edge-Triggered Latch (RSINT) 8-51
OR-Gate 8-51
Power-Fail Interrupt (PFINT) 8-51
Multibus Interrupt Drivers 8-52
Bus Drive (BUSDRIVE1*, BUSDRIVE2*) 8-52
Parity Error (PER*) 8-52
Interrupt Programming 8-53
82C59A PIC Polled Mode 8-53
Direct-Vectored Mode 8-53
Light-Emitting Diodes (LEDs) 8-55
Questions and Answers 8-57

Chapter 9. Connectors and Cables

Introduction 9-2
User-Supplied Connectors 9-5
User-Supplied Cables 9-6
Multibus Interface (Connector P1) 9-7
Auxiliary Interface (Connector P2) 9-12
Memory Interface (Connector P3) 9-14
Parallel Interface (Connector J1) 9-15
SCSI Interface (Connector J2) 9-18
Front Panel Interface (Connector J3) 9-21
Serial Interfaces (Connectors J4L and J4U) 9-22
SBX Interfaces (Connectors J5 and J6) 9-29

Chapter 10. Service Information

Services Offered 10-2
System Engineering Consulting Services 10-2
Customer Training 10-2
Hardware On-Site Support 10-2
Mail-In Service 10-2
How to Get Service 10-3
Call This Number for Service 10-3
Package Your Board for Shipment 10-4
Send Your Board to This Address 10-4

Appendix A. Configuration Worksheets

About the Worksheets A-1
Installation Checklist A-2
Real Mode Memory Worksheet A-3
Real Mode Memory Map Worksheet A-4
Protected Mode Memory Map Worksheet A-5
Protected Mode Memory Worksheet A-6
System Memory Worksheet A-7
System Memory Map Worksheet A-8
I/O Configuration Worksheet A-9
CPU Configuration Worksheet A-12
Jumper Configuration Worksheet A-13
Programming Checklist A-14
Hexadecimal Conversion Table A-15

Appendix B. Jumper Information

Default Jumper Settings B-1
Stake Pin Matrix With Stake Pin Names B-18
Default Wire List B-22
Default Components B-24

Appendix C. Multidrop Information

Introduction C-1
Multidrop Configuration Sequence C-2
Multidrop Bias Resistors RP5 and RP6 ..... C-3
Case 1: Lines Floating C-5
Case 2: Lines Driven C-7
Multidrop Termination Resistor RP4 ..... C-9

Appendix D. Compatibility Notes

Compatibility Notes D-1

Appendix E. iRMX® Documentation Roadmap

Glossary

Index

Figures

2-1. iSBC 486/12S Block Diagram 2-3
2-2. CPU and Memory Block Diagram 2-4
3-1. Memory Module Installation Exploded View 3-6
3-2. EPROM Installation 3-17
3-3. SCSI Termination Resistors and Fuse 3-18
3-4. Multibus Termination Resistor Pack 3-19
3-5. Orientation of RP6 for RS-422A Operation 3-20
3-6. 3487 RS-422A Line Driver and Termination Resistor Sockets 3-21
3-7. Single and Double-Wide SBX Modules 3-22
3-8. Single-Wide SBX Module Installation 3-24
3-9. Double-Wide SBX Module Installation 3-26
4-1. Default Real Mode Memory Map 4-3
5-1. Default Protected Mode Memory Map 5-4
5-2. Multibus Address Space in the 4 Gbyte Address Space 5-12
6-1. Default System Memory Map 6-4
6-2. Using the Dual-Port 16 Mbyte Page Register 6-6
6-3. Using the Dual-Port Starting and Ending Addresses 6-8
6-4. Using the Dual-Port Alias 6-9
6-5. Dual-Port Alias Wrap-Around at the top of the 16 Mbyte Page 6-10
6-6. Aliasing Example 6-18
6-7. Real Mode Configuration Example 1 6-23
6-8. Real Mode Configuration Example 2 6-24
6-9. Protected Mode Multiprocessing Example 6-26
6-10. Example of Dual Port Alias in Protected Mode Multiprocessing Example 6-27
7-1. Default I/O Address Map 7-8
7-2. 82C54 PIT Default Configuration 7-12
7-3. 82C54 Programming Example 7-15
7-4. 82C55A Parallel Peripheral Interface Logic Diagram 7-18
7-5. Line Printer Interface (Port A) 7-20
7-6. Using PC4 to Control Port A Transceiver Direction 7-22
7-7. User-Defined Code 7-23
7-8. LPT Interrupt Latch 7-24
7-9. Power-Fail Sense PFSN* Signal 7-25
7-10. Dual-Port Lock 7-27
7-11. NMI Mask Logic 7-28
7-12. Multibus Time-Out Clear 7-29
7-13. RSINT User-Defined Interrupt 7-30
7-14. Multibus Interrupt Drivers (BUSDRIVE1*, BUSDRIVE2*) 7-31
7-15. LPT Interrupt Latch Clear 7-32
7-16. 82C55A Control Words 7-35
7-17. 82C55A Programming Example 7-37
7-18. Port C Bit Set/Reset Example 7-39
7-19. Centronics-Compatible Printer Timing 7-40
7-20. General Timing Parameters for Centronics-Compatible Printer 7-41
7-21. SCSI Interface Block Diagram 7-45

Figures (continued)

7-22. SCSI Connector J2 Pin-Out 7-47
7-23. SCSI Termination Resistor Packs RP1, RP2, RP3 7-49
7-24. Channel A Configured for DTE Operation and Channel B Configured for DCE Operation 7-55
7-25. Channel A and B Configured for DCE Operation 7-56
7-26. RS-422A Multidrop Operation 7-57
7-27. Channel A RS-232C DCE 7-58
7-28. Channel B RS-232C DCE 7-59
7-29. Channel A RS-232C DTE 7-60
7-30. Channel A RS-422A DCE 7-61
7-31. 8274 MPSC Programming Example 7-70
7-32. SBX DMA Configuration 7-78
7-33. INIT* and HWRESET* Signal 7-82
7-34. Location of Multibus Termination Resistor Pack RP7 7-83
7-35. Multibus Serial and Parallel Arbitration 7-86
8-1. Using BUSREQE as an Interrupt Source 8-10
8-2. ADMA Configuration 8-13
8-3. Overview of ADMA and DAG 8-18
8-4. DAG Registers 8-19
8-5. 82C59A Programmable Interrupt Controllers Overview 8-36
8-6. 82C59A PIC Interrupt Configuration 8-38
8-7. NMI Interrupt Logic and Spare Gates 8-39
8-8. Multibus Interrupt Stake Pins 8-46
8-9. Multibus Time-Out Interrupt 8-47
8-10. Latching the BUSREQE Signal 8-49
8-11. Line Printer Interrupt 8-50
8-12. BUSDRIVE1* and BUSDRIVE2* Signals 8-52
8-13. 82C59A PIC Initialization Programming Example 8-54
8-14. Light-Emitting Diodes (LEDs) 8-55
9-1. Location of Connectors 9-3
9-2. Connectors and Cables 9-4
9-3. Pin Numbering on Connectors P1 and P2. 9-11
9-4. Cable From J1 to Back Chassis Panel 9-16
9-5. Printer Cable Construction 9-17
9-6. SCSI Cable 9-19
9-7. SCSI Cable Construction 9-20
9-8. Connector J3 Pin Locations 9-21
9-9. Connector J4 Pin Locations 9-22
9-10. Connector J4U and J4L RS-232C Cable to Back Panel 9-25
9-11. RS-232C Cable Construction 9-26
9-12. Connector J4L RS-442A/449 Cable to Back Chassis Panel 9-27
9-13. RS-442A/449 Cable Construction 9-28
9-14. SBX Pin Numbering 9-33

Figures (continued)

B-1. Default Jumpers (iSBC 486/12 Series with 2 Mbytes of DRAM) B-2
B-2. Default Jumpers (iSBC 486/12 Series with 4 Mbytes of DRAM) B-3
B-3. Default Jumpers (iSBC 486/12 Series with 8 Mbytes of DRAM) B-4
B-4. Default Jumpers (iSBC 486/12 Series with 16 or 32 Mbytes of DRAM) B-5
C-1. Full-Duplex RS-422A/449 Multidrop Configuration Example C-2
C-2. Bias and Terminator Resistor Placement Example C-4
C-3. Case 1 Configuration Example (Lines Floating) C-6
C-4. Case 2 Configuration Example (Lines Driven) C-8

Tables

1-1. iSBC 486/12 Series Performance Specifications 1-3
1-2. General Specifications 1-4
1-3. Related Publications 1-13
3-1. Resistor Pack and Fuse Data 3-2
3-2. Parts and Tools for Memory Module Installation 3-7
3-3. Procedure A Combinations 3-8
3-4. Installation Procedure B Combinations 3-12
3-5. Parts and Tools for Single-Wide SBX Module Installation 3-23
3-6. Parts and Tools for Double-Wide SBX Module Installation 3-25
4-1. Multibus Window Starting Address Jumper Configuration 4-6
4-2. Multibus Window Ending Address Jumper Configuration 4-7
4-3. DRAM Ending Address Jumper Configuration 4-9
4-4. Supported EPROM Sizes 4-10
4-5. EPROM Jumper Configuration 4-12
4-6. Size of Multibus Interface Address Space (Kbytes) 4-14
5-1. DRAM Ending Address, 1 Mbyte Page 5-7
5-2. DRAM Ending Address, 64 Kbyte Page 5-8
5-3. EPROM Starting Address, PVAM 5-9
5-4. EPROM Jumper Configuration, Protected Mode 5-10
5-5. iSBC 486/12 Board to Multibus Address Mapping 5-11
6-1. Dual-Port Starting Address (Mbyte Boundary) 6-12
6-2. Dual-Port Starting Address (64 Kbyte Boundary) 6-13
6-3. Dual-Port Ending Address (Mbyte Boundary) 6-14
6-4. Dual-Port Ending Address (64 Kbyte Boundary) 6-15
6-5. Dual-Port Alias Example 6-16
6-6. Dual-Port Alias Jumpers 6-19
6-7. Dual-Port 16 Mbyte Page Register 6-19
6-8. Starting and Ending Addresses of Dual-Port Pages 6-20
6-9. Real Mode Multiprocessing Example #1 6-22
6-10. Protected Mode Multiprocessing Example 6-25
7-1. I/O Port Addresses 7-6
7-2. Special I/O Addresses 7-9
7-3. 82C54 Configuration Options 7-11
7-4. 82C54 PIT Jumpers 7-12

Tables (continued)

7-5. 82C54 PIT I/O Port Addresses ... 7-14
7-6. Baud Rate Generation ... 7-16
7-7. 82C54 Count Values for Real Time Clock ... 7-17
7-8. Parallel Port Bit Assignments for the 82C55A PPI ... 7-19
7-9. User-Defined Code Jumpers ... 7-24
7-10. Jumper Configuration for Control of Port A Transceiver ... 7-26
7-11. 82C55A PPI Stake Pins ... 7-33
7-12. 82C55A I/O Addresses ... 7-34
7-13. Port C Bit Set/Reset Control Word ... 7-38
7-14. Centronics-Compatible Signals ... 7-39
7-15. Timing Parameters for Centronics-Compatible Printers ... 7-41
7-16. SCSI Signals ... 7-46
7-17. SCSI ID Jumpers ... 7-48
7-18. SCZID General Purpose Bit GP3 Jumpers ... 7-48
7-19. SPC I/O Register Addresses ... 7-50
7-20. SCSI Control Register Addresses ... 7-52
7-21. SCSI ID Register Bits<3:0> ... 7-52
7-22. Blast Register Functions ... 7-53
7-23. Serial Channel Configurations ... 7-57
7-24. Serial Interface Connectors (J4) ... 7-62
7-25. 8274 MPSC I/O Addresses ... 7-62
7-26. Default Serial Jumpers ... 7-63
7-27. Default Serial Components ... 7-64
7-28. Channel A RS-232C DCE Configuration Options (Connector J4L) ... 7-64
7-29. Channel A RS-232C DTE Jumpers (Connector J4L) ... 7-65
7-30. RS-232C DTE Configuration Options (Connector J4L) ... 7-65
7-31. Configuration of RS-232C DTE Serial Components (Connector J4L) ... 7-66
7-32. Channel A RS-422A DCE Jumpers (Connector J4L) ... 7-66
7-33. RS-422A Configuration Options ... 7-67
7-34. Configuration of RS-422A Serial Components ... 7-67
7-35. RS-422A Multidrop Configuration Jumpers ... 7-68
7-36. Default SBX I/O Addresses (Connector J5) ... 7-72
7-37. Default SBX I/O Addresses (Connector J6) ... 7-73
7-38. SBX Address Swap Jumpers ... 7-74
7-39. SBX I/O Addresses Swap (Connector J6) ... 7-75
7-40. SBX I/O Address Swap (Connector J5) ... 7-76
7-41. SBX Option Lines Stake Pins ... 7-79
7-42. SBX Interrupt Stake Pins ... 7-79
7-43. BCLK* Signal Jumpers ... 7-80
7-44. CCLK* Signal Jumpers ... 7-81
7-45. INIT* Signal Jumpers ... 7-82
7-46. Software Reset Jumpers ... 7-83
7-47. Multibus Address Line Termination Resistors ... 7-84
7-48. Multibus Interrupt Stake Pins ... 7-84
7-49. Multibus Buffered I/O Write Jumper ... 7-85
7-50. Multibus Arbitration Priority Jumpers ... 7-87
7-51. Multibus Arbitration Mode Jumpers ... 7-87

Tables (continued)

8-1. Board ID Register 8-6
8-2. iSBC 486/12 Board Reset Actions 8-8
8-3. Multibus Time-Out Response Jumper 8-9
8-4. Default ADMA Jumpers 8-14
8-5. ADMA Jumper Options 8-14
8-6. ADMA Internal Register I/O Addresses 8-15
8-7. ADMA General Mode Register Initialization 8-16
8-8. Use of DAG Registers 8-20
8-9. I/O Addresses of DAG Registers 8-21
8-10. Significant Bits of One-Cycle Page Register 8-24
8-11. Default Memory Address Translation 8-25
8-12. Maximum DMA Transfer Rate (One-Cycle Transfers Between DRAM and SCSI FIFOs) 8-26
8-13. Maximum DMA Transfer Rate (Two-Cycle Transfers) 8-26
8-14. DMA Transfer Specifications (One-Cycle Transfers) 8-27
8-15. DMA Transfer Specifications (Two-Cycle Transfers) 8-28
8-16. Supported One-Cycle Transfer Modes 8-28
8-17. Default Interrupt Assignments 8-40
8-18. Interrupt Destinations 8-41
8-19. NMI Mask 8-42
8-20. Multibus Interrupt Destinations 8-43
8-21. Interrupt Sources 8-44
8-22. Multibus Interrupt Drivers 8-52
8-23. LED Functions 8-56
9-1. Connectors 9-2
9-2. User-Supplied Connectors 9-5
9-3. User-Supplied Cable Parts 9-6
9-4. Multibus Interface Signals (Connector P1) 9-7
9-5. Connector P1 Pin Assignments 9-10
9-6. Auxiliary Interface Signals (Connectors P2 and J3) 9-13
9-7. Connector P2 Pin Assignments 9-14
9-8. Connector J1 Pin Assignments 9-15
9-9. Connector J2 Pin Assignments 9-18
9-10. Connector J3 Pin Assignments 9-21
9-11. Serial Connector J4 Cross-Reference 9-22
9-12. Connectors J4L and J4U RS-232C Pin Assignments 9-23
9-13. Connector J4L RS-442A/449 Pin Assignments 9-24
9-14. SBX Interface Signals (Connectors J5 and J6) 9-30
9-15. Connector J5 and J6 Pin Assignments 9-32
B-1. Stake Pin and Jumper Listing (Numerical Order) B-6
B-2. Stake Pin and Jumper Listing (Alphabetical Order) B-12
B-3. Additional Jumpers due to Memory Module Type B-21
B-4. Default Wire List B-22
B-5. Default Components B-24
D-1. Compatibility Notes D-1

***

Introduction to the iSBC® 486/12 Series Products 1

Chapter Contents

Introduction 1-2
System Considerations 1-3
Specifications 1-3
About This Manual 1-7
What's in This Manual 1-7
Notational Conventions 1-10
Active-Low Signal Name Conventions 1-10
Component Identifiers 1-10
Hexadecimal Numbers 1-10
Bit and Byte Order 1-11
Ranges of Bits or Signal Lines 1-11
Undefined Bits and Software Compatibility 1-11
Special Symbols 1-12
Related Publications 1-13
How To Report Errors in This Manual 1-14
About Intel 1-15
Intel Real-Time Operating Systems 1-15
Intel Customer Services 1-15
Intel Multibus Products 1-16

Introduction

The Intel iSBC® 486/12 Series Single Board Computers are Intel Corporation's highest performance Multibus I CPU boards. The iSBC 486/12 family includes models with the Intel486™ DX2 or Intel486, with or without SCSI/ADMA, and with various amounts of on-board DRAM.

Features of the iSBC 486/12 family include the following:

- 66 MHz Intel486 DX2 or 33 MHz Intel486 microprocessor
- single-ended SCSI interface (iSBC 486/12S series only)
- Advanced DMA (ADMA) controller (iSBC 486/12S series only)
- 4-, 8-, 16-, or 32-Mbyte DRAM module shipped with baseboard
- a second DRAM module can be installed by the customer
- two serial ports
- one parallel port
- two SBX interfaces
- compatibility with iSBC 286/12, iSBC 386/12, and iSBC 386/12S Single Board Computers
- up to 1 Mbyte of EPROM
- all on-board DRAM can be configured as dual-port DRAM
- one 82C54 Programmable Interval Timer (three timers/counters)
- 16 interrupt levels controlled by two 82C59A Programmable Interrupt Controllers

The iSBC 486/12 products deliver the power of the Intel486 DX2 or Intel486 microprocessor. These boards offer maximum performance while being compatible with other 286 and Intel386™ CPU Multibus boards.

The iSBC 486/12 boards have three options that are selected when ordering the board:

- Intel486 DX2 microprocessor operating internally at 66 MHz or Intel486 microprocessor operating at 33 MHz
- size of on-board DRAM module
- SCSI option (includes SCSI and ADMA)

This manual presents information for all the configuration options of the iSBC 486/12 products. References to the iSBC 486/12 include the 486/12 DX2, 486/12S, and 486/12S DX2, unless otherwise noted. All references to the 486/12S include the 486/12S DX2, unless otherwise noted.

All references to the Intel486 microprocessor include the Intel486 DX2, unless otherwise noted.

This manual was written for engineers and technicians who need to integrate the iSBC 486/12 Single Board Computer into a Multibus I system. This manual includes information on board operation, configuration, I/O interfaces and connectors, and device programming.

System Considerations

NOTE

The faster clock rate of the Intel486 DX2 CPU will speed up software timing loops. This might be a consideration in porting software from a board with a 33 MHz CPU to one with a 66 MHz CPU.

Specifications

Table 1-1. iSBC® 486/12 Series Performance Specifications

DRAM	Write, DRAM page miss	0 wait states
	Write, DRAM page hit	0 wait states
	Read, DRAM page miss	7 wait states
	Read, DRAM page hit	5 wait states
	Burst Read[1], DRAM page miss	10 wait states
	Burst Read[1], DRAM page hit	8 wait states
SCSI	Synchronous	5 Mbytes/s
	Asynchronous	3 Mbytes/s

Note:
[1] Burst read of 16 bytes

Table 1-2. General Specifications

IEEE 796 compliance level (Multibus)	Master D16/M24/I16/V0 EL	
	This translates to:	
	Master	a system master
	D16	8- or 16-bit data transfers
	M24	24-bit memory addressing
	I16	8-, 16-bit I/O transfers
	V0	non-vectored interrupts
	EL	level or edge-triggered interrupts

Table 1-2. General Specifications (Continued)

IEEE 959 compliance level (SBX)	D16/D16/DMA This translates to: D16 16-bit baseboard addresses D16 8- or 16-bit expansion modules DMA DMA transfers (iSBC 486/12S series only)		
Clock speeds	Clock:	Frequency (MHz):	Comments:
	CPU	33	Intel486 DX2 runs internally at 66 MHz
	82258 ADMA	20	10 MHz ADMA
	MB87033B SPC	10	--
	MCLK	9.83	SBX interface
	BCLK	9.83	Multibus interface
	CCLK	9.83	Multibus interface
	SERCLK	4.00	used by 82C54 PIT
	TCLK	1.228	used by 82C54 PIT
LEDs	Designator:	Color:	Description:
	DS1	green	run indicator
	DS2	red	user defined
	DS3	yellow	Multibus time-out
	DS4	red	user defined
	DS5	red	DRAM parity error
	DS6	yellow	SCSI activity
DRAM	• uses one or two MM30x memory modules • one 4-, 8-, 16-, or 32-Mbyte module shipped with baseboard • optional 8, 16, or 32 Mbyte DRAM module can be installed • all DRAM can be configured as dual-port memory • maximum of 64 Mbytes of DRAM		
EPROM	• two 28- or 32-pin JEDEC EPROM sockets • 32 Kbytes to 1 Mbyte of EPROM • 250 ns access time or faster required		
Operand size	Intel486 CPU: 8-, 16-, 24-, or 32-bits 82258 ADMA: 8- or 16-bits		

Table 1-2. General Specifications (Continued)

<table>
<tr><td>Serial ports</td><td colspan="3">• 2 serial ports controlled by 8274 MPSC
• synchronous or asynchronous operation

Synchronous:

600 to 615,000 baud
5- to 8-bit characters
even or odd parity
1, 1½, or 2 stop bits
HDLC/SDLC sync
automatic sync insertion

Asynchronous:

75 to 19,200 baud
5- to 8-bit characters
even or odd parity
1, 1½, or 2 stop bits
false start bit detection

• external break detection circuit and interrupt
• RS-232C or RS-422A/449 operation

Channel A:

RS-232C DCE
RS-232C DTE
RS-422A/449 DCE
multidrop operation

Channel B:

RS-232C DCE</td></tr>
<tr><td>Parallel port</td><td colspan="3">• configurable as a Centronics-compatible printer port or a general purpose parallel I/O port
• controlled by 82C55A PPI</td></tr>
<tr><td>SCSI port</td><td colspan="3">• controlled by Fujitsu MB87033B SPC
• single-ended interface
• maximum cable length for single-ended interface is 6 m
• synchronous or asynchronous interface

Synchronous:
8-bit data
5.0 Mbyte/s bandwidth

Asynchronous:
8-bit data
3.0 Mbyte/s bandwidth</td></tr>
<tr><td>Environmental</td><td>temperature

relative humidity
air flow</td><td>0 to 55 °C (32 to 130 °F)
-40 to 70 °C
(-40 to 158 °F)
90%, non-condensing
200 LFM, minimum</td><td>operating
storage

operating
operating</td></tr>
<tr><td>Physical</td><td colspan="3">width 30.48 cm (12.0 in.)
length 18.00 cm (7.05 in.)
height 2.18 cm (0.86 in.) with one memory module
4.11 cm (1.62 in.) with two memory modules</td></tr>
</table>

Table 1-2. General Specifications (Continued)

Maximum power requirements (Does not include SBX modules or EPROM power requirements.)	iSBC 486/12 baseboard	+5.25 V ±12 V	11.5 A 50 mA	(60.4 W) (0.6 W)
	iSBC 486/12 DX2 baseboard	+5.25 V ±12 V	11.73 A 50 mA	(61.6 W) (0.6 W)
	iSBC 486/12S baseboard	+5.25 V ±12 V	13.3 A 50 mA	(69.8 W) (0.6 W)
	iSBC 486/12S DX2 baseboard	+5.25 V ±12 V	13.53 A 50 mA	(71 W) (0.6 W)
	MM302 memory module	+5.25 V	2.8 A	(14.8 W)
	MM304 memory module	+5.25 V	2.8 A	(14.9 W)
	MM308 memory module	+5.25 V	3.2 A	(17.0 W)

About This Manual

What's in This Manual

The manual is organized into the following groups:

- Introduction
- Operation and Service Information
- Configuration Information
- Installation, Cable Information, and Jumper Reference Table
- Reference Information

Introduction

Chapter 1 **Introduction to the iSBC 486/12 Series** contains an overview of the features of the iSBC 486/12 and discusses the contents of this manual.

Operation and Service Information

Chapter 2 **Board Operation** provides a brief overview of each major component on the board.

Chapter 10 **Service Information** tells how to contact Intel for service.

Configuration Information

Chapter 4 **Real Mode Memory Configuration** contains information on how to configure the memory for real mode operation.

Chapter 5 **Protected Mode Memory Configuration** contains information on how to configure your memory for protected mode operation.

Chapter 6 **System Memory Configuration** contains information on how to configure your dual port memory for access by other Multibus agents.

Chapter 7 **I/O Subsystem Configuration** contains information on how to configure your serial, parallel, SBX, and Multibus interfaces.

Chapter 8 **CPU Subsystem Configuration** contains information on how to configure the CPU, interrupts, and DMA components.

Appendix A **iSBC® 486/12 Configuration Worksheets** This appendix contains blank configuration worksheets for your use.

Installation, Cable Information, and Jumper Reference Table

Chapter 3 **Installation** contains information on how to install the user-provided components on the board and how to install the board in your system.

Chapter 9 **Cables and Connector Information** contains information about the connectors and cables used with the iSBC 486/12 series boards.

Appendix B **Jumper Information** contains a list of all the jumpers on the iSBC 486/12 series boards.

Reference Information

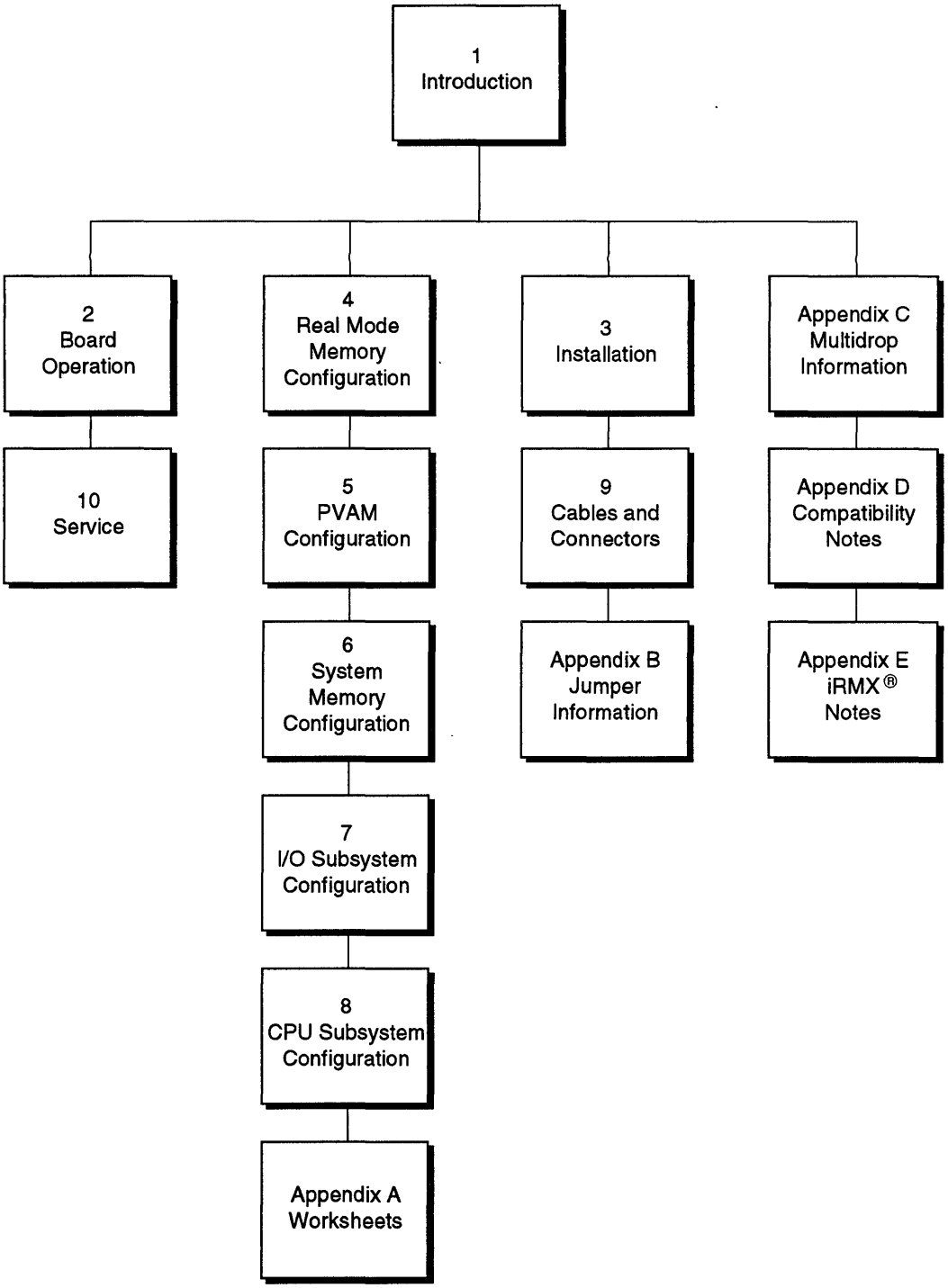
Appendix C **Multidrop Information** contains information to help you configure your RS-422A/449 interface for multidrop operation.

Appendix D **Compatibility Notes** contains a short list of items that are different between the iSBC 368/12 and the iSBC 486/12.

Appendix E **iRMX® Documentation Roadmap** directs users of Intel's iRMX operating systems to information on the iSBC 486/12 products.

Glossary The Glossary defines terms and abbreviations that are used in this manual. You can also find a mini-glossary at the beginning of each chapter.

The following "roadmap" shows the organization of this manual.



Notational Conventions

This manual uses special notational conventions for signal names, default jumpers, hexadecimal numbers, byte numbering, and symbols.

Active-Low Signal Name Conventions

NAME*
NAME#

An asterisk (or pound sign) is used after a signal mnemonic to identify an active-low signal: DACK* (or DACK#) indicates an active-low signal. This manual uses the asterisk notation; the schematics use the pound sign notation.

Component Identifiers

Example	Meaning
U43	The "U" identifies an integrated circuit.
RP4	The "RP" prefix identifies a resistor pack.
E999	The "E" prefix identifies a stake pin.
E999-E000§	A hyphen between two stake pins indicates a jumper block is installed between the two stake pins. A section mark "§" is used after a jumper pair to indicate the factory-installed default jumper.
J4U-5	The "J" prefix identifies a female connector. The "P" prefix identifies a male connector. When two connectors are stacked, "U" identifies the upper connector and "L" identifies the lower connector. Pin numbers are appended after the connector identifier. For example, J4U-5 refers to connector J4 upper, pin 5.

Hexadecimal Numbers

When hexadecimal (base 16) numbers are used, they are indicated by an "H" suffix. A leading zero is added if the number would otherwise begin with one of the hexadecimal digits A through F. For example: 0H, 1H,..., 0AH, 0BH,..., 0FH. Hexadecimal numbers in tables and figures may be padded with leading zeros to align the numbers in a column.

The number of hexadecimal digits does not indicate the size of the operand. For example, 0FFH could be a byte, word, or double word operand. Likewise for addresses, 0H can represent a 20-, 24-, or 32-bit address.

Hexadecimal numbers are shown with a space between every four digits to make it easier to recognize the number. For example, 0FFFFFFFFH is written as 0 FFFF FFFFH. Do not include the spaces when using hexadecimal numbers in program statements.

Bit and Byte Order

The illustrations in this manual follow the same conventions as in the *Intel486™ Microprocessor Family Programmer's Reference Manual* for bit positions and byte order. In illustrations of address spaces, smaller addresses appear toward the bottom of the figure; addresses increase toward the top. Bit positions are numbered from right to left, starting with bit 0 as the rightmost bit. For example, A23 is the left-most bit of a 24-bit address; A0 is the right-most bit.

Ranges of Bits or Signal Lines

In this manual, ranges of bits are indicated using two dots: bits 7..0 indicates bits 7 through bit 0 (the leftmost bit is listed first). A group of signal lines is indicated using angle brackets and a colon: A<23:2> indicates signal lines A23 through A2.

In the schematics, a range of signal lines is indicated using square brackets and a colon: A[23:2] indicates signal lines A23 through A2.

Undefined Bits and Software Compatibility

Some I/O registers and memory regions are marked as *reserved*. For software compatibility with future versions of the iSBC 486/12 family, it is essential that your software avoid using all reserved bits, registers, or address spaces.

Special Symbols

This manual includes several special symbols to help you find important information.

*** The three asterisks are the end-of-chapter mark.

CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY!

Electrostatic Discharge Damage. This symbol is placed next to important information that will help you avoid damage to the board by static electricity.

NOTE

A NOTE contains information that you should read before continuing.

CAUTION

A CAUTION contains information that will help you avoid software failures or damage to your hardware.

Related Publications

Table 1-3. Related Publications

Subject	Publication	Order No.
Intel literature	*Customer Literature Price List (US and Canada),* Intel	210620
	International Literature Guide, Intel	E00029
Intel486 microprocessor	*Intel486™ DX Microprocessor Data Book,* Intel	240440
	Intel486™ Microprocessor Family Programmer's Reference Manual, Intel	240486
82258 ADMA	*82258 ADMA User's Guide*$, Intel	231297
EPROMs	*Programmable Logic* Handbook$, Intel	296083
8255A PPI	*Peripheral Components* Handbook$, Intel	296467
82C54 PIT	*Peripheral Components* Handbook$, Intel	296467
82C59A PIC	*Peripheral Components* Handbook$, Intel	296467
8274 MPSC	*Microcommunications* Handbook$, Intel	231658
Intel486 support tools	*Intel386™/486™ Family Development Support* fact sheet, Intel	280808
Intel Multibus boards and software	*Microcomputer Products* Handbook, Intel	280407
MB87033B programming	*Fast Track to SCSI*$, Fujitsu	contact Fujitsu
SCSI standard	*SCSI Specification*$, American National Standards Institute	ANSI X3.131-1986
Serial interface standards	*EIA Standard RS-232-C*$, Electronic Industries Association	EIA RS-232-C
	EIA Standard RS-422-A$, Electronic Industries Association	EIA RS-422-A
	EIA Standard RS-449$, Electronic Industries Association	EIA RS-449
Floating point standard	*IEEE Standard for Binary Floating Point Arithmetic*$, IEEE	IEEE 754-1985
Multibus standards	IEEE 796 Multibus Specification$, IEEE	IEEE 796-1988
	IEEE 959 SBX Bus Specification$, IEEE	IEEE 959-1988

$ Contact the listed publisher for current prices.

You can order the related publications from the following publishers:

Intel	Intel Literature Sales P.O. Box 7641 Mt. Prospect, IL 60056-7641 Phone toll-free 1-800-548-4725 (U.S. and Canada only)
IEEE	The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street New York, NY 10017 Phone (212) 705-7900
ANSI	American National Standards Institute, Inc. 1430 Broadway New York, NY 10018 Phone (212) 642-4900
Fujitsu	Fujitsu Microelectronics, Inc. Integrated Circuits Division 3545 North First Street San Jose, CA 95134 Phone (408) 922-9000

How To Report Errors in This Manual

Please use the reader comment card included in the back of this manual to report problems with this manual.

About Intel

Intel Real-Time Operating Systems

Intel offers a complete range of real-time operating systems and development tools for its Multibus Single Board Computers. The Intel family of real-time operating systems includes the following:

- iRMX I (a 16-bit real-time operating system that runs in real mode)
- iRMX II (a 16-bit real-time operating system that runs in protected mode)
- iRMX III (a full 32-bit real-time operating system that runs in protected mode)
- iRMK (a multi-tasking 32-bit kernel)

Appendix E provides a "roadmap" to iRMX documentation to help you find the iRMX documentation that is related to the iSBC 486/12 products.

Intel Customer Services

Intel offers world-wide customer services, including the following:

- hands-on training workshops in Intel systems hardware, operating systems, languages, tools, and networking products
- consulting by Intel Systems Engineering Services for both hardware and software design solutions
- over 100 Intel service and training centers around the world
- custom system integration
- networking design, installation, and maintenance
- hardware maintenance programs with prompt, world-wide response
- software support that includes updates, telephone hotline support, troubleshooting guides, and the *;COMMENTS* newsletter

For complete information on Intel Customer Services, contact the nearest Intel Sales Office listed in the back of this manual.

Intel Multibus Products

Intel offers a complete line of Multibus products, including:

- Multibus development systems
- a wide variety of CPU boards
- high-performance memory expansion modules
- SBX I/O expansion modules
- peripheral controllers
- LAN and serial communication boards
- system chassis
- digital and analog I/O boards
- iRMX real-time operating systems
- iRMK real-time kernel

For more information, refer to the *Microcomputer Products* Handbook (Intel order number 280407), or contact your local Intel Sales Office.

***

Board Operation 2

Chapter Contents

Introduction 2-2
CPU and Memory 2-4
ADMA, DAG, and SPC 2-5
I/O Subsystem 2-5
Multibus/Dual-Port Memory 2-6

Introduction

Figure 2-1 shows a block diagram for the board.

The board is divided into the following functional areas:

- CPU and Memory
- ADMA, DAG, and SCSI
- I/O
- Multibus and Dual-port

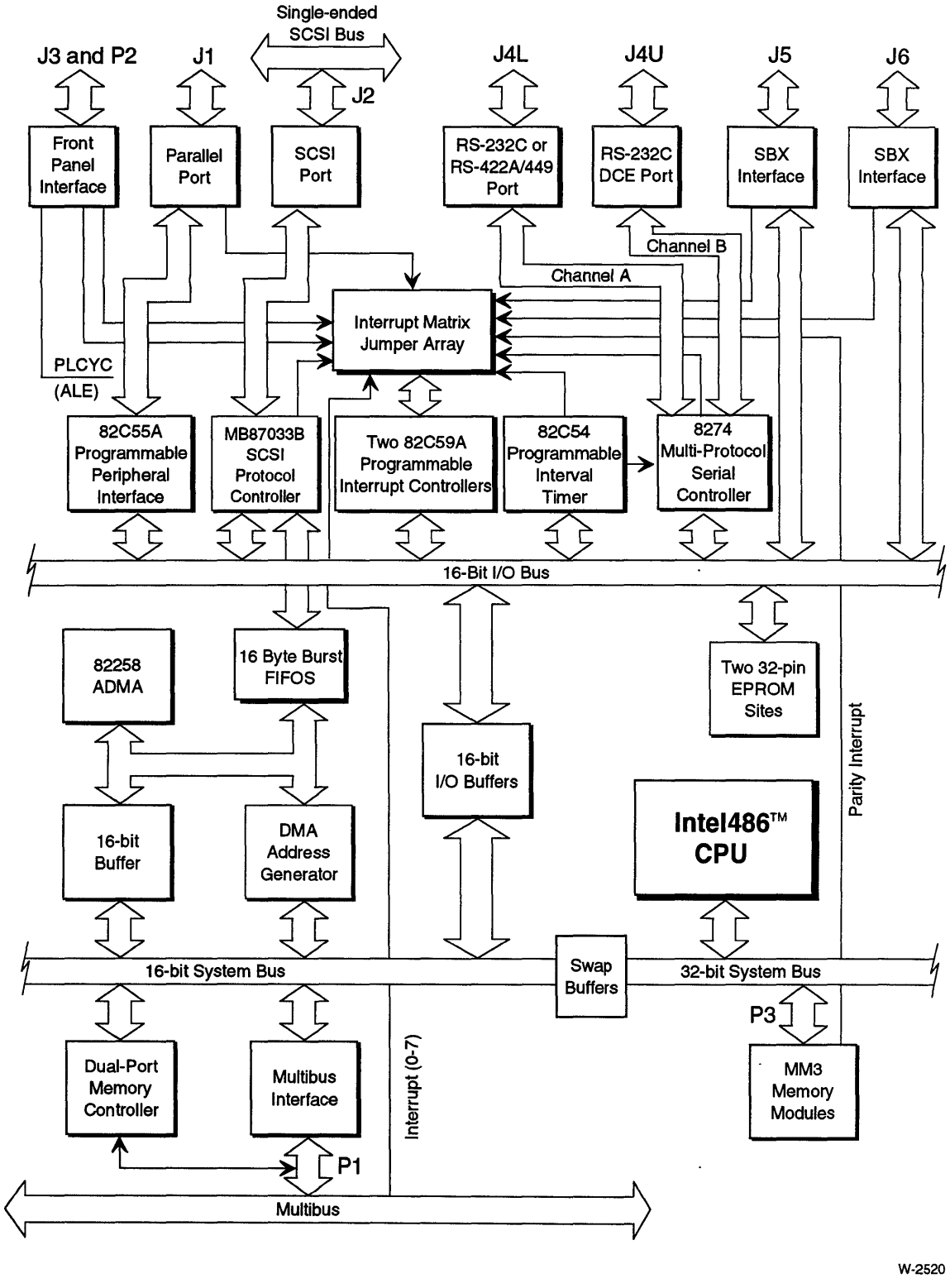
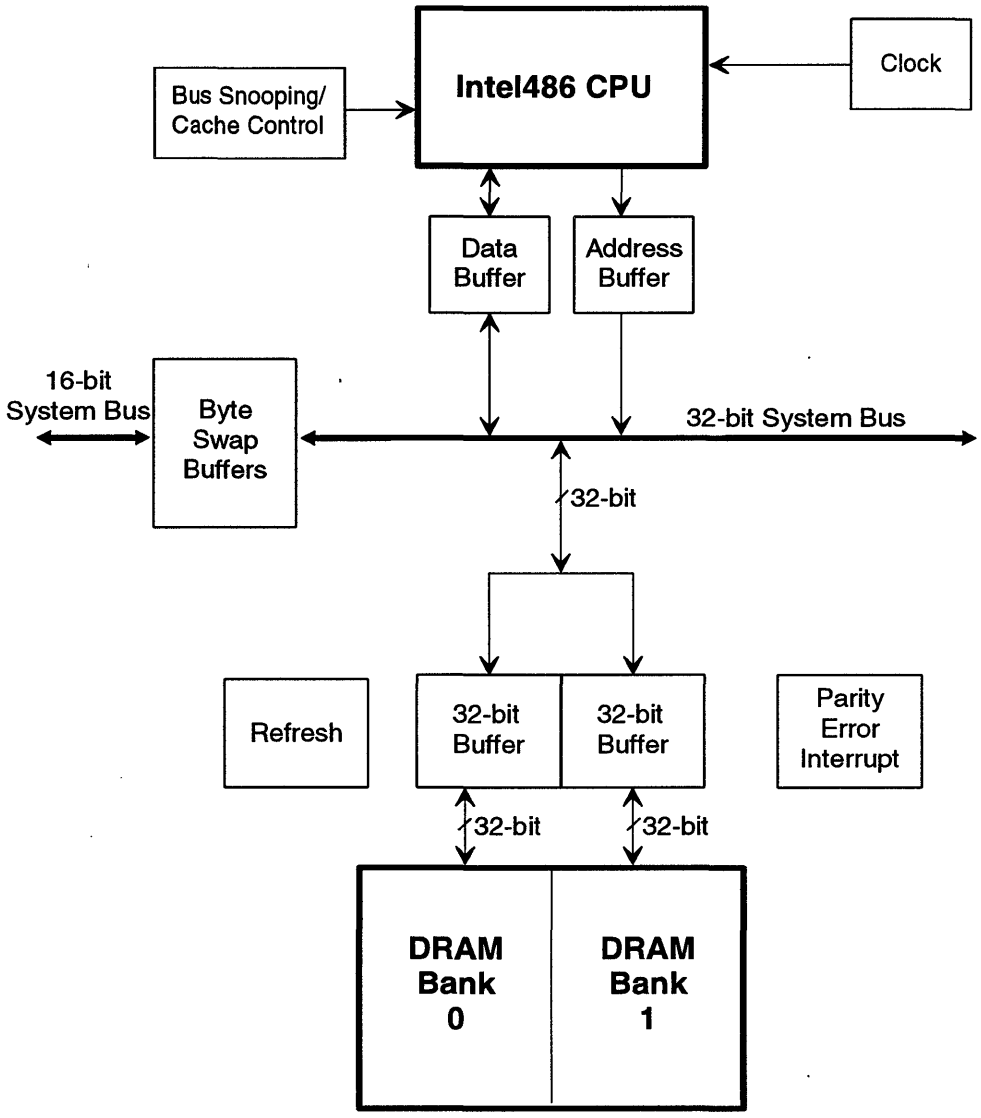


Figure 2-1. iSBC® 486/12S Block Diagram

CPU and Memory



Intel486 is a trademark of Intel Corporation.

W-2537

Figure 2-2. CPU and Memory Block Diagram

The Intel486 microprocessor and DRAM reside on the 32-bit system bus. This bus provides a high-speed interface between the CPU and memory. The 32-bit system bus is connected to the 16-bit system bus and the 16-bit I/O bus by buffers. The ADMA, CPU, and dual-port memory controller must arbitrate for this internal bus. Only one bus master can own the internal bus at any one time.

The address decode logic supports up to 64 Mbytes of DRAM. The DRAM modules can be stacked up to two-high on connector P3. To increase performance, the DRAM array is divided into two interleaved 32-bit banks. Connector P3 provides a 64-bit data path to the MM3 memory modules.

The Intel486 microprocessor has a unified code and data cache. The cache is software transparent to maintain binary compatibility with previous generations of the x86 architecture. The iSBC 486/12 includes "bus snooping" logic to detect when the data in the cache has been invalidated by a DMA or Multibus write to DRAM. Only DRAM accesses are cached. To preserve cache coherency, the write strategy of the cache is write-through. The cache is disabled when the board is reset. The cache is enabled by writing to the CR0 register (an Intel486 CPU register).

ADMA, DAG, and SPC

The 82258 Advanced Direct Memory Access (ADMA) Controller provides four, independently programmable channels for high-performance DMA operations. The ADMA supports memory-to-memory, memory-to-I/O, and I/O-to-I/O transfers. The ADMA can access the DRAM, on-board I/O (including the SBX I/O address space), EPROM, SCSI FIFO, and Multibus memory and I/O.

The DMA Address Generator (DAG) is a custom gate array that extends the capabilities of the ADMA. The ADMA/DAG combination supports the full 4 Gbyte address range of the Intel486 microprocessor. It also supports high-speed burst transfers between the SCSI data FIFOs and memory.

The 82258 ADMA controller and the DAG are available on the high integration "S" series (iSBC 486/12S series).

I/O Subsystem

The iSBC 486/12 series include the following on-board I/O resources:

- 3-counter 82C54A Programmable Interval Timer (PIT)
- front panel interface
- 24-bit 82C55A Programmable Peripheral Interface (PPI)
- SCSI single-ended interface (on "S" models)
- two-channel 8274 Multi-protocol Serial Controller (MPSC)
- two SBX interfaces

Multibus/Dual-Port Memory

The Multibus interface supports 8- and 16-bit memory and I/O transfers. To support multi-processor operations, all on-board DRAM can be configured as dual-port memory. Up to 64 Mbytes (four 16-Mbyte pages) of dual-port memory can be accessed by other agents on the Multibus interface.

The Multibus interrupt lines are available at the interrupt jumper matrix. Two signals from the 82C55A PPI, and the memory parity error signal are provided to drive the Multibus interrupts. Bus-vectored interrupts are not supported.

***

Installation 3

Chapter Contents

Introduction 3-2
Resistor Pack and Fuse Data 3-2
Electrostatic Discharge Damage 3-3
Unpacking and Inspection 3-4
Component and Module Installation 3-4
MM3 Memory Module Installation 3-5
Parts Required for Installing MM3 Memory Modules 3-7
Memory Module Sizes 3-7
MM3 Installation Procedure A 3-8
MM3 Installation Procedure B 3-12
EPROM Installation 3-16
SCSI Termination Resistors and Fuse 3-18
Multibus Termination Resistors 3-19
RS-422A/449 Termination Resistors 3-20
3487 RS-422A/449 Line Driver 3-21
SBX Board Installation 3-21
Single-Wide Module Installation 3-23
Double-Wide SBX Module Installation 3-25
Board Installation Procedure 3-27

Introduction

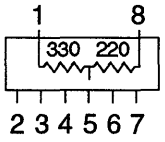
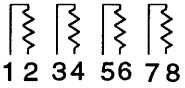
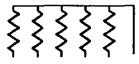
This chapter describes how to install the following components:

- resistor packs
- memory modules
- EPROMs
- SBX modules

Appendix A includes an installation checklist to help you during the installation process.

Resistor Pack and Fuse Data

Table 3-1. Resistor Pack and Fuse Data

Component	Specification
F1	SCSI termination power fuse, 1 A, 125 V.
RP1, RP2, RP3	 W-2545 SCSI termination resistor pack, 1 W, 5%, 220/330 Ω pull-up/pull-down.
RP4	 W-2544 RS-422A termination resistor pack, 1 W, 2%, refer to Appendix C for value.
RP5, RP6	 W-2543 RS-422A bias resistor packs, 3/4 W, 2%, refer to Appendix C for value.
RP7	RP7 is the Multibus address line termination resistor pack, 2.2 kΩ, 3/4 W, 2%.

Electrostatic Discharge Damage

CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY!

Because of the CMOS design, this board is sensitive to ESD (electrostatic discharge). When you handle the board, both you and the work station must be grounded (earth ground) to protect the CMOS devices. Use an antistatic bag to transport the board. If you do not follow these instructions, you can damage the board.

Until the iSBC 486/12 board is integrated into a system, it can be damaged by electrostatic discharges. We recommend you take the following steps to eliminate electrostatic discharge damage (ESD) to the board during your manufacturing process:

- Transport the board in the static shielding envelope provided by Intel.
- Require personnel who touch the board to wear special anti-static smocks.
- Use conductive table mat. This mat is grounded through a current-limiting resistor to assure personnel safety in case of a power line fault. The mat's top surface resistance is designed such that sliding a computer board across its surface will not generate more than 100 volts. The mat should have two swivel/snap connectors for the connection of grounding straps: one grounds the mat, the other grounds the operator to the mat.
- Use conductive wrist strap. This strap is kept in constant contact with bare skin and has a cable for attaching to the conductive table mat. The purpose of the wrist strap is to drain the operator's static charge as well as any static charge found on the parts container. The wrist strap cable has an integral current-limiting resistor for personnel safety. Wrist straps must be tested frequently to ensure that they are undamaged and operating correctly.
- Work on a conductive floor mat. This mat must also be grounded through a current-limiting resistor.
- Keep the work area free of static charge generators: plastic wrappers, cigarette packages, sandwich bags, foam drink cups, plastic binders.

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 the packing material for the agent's inspection.

For repair of a product damaged in shipment, contact Intel Customer Service Marketing Administration to obtain a Return Authorization Number and further instructions. A copy of the purchase order is required to complete the repair. A copy of the purchase order should also be submitted to the carrier with your claim. Refer to Chapter 10, "Service Information," for more details.

Save the shipping cartons and packing material in the event that the product must be returned at a future date.

Component and Module Installation

You may need to supply the following items to connect the board to your system, or to change the board for your application. The following sections show you how to install these items:

- additional MM3 memory module
- EPROMs
- SCSI termination resistors (iSBC 486/12S series only)
- SCSI fuse
- Multibus termination resistors
- RS-422A/449 bias and termination resistors
- 3487 RS-422A/449 line driver
- SBX boards

For information on cables and connectors, refer to Chapter 9, "Connectors and Cables."

MM3 Memory Module Installation

When you receive your board, the factory has installed one MM3 memory module. You can purchase and install an additional MM3 memory module or swap the existing memory module for a larger module. For example, to upgrade two iSBC 486/12 boards from 4 Mbytes to 8 Mbytes, you could combine the two MM304 memory modules onto one baseboard and then add an MM3008 memory module to the second board.

Figure 3-1 shows the location of connector P3 and an exploded view of two memory modules installed on the baseboard. This chapter describes two installation procedures:

A. **MM3 Installation Procedure A (page 3-8).** Use this procedure if you have a lower density or capacity memory module on the base board and you want to add a higher density or capacity memory module. In this case you will need to put the higher density or capacity module on the bottom and put the lower density or capacity module on top. This procedure shows you how to swap modules.

B. **MM3 Installation Procedure B (page 3-12).** Use this procedure if you have a memory module on the base board that has a density or capacity equal to or greater than the module you want to add.

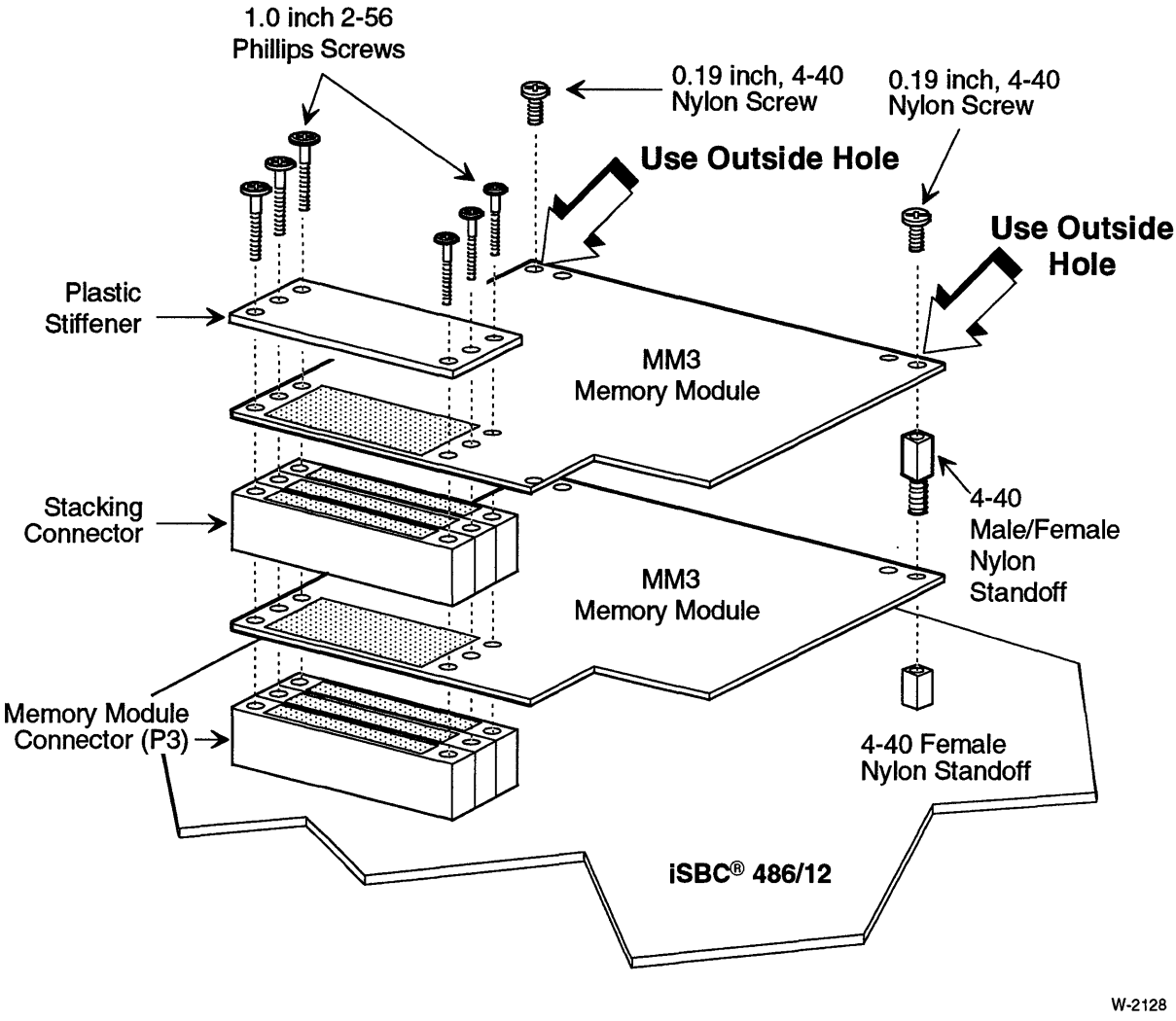


Figure 3-1. Memory Module Installation Exploded View

Parts Required for Installing MM3 Memory Modules

To add an additional memory module to the board, use the hardware kit that comes with the MM3 memory module.

Table 3-2. Parts and Tools for Memory Module Installation

Quantity	Description
1	Memory module
1	Stacking connector
2	4-40 male/female nylon stand-offs
1	No. 1 Phillips torque screw driver
1	Small straight-blade screw driver

Memory Module Sizes

The MM3 memory modules come in the following sizes:

Lower Density Modules

MM302	2 Mbytes	No longer available.
MM304	4 Mbytes	Can only be ordered with baseboard.
MM308	8 Mbytes	Replaced by MM3008.

Higher Density Modules
(Can be ordered separately or with baseboard.[1])

MM3008	8 Mbytes
MM3016	16 Mbyte
MM3032	32 Mbytes

[1]Contact Intel for availability.

CAUTION

When two modules are installed, the higher density or capacity module must be installed on the bottom.

MM3 Installation Procedure A

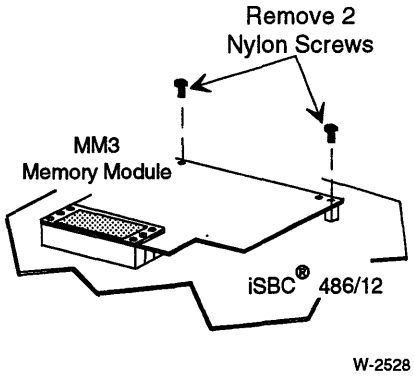
Use this procedure if you have a lower density or capacity memory module on the base board and you want to add a higher density or capacity memory module. In this case you will need to put the higher density or capacity memory module on the bottom and put the lower density or capacity module on top. This procedure shows you how to swap modules.

Table 3-3. Procedure A Combinations

Before Installation		After Installation	
Existing Memory Module on Baseboard	**Additional Memory Module**	**Bottom Memory Module**	**Top Memory Module**
MM302	MM304, MM304, MM308, MM3008, MM3016, or MM3032	MM304, MM304, MM308, MM3008, MM3016, or MM3032	MM302
MM304	MM308, MM3008, MM3016, or MM3032	MM308, MM3008, MM3016, or MM3032	MM304
MM308	MM3008, MM3016, or MM3032	MM3008, MM3016, or MM3032	MM308
MM3008	MM3016 or MM3032	MM3016 or MM3032	MM3008
MM3016	MM3032	MM3032	MM3016

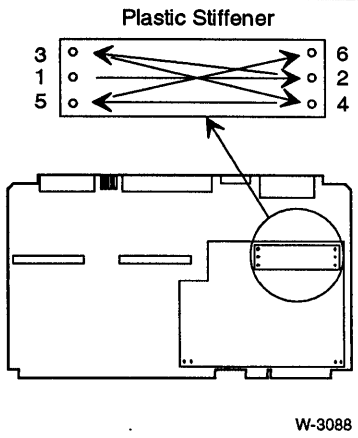
1. Place the iSBC 486/12 board, component-side up, on a grounded workbench or a piece of conductive foam.

2. Remove the two nylon screws from the top side of the standoff supports.

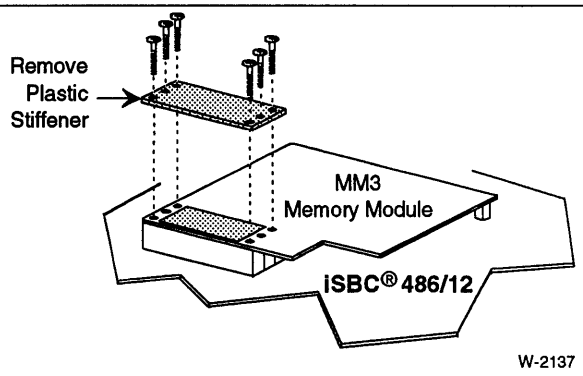


(continued)

3. Locate the black stiffener bar then loosen the six Phillips screws one turn at a time, alternating in a diagonal pattern.

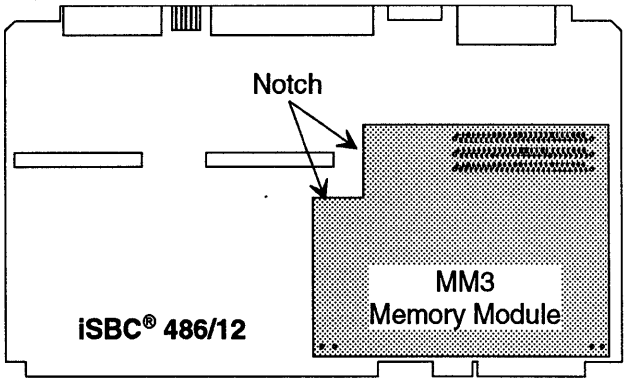


4. Remove the six Phillips screws then lift the black plastic stiffener off the module. Save the stiffener and screws.



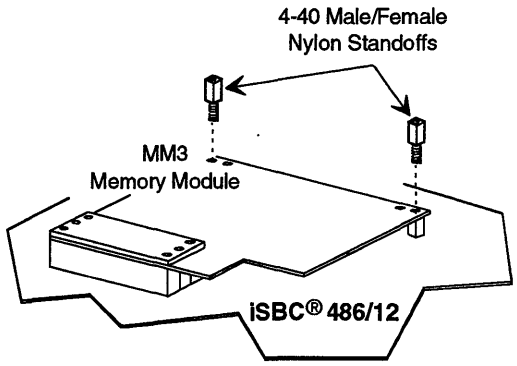
5. Lift off the memory module and place it in an anti-static bag.

6. Place the higher capacity memory module on the bottom connector and standoffs. Align the notch as shown in the figure.

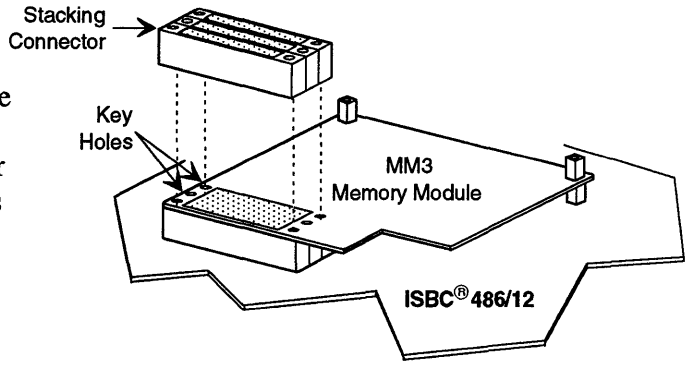


(continued)

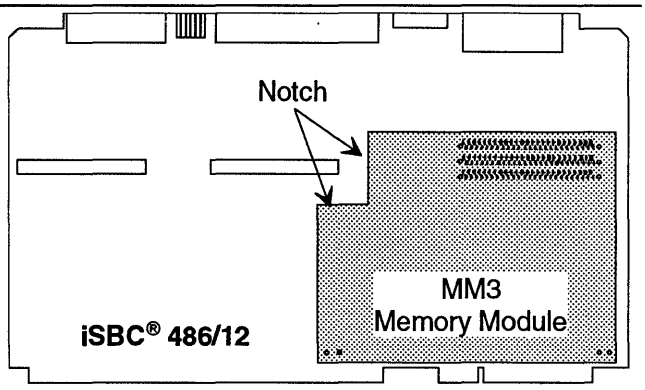
7. Screw a male/female standoff through the memory module into each of the existing standoffs.



8. Place the stacking connector on the top-side contact pads of the second memory module. Be sure the two alignment studs on the bottom of the stacking connector are in the largest mounting holes of the memory module.



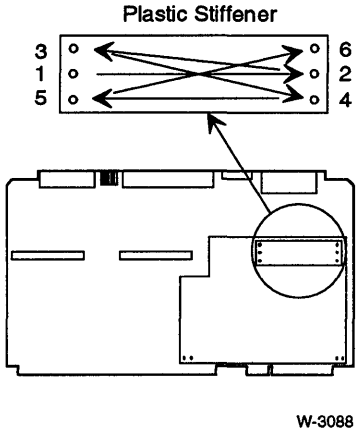
9. Place the lower capacity memory module with the primary side up on the stacking connector and the two standoff posts. Align the notch as shown in the figure.



10. Place the black stiffener on the top memory module contact pads with the recessed holes face up.

(continued)

11. Loosely install the six screws in the stiffener in the sequence shown in the figure.



CAUTION

Do not over tighten the screws. Over tightening can damage the connector and screws.

Use a No. 1 Phillips torque screw driver, set it to 3.5 inch-pounds, and tighten the six screws equally, one turn at a time, alternating in a diagonal pattern.

12. Install and tighten the two nylon screws on the standoffs. Use the ***outside*** holes.

MM3 Installation Procedure B

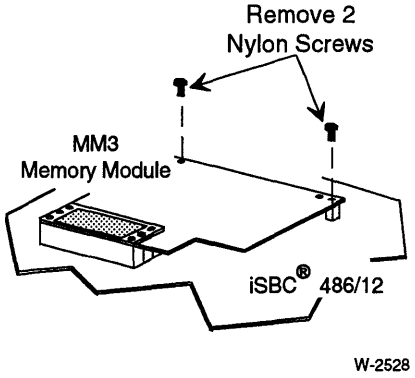
Use this procedure if you have a memory module on the base board that has a capacity equal to or greater than the module you want to add.

Table 3-4. Installation Procedure B Combinations

Before Installation		After Installation	
Existing Memory Module on Baseboard	**Additional Memory Module**	**Bottom Memory Module**	**Top Memory Module**
MM302	MM302	MM302	MM302
MM304	MM302 or MM304	MM304	MM302 or MM304
MM308	MM302, MM304, or MM308	MM308	MM302, MM304, or MM308
MM3008	MM302, MM304, MM308, or MM3008	MM3008	MM302, MM304, MM308, or MM3008
MM3016	MM302, MM304, MM308, MM3008, or MM3016	MM3016	MM302, MM304, MM308, MM3008, or MM3016
MM3032	MM302, MM304, MM308, MM3008, MM3016, or MM3032	MM3032	MM302, MM304, MM308, MM3008, MM3016, or MM3032

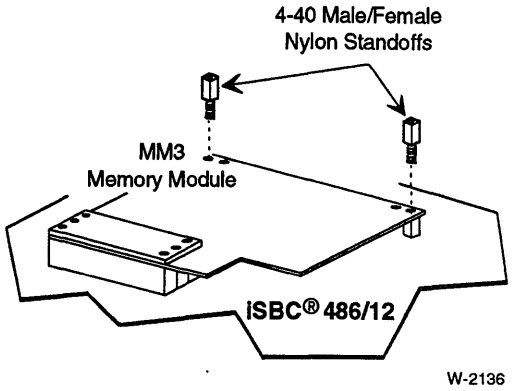
1. Place the iSBC 486/12 board, component-side up, on a grounded workbench or a piece of conductive foam.

2. Remove the two nylon screws from the top side of the standoff supports.

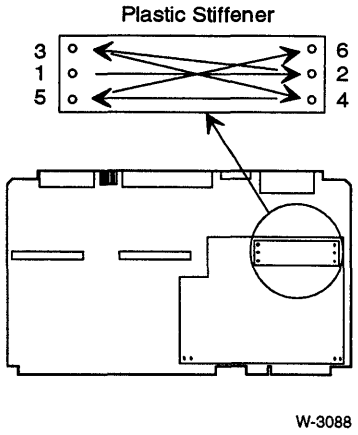


(continued)

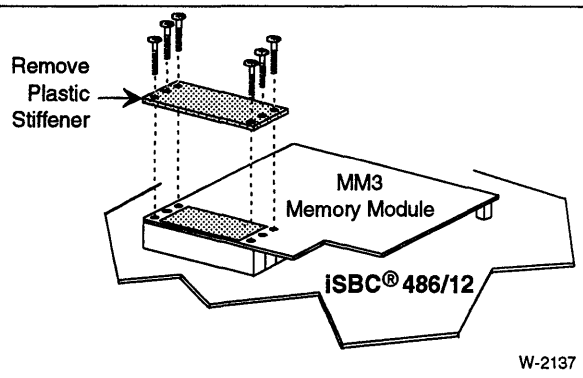
3. Screw a new male/female standoff on each of the existing standoffs through the original memory module.



4. Remove the six screws from the black stiffener bar on connector P3 of the bottom module, one turn at a time, alternating in a diagonal pattern.

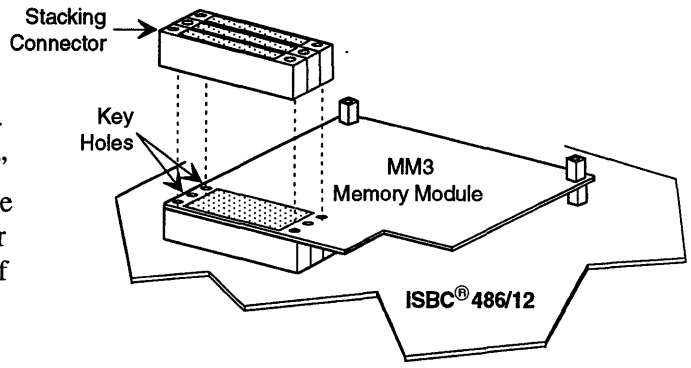


5. Lift the plastic stiffener off the module.



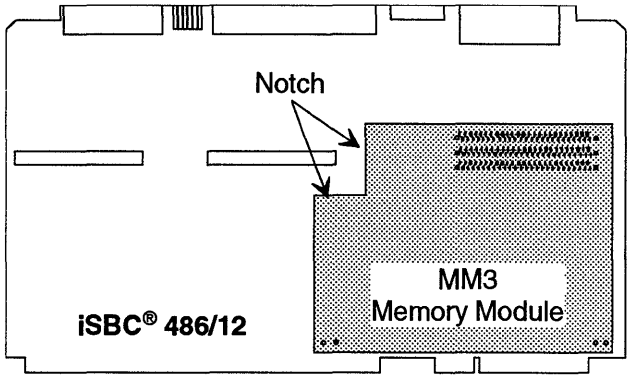
(continued)

6. Place the stacking connector where the stiffener was, on the top-side contact pads of the bottom module. The top-side of the module is marked "Primary." Be sure the alignment stud on the bottom of the stacking connector is in the largest mounting hole of the memory module.



W-2127

7. Place the new MM308 memory module with the primary side up on the stacking connector and the two standoff posts.

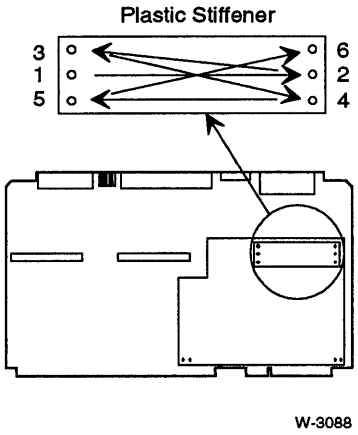


W-2530

8. Place the stiffener on the top memory module contact pads with the recessed holes face up.

(continued)

9. Loosely install the six screws in the stiffener in the sequence shown in the figure.



CAUTION

When installing memory modules, do not over tighten the screws. Over tightening can damage the connector and screws.

Use a No. 1 Phillips torque screw driver, set it to 3.5 inch-pounds, and tighten the six screws equally, one turn at a time, alternating in a diagonal pattern.

10. Install and tighten the two nylon screws on the standoffs. Use the ***outside*** holes.

EPROM Installation

For normal board operation, you must supply two EPROMs in the JEDEC sockets U28 (even-address byte) and U43 (odd-address byte). Figure 3-2 shows the installation of EPROMs in the 32-pin sockets.

Chapter 4, "Real Mode Memory Configuration," will help you select an EPROM for your application.

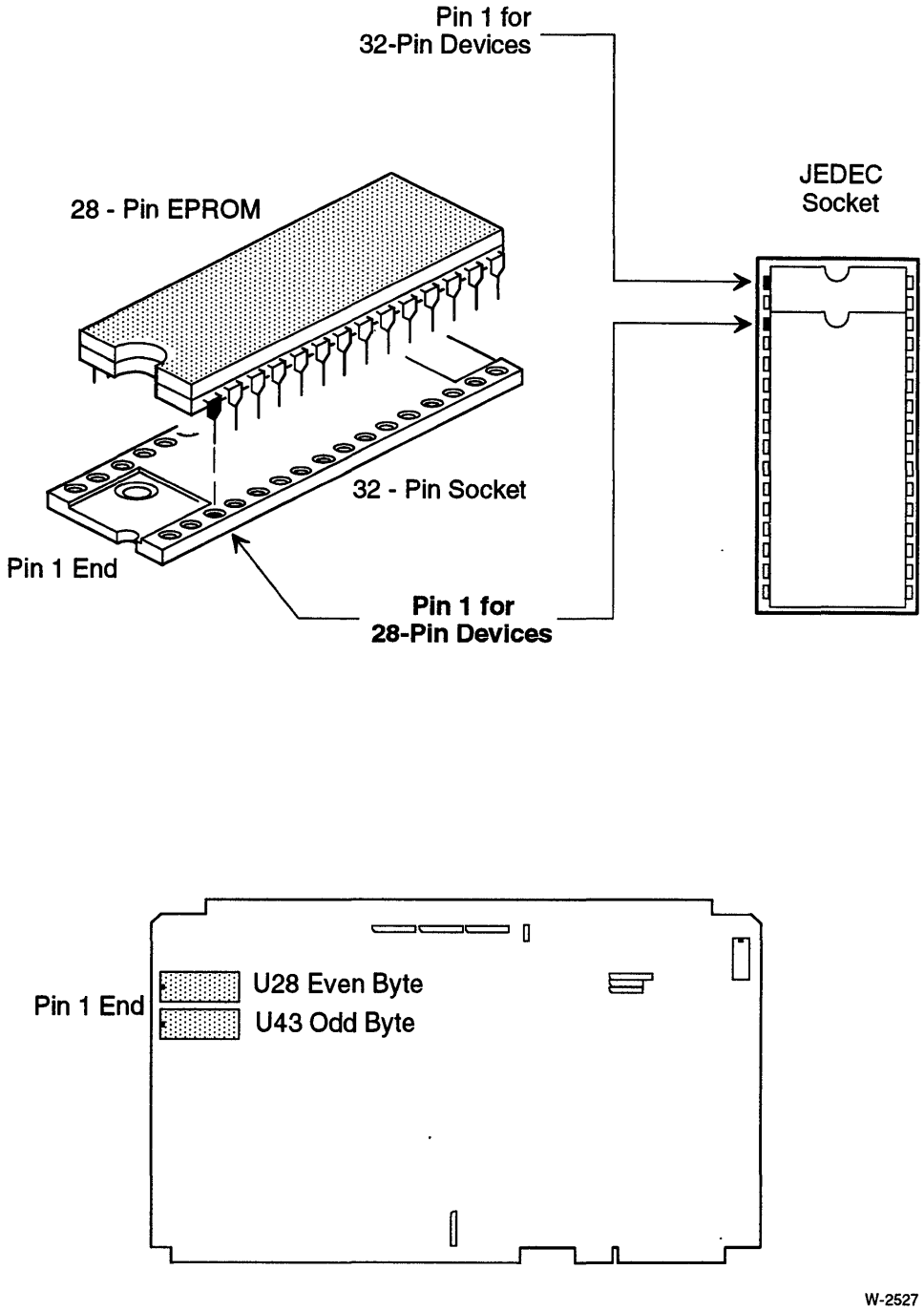


Figure 3-2. EPROM Installation

SCSI Termination Resistors and Fuse

Install the SCSI termination resistors if the iSBC 486/12S board is at the end of the SCSI cable, otherwise remove the resistors.

Install the SCSI termination power fuse if you want the iSBC 486/12S board to power the termination resistors on the SCSI cable when the other SCSI devices are powered off.

Refer to Chapter 7, "I/O Subsystem Configuration," for more information.

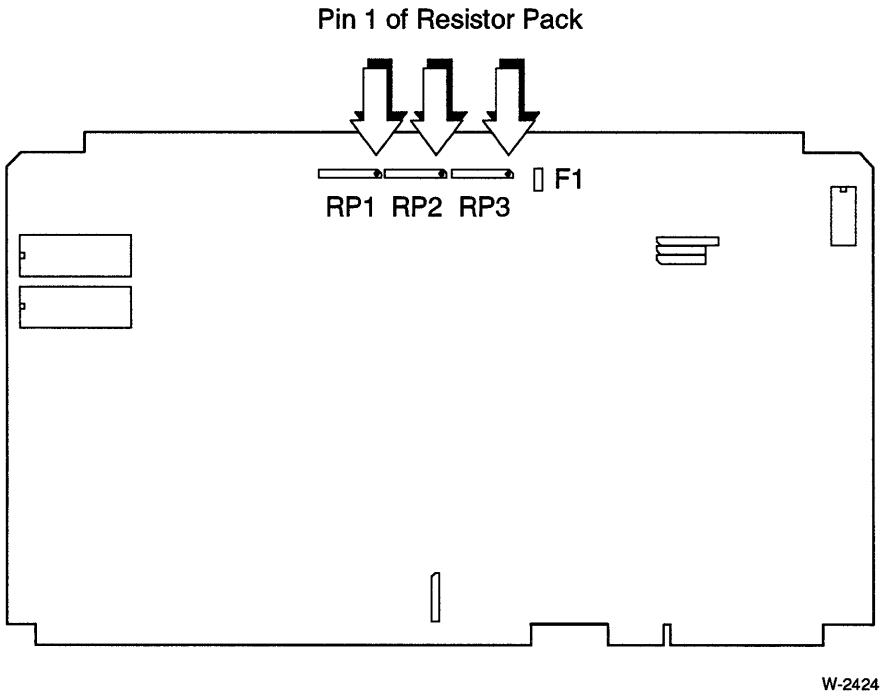


Figure 3-3. SCSI Termination Resistors and Fuse

Multibus Termination Resistors

Install the Multibus termination resistor pack RP7 if the iSBC 486/12 board is selected to terminate the Multibus address lines.

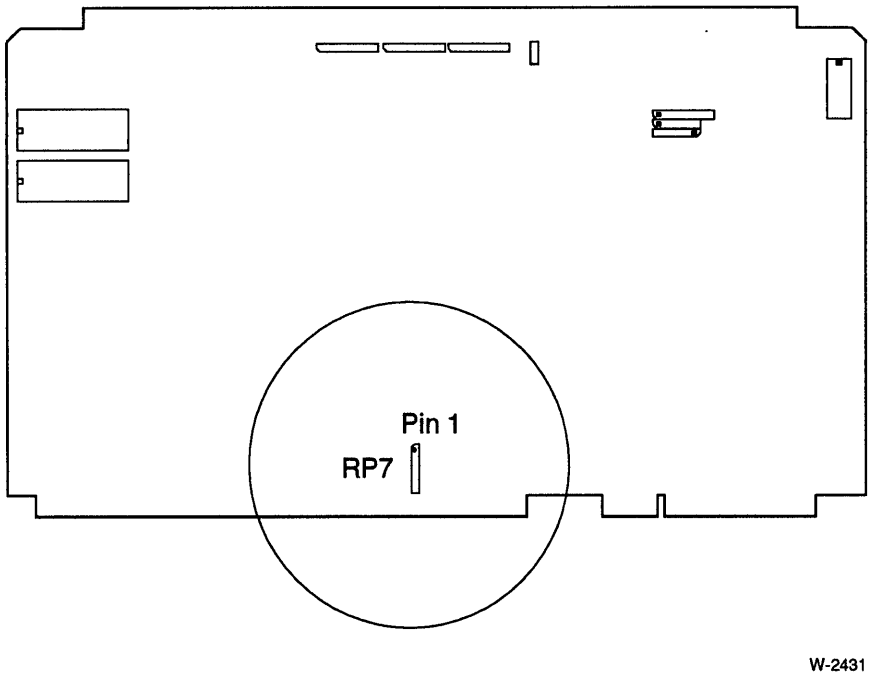


Figure 3-4. Multibus Termination Resistor Pack

RS-422A/449 Termination Resistors

You can install bias and termination resistors in sockets RP4, RP5, and RP6 to bias the TR, TT, SD, and RS signal lines in an RS-422A/449 application. Figure 3-6 (page 3-21) shows the location and orientation of the sockets.

CAUTIONS

1. **For RS-422A/449 operation, pin 1 of RP6 must be installed in pin 6 of the RP6 socket.**

2. **For RS-422A/449 operation as the most remote slave on a multidrop line, install RP4 and remove RP5 and RP6. For RS-422A/449 operation as the master on a multidrop line, remove RP4 and install RP5 and RP6.**

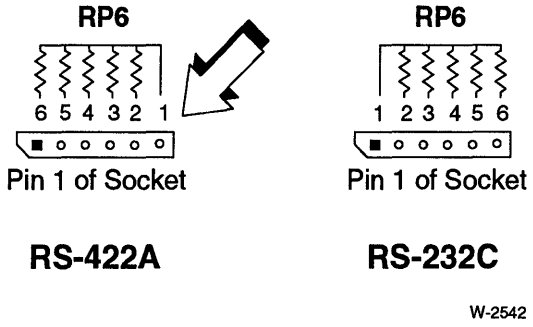


Figure 3-5. Orientation of RP6 for RS-422A Operation

Refer to Chapter 7, "I/O Subsystem Configuration," for serial configuration information. Refer to Appendix C for information on how to compute the value of RP4, RP5, and RP6 for multi-drop operation.

3487 RS-422A/449 Line Driver

If you configure connector J4L for RS-422A operation, you must install a DS3487N RS-422A line driver in socket U14. National Semiconductor produces a 3487 device that is compatible with the board.

Figure 3-6 shows the location of socket U14. Refer to Chapter 7, "I/O Subsystem Configuration," for serial configuration information. Refer to Appendix C for information on how to compute the value of RP4, RP5, and RP6 for multi-drop operation.

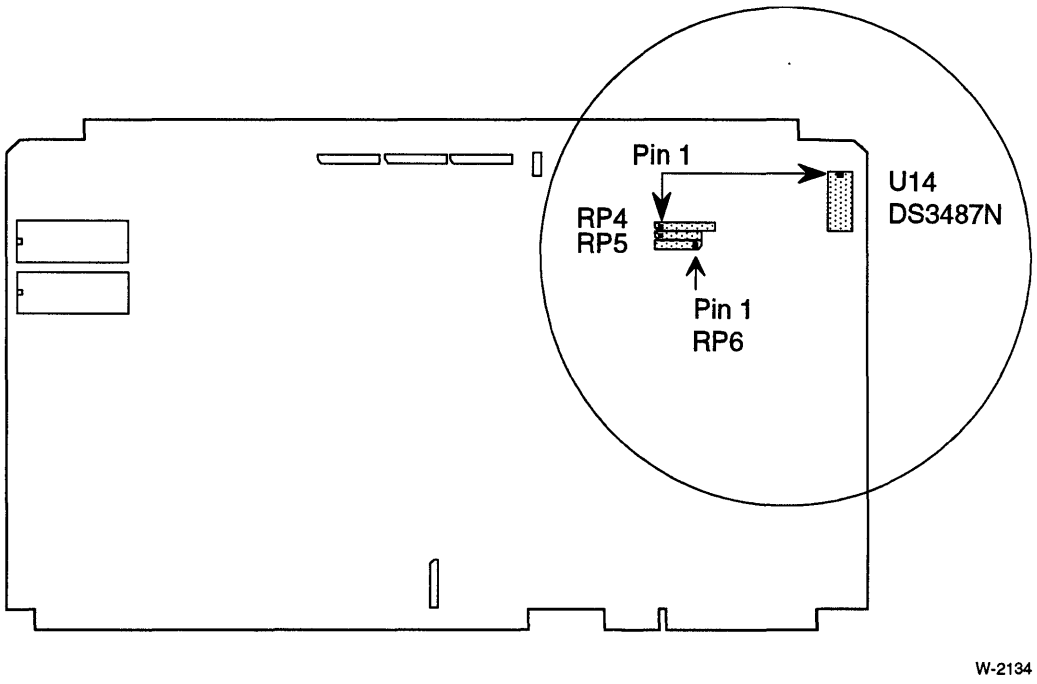


Figure 3-6. 3487 RS-422A Line Driver and Termination Resistor Sockets

SBX Board Installation

To expand the iSBC 486/12 board's I/O capabilities, you can add up to two SBX boards in connectors J5 and J6. Both connectors support 8-bit and 16-bit operation. Each connector supports DMA transfers.

Refer to Chapter 7, "I/O Subsystem Configuration," for SBX configuration information.

Connector J6 supports single-wide and double-wide SBX boards. Due to the physical interference with the memory module, connector J5 supports only single-wide SBX boards. If you install a double-wide SBX module on connector J6, you cannot install a single-wide SBX module on connector J5, due to the physical interference of the double-wide module with connector J5. Figure 3-7 shows the single- and double-wide SBX boards installed on the iSBC 486/12 baseboard.

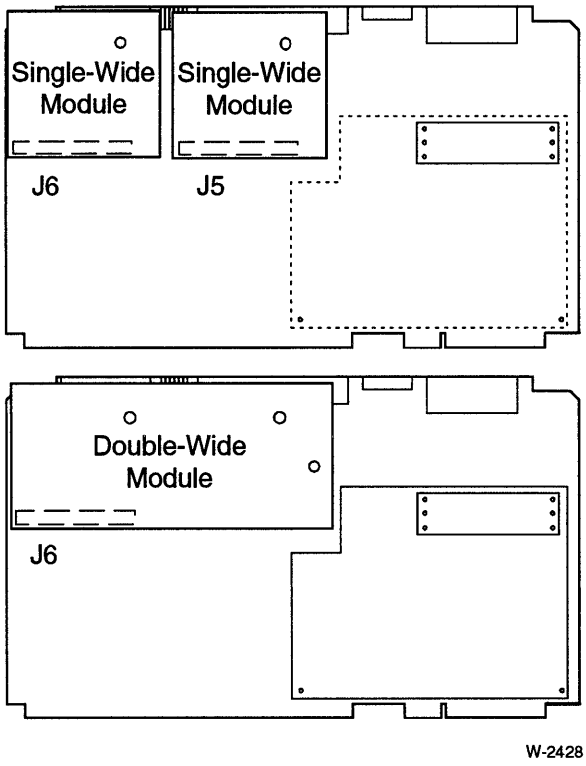


Figure 3-7. Single and Double-Wide SBX Modules

Single-Wide Module Installation

Table 3-5 lists the parts you will need to install a single-wide SBX module on connector J5 or J6.

Table 3-5. Parts and Tools for Single-Wide SBX Module Installation

Quantity	Description
1	Single-wide SBX module
1	0.5-inch nylon spacers
2	0.25-inch 6-32 nylon screws
1	Straight-blade screwdriver

To install a single-wide SBX module on connector J5 or J6, do the following:

1. Attach the nylon spacer to the baseboard using one nylon screw inserted through the bottom of the baseboard.
2. Position the SBX module over the SBX connector and press down directly over the connector to seat it firmly.
3. Insert the nylon screw through the SBX module and attach it to the nylon standoff.

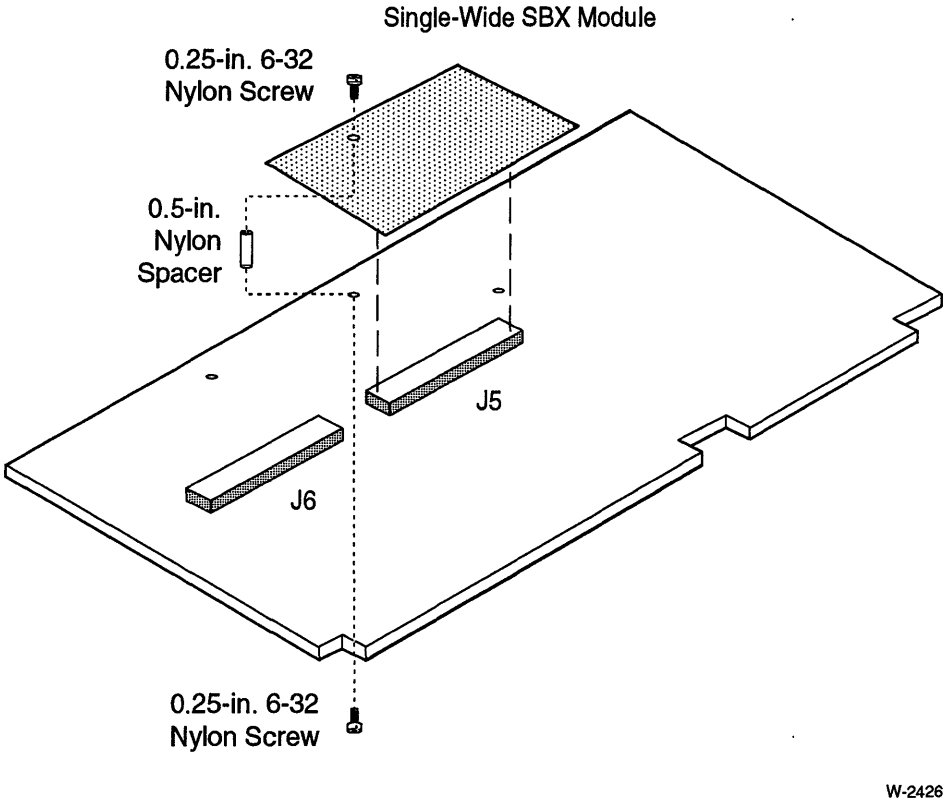


Figure 3-8. Single-Wide SBX Module Installation

Double-Wide SBX Module Installation

Table 3-6 lists the parts you will need to install a double-wide SBX module on connector J6.

Table 3-6. Parts and Tools for Double-Wide SBX Module Installation

Quantity	Description
1	Double-wide SBX module
3	0.5-inch nylon spacers
6	0.25-inch 6-32 nylon screws
1	Straight-blade screwdriver

To install a double-wide SBX module on connector J6, do the following:

1. Attach the three nylon spacers to the baseboard.
2. Position the SBX module over connector J6 and press down directly over the connector to seat it firmly.
3. Insert the three screws through the SBX module and attach them to the nylon standoffs.

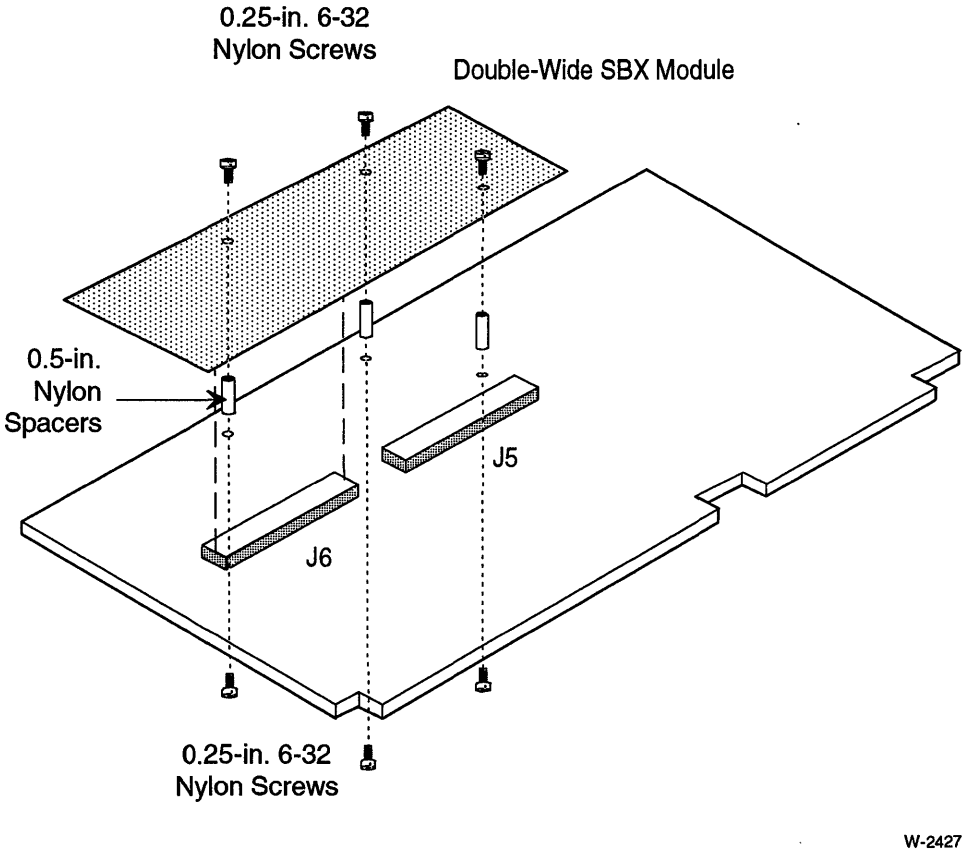


Figure 3-9. Double-Wide SBX Module Installation

Board Installation Procedure

CAUTION
SUBJECT TO DAMAGE BY STATIC ELECTRICITY!

Because of the CMOS design, the board is sensitive to ESD (electrostatic discharge damage). Refer to page 3-3 for recommended ESD measures.

Follow these steps to install the board in your system:

1. Turn the system power off.
2. Connect your body (electrically) to an earth ground device.
3. Remove the board from the antistatic bag.
4. Place the board on a piece of ESD-grounded conductive foam.
5. Verify that the board has the proper jumper configuration for your system use.
6. Insert your EPROMs, SBX modules, and other devices as described in this chapter.
7. Slide the board in a card slot in the system cardcage and firmly seat the Multibus connectors (P1 and P2).
8. Connect your cables.
9. Turn the power on.

***

Real Mode Memory Configuration 4

Chapter Contents

Introduction 4-2
- Related Information 4-2
- New Terms in this Chapter 4-2
- Default Real Mode Memory Map 4-3
- Configuration Priority 4-3
- Switching Between Real and Protected Modes 4-4
- Configuration Procedure 4-5

1st Priority: Multibus Window Configuration 4-5
- Window Size 4-6
- Local Address to Multibus Window Address Mapping 4-6
- Window Jumper Configuration 4-6

2nd Priority: DRAM Configuration (Real Mode) 4-8
- DRAM Size 4-8
- DRAM Starting and Ending Addresses 4-8
- DRAM Access from the Multibus Interface 4-8
- DRAM Jumper Configuration 4-8

3rd Priority: EPROM Configuration 4-10
- EPROM Size 4-10
- EPROM Access Time 4-11
- EPROM Address Range (Real Mode) 4-11
- EPROM Access from the Multibus Interface 4-11
- EPROM Jumper Configuration 4-12

4th Priority: Multibus Memory Space Configuration 4-13
Sample Worksheet 4-15
Configuration Questions and Answers 4-17

Introduction

The Intel486 microprocessor has two modes of operation: *real address mode* (real mode) and *protected virtual address mode* (also called protected mode or PVAM). This chapter discusses the real mode memory organization of the iSBC 486/12 board. In real mode, the Intel486 microprocessor operates as a very fast 8086 processor. Real mode is required primarily to set up the processor and board for protected mode operation. The addressing limit in real mode is 1 Mbyte of memory.

Related Information

The following topics are related to real mode memory configuration:

- protected mode memory configuration, see Chapter 5
- system memory configuration, see Chapter 6
- real mode multiprocessing example, see Chapter 6
- details of changing between real mode and protected mode, see Chapter 7
- real mode programming details, see the *Intel486™ DX Microprocessor Data Book* and the *Intel486™ Microprocessor Family Programmer's Reference Manual*

New Terms in this Chapter

dual-port memory — Memory that can be accessed by both the local CPU and from the Multibus interface.

local DRAM — On-board DRAM that is accessible to the local CPU.

Multibus interface address space — An address space mapped to the Multibus interface. When the iSBC 486/12 board is in real mode, the address space between the end of DRAM and the start of EPROM is mapped to the Multibus interface.

Multibus window — The Multibus "window" is a contiguous block of memory addresses in real mode that is mapped to the Multibus interface memory space. The Multibus window can start and end on any 64-Kbyte boundary in the real address memory space (0H to 0F FFFFH).

multiprocessing — Two or more CPU boards operating together across the Multibus interface. You can use dual-port memory to share data and control information between the multiple CPUs.

real address mode — The default addressing mode for the x86 family of microprocessors. The Intel486 microprocessor in real mode is compatible with the Intel 8086 microprocessor.

system memory — Memory resources on the Multibus interface.

Default Real Mode Memory Map

In real mode, the default memory configuration is as follows:

- 128 Kbytes of EPROM at address range 0E 0000H to 0F FFFFH
- 896 Kbytes of local DRAM at address range 0H to 0D FFFFH
- 0 Kbytes of the Multibus interface address space
- Multibus window is disabled

Figure 4-1 shows the default real mode memory map.

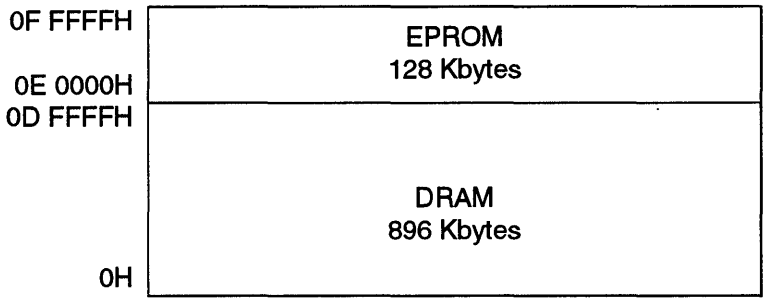


Figure 4-1. Default Real Mode Memory Map

Configuration Priority

If you overlap the address spaces for different types of memory resources, the address decode logic selects the memory resources in the following order:

1st	highest priority	Multibus window
2nd		DRAM
3rd		EPROM
4th	lowest priority	Multibus interface address space

CAUTION

Do not set the real mode ending DRAM address to 0F FFFFH. In this configuration, DRAM extends to the top of memory and the EPROM address space is masked. (Although the Intel486 microprocessor always starts at 0FFFF FFF0H and remains in the last 64 Kbytes of the 4 Gbyte address space until the first long jump or call, the iSBC 486/12 board ignores the upper address lines and only uses 20-bit addressing while in real mode.)

NOTE

Be careful when creating a Multibus window that uses the entire 1 Mbyte address space. In this configuration, the iSBC 486/12 board's CPU must rely on other boards in the system for all memory resources.

Switching Between Real and Protected Modes

Both the Intel486 microprocessor and the iSBC 486/12 board have two modes of operation: real address mode and protected virtual address mode. Both the CPU and the board should be switched to the same mode. After a cold reset, both the CPU and the board are in real mode.

To switch the CPU between real and protected mode, you must change the PE bit in the CR0 register (a register inside the Intel486 CPU). You must also, in most cases, do some initialization. Refer to the *Intel486™ Microprocessor Family Programmer's Reference Manual* for information on switching the Intel486 microprocessor to protected mode.

To switch the board between real and protected mode, you must write to an I/O address. Refer to Chapter 7 for the details of changing the board between real mode and protected mode.

Configuration Procedure

To configure your board for real address mode, do the following:

1. Copy the Real Mode Memory Configuration Worksheets in Appendix A.
2. Copy the Jumper Configuration Worksheet in Appendix A.
3. Follow the worksheet and check-off each step. Pencil-in the jumpers on the jumper worksheet as you select each option. If you get lost, refer to the Table of contents at the beginning of the chapter.
4. Go on to Chapter 5, "Protected Mode Memory Configuration."

The information in this chapter is arranged in the same order as the priority used by the address decode logic. This chapter contains the following subsections:

- 1st Priority: Multibus Window Configuration
- 2nd Priority: DRAM Configuration (Real Mode)
- 3rd Priority: EPROM Configuration
- 4th Priority: Multibus Memory Space Configuration

You may also want to look at:

- the sample worksheets starting on page 4-15
- configuration question and answers on page 4-17

1st Priority: Multibus Window Configuration

The Multibus interface "window" is a contiguous block of addresses in real mode that is mapped to the memory space of the Multibus interface. This window allows the Intel486 CPU to access memory-mapped resources over the Multibus interface. The window does *not* allow other agents on the Multibus interface to access local memory resources. The Multibus window has priority over all other memory resources.

NOTE

The Multibus interface window is a feature of the iSBC 486/12 board that only exists when the board is in real mode. In protected mode, the iSBC 486/12 board has other features that make the Multibus window unnecessary.

You can use the Multibus window to support a multiprocessing environment. Refer to "System Memory Configuration" (Chapter 6) for configuration details.

In addition to the Multibus window, the address space between the end of DRAM and the start of EPROM is, by default, also mapped to the Multibus interface. You can use both the Multibus window and the default Multibus address space in the same way.

Window Size

You can configure the starting and ending address of this window on any 64-Kbyte boundary. Set the starting address higher than the ending address to disable the Multibus window.

Local Address to Multibus Window Address Mapping

All addresses in the Multibus window are mapped 1-to-1 to the same addresses in the Multibus memory space. For example, a Multibus window from 0D 0000H to 0D FFFFH maps to the same address on the Multibus interface.

Window Jumper Configuration

Refer to Tables 4-1 and 4-2 to set the jumpers for the Multibus window starting and ending addresses. Record your jumper settings on a copy of the Jumper Worksheet in Appendix A.

Table 4-1. Multibus Window Starting Address Jumper Configuration

Starting Address	E093-E103	E113-E123	E133-E143	E153-E163
00 0000H	OUT	OUT	OUT	OUT
01 0000H §	OUT	OUT	OUT	IN
02 0000H	OUT	OUT	IN	OUT
03 0000H	OUT	OUT	IN	IN
04 0000H	OUT	IN	OUT	OUT
05 0000H	OUT	IN	OUT	IN
06 0000H	OUT	IN	IN	OUT
07 0000H	OUT	IN	IN	IN
08 0000H	IN	OUT	OUT	OUT
09 0000H	IN	OUT	OUT	IN
0A 0000H	IN	OUT	IN	OUT
0B 0000H	IN	OUT	IN	IN
0C 0000H	IN	IN	OUT	OUT
0D 0000H	IN	IN	OUT	IN
0E 0000H	IN	IN	IN	OUT
0F 0000H	IN	IN	IN	IN

Note:
§ default configuration (Multibus window disabled)

Table 4-2. Multibus Window Ending Address Jumper Configuration

Ending Address	E094-E104	E114-E124	E134-E144	E154-E164
00 FFFFH §	OUT	OUT	OUT	OUT
01 FFFFH	OUT	OUT	OUT	IN
02 FFFFH	OUT	OUT	IN	OUT
03 FFFFH	OUT	OUT	IN	IN
04 FFFFH	OUT	IN	OUT	OUT
05 FFFFH	OUT	IN	OUT	IN
06 FFFFH	OUT	IN	IN	OUT
07 FFFFH	OUT	IN	IN	IN
08 FFFFH	IN	OUT	OUT	OUT
09 FFFFH	IN	OUT	OUT	IN
0A FFFFH	IN	OUT	IN	OUT
0B FFFFH	IN	OUT	IN	IN
0C FFFFH	IN	IN	OUT	OUT
0D FFFFH	IN	IN	OUT	IN
0E FFFFH	IN	IN	IN	OUT
0F FFFFH[1]	IN	IN	IN	IN

Notes:
§ default configuration (Multibus window disabled)
[1] Be careful when creating a Multibus window that uses the entire 1 Mbyte address space. In this configuration, the iSBC 486/12 board must rely on other boards in the system for all memory resources.

2nd Priority: DRAM Configuration (Real Mode)

DRAM Size

The iSBC 486/12 board can have from 2 to 64 Mbytes of on-board DRAM. In real mode, however, the Intel486 CPU can only address the first 1 Mbyte of memory. (To access more memory, you must switch the iSBC 486/12 board and the Intel486 CPU to protected mode.)

DRAM Starting and Ending Addresses

The local DRAM starting address is always 0H. You have the option to configure the ending address of DRAM in real mode.

The ending DRAM address can range from 0 FFFFH to 0F FFFFH. The ending address must be on a 64 Kbyte boundary. The space between the end of DRAM and the start of EPROM is mapped to the Multibus interface.

NOTES

1. Remember that the priority of the Multibus window is higher than DRAM, whereas the priority of EPROM is lower than DRAM. This means that if you overlap the DRAM and EPROM, memory accesses will be mapped to DRAM. If you overlap DRAM and the Multibus window, memory accesses will be mapped to the Multibus interface.

2. The real mode DRAM ending address is configured independently of the protected mode DRAM ending address.

DRAM Access from the Multibus Interface

All dual-ported DRAM is accessible to other agents on the Multibus interface. Chapter 6 provides information on how to configure the dual-port memory. The dual-port configuration is the same for both real mode and protected mode.

DRAM Jumper Configuration

Table 4-3 shows the jumper settings used to configure the DRAM ending address.

Table 4-3. DRAM Ending Address Jumper Configuration

Ending Address	E092-E102	E112-E122	E132-E142	E152-E162
00 FFFFH	OUT	OUT	OUT	OUT
01 FFFFH	OUT	OUT	OUT	IN
02 FFFFH	OUT	OUT	IN	OUT
03 FFFFH	OUT	OUT	IN	IN
04 FFFFH	OUT	IN	OUT	OUT
05 FFFFH	OUT	IN	OUT	IN
06 FFFFH	OUT	IN	IN	OUT
07 FFFFH	OUT	IN	IN	IN
08 FFFFH	IN	OUT	OUT	OUT
09 FFFFH	IN	OUT	OUT	IN
0A FFFFH	IN	OUT	IN	OUT
0B FFFFH	IN	OUT	IN	IN
0C FFFFH	IN	IN	OUT	OUT
0D FFFFH §	IN ✓	IN ✓	OUT ✓	IN ✓
0E FFFFH	IN	IN	IN	OUT
0F FFFFH [1]	IN	IN	IN	IN

Notes:
§ default
[1] Be careful when setting the ending DRAM address to 0F FFFFH. In this configuration, DRAM extends to the top of memory and the EPROM address space in the first 1 Mbyte is masked. Generally this configuration will not work unless you have a Multibus window that allows the iSBC 486/12 CPU to boot from other memory resources in your system. (This is not a recommended configuration.)

NOTE

While the Intel486 microprocessor always starts at 0FFFF FFF0H and remains in the last 64 Kbytes of the 4 Gbyte address space until the first long jump or call, the iSBC 486/12 board ignores the upper address lines and uses 20-bit addressing while in real mode.

3rd Priority: EPROM Configuration

This section discusses the following topics:

- supported EPROM sizes
- EPROM access time requirements
- EPROM address range in real mode
- access to EPROM from the Multibus interface
- configuration of the EPROM jumpers

EPROM Size

The iSBC 486/12 board has two JEDEC EPROM sites that can be configured to accept 28- or 32-pin EPROMs. Table 4-4 shows the supported sizes.

Table 4-4. Supported EPROM Sizes

EPROM	Device Density (bits)	Total Size (Kbytes)	Total Size (bytes)
27128	16K x 8	32	8FFFH
27256	32K x 8	64	0 FFFFH
27512	64K x 8	128 (default)	1 FFFFH
27010	128K x 8	256	3 FFFFH
27020	256K x 8	512	7 FFFFH
27040	512K x 8	960 [1]	0E FFFFH

Note:
[1] 64 Kbytes are always mapped to DRAM or the Multibus window. The usable size of the EPROM address space is therefore reduced to 960 Kbytes.

EPROM Access Time

The iSBC 486/12 board does not have jumper-configured EPROM access speed. All EPROMs must meet the following parameters:

$T_{acc} \leq 250$ ns (address to output delay)

$T_{ce} \leq 250$ ns (chip enable to output delay)

$T_{oe} \leq 250$ ns (output enable to output delay)

NOTE

For reliable operation, you must use EPROMs with $T_{acc} \leq 250$ ns.

EPROM Address Range (Real Mode)

In real mode, EPROM address space is always justified at the top of the 1 Mbyte memory space. The default EPROM address range is from 0D 0000H to 0F FFFFH (128 Kbytes).

In real mode, the board supports from 32 Kbytes to 960 Kbytes of EPROM. (You are limited to 960 Kbytes because the minimum DRAM configuration is 64 Kbytes.)

NOTE

Remember that both the Multibus window and the DRAM address space have priority over EPROM in the memory decode scheme.

EPROM Access from the Multibus Interface

The EPROM address space is not accessible to other agents on the Multibus interface.

EPROM Jumper Configuration

Refer to Table 4-5 for EPROM jumper configuration information. Record the information on a copy of the Jumper Worksheet provided in Appendix A.

Table 4-5. EPROM Jumper Configuration

Jumper	EPROM Device Type					
	27128	27256	27512§	27010	27020	27040
E091-E101	OUT	OUT	OUT	IN	IN	IN
E111-E121	OUT	IN	IN	OUT	OUT	IN
E131-E141	IN	OUT	IN	OUT	IN	OUT
E234-E239	IN	OUT	OUT	OUT	OUT	OUT
E239-E244	OUT	IN	IN	IN	IN	IN
E233-E238	IN	IN	OUT	OUT	OUT	OUT
E238-E243	OUT	OUT	IN	IN	IN	IN
E232-E237	IN	IN	IN	OUT	OUT	OUT
E237-E242	OUT	OUT	OUT	IN	IN	IN
E231-E236	X	X	X	IN	IN	OUT
E236-E241	X	X	X	OUT	OUT	IN
Capacity (Bytes)	16K	32K	64K	128K	256K	512K
Total Bytes	32K	64K	128K	256K	512K	1M (960)[1]
Starting Address	0F 8000H	0F 0000H	0E 0000H	0C 0000H	08 0000H	01 0000H[1]

Notes:
§ default
X means "don't care"
[1] The first 64 Kbytes are reserved for local DRAM (0H to 0 FFFFH) in real mode.

4th Priority: Multibus Memory Space Configuration

You can access system memory or memory-mapped I/O resources through the Multibus memory space. The memory address range that is not configured as EPROM or DRAM maps to the Multibus interface. In addition, you can configure a separate, noncontiguous Multibus window. If the Multibus window meets or overlaps the Multibus interface address space, the two form one contiguous block.

You can configure the board for up to 928 Kbytes of Multibus address space (or 1 Mbyte using a Multibus window). The maximum address range involves the following requirements:

- minimum amount of DRAM: 64 Kbytes
- minimum amount of EPROM: 32 Kbytes

You can use the Multibus interface address space and the Multibus interface window to support a multiprocessing environment. Refer to "System Memory Configuration" (Chapter 6) for details.

The following Table shows the size of the Multibus interface address space for all the combinations of EPROM size and DRAM ending address.

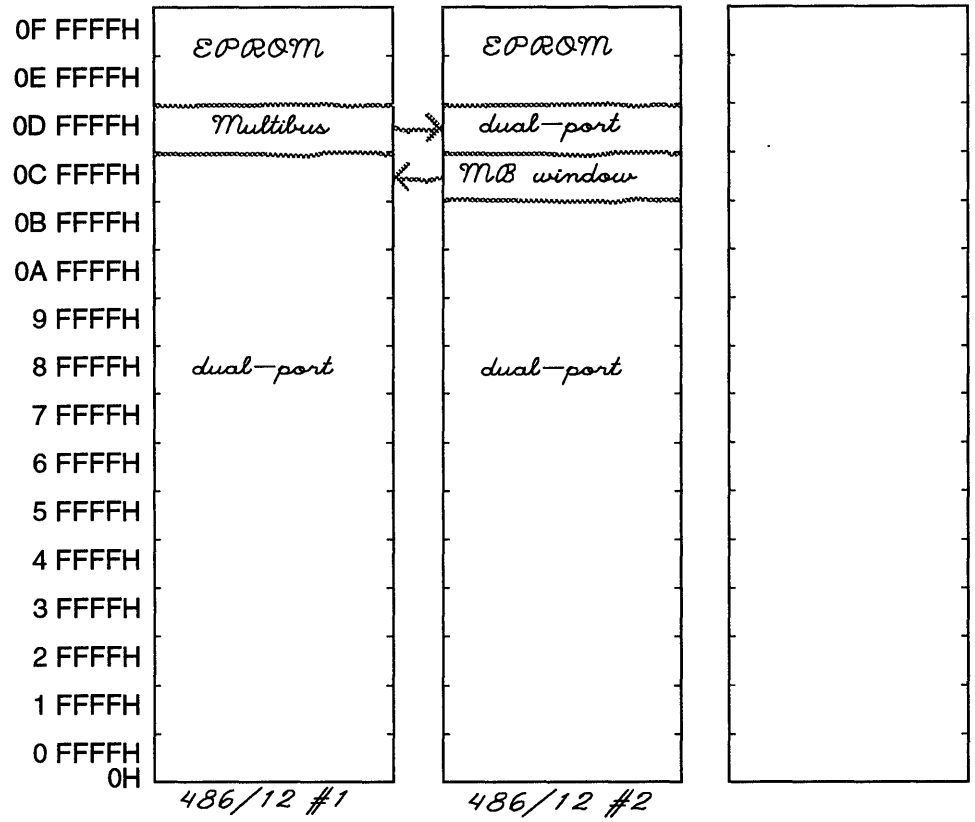
Table 4-6. Size of Multibus Interface Address Space (Kbytes)

	EPROM Size (Bytes)					
	32K	64K	128K§	256K	512K	1M
	EPROM Starting Address					
	0F 8000H	0F 0000H	0E 0000H§	0C 0000H	8 0000H	1 0000H
DRAM Ending Address	Multibus Interface Address Space (Kbytes)					
0F FFFFH	0	0	0	0	0	0
0E FFFFH	32	0	0	0	0	0
0D FFFFH §	96	64	0§	0	0	0
0C FFFFH	160	128	64	0	0	0
0B FFFFH	224	192	128	0	0	0
0A FFFFH	288	256	192	64	0	0
09 FFFFH	352	320	256	128	0	0
08 FFFFH	416	384	320	192	0	0
07 FFFFH	480	448	384	256	0	0
06 FFFFH	544	512	448	320	64	0
05 FFFFH	608	576	512	384	128	0
04 FFFFH	672	640	576	448	192	0
03 FFFFH	736	704	640	512	256	0
02 FFFFH	800	768	704	576	320	0
01 FFFFH	864	832	768	640	384	0
00 FFFFH	928	896	832	704	448	0

Note:
§ default

Sample Worksheet

Real Mode Memory Map Worksheet



NOTES:

486/12 #1:
- 128 Kbytes EPROM
- 64 Kbyte Multibus interface address space
- 832 Kbytes of dual-port DRAM
- no Multibus window

486/12 #2:
- 128 Kbytes EPROM
- no Multibus interface address space
- 832 Kbytes of dual-port DRAM (non-contiguous)
- 64 Kbytes Multibus windows

W-2151

Real Mode Memory Configuration Worksheet

Step 1. Configure the Multibus window

- ☑ Starting address = 1 0000 H
- ☑ Ending address = 0 FFFF H
- ☑ Size of Multibus window = 0 Kbytes
- ☑ Record the Multibus window on the Real Mode Memory Map
- ☑ Record the Multibus window jumpers on your jumper Worksheet

Step 2. Configure DRAM

- ☑ DRAM ending address = 0C FFFF H
- ☑ Size of DRAM = 832 Kbytes
- ☑ Record DRAM ending address on the Real Mode Memory Map
- ☑ Record DRAM ending address jumpers your Jumper Worksheet

Step 3. Configure EPROM

- ☑ Total EPROM size = 128 Kbytes
- ☑ EPROM starting address = 0E 0000 H
- ☑ Record the EPROM memory space on the Real Mode Memory Map
- ☑ Record the EPROM jumpers on your Jumper Worksheet

Step 4. Record the Multibus interface address space

- ☑ Size of Multibus interface address space = 64 Kbytes
- ☑ Record the Multibus interface address space on the Real Mode Memory Map

NOTES:

W-2153

Configuration Questions and Answers

Q. If I have a 4 Mbyte memory module, what happens to the memory above 0F FFFFH?

A. The Intel486 microprocessor cannot access any DRAM above 0F FFFFH while in real mode. You can, however, configure the entire 4 Mbytes of DRAM to make it accessible to other agents on the Multibus interface. To do this, set the dual-port ending address to 3F FFFFH. System memory configuration is discussed in Chapter 6.

Q. Is the dual-port DRAM overlaid by the EPROM address space accessible from the Multibus interface?

A. Yes. Any "dual-port" DRAM that is overlaid by the EPROM address space will be accessible from the Multibus interface. This portion of DRAM will not really be "dual-port" DRAM because the iSBC 486/12 CPU will not be able to access it. To the Multibus interface, however, this region is the same as any other part of memory.

Q. How can I initialize the DRAM that is overlaid by the EPROM?

A. If another Multibus agent is using this memory, have that agent initialize the memory.

Q. Can I use 27040 EPROMs (512K x 8)?

A. Yes, but you can use only 960 Kbytes of EPROM. In real mode, the minimum DRAM size is 64 Kbytes (between 0H and 0 FFFFH). The EPROM address space is therefore reduced to 960 Kbytes (1024 - 64 Kbytes). The starting address of EPROM will be 1 0000H. The 64 Kbyte DRAM area can be overlaid by a Multibus window if you want to access system memory resources. To use the entire 1 Mbyte of EPROM, you must switch to protected mode.

Q. Is the EPROM address space accessible from the Multibus interface?

A. No.

Q. How is the "Multibus interface address space" different from the "Multibus window"?

A. There are two differences: the position in the memory space and the priority in the address decode scheme. In real mode, the "Multibus interface address space" is always between the ending address of DRAM and the starting address of EPROM. The Multibus window, on the other hand, can start and end on any 64 Kbyte boundary. The second difference, priority of the memory resource in the address decode scheme, is important when two resources overlap. The Multibus interface address space has the lowest priority; the Multibus window has the highest priority. This means that the Multibus window has priority if it overlaps the EPROM or DRAM. Similarly, the Multibus interface address space has the lowest priority and can be completely masked by the EPROM and DRAM address spaces.

Q. Can I access system memory above 0F FFFFH by using the hardware page switching register like on the iSBC 386/12?

A. No. The iSBC 486/12 board does not have the Megabyte Page Register used on the iSBC 386/12. You can access the entire 16 Mbyte Multibus address space in protected mode.

***

Protected Mode Memory Configuration 5

Chapter Contents

Introduction 5-2
Related Information 5-2
New Terms in This Chapter 5-2
Configuration Procedure 5-2
Default Protected Mode Memory Map 5-3
Configuration Priority 5-5
Switching Between Real and Protected Modes 5-6
1st Priority: DRAM Configuration 5-6
DRAM Size 5-6
DRAM Starting and Ending Addresses (First 16 Mbyte Page) 5-6
DRAM Access from the Multibus Interface 5-7
DRAM Jumper Configuration 5-7
2nd Priority: EPROM Configuration 5-8
EPROM Size and Access Time 5-8
EPROM Address Range (PVAM) 5-9
EPROM Access from the Multibus Interface 5-9
EPROM Jumper Configuration 5-10
3rd Priority: Multibus Memory Space Configuration 5-10
Multibus Memory Space in the Third Gbyte Page 5-11
Multibus Memory Space in the First 16 Mbyte Page 5-12
Sample Worksheet 5-13
Configuration Questions and Answers 5-15

Introduction

This chapter discusses the protected mode memory configuration of the iSBC 486/12 board. In protected mode, also called Protected Virtual Address Mode or PVAM, the Intel486 CPU and iSBC 486/12 board can address memory up to 4 Gbytes. For protected mode operation, you have two configuration choices:

- you can set the DRAM ending address on any 64-Kbyte boundary in the first 16 Mbytes
- you can map the EPROM address space at the top of the 16th Mbyte page

Related Information

The following topics are related to protected mode memory configuration:

- for details of changing the board between real mode and protected mode, see Chapter 7, "I/O Subsystem Configuration"
- for information on protected mode initialization and programming, see the *Intel486™ Microprocessor Family Programmer's Reference Manual*

New Terms in This Chapter

protected virtual address mode — Refers to the mode of addressing for both the Intel486 microprocessor and the iSBC 486/12 board. The address range of the CPU and board is 4 Gbytes in protected mode.

Configuration Procedure

To configure your board for protected mode, do the following:

1. Complete the real mode memory configuration.
2. Copy the Protected Mode Memory Configuration Worksheets in Appendix A.
3. Use the copy of the Jumper Configuration Worksheet that you used in previous chapters.
4. Follow the worksheet and check-off each step. Pencil-in the jumpers on the jumper worksheet as you select each option. If you get lost, refer to the table of contents at the beginning of the chapter.
5. Go on to Chapter 6, "System Memory Configuration."

You may also want to look at:

- the sample PVAM worksheets starting on page 5-13
- PVAM configuration question and answers on page 5-15

Default Protected Mode Memory Map

The default configuration varies depending on the size of the memory module that you ordered with the board.

The default board in protected mode is configured as follows:

- 128 Kbytes of EPROM from 0 FFFE 0000H to 0 FFFF FFFFH (4th Gbyte page) and from 0FE 0000H to 0FF FFFFH (16th Mbyte page).
- DRAM starting address is always at 0H.
- DRAM ending address (first 16 Mbyte page only) is at base of Multibus interface address space. (The DRAM ending address does not affect the DRAM above 0FF FFFFH.)
- Multibus interface address space between the top of DRAM and the bottom of EPROM.

Size of Memory Module	Multibus Starting Address	Multibus Ending Address
2 Mbyte	20 0000H	0FD FFFFH
4 Mbyte	40 0000H	0FD FFFFH
8 Mbyte	80 0000H	0FD FFFFH
16 Mbyte	0F8 0000H	0FD FFFFH
32 Mbyte	0F8 0000H	0FD FFFFH

- The full 16 Mbyte Multibus address range in the 3rd Gbyte page, starting at 8000 0000H and repeating every 16 Mbytes to 0 BFFF FFFFH.

Figure 5-1 shows the default protected mode memory map.

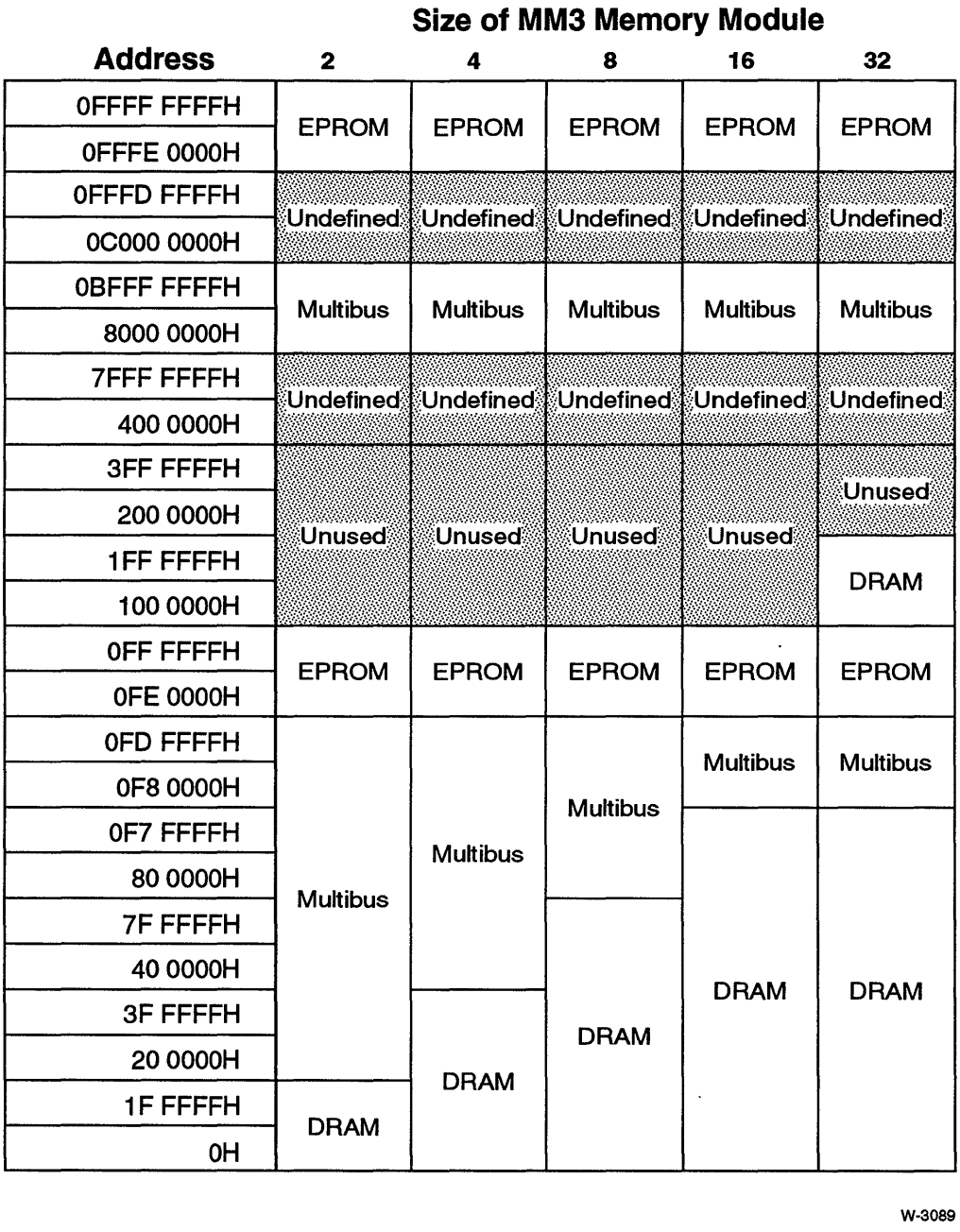


Figure 5-1. Default Protected Mode Memory Map

See Chapter 6, "System Memory Configuration," for a discussion of dual-port memory.

NOTES

1. The Multibus window option in real mode is not supported in protected mode.

2. With the exception of EPROM and dual-port memory, the real mode and protected mode memory configurations operate independently of each other. You must configure real mode memory space, even if you operate in protected mode. The board operates in real mode after power-up or reset.

Configuration Priority

In protected mode, if you overlap the address space configuration for the different types of memory resources, the board selects the memory resource with the higher priority. The priority scheme, from first to last, is as follows:

1st	highest	DRAM
2nd		EPROM
3rd	lowest	Multibus interface address space

Switching Between Real and Protected Modes

Both the Intel486 microprocessor and the iSBC 486/12 board have two modes of operation: real address mode, and protected virtual address mode. Both the CPU and the board should be switched to the same mode.

After a cold reset, both the CPU and the board are in real mode.

To switch the CPU between real and protected mode, you must change the PE bit in the CR0 register (a register inside the Intel486 CPU). You must also, in most cases, do some initialization. Refer to the *Intel486™ Microprocessor Family Programmer's Reference Manual* for information on switching the Intel486 microprocessor to protected mode.

To switch the board between real and protected mode, you must write to an I/O address. Refer to Chapter 7, "I/O Subsystem Configuration," for the details of changing the board between real mode and protected mode.

1st Priority: DRAM Configuration

DRAM Size

DRAM size is configured by installing one or more MM3 memory modules and setting the DRAM ending address. All boards must have at least one MM3 memory module. The address decode logic on the iSBC 486/12 board currently supports up to 64 Mbytes of DRAM.

The first 256 Mbytes of the 4 Gbyte memory space are allocated to on-board DRAM in the address decode logic. Currently, only the first 64 Mbytes are available. Future products may extend the amount of DRAM to the full 256 Mbytes as DRAM memory technology develops. Contact your local Intel sales office for information about the available sizes of MM3 memory modules.

DRAM Starting and Ending Addresses (First 16 Mbyte Page)

DRAM starting address is always at 0H.

You can set the DRAM ending address to any 64 Kbyte boundary in the first 16 Mbytes of address space (DRAM above the first 16 Mbytes is not affected).

CAUTION

Do not configure the DRAM ending address range to exceed the physical memory size of the MM3 memory modules you have installed.

The address range between the end of DRAM and either the start of EPROM (if configured for the 16th Mbyte page) or the top of the 16th Mbyte page, is mapped to the Multibus interface. This feature is used to support multiprocessing in some applications.

DRAM Access from the Multibus Interface

All dual-ported DRAM is accessible to other agents on the Multibus interface. Chapter 6 provides information on how to configure dual-port memory. The dual-port configuration is the same for both real mode and protected mode.

DRAM Jumper Configuration

Refer to Tables 5-1 and 5-2 to set the jumpers for the DRAM ending address in protected mode. These jumpers operate independently of the real mode DRAM ending address jumpers. Record your jumper settings on the Jumper Worksheet in Appendix A.

Table 5-1. DRAM Ending Address, 1 Mbyte Page

Ending Address	E095-E105	E115-E125	E135-E145	E155-E165
0X FFFFH	OUT	OUT	OUT	OUT
1X FFFFH[1]	OUT	OUT	OUT	IN
2X FFFFH	OUT	OUT	IN	OUT
3X FFFFH[2]	OUT	OUT	IN	IN
4X FFFFH	OUT	IN	OUT	OUT
5X FFFFH	OUT	IN	OUT	IN
6X FFFFH	OUT	IN	IN	OUT
7X FFFFH[3]	OUT	IN	IN	IN
8X FFFFH	IN	OUT	OUT	OUT
9X FFFFH	IN	OUT	OUT	IN
AX FFFFH	IN	OUT	IN	OUT
BX FFFFH	IN	OUT	IN	IN
CX FFFFH	IN	IN	OUT	OUT
DX FFFFH	IN	IN	OUT	IN
EX FFFFH	IN	IN	IN	OUT
FX FFFFH[4]	IN	IN	IN	IN

Notes:

"X" is determined by jumper configuration shown in Table 5-2.

[1] default (2 Mbyte memory module)

[2] default (4 Mbyte memory module)

[3] default (8 Mbyte memory module)

[4] default (16 or 32 Mbyte memory module)

Table 5-2. DRAM Ending Address, 64 Kbyte Page

Ending Address	E096-E106	E116-E126	E136-E146	E156-E166
X0 FFFFH	OUT	OUT	OUT	OUT
X1 FFFFH	OUT	OUT	OUT	IN
X2 FFFFH	OUT	OUT	IN	OUT
X3 FFFFH	OUT	OUT	IN	IN
X4 FFFFH	OUT	IN	OUT	OUT
X5 FFFFH	OUT	IN	OUT	IN
X6 FFFFH	OUT	IN	IN	OUT
X7 FFFFH[1]	OUT	IN	IN	IN
X8 FFFFH	IN	OUT	OUT	OUT
X9 FFFFH	IN	OUT	OUT	IN
XA FFFFH	IN	OUT	IN	OUT
XB FFFFH	IN	OUT	IN	IN
XC FFFFH	IN	IN	OUT	OUT
XD FFFFH	IN	IN	OUT	IN
XE FFFFH	IN	IN	IN	OUT
XF FFFFH[2]	IN	IN	IN	IN

Notes:

[1] default for baseboards with 16 or 32 Mbyte memory modules

[2] default for baseboards with 2, 4, or 8 Mbyte memory modules

"X" is determined by jumper configuration shown in Table 5-1.

2nd Priority: EPROM Configuration

EPROM Size and Access Time

The EPROM size configuration is discussed in Chapter 4, Real Mode Memory Configuration. The EPROM access time requirements are the same for real and protected modes. Refer to the Real Mode Configuration Worksheet for EPROM size and access time configuration.

Table 5-3 lists the starting address of EPROM in protected mode.

Table 5-3. EPROM Starting Address, PVAM

EPROM	Size (Kbytes)	Total Size (Kbytes)	Starting Address (4th Gbyte Page)	Starting Address (16 Mbyte Page)
27128	16	32	0 FFFF 8000H	0FF 8000H
27256	32	64	0 FFFF 0000H	0FF 0000H
27512§	64	128	0 FFFE 0000H	0FE 0000H
27010	128	256	0 FFFC 0000H	0FC 0000H
27020	256	512	0 FFF8 0000H	0F8 0000H
27040	512	1024	0 FFF0 0000H	0F0 0000H

Note: § default

NOTE

If you have selected the 27040 EPROMs (512K x 8), you can access the entire 1 Mbyte address space in protected mode. In real mode, the minimum DRAM size is 64 Kbytes so the usable EPROM address space is reduced to 960 Kbytes.

EPROM Address Range (PVAM)

In protected mode, the EPROM address range is always at the top of the 4 Gbyte address range. In addition, you can map the EPROM address space in the top Mbyte of the 16th Mbyte page by installing jumper E151-E161.

EPROM Access from the Multibus Interface

The EPROM address space is not accessible from the Multibus interface.

EPROM Jumper Configuration

Refer to the Chapter 4, "Real Mode Memory Configuration," for EPROM Size jumpers.

Refer to Table 5-4 and set the jumpers to configure the location of the EPROM address space in protected mode.

Table 5-4. EPROM Jumper Configuration, Protected Mode

Configuration	Jumper E151-E161
EPROM only at top of 4th Gbyte page	OUT
EPROM at 4th Gbyte and 16th Mbyte page	IN (default)

3rd Priority: Multibus Memory Space Configuration

In protected mode, the Multibus memory address space is mapped to two address ranges:

- the address range starting at 8000 0000H and repeating every 16 Mbytes to 0 BFFF FFFFH (this provides access to the full 16 Mbyte Multibus address space regardless of how the first 16 Mbyte page is configured)
- the address range between the end of DRAM and either the start of EPROM (if configured for the 16th Mbyte page), or the top of the 16th Mbyte page (0FF FFFFH)

You can use the Multibus interface address space and the dual-port memory to configure a multiprocessing environment for your system. Refer to Chapter 6, "System Memory Configuration," for multiprocessing details.

Multibus Memory Space in the Third Gbyte Page

The 16 Mbyte Multibus address space is also mapped to the address range starting at 8000 0000H and repeating every 16 Mbytes to 0 BFFF FFFFH. This address range provides access to the full 16 Mbyte Multibus address space. Table 5-5 lists the address mapping between the physical address from the iSBC 486/12 board to the Multibus interface.

Table 5-5. iSBC® 486/12 Board to Multibus Address Mapping

iSBC 486/12 Address	Multibus Address	Address Offset
8000 0000H 80FF FFFFH	0H 0FF FFFFH	8000 0000H
8100 0000H 81FF FFFFH	0H 0FF FFFFH	8100 0000H
...	...	...
xx00 0000H xxFF FFFFH	0H 0FF FFFFH	xx00 0000H
...	...	...
0 BF00 0000H 0 BFFF FFFFH	0H 0FF FFFFH	0 BF00 0000H

Note:
xx = two hexadecimal digits from 80H to 0BFH

Figure 5-2 shows the Multibus address space in the 3rd Gbyte page of the physical address space of the iSBC 486/12 board.

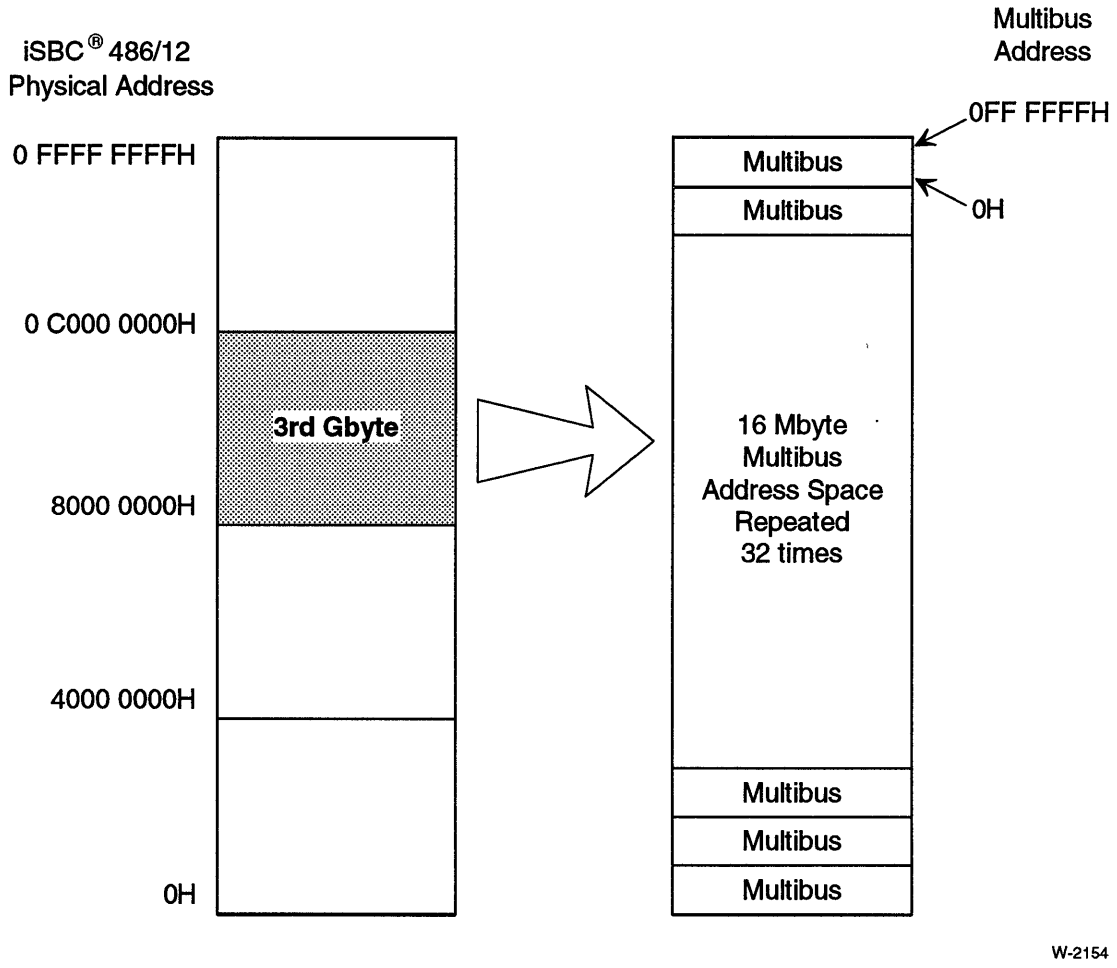


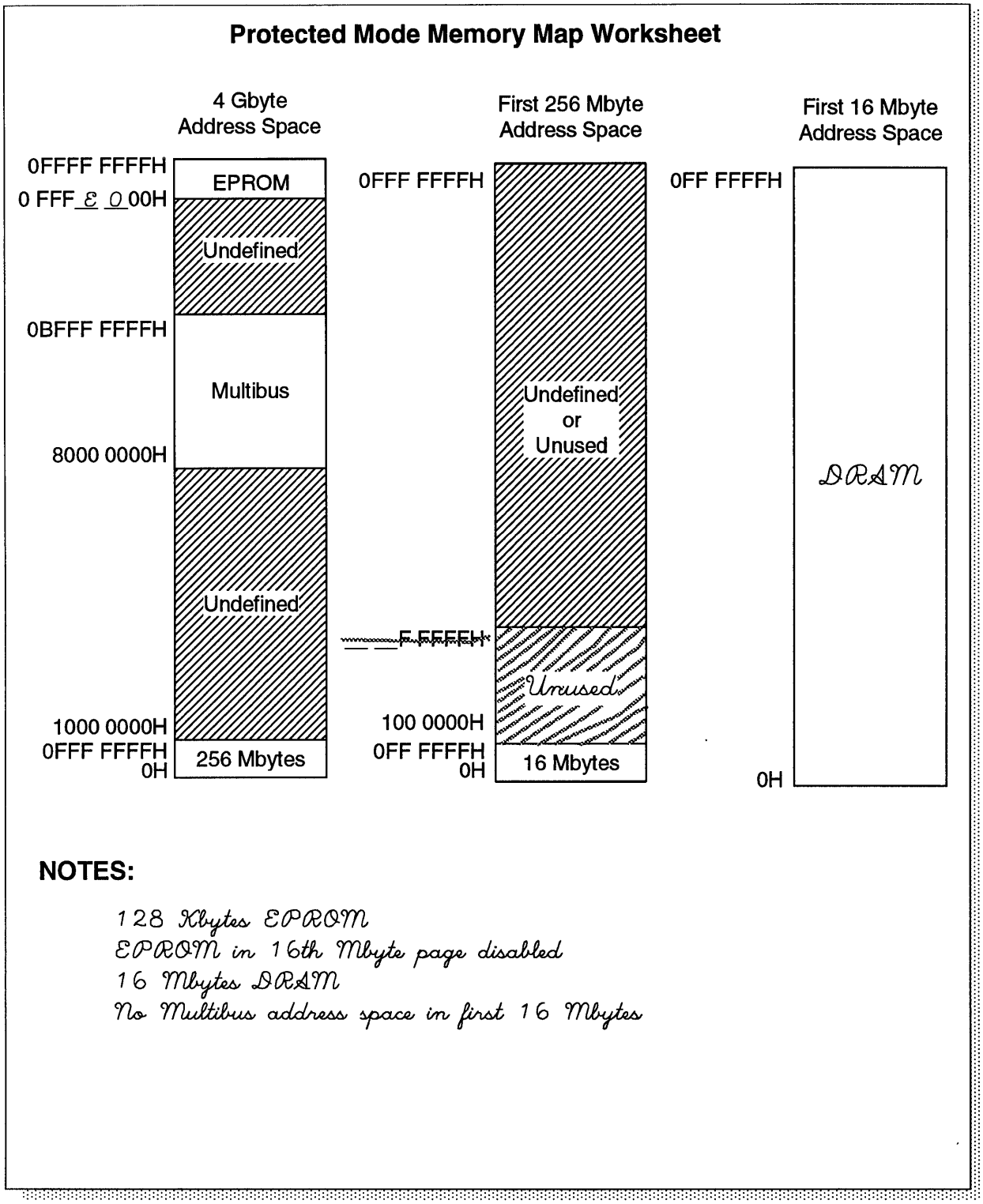
Figure 5-2. Multibus Address Space in the 4 Gbyte Address Space

Multibus Memory Space in the First 16 Mbyte Page

The address range between the end of DRAM and either the start of EPROM (if configured for the 16th Mbyte page) or the top of the 16th Mbyte page (0FF FFFFH) is mapped to the Multibus interface. The address mapping is one-to-one. Set the end of DRAM to adjust the amount of Multibus address space.

In protected mode, the iSBC 486/12 board cannot use the Multibus window you configured in real mode. The Multibus window jumpers are ignored in protected mode.

Sample Worksheet



Protected Mode Worksheet

Step 1. Select the DRAM ending address

- ☑ Total size of DRAM = 16 Mbytes 0FF FFFFH bytes
- ☑ DRAM ending address = 0FF FFFFH
- ☑ Record DRAM on memory map
- ☑ Record DRAM ending address jumpers on Jumper Worksheet

Step 2. EPROM configuration

- ☑ EPROM size = 128 Kbytes 1 FFFF H bytes
- ☑ EPROM starting address in 4th Gbyte = 0FFFE 0000 H
- ☑ EPROM enabled at top of 16th Mbyte ( ) Yes (☑) No
- ☑ EPROM starting address in 16th Mbyte = _____ (☑) Not Enabled
- ☑ Record EPROM address space on memory map worksheet
- ☑ Record EPROM jumpers on your Jumper Worksheet

Step 3. Record Multibus address space

- ☑ Record the Multibus address space on the memory map worksheet

W-2160

Configuration Questions and Answers

Q. Why is there an option to configure the EPROM address space at the top of the 16th Mbyte page?

A. This option is useful to maintain compatibility with software written for the Intel 80286 microprocessor.

If you have 16 Mbytes (or more) of DRAM, configuring the EPROM address space in the 16th Mbyte page will overlap DRAM. To access the EPROM, you must adjust the DRAM ending address equal to, or less than, the EPROM starting address.

Q. If I have 32 Mbytes of DRAM, what happens to the memory above 16 Mbytes when I set the ending DRAM address?

A. DRAM from 100 0000H and above is always accessible to the CPU in protected mode: it is not affected by the DRAM ending address jumpers. The function of the DRAM ending address jumpers is to allow access to the Multibus address space in the first 16 Mbyte page of the iSBC 486/12 address space. This is useful for compatibility with the Intel 80286 microprocessor. Because the entire 16 Mbyte Multibus address space is also mapped into the 3rd Gbyte page, you generally don't need to create a Multibus address space in the first 16 Mbyte page.

Q. Is the DRAM ending address in PVAM the same as the ending address in real mode?

A. No. The two ending addresses are configured independently.

Q. Do I need to change the configuration when I add a second MM3 memory module?

A. You should check the DRAM ending address and reset it to a higher value. You should also check to determine whether or not you need to disable the EPROM address space in the 16th Mbyte page.

Q. If the addresses mapped to the EPROMs repeat in the forth Gbyte page, can I access EPROM anywhere in the forth Gbyte?

A. For software compatibility with future products, we recommend using only the EPROM address space that ends at 0 FFFF FFFFH. Having the EPROM address space repeat in the fourth Gbyte page is a by-product of the address decode logic and not a required feature of the iSBC 486/12. Intel, therefore, does not guarantee that future versions of the iSBC 486/12 family will have this feature.

Q. How is the protected mode memory configuration of the iSBC 486/12 different from the iSBC 386/12?

A. In many cases the iSBC 386/12 and iSBC 486/12 board's protected mode memory maps can be configured identically. There are a few differences between the models that you should keep in mind if you want both boards to have identical memory maps:

- The iSBC 486/12 board does not have an iLBX™ interface to configure.
- You cannot configure the starting address of DRAM on the iSBC 486/12.
- On the iSBC 486/12 board, the memory map of the first 64 Mbytes is *not* repeated above 3FFF FFFFH.
- On the iSBC 486/12 board, the 16 Mbyte Multibus address space is accessible in the third Gbyte page.
- The iSBC 486/12 board can have up to 64 Mbytes of DRAM.
- The iSBC 486/12 board does *not* support 2764 (8K x 8) EPROMs.
- On the iSBC 486/12 board, the first 16 Mbytes of DRAM does *not* repeat between 100 0000H and 3FF FFFFH.

For iSBC 386/12 configuration information, refer to the *iSBC® 386/12S Single Board Computer User's Manual*, Intel order number 459913.

***

System Memory Configuration 6

Chapter Contents

Introduction 6-2
Related Information 6-2
New Terms in this Chapter 6-2
Configuration Procedure 6-3
Default System Memory Map 6-4
Configuration Priority 6-5
Dual-Port Configuration 6-5
Overview 6-5
Dual-Port 16 Mbyte Page Register 6-5
Dual-Port Starting and Ending Addresses 6-7
Dual-Port Alias 6-8
Starting and Ending Addresses Configuration 6-11
Dual-Port Alias Configuration 6-15
Dual-Port Alias Example 6-16
Dual-Port Alias Jumpers 6-19
Dual-Port 16 Mbyte Page Register 6-19
Dual-Port Lock 6-20
Real Mode Multiprocessing Configuration 6-21
Real Mode Multiprocessing Configuration Options 6-21
Real Mode Multiprocessing Examples 6-21
Protected Mode Multiprocessing Configuration 6-25
Protected Mode Multiprocessing Configuration Options 6-25
Protected Mode Multiprocessing Example 6-25
Sample Worksheets 6-28
Configuration Question and Answers 6-30

Introduction

System memory is memory that is accessible from the Multibus interface. You can configure part or all of the iSBC 486/12 board's dual-port memory to be accessible to the Multibus interface. This chapter discusses the configuration of the dual-port memory on the iSBC 486/12 board and provides examples of how dual-port memory can be configured to support multiprocessing.

When configuring dual-port memory, you have the following options:

- starting and ending address of dual-port memory, as seen from the Multibus interface
- the dual-port alias (an address offset that is added to the Multibus address to calculate the physical address in DRAM)
- an I/O register that is used to select one of the 16 Mbyte pages in the DRAM address space

Related Information

The following topics are related to system memory configuration:

- real mode memory configuration, see Chapter 4
- protected mode memory configuration, see Chapter 5

New Terms in this Chapter

dual-port memory — Memory that can be accessed both from the local CPU and from other agents on the Multibus interface.

dual-port alias — An address offset that is added to the Multibus address to give the physical address of dual-port DRAM.

dual-port lock — A control bit in I/O space that locks the dual-port DRAM to the local bus.

local DRAM — On-board DRAM that is not configured as dual-port DRAM.

multiprocessing — Two or more processors working together across the Multibus interface. Data and control information can be exchanged through dual-port memory.

system memory — System memory is memory that is accessible from the Multibus interface.

Configuration Procedure

To configure your system memory, we recommend you do the following:

1. Complete the real mode and PVAM memory configuration.
2. Copy the System Memory Configuration Worksheets in Appendix A.
3. Use the copy of the Jumper Configuration Worksheet that you used in previous chapters.
4. Follow the worksheet and check-off each step. Pencil-in the jumpers on the jumper worksheet as you select each option. If you get lost, refer to the table of contents at the beginning of the chapter.
5. Go on to Chapter 7, "I/O Subsystem Configuration."

You may also want to look at:

- the real mode and PVAM multiprocessing examples starting on page 6-21
- the sample system memory worksheets starting on page 6-28
- system memory configuration question and answers on page 6-30

Default System Memory Map

The default system memory configuration is as follows:

- dual-port starting address is 0H
- dual-port ending address in the first 16 Mbyte page is equal to:
  - 1F FFFFH for 2 Mbytes of DRAM
  - 3F FFFFH for 4 Mbytes of DRAM
  - 7F FFFFH for 8 Mbytes of DRAM
  - 0F7 FFFFH for 16 Mbytes of DRAM
  - 0F7 FFFFH for 32 Mbytes of DRAM
- dual-port alias is 0H
- the Dual-Port 16 Mbyte Page Register is set to 0H

Figure 6-1 shows the default system memory map for the iSBC 486/12 board with 2, 4, 8, 16, and 32 Mbytes of DRAM.

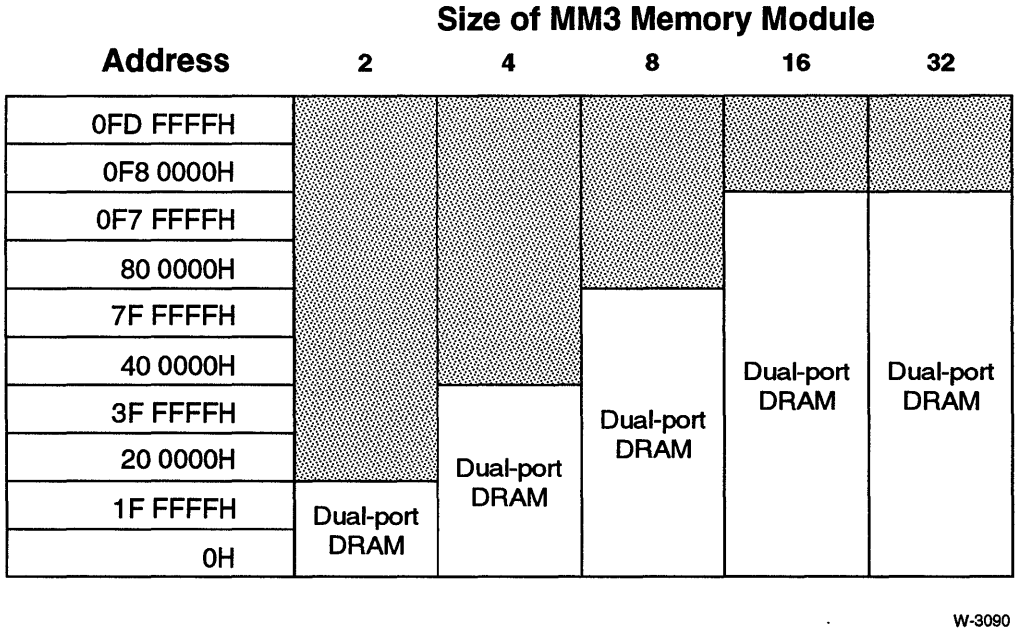


Figure 6-1. Default System Memory Map

Configuration Priority

The dual-port memory configuration is independent of all other configuration options. Dual-port memory operation is not affected by the mode (real mode or protected mode) in which the board is operating.

Dual-Port Configuration

Overview

Dual-port memory is memory that is accessible from both the local CPU and the Multibus interface. It is called system memory because it is a memory resource on the Multibus interface. The configuration of the dual-port memory is independent of whether the iSBC 486/12 board is in real or protected mode.

Dual-port memory is configured by four parameters:

- dual-port starting address
- dual-port ending address
- dual-port alias
- 16 Mbyte Page Register value

By adjusting these parameters, you can map a range of addresses from the 16 Mbyte Multibus space to a range of DRAM addresses in the first 256 Mbytes of the CPU's address space. (Currently, only the first 64 Mbytes of the 256 Mbyte address space supports DRAM, the remaining memory is reserved for future expansion.)

To understand how to configure the iSBC 486/12 dual-port DRAM, you need to understand how the four parameters mentioned above interact. The next four sections will explain these concepts.

Dual-Port 16 Mbyte Page Register

The Dual-Port 16 Mbyte Page Register (I/O address 0EEH) selects which 16 Mbyte page in physical memory will be used for dual-port accesses. When programmed with the correct value, this register allows the entire DRAM address space on the iSBC 486/12 board to be configured as dual-port memory. Because the on-board I/O is not accessible from the Multibus interface, the value of this register can only be changed by the iSBC 486/12 CPU or ADMA. Figure 6-2 shows the iSBC 486/12 address space and the pages selected by the Dual-Port 16 Mbyte Page Register.

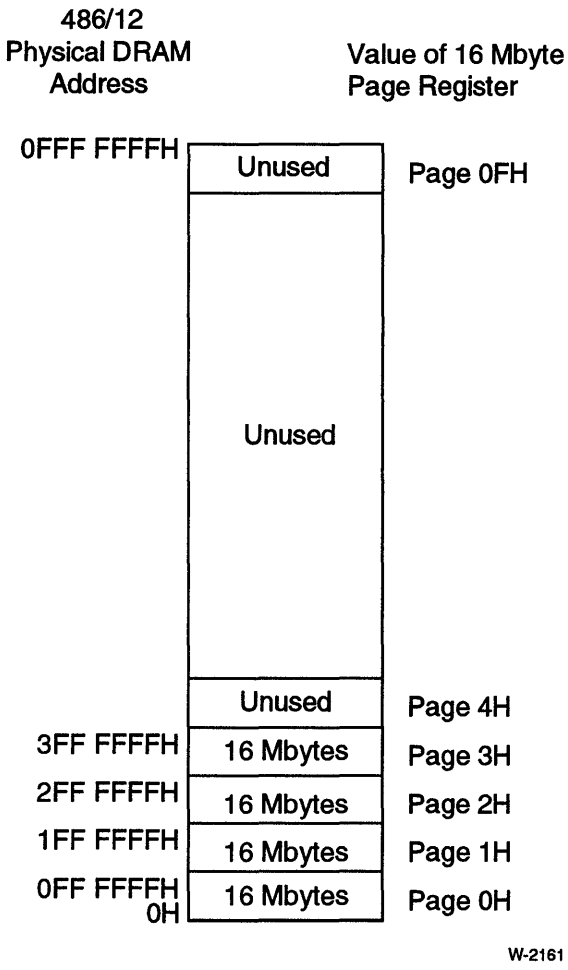


Figure 6-2. Using the Dual-Port 16 Mbyte Page Register

Each page is affected by the dual-port starting and ending addresses in the same way. For example, if the starting and ending dual-port addresses are configured as follows:

- total DRAM = 64 Mbytes
- dual-port starting address = 0H
- dual-port ending address = 0EF FFFFH

Then the first 15 Mbytes of each 16 Mbyte page will be dual-port DRAM. The resulting configuration is:

- total dual-port DRAM = 60 Mbytes
- page 0 dual-port DRAM from 0H to 0EF FFFFH
- page 1 dual-port DRAM from 100 0000H to 1EF FFFFH
- page 2 dual-port DRAM from 200 0000H to 2EF FFFFH
- page 3 dual-port DRAM from 300 0000H to 3EF FFFFH

Dual-Port Starting and Ending Addresses

The dual-port starting and ending addresses are set by jumpers. These jumpers specify the starting and ending addresses of dual-port memory *as viewed from the Multibus interface.* Both the starting and ending address can be set on any 64 Kbyte boundary over the entire 16 Mbyte Multibus address space. Figure 6-3 shows the dual-port starting and ending addresses and their relationship to the 16 Mbyte Multibus address space.

Dual-port memory is disabled by setting the starting address greater than the ending address.

The starting and ending addresses are relative to the Multibus address space. These addresses are the same for each page selected by the Dual Port 16 Mbyte Page Register.

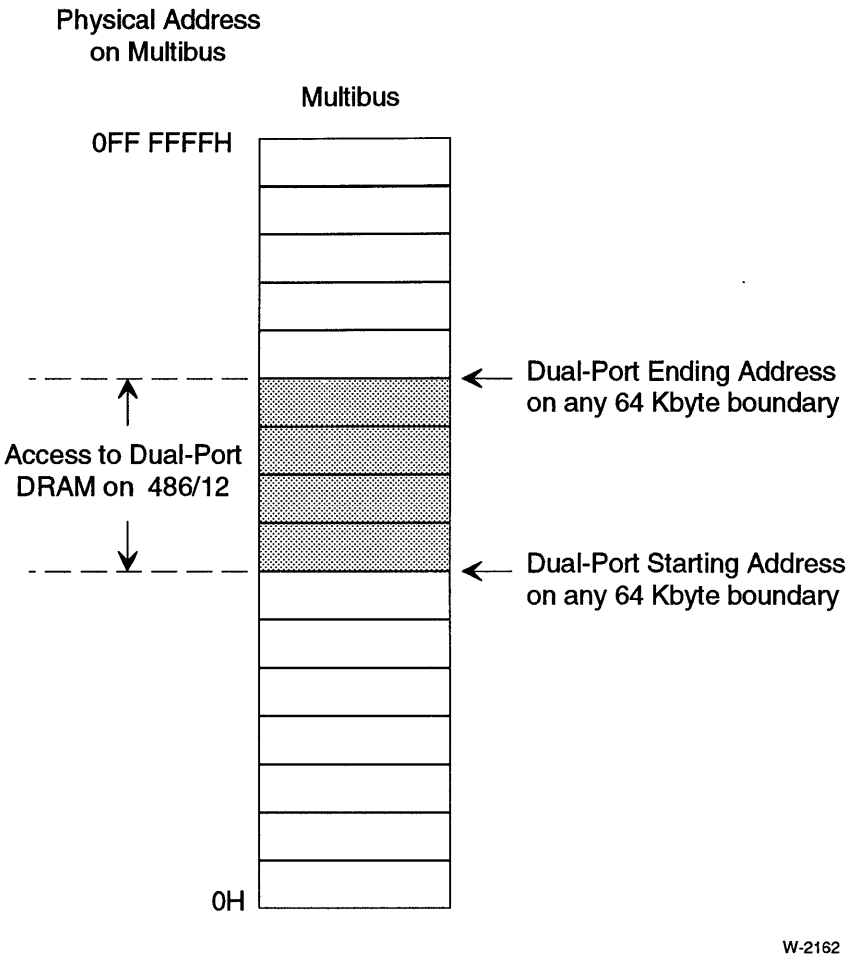


Figure 6-3. Using the Dual-Port Starting and Ending Addresses

Dual-Port Alias

The dual-port alias is an offset that is added to the Multibus address to give the physical address of dual-port DRAM (relative to the starting address of the page selected by the Dual-Port 16 Mbyte Page Register). Figure 6-4 shows how the dual-port alias is used. If the resulting address would be greater than 0FF FFFFH, the address is "wrapped around" as shown in Figure 6-5.

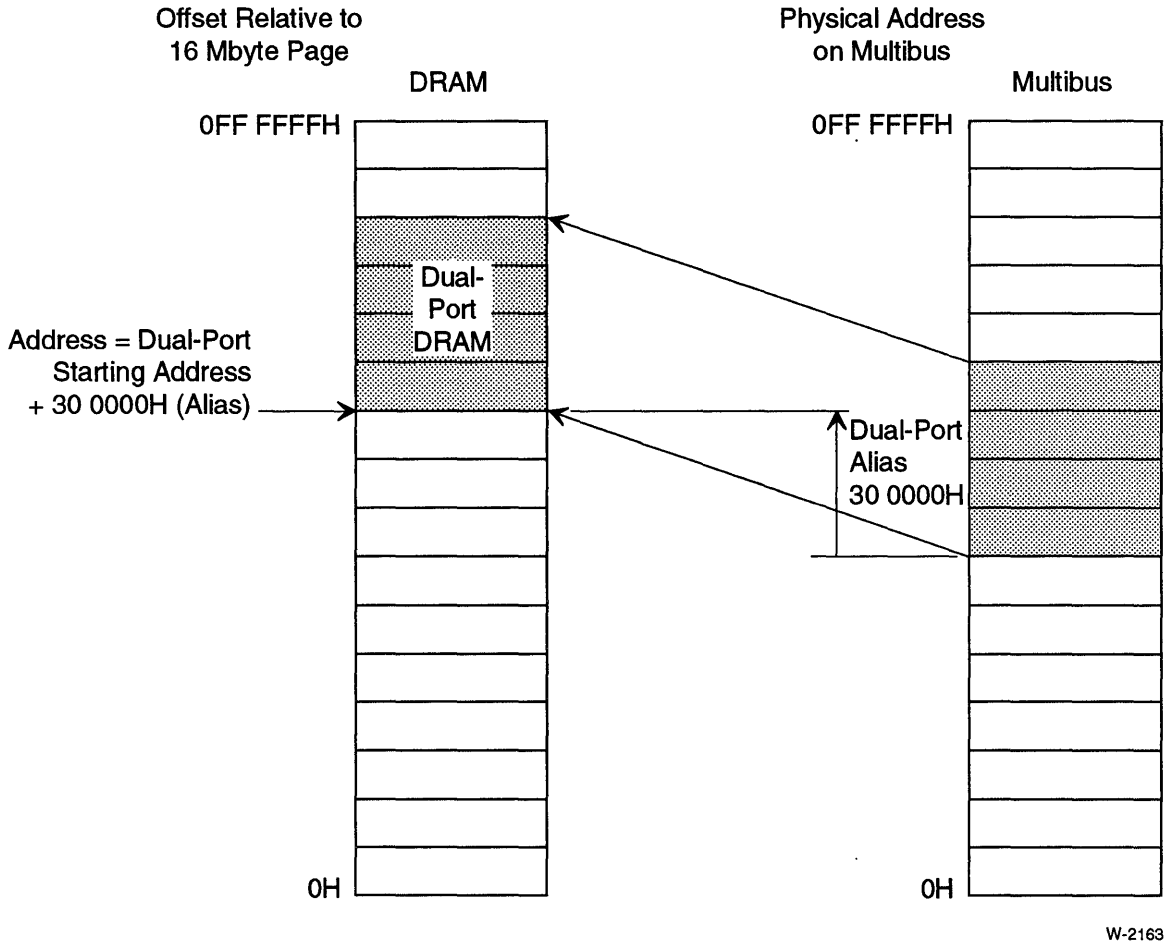


Figure 6-4. Using the Dual-Port Alias

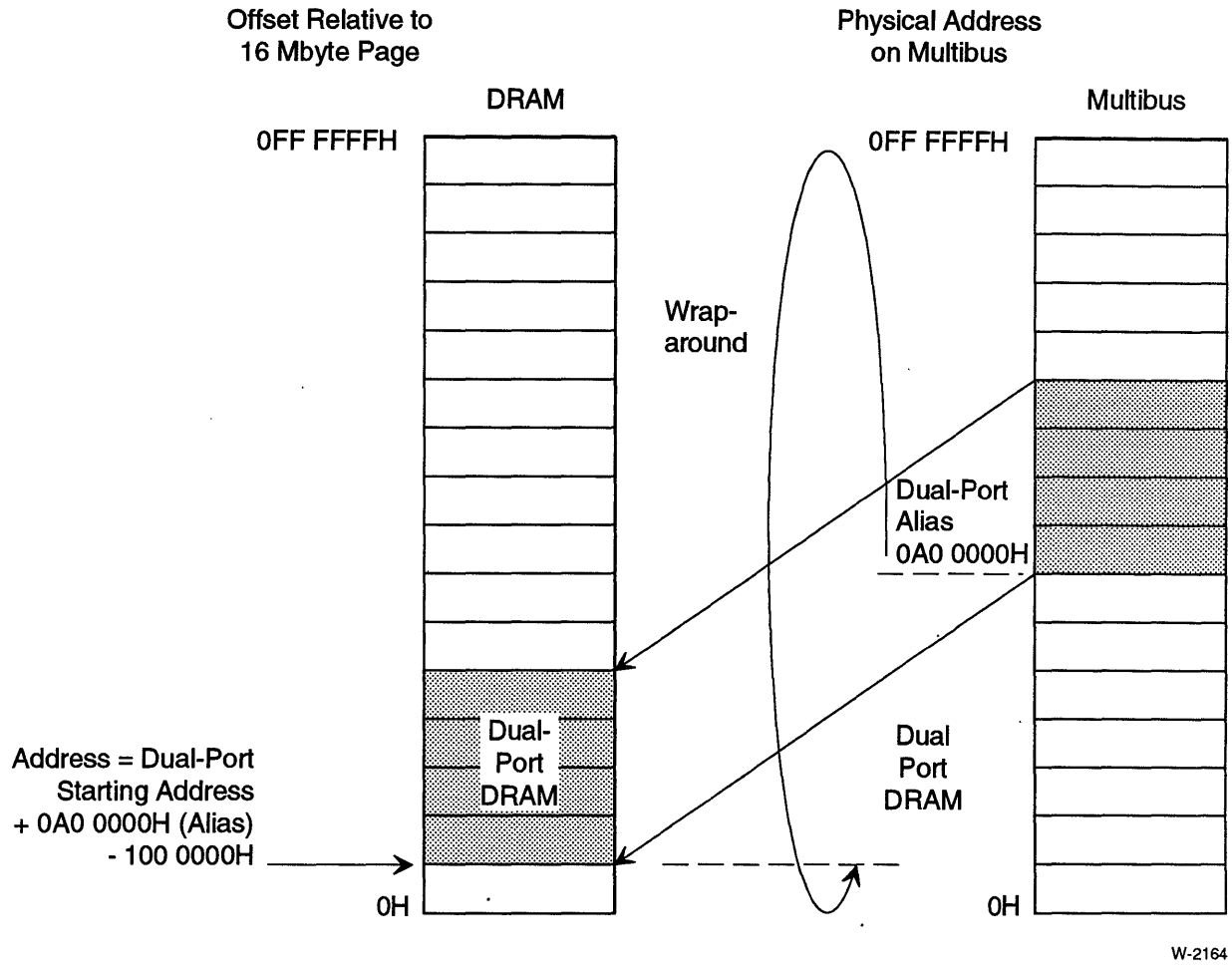


Figure 6-5. Dual-Port Alias Wrap-Around at the top of the 16 Mbyte Page

Starting and Ending Addresses Configuration

The dual-port starting and ending addresses are set using two sets of jumpers. You can configure the starting and ending addresses on any 64 Kbyte boundary in the 16 Mbyte Multibus address space. Select the starting and ending addresses from the viewpoint of the Multibus agent.

NOTE

To disable dual-port memory, set the starting address higher than the ending address.

For reliable operation, keep the following restrictions in mind when you configure dual-port memory:

- The size of the dual-port memory should be less than, or equal to, the amount of DRAM on the iSBC 486/12 board.
- Avoid having the dual-port memory overlap the Multibus interface address space. (The Multibus interface address space is the space between the end of DRAM and either the start of EPROM or 0FF FFFFH if EPROM is not configured in the 16th Mbyte page in protected mode.)
- Dual-port memory overlaid by the EPROM address space can only be accessed by other Multibus agents. Because the CPU is configured to access the EPROM, this DRAM will only be accessible to other Multibus agents.

The dual-port starting and ending jumpers are listed in the following tables:

- Table 6-1 Dual-Port Starting Address (Mbyte Boundary)
- Table 6-2 Dual-Port Starting Address (64 Kbyte Boundary)
- Table 6-3 Dual-Port Ending Address, (Mbyte Boundary)
- Table 6-4 Dual-Port Ending Address (64 Kbyte Boundary)

Table 6-1. Dual-Port Starting Address (Mbyte Boundary)

Starting Address	E097-E107	E117-E127	E137-E147	E157-E167
0X 0000H §	OUT	OUT	OUT	OUT
1X 0000H	OUT	OUT	OUT	IN
2X 0000H	OUT	OUT	IN	OUT
3X 0000H	OUT	OUT	IN	IN
4X 0000H	OUT	IN	OUT	OUT
5X 0000H	OUT	IN	OUT	IN
6X 0000H	OUT	IN	IN	OUT
7X 0000H	OUT	IN	IN	IN
8X 0000H	IN	OUT	OUT	OUT
9X 0000H	IN	OUT	OUT	IN
0AX 0000H	IN	OUT	IN	OUT
0BX 0000H	IN	OUT	IN	IN
0CX 0000H	IN	IN	OUT	OUT
0DX 0000H	IN	IN	OUT	IN
0EX 0000H	IN	IN	IN	OUT
0FX 0000H	IN	IN	IN	IN

Notes:
§ default

"X" is determined by Table 6-2.

Table 6-2. Dual-Port Starting Address (64 Kbyte Boundary)

Starting Address	E098-E108	E118-E128	E138-E148	E158-E168
X0 0000H §	OUT	OUT	OUT	OUT
X1 0000H	OUT	OUT	OUT	IN
X2 0000H	OUT	OUT	IN	OUT
X3 0000H	OUT	OUT	IN	IN
X4 0000H	OUT	IN	OUT	OUT
X5 0000H	OUT	IN	OUT	IN
X6 0000H	OUT	IN	IN	OUT
X7 0000H	OUT	IN	IN	IN
X8 0000H	IN	OUT	OUT	OUT
X9 0000H	IN	OUT	OUT	IN
XA 0000H	IN	OUT	IN	OUT
XB 0000H	IN	OUT	IN	IN
XC 0000H	IN	IN	OUT	OUT
XD 0000H	IN	IN	OUT	IN
XE 0000H	IN	IN	IN	OUT
XF 0000H	IN	IN	IN	IN

Notes:
§ default

"X" is determined by Table 6-1.

Table 6-3. Dual-Port Ending Address (Mbyte Boundary)

Ending Address	E099-E109	E119-E129	E139-E149	E159-E169
0X FFFFH	OUT	OUT	OUT	OUT
1X FFFFH[1]	OUT	OUT	OUT	IN
2X FFFFH	OUT	OUT	IN	OUT
3X FFFFH[2]	OUT	OUT	IN	IN
4X FFFFH	OUT	IN	OUT	OUT
5X FFFFH	OUT	IN	OUT	IN
6X FFFFH	OUT	IN	IN	OUT
7X FFFFH[3]	OUT	IN	IN	IN
8X FFFFH	IN	OUT	OUT	OUT
9X FFFFH	IN	OUT	OUT	IN
0AX FFFFH	IN	OUT	IN	OUT
0BX FFFFH	IN	OUT	IN	IN
0CX FFFFH	IN	IN	OUT	OUT
0DX FFFFH	IN	IN	OUT	IN
0EX FFFFH	IN	IN	IN	OUT
0FX FFFFH[4]	IN	IN	IN	IN

Notes:
"X" is determined by Table 6-4.

[1] default (with a 2 Mbyte memory module)

[2] default (with a 4 Mbyte memory module)

[3] default (with a 8 Mbyte memory module)

[4] default (with a 16 or 32 Mbyte memory module)

Table 6-4. Dual-Port Ending Address (64 Kbyte Boundary)

Ending Address	E100-E110	E120-E130	E140-E150	E160-E170
X0 FFFFH	OUT	OUT	OUT	OUT
X1 FFFFH	OUT	OUT	OUT	IN
X2 FFFFH	OUT	OUT	IN	OUT
X3 FFFFH	OUT	OUT	IN	IN
X4 FFFFH	OUT	IN	OUT	OUT
X5 FFFFH	OUT	IN	OUT	IN
X6 FFFFH	OUT	IN	IN	OUT
X7 FFFFH[1]	OUT	IN	IN	IN
X8 FFFFH	IN	OUT	OUT	OUT
X9 FFFFH	IN	OUT	OUT	IN
XA FFFFH	IN	OUT	IN	OUT
XB FFFFH	IN	OUT	IN	IN
XC FFFFH	IN	IN	OUT	OUT
XD FFFFH	IN	IN	OUT	IN
XE FFFFH	IN	IN	IN	OUT
XF FFFFH[2]	IN	IN	IN	IN

Notes:
[1] default (with a 16 or 32 Mbyte memory module)

[2] default (with a 2, 4, or 8 Mbyte memory module)

"X" is determined by Table 6-3.

Dual-Port Alias Configuration

Dual-port aliasing enables an off-board agent to access dual-port DRAM at a different address than the iSBC 486/12 board. The alias is added to the Multibus address to determine the on-board DRAM address (see Figure 6-4). If the aliased address is greater than 0FF FFFFH, the address wraps around in the 16 Mbyte address range. The dual-port alias can be set in 1 Mbyte increments. Dual-port address aliasing is set to zero (disabled) in the default configuration.

Dual-port aliasing is used to support multiprocessor systems. The advantage of aliasing is that although DRAM may be located at the same addresses on different boards, the software does not need to be changed to align the Multibus interface address space of one board with the dual-port memory of another. All boards in the system can have exactly the same memory maps and software. Each board will, of course, need different dual-port aliases.

NOTE

For multiprocessing applications, aliasing can only be used in protected mode because the minimum alias is 1 Mbyte which would offset the dual-port memory beyond the address range of the iSBC 486/12 CPU when it is in real mode. Aliasing can, however, be used in real mode to allow other boards in the system to access dual-port memory at different addresses.

Dual-Port Alias Example

The following example shows a typical multiprocessing application using three iSBC 486/12 boards. The boards exchange data and control information though dual-port memory. This example shows how you can use dual-port aliasing to configure all three boards with the same memory maps, while allowing each board to access 1 Mbyte of dual-port memory on each of the other boards. Table 6-5 gives the configuration of each board.

Table 6-5. Dual-Port Alias Example

Parameter	Board 1	Board 2	Board 3
Board type	486/12	486/12	486/12
On-board DRAM	2 Mbytes	2 Mbytes	2 Mbytes
Dual-port starting address	20 0000H	30 0000H	40 0000H
Dual-port ending address	2F FFFFH	3F FFFFH	4F FFFFH
Alias offset	0F0 0000H	0E0 0000H	0D0 0000H
Local DRAM starting address	0H	0H	0H
Local DRAM ending address	0F FFFFH	0F FFFFH	0F FFFFH
Physical starting address of dual-port DRAM	10 0000H	10 0000H	10 0000H
Physical ending address of dual-port DRAM	1F FFFFH	1F FFFFH	1F FFFFH
Multibus address space mapped to board 1	not applicable	20 0000H 2F FFFFH	20 0000H 2F FFFFH
Multibus address space mapped to board 2	30 0000H 3F FFFFH	not applicable	30 0000H 3F FFFFH
Multibus address space mapped to board 3	40 0000H 4F FFFFH	40 0000H 4F FFFFH	not applicable

To illustrate how the dual-port alias is used, let's look at two of the boards in this example. If board 1 wants to pass data to board 2, it does so either by writing to board 2's dual-port memory, or by writing to its own dual-port memory.

To write to board 2's dual-port memory, board 1 writes to addresses between 30 0000H and 3F FFFFH. This address range falls between board 2's dual-port starting and ending addresses. Board 2 then adds the dual-port alias of 0E0 0000H to the Multibus address to find the physical address of DRAM on board 2. For example, if board 1 writes to 3A 0000H, the resulting DRAM address is:

3A 0000H + 0E0 0000H - 100 0000H = 1A 0000H

Note that the address wraps around in this example (that is why 100 0000H was subtracted).

Figure 6-6 shows the memory maps of the three boards.

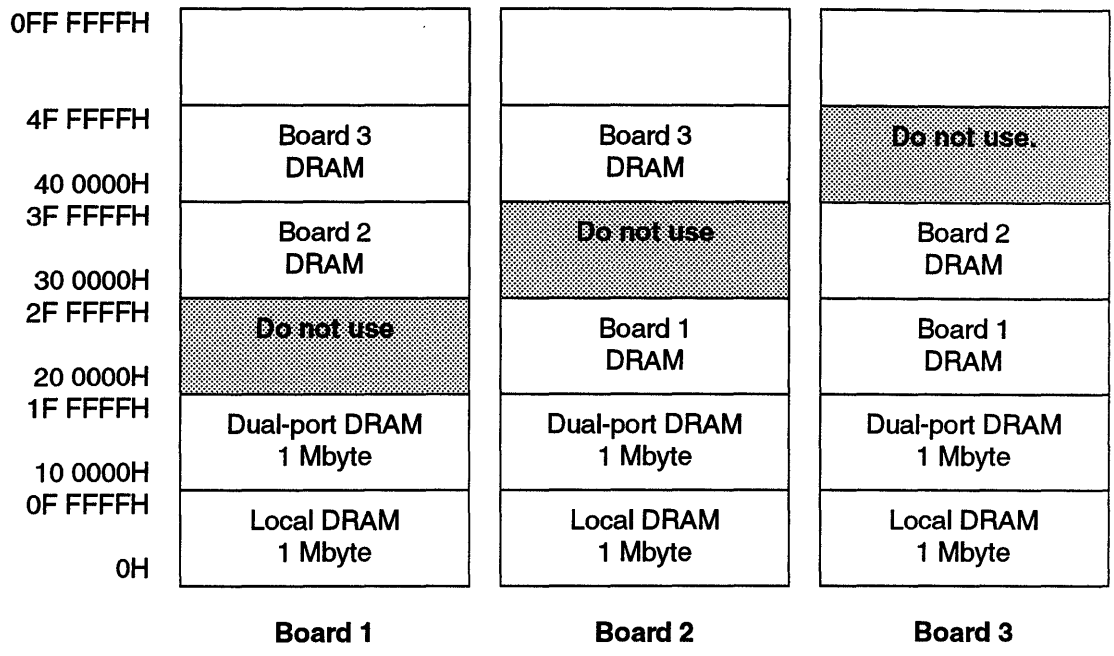


Figure 6-6. Aliasing Example

NOTE

The areas marked "Do not use" in Figure 6-6 are the regions in the Multibus address space that are mapped to the boards' own dual-port DRAM. The iSBC 486/12 board cannot go through the Multibus interface to access its own dual-port DRAM.

CAUTION

Accesses from a board to an area marked "Do not use" in its own address space is not allowed. For example, board number 1 in the preceding example must avoid reading from or writing to addresses between 20 0000H and 2F FFFFH because this region on the Multibus address space is mapped to its own dual-port DRAM.

Dual-Port Alias Jumpers

Table 6-6. Dual-Port Alias Jumpers

Offset	E235-E240	E245-E250	E255-E260	E265-E270
0H §	OUT	OUT	OUT	OUT
10 0000H	OUT	OUT	OUT	IN
20 0000H	OUT	OUT	IN	OUT
30 0000H	OUT	OUT	IN	IN
40 0000H	OUT	IN	OUT	OUT
50 0000H	OUT	IN	OUT	IN
60 0000H	OUT	IN	IN	OUT
70 0000H	OUT	IN	IN	IN
80 0000H	IN	OUT	OUT	OUT
90 0000H	IN	OUT	OUT	IN
0A0 0000H	IN	OUT	IN	OUT
0B0 0000H	IN	OUT	IN	IN
0C0 0000H	IN	IN	OUT	OUT
0D0 0000H	IN	IN	OUT	IN
0E0 0000H	IN	IN	IN	OUT
0F0 0000H	IN	IN	IN	IN

Note:
§ default

Dual-Port 16 Mbyte Page Register

The Dual-Port 16 Mbyte Page Register selects which 16 Mbyte page in physical memory will be used for dual-port accesses. The Dual-Port 16 Mbyte Page Register is byte-wide, write-only register at I/O address 0EEH.

The power-up and hardware reset value is 0H.

Table 6-7 shows the bit definitions of the Dual-Port 16 Mbyte Page Register.

Table 6-7. Dual-Port 16 Mbyte Page Register

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0				DP_PAGE			

bits 7..4 Unused

bits 3..0 DP_PAGE is a hexadecimal digit from 0H to 3H (values from 4H to 0FH are reserved)

Table 6-8. Starting and Ending Addresses of Dual-Port Pages

DP_Page	Physical DRAM Starting Address of Dual-Port Page	Physical DRAM Ending Address of Dual-Port Page
0H§	0H	0FF FFFFH
1H	100 0000H	1FF FFFFH
2H	200 0000H	2FF FFFFH
3H	300 0000H	3FF FFFFH
4H..0FH	Reserved	Reserved

Note:
§ default (power-up or hardware reset value)

Dual-Port Lock

Dual-port memory can be "locked" to the local bus by clearing bit 1, port C, on the 82C55A PPI. The dual-port memory remains locked until this bit is set.

NOTE

To prevent a deadlock situation, the board surrenders the dual-port memory to a requesting master if the request coincides with an on-board Multibus interface request, *regardless of dual-port lock.* The iSBC 486/12 board's Multibus access is not completed until the off-board master has completed or aborted its cycle.

Dual-port memory is granted to off-board Multibus agents under the following conditions:

- dual-port memory is not locked
- the iSBC 486/12 board starts a Multibus interface cycle simultaneously with the dual-port request from the off-board master, regardless of dual-port lock

The dual-port lock operates independently of the Multibus lock signal. The iSBC 486/12 CPU can assert the Multibus lock signal to lock off-board memory resources during Multibus accesses. Off-board masters can assert the Multibus lock signal to prevent the iSBC 486/12 CPU or ADMA from accessing the dual-port DRAM.

Real Mode Multiprocessing Configuration

Real Mode Multiprocessing Configuration Options

In real mode, you can configure the Multibus interface address space, dual-port memory, and the Multibus window to create a multiprocessing environment for your system.

Multiprocessing is the method by which multiple boards perform different related tasks and exchange data to complete those tasks. To exchange data in real mode, the dual-port memory of one board must be aligned with the another board's Multibus interface address space or Multibus window.

Dual-port aliasing is not used in real mode because the minimum alias offset is 1 Mbyte. If used, this would offset the dual-port memory beyond the address range of the CPU in real mode. The dual-port alias should be set to 0H for real mode operation.

Real Mode Multiprocessing Examples

The following examples shows an iSBC 486/12 board passing information in a multiprocessing environment with three other iSBC 486/12 boards.

Example 1. This example illustrates the point that the dual-port memory of one board must be aligned with the Multibus address space of another board in order to exchange data. It also illustrates how the Multibus interface address space works together with the Multibus window in real mode to support multiprocessing.

Table 6-9 lists the configuration parameters in this example.

Table 6-9. Real Mode Multiprocessing Example #1

Configuration	Board 1	Board 2	Board 3	Board 4
EPROM size	128 Kbytes	128 Kbytes	128 Kbytes	128 Kbytes
DRAM ending address	0D FFFFH	0C FFFFH	0B FFFFH	0A FFFFH
Dual-port ending address	0D FFFFH	0C FFFFH	0B FFFFH	0A FFFFH
Dual-port starting address	0D 0000H	0C 0000H	0B 0000H	0A 0000H
Multibus window starting address	0A 0000H	0A 0000H	0A 0000H	2 0000H
Multibus window ending address	0C FFFFH	0B FFFFH	0A FFFFH	0 FFFFH[1]
Dual-port alias	0H	0H	0H	0H
Multibus interface address space	0	64 Kbytes	128 Kbytes	192 Kbytes
Dual-port memory	64 Kbytes	64 Kbytes	64 Kbytes	64 Kbytes
Multibus window	192 Kbytes	128 Kbytes	64 Kbytes	0 Kbytes
Local DRAM	640 Kbytes	640 Kbytes	640 Kbytes	640 Kbytes

Note:
[1] Multibus window is disabled.

You can adjust the size of the dual-port memory in increments of 64 Kbytes to meet the needs of your application. The size of the dual-port memory on each board may be different.

The maximum number of boards that can be configured in this scheme depends on the size of your EPROM address space and how much local DRAM your application requires. For 128 Kbytes of EPROM, 64 Kbytes of dual-port memory on each board, and 64 Kbytes of local DRAM, the maximum number of boards is 13.

Figure 6-7 shows the memory map for the real mode example.

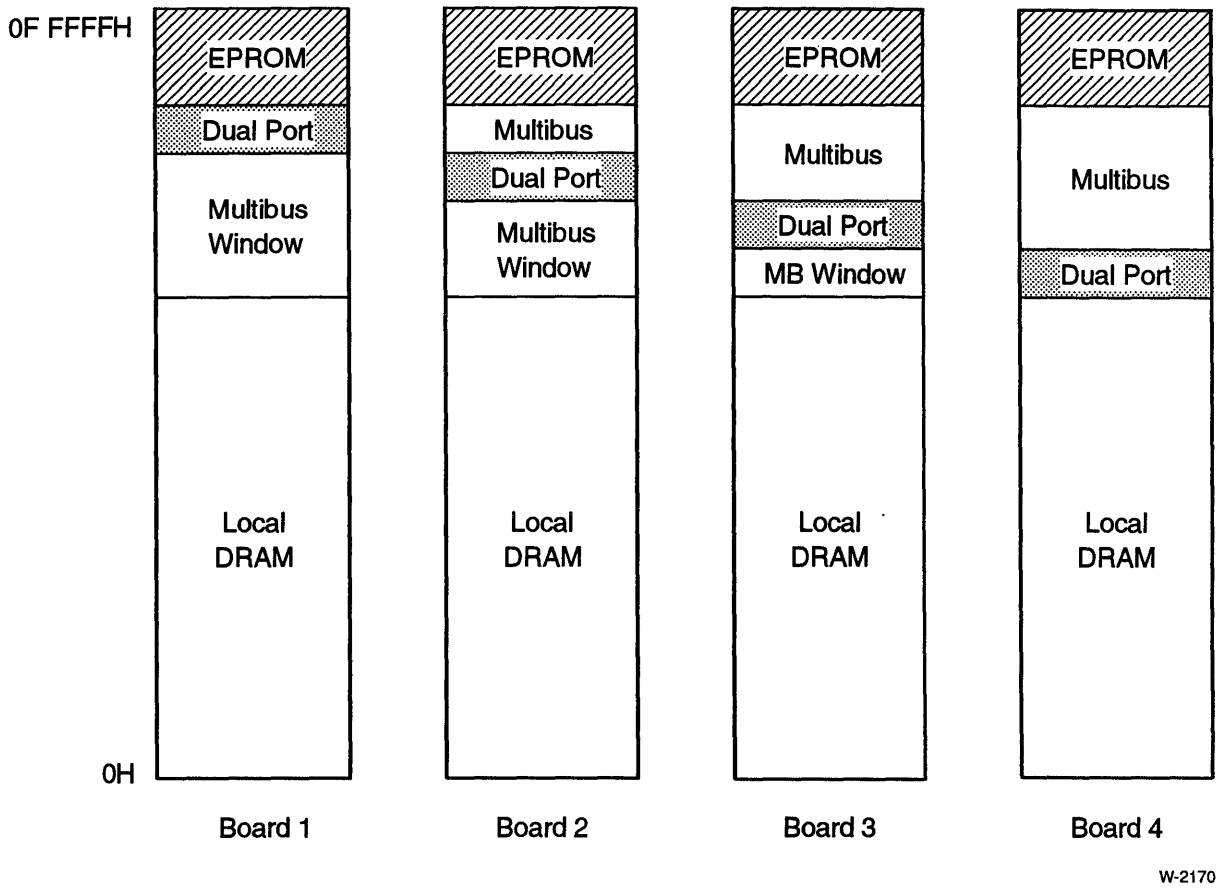


Figure 6-7. Real Mode Configuration Example 1

Example 1 shows an example where each board has a 64 Kbyte block of dual-port memory and each board can access the dual-port memory on each of the other boards.

Example 2. The next example shows a simpler case. In Example #2, only board 1 can access the dual-port memory on boards 2, 3, and 4. In this case, board 1 passes data and control blocks to the dual-port memory on boards 2, 3, and 4. Boards 2, 3, and 4 pass data only to board 1. This gives boards 2, 3, and 4 more local DRAM. This approach can be used when boards 2, 3, and 4 don't need to directly exchange data.

Figure 6-8 shows the memory map for this example.

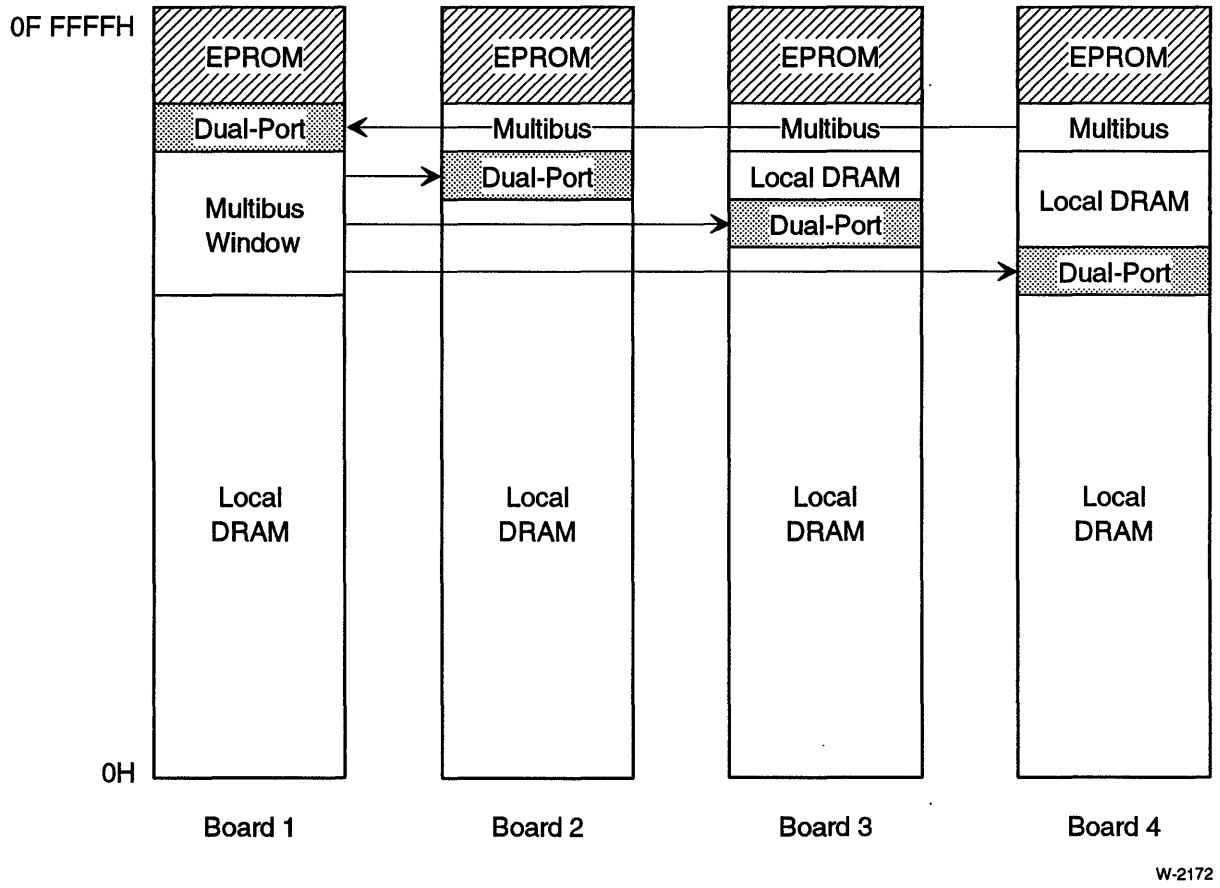


Figure 6-8. Real Mode Configuration Example 2

Example 3 (not illustrated). To simplify this example further, we can eliminate the dual-port memory on board 1. In this case, board 1 passes data and control blocks to the dual-port memory on boards 2, 3, and 4. This configuration maximizes the amount of local DRAM on boards 2, 3, and 4.

Since every application is unique, you may need to use a combination of the configurations illustrated in these examples. If your application requires more than 1 Mbyte of memory, look at the protected mode examples in the next section.

Protected Mode Multiprocessing Configuration

Protected Mode Multiprocessing Configuration Options

In protected mode, you can configure the Multibus interface address space, dual-port memory, and dual-port alias to create a multiprocessing environment for your system. The Multibus window option does not exist in protected mode.

Protected Mode Multiprocessing Example

This example shows four iSBC 486/12 boards passing information in a multiprocessing environment. Each board operates in protected mode and has the same memory map.

Table 6-10 lists the configuration parameters for this example.

Table 6-10. Protected Mode Multiprocessing Example

Parameter	Board 1	Board 2	Board 3	Board 4
EPROM Size	128 Kbytes	128 Kbytes	128 Kbytes	128 Kbytes
DRAM ending address	7F FFFFH	7F FFFFH	7F FFFFH	7F FFFFH
Dual-port ending address	83 FFFFH	82 FFFFH	81 FFFFH	80 FFFFH
Dual-port starting address	83 0000H	82 0000H	81 0000H	80 0000H
Dual port alias	0FC 0000H	0FF 0000H	0FE 0000H	0FD 0000H
Multibus interface address space	8064 Kbytes[1]	8064 Kbytes[1]	8064 Kbytes[1]	8064 Kbytes[1]
Dual port memory starting address ending address	64 Kbytes 7F 0000H 7F FFFFH	64 Kbytes 7F 0000H 7F FFFFH	64 Kbytes 7F 0000H 7F FFFFH	64 Kbytes 7F 0000H 7F FFFFH
Local DRAM	8128 Kbytes[2]	8128 Kbytes[2]	8128 Kbytes[2]	8128 Kbytes[2]
Dual-Port 16 Mbyte Page	0H	0H	0H	0H

Notes:
[1] 8 Mbytes minus 128 Kbytes of EPROM
[2] 8 Mbytes minus 64 Kbytes of dual-port memory

Figure 6-9 shows the memory maps for the boards in this example.

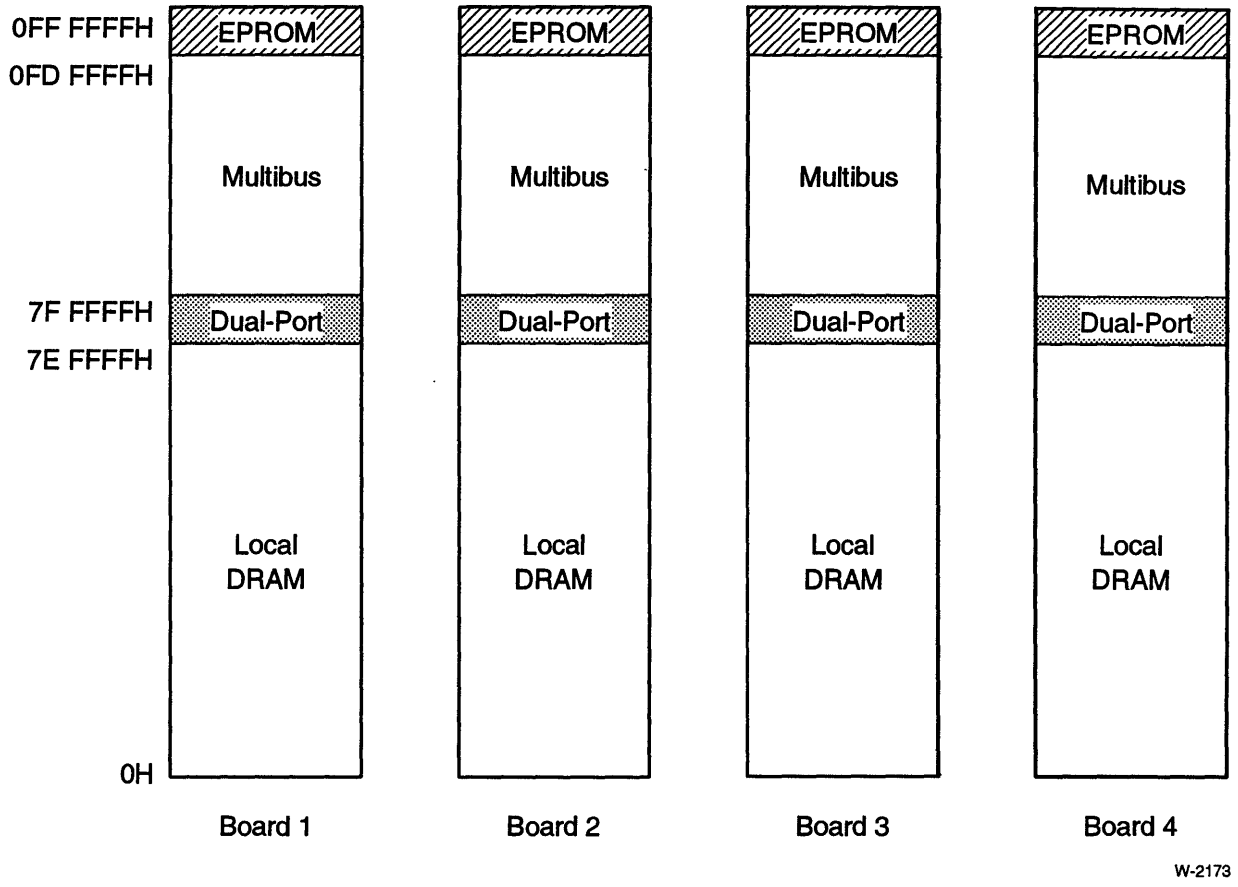


Figure 6-9. Protected Mode Multiprocessing Example

Figure 6-10 shows how the dual-port alias is used in this example. This figure illustrates, for example, that if board 1 writes to 81 0020H, the resulting DRAM address on board 3 is:

81 0020H + 0FE 0000H - 100 0000H = 7F 0020H

Notice that if the memory maps are identical for every board, then each board must have a unique dual-port alias.

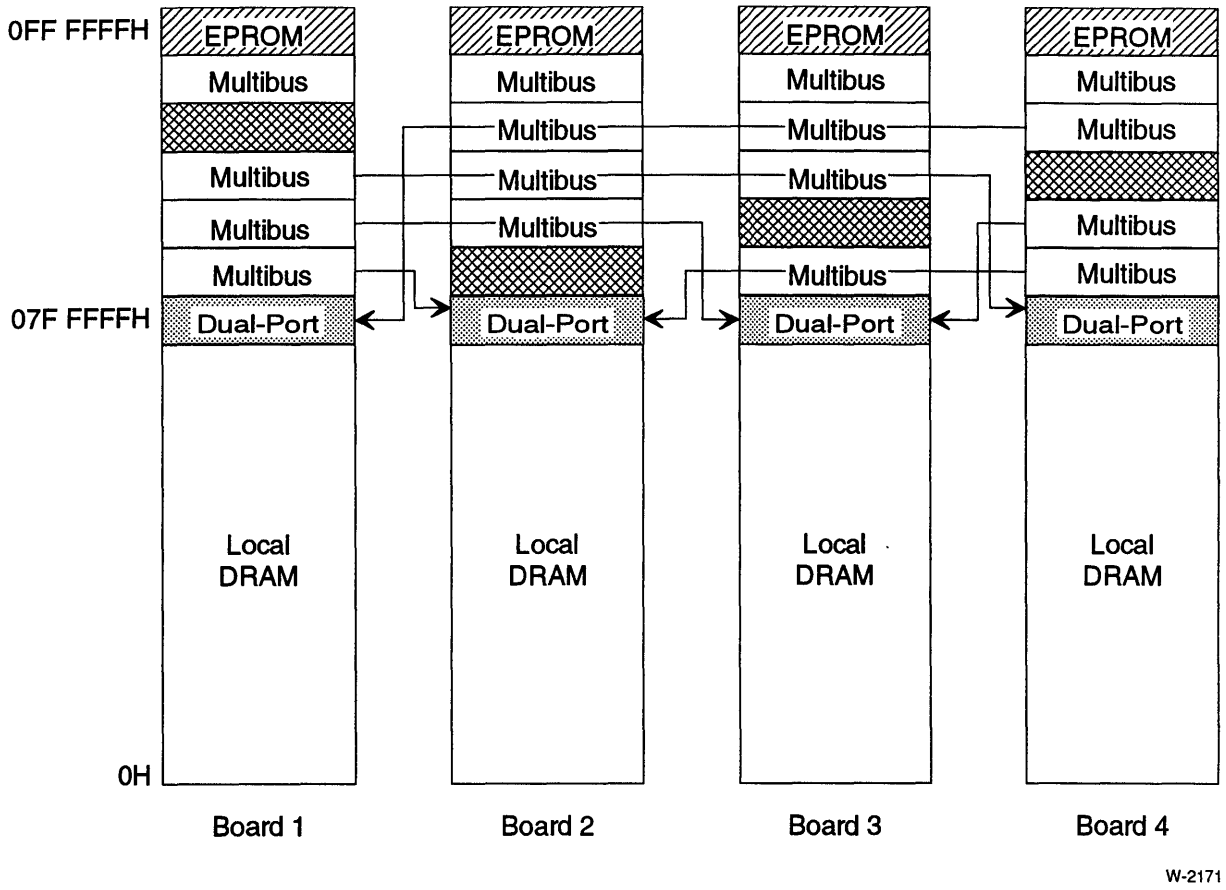
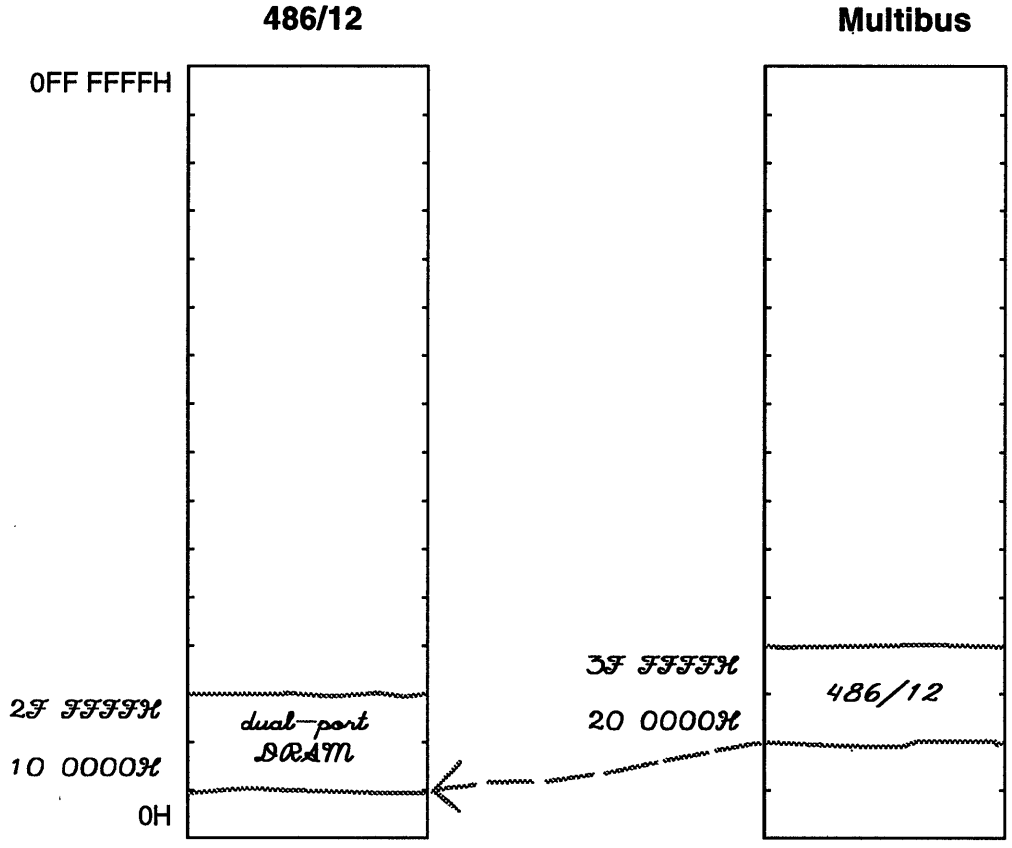


Figure 6-10. Example of Dual Port Alias in Protected Mode Multiprocessing Example

Sample Worksheets

System Memory Map Worksheet



NOTES:

Alias = 0F0 0000H
dual-port start = 20 0000H
dual-port end = 3F FFFFH
dual-port 16 Mbyte Page = 0H

W-2167

System Memory Worksheet

Step 1. Dual-port starting and ending addresses

__ Dual port starting address = 20 0000 H
__ Dual port ending address = 3F FFFF H
__ Record dual port starting and ending addresses on the System Memory Map Worksheet
__ Record starting and ending address jumpers on the Jumper Worksheet

Step 2. Dual-port alias

__ Dual port alias = 0F0 0000 H
__ Record dual port alias on the System Memory Map Worksheet
__ Record dual port alias jumpers on your Jumper Worksheet

W-2169

Configuration Question and Answers

Q. How is the dual-port memory configuration of the iSBC 486/12 board different from the iSBC 386/12 board?

A. The iSBC 486/12 board is 100 percent compatible with all iSBC 386/12 dual-port memory configurations. In addition, the iSBC 486/12 board can use the Dual-Port 16 Mbyte Page Register to map the dual-port memory anywhere in the on-board DRAM. The iSBC 486/12 board supports up to 64 Mbytes of on-board DRAM.

Q. How should I configure system memory if I want to use the iRMX II/III operating system?

A. You should use the default configuration for iRMX II/III.

Q. Can other Multibus agents switch the dual-port pages using the Dual-Port 16 Mbyte Page Register?

A. No. The on-board I/O space is not accessible to the Multibus interface.

Q. Can the Intel486 CPU or the 82258 ADMA use the local bus while another bus master is accessing dual-port memory and asserting the Multibus Lock signal?

A. The Intel486 CPU will continue to execute out of the internal cache until it needs the local bus. The Intel486 CPU and ADMA cannot use the local bus while the dual-port memory is locked by the Multibus.

Q. How can I disable dual-port memory?

A. Dual-port memory is disabled by setting the dual-port starting address greater than the dual-port ending address.

Q. In protected mode, can I shift the starting address of DRAM to support my multiprocessing application?

A. No. Unlike the iSBC 386/2X, iSBC 386/3X, and iSBC 386/12 boards, the iSBC 486/12 board does not support this feature. Use dual-port aliasing to support multiprocessing in protected mode.

Q. How is the dual-port ending address and the ending address of physical DRAM related?

A. Avoid mapping memory addresses beyond physical memory to the Multibus interface. You should check your setting of the dual-port ending address, dual-port alias, and the value in the 16 Mbyte Page register to be sure this is avoided.

Q. Are the starting and ending dual-port addresses the same for each page selected by the 16 Mbyte Page Register?

A. Yes. The dual-port alias is the same too.

Q. Can I access the dual-port memory on my own board through the Multibus address space in the 3rd Gbyte page?

A. No. The iSBC 486/12 board cannot access its own DRAM through the Multibus interface.

Q. I have 16 Mbytes of DRAM. I set the DRAM ending address at 7F FFFFH so that I can have an 8 Mbyte Multibus address space. What should I do to maximize the amount of dual-port memory? My operating system can only run in the first 16 Mbytes of memory, but I would like to make as much DRAM as possible available to other boards in the system.

A. You can't have any dual-port memory in the region used by your Multibus address space (Multibus addresses 80 0000H to 0FF FFFFH). The best you can do is to have 8 Mbytes of dual-port memory. Set your dual-port starting and ending addresses at 0H and 7F FFFFH, respectively. You can choose any dual-port alias you want, depending on how much dual port memory you want in the first 8 Mbytes. For example, if you want 0H to 3F FFFFH to be local DRAM, and 40 0000H to 7F FFFFH to be dual-port memory (as seen from the iSBC 486/12 board), you would set your dual-port alias to 40 0000H.

***

I/O Subsystem Configuration 7

Chapter Contents

Introduction 7-4
Related Information 7-4
New Terms in this Chapter 7-4
Configuration Procedure 7-5
Default Configuration 7-5
Default I/O Addresses 7-6
Special I/O Addresses 7-9
I/O Configuration 7-10
82C54 Programmable Interval Timer (PIT) 7-10
82C54 PIT Default Configuration 7-11
82C54 PIT Jumpers 7-12
82C54 PIT Programming 7-13
82C54 PIT Modes of Operation 7-13
82C54 PIT Initialization 7-13
82C54 I/O Port Addresses 7-13
82C54 Programming Example 7-14
82C54 PIT Count Values 7-16
82C55A Programmable Peripheral Interface (PPI) 7-17
82C55A PPI Overview 7-17
82C55A PPI Default Configuration 7-19
Port A Configuration 7-20
Port A Configuration for Output 7-20
Port A Configuration for Input 7-20
Port A Configuration for Bidirectional I/O 7-21
Port B Configuration 7-23
PB<2:0> User-Defined Code 7-23
PB3 LPT Interrupt Input 7-24
PB4 LPT Acknowledge* Input 7-25
PB5 Power-Fail Sense (PFSN*) Signal 7-25
PB6 LPT Error Input 7-26
PB7 Used To Control Direction of 74ALS245 (Optional) 7-26
PB7 LPT Busy Input 7-26
Port C Configuration 7-27
PC0 LPT Strobe* Output 7-27
PC1 Dual-Port Lock Output 7-27
PC2 NMI Mask Output 7-28
PC3 Multibus Time-Out Clear Output 7-28
PC4 CLREDGE1* Output (Optional) 7-29

PC<6:5> Multibus Interrupt Drivers 7-30
PC7 LPT Interrupt Latch Clear Output 7-31
82C55A PPI Stake Pins 7-33
82C55A Programming 7-34
82C55A PPI I/O Addresses 7-34
Supported Modes of Operation 7-34
82C55A PPI Initialization 7-36
Bit Set/Reset Control of Port C 7-38
82C55A PPI Applications 7-39
Centronics-Compatible Printer 7-39
NMI Mask Control 7-42
Switching the Board Between Real and Protected Modes 7-42
MB87033B SCSI Peripheral Controller (SPC) 7-44
SCSI Interface Overview 7-44
iSBC˜ 486/12S SCSI Block Diagram 7-44
SCSI Signals 7-46
SCSI ID Jumpers and GP3 Bit Jumper 7-48
SCSI Termination Resistors 7-48
SCSI Termination Power 7-49
MB87033B SPC Registers 7-49
MB87033B SPC Register I/O Addresses. 7-50
MB87033B SPC Programming 7-51
SCSI Control Registers 7-51
SCZID Register (1A1H - Read) 7-52
SCZINTDIS Register (1A1H - Write) 7-53
BLAST Register (1A3H) 7-53
HIN Register (1A5H) 7-54
BYTXFER Register (1A7H) 7-54
TURBO Register (1A9H) 7-54
SCZRST Register (1ABH - Write Only) 7-54
FIFO Data (1AEH) 7-54
8274 Multiple-Protocol Serial Controller (MPSC) 7-55
Serial Channel Configurations 7-57
Serial Connectors 7-62
MPSC I/O Ports 7-62
MPSC Configuration 7-62
Default Serial Configuration 7-62
Channel A RS-232C DCE Configuration 7-64
Channel A RS-232C DTE Configuration 7-64
Channel A RS-422A/449 DCE Configuration 7-66
Channel A RS-422A Multidrop Configuration 7-68
Serial Interrupt Configuration 7-68
Serial Break Detect Circuit 7-68
8274 MPSC Programming 7-68
SBX Interface 7-71
SBX I/O Addresses 7-71
SBX I/O Command Recovery Time 7-77
SBX DMA Operations 7-77
SBX Option Lines 7-79

SBX Interrupts 7-79
Multibus Interface (Connector P1) 7-80
BCLK* Signal Configuration 7-80
CCLK* Signal Configuration 7-80
INIT* Signal Configuration 7-81
Software Reset Configuration 7-82
Multibus Address Line Termination Resistors 7-83
Multibus Interrupts 7-84
Multibus LOCK* Signal 7-85
Multibus Buffered I/O Write Option 7-85
Multibus Arbitration 7-85
Configuration Questions and Answers 7-88

Introduction

This chapter includes the I/O address map for the board, discusses how to configure the PIT, PPI, MPSC, SPC, SBX interface, and the Multibus interface. The next chapter, Chapter 8, discusses the CPU, ADMA, DAG, PIC, and LEDs.

Related Information

Appendix A contains an I/O configuration worksheet to help you plan and record the I/O configuration. Appendix A also contains a jumper worksheet you can use to record your jumper configuration.

The following topics are related to I/O configuration:

- For ADMA and interrupt configuration, see Chapter 8, "CPU Subsystem Configuration."
- For component installation instructions, see Chapter 3, "Installation."
- For I/O connectors and cables, see Chapter 9, "Connectors and Cables."
- For RS-449/422 multidrop operation, see Appendix C, "Multidrop Information."

New Terms in this Chapter

MB87033B SPC — Fujitsu MB87033B SCSI Protocol Controller (SPC)

82C54 PIT — 82C54 Programmable Interval Timer (PIT). The PIT has three counters/timers.

82C55A PPI — 82C55A Programmable Peripheral Interface (PPI). The PPI has three 8-bit ports.

8274 MPSC — 8274 Multiple-Protocol Serial Controller (MPSC). The MPSC has two full-duplex serial channels.

82C59A PIC — 82C59A Programmable Interrupt Controller (PIC). The PIC has 8 interrupt request levels. The iSBC 486/12 board has two PICs that are cascaded to provide 15 interrupt request levels. Chapter 8, "CPU Subsystem Configuration," discusses the interrupt configuration.

break detect logic — The break detect logic is a circuit that is external to the 8274 MPSC. This circuit detects a serial break condition on either of the two serial channels. The output of the break detect logic is an interrupt signal that can be used as an interrupt source for the 82C59A PIC. The break detect logic can be used to generate an interrupt before the 8274 MPSC is initialized.

SBX — IEEE 959 SBX (System Bus Expansion) interface

DCE — Data Communication Equipment

DTE — Data Terminal Equipment

turbo Refers to a DMA transfer mode or a board register that controls the DMA transfers to the SCSI FIFOs. In turbo mode, the DMA request is asserted only after the FIFOs have reached a 128-byte threshold. The turbo mode is used together with the blast mode to control the high-speed DMA transfers to and from the SCSI FIFOs.

blast Refers to a DMA transfer mode or a board register that controls the DMA transfers to the SCSI FIFOs. In blast mode the ADMA/DAG will transfer 16 bytes of data during each ADMA cycle.

FIFO Refers to the SCSI FIFO buffers.

Configuration Procedure

To configure the I/O on the board, we suggest you do the following:

1. Copy the I/O Configuration Worksheet in Appendix A.
2. Use the copy of the Jumper Configuration Worksheet that you used in previous chapters.
3. Follow the worksheet and check-off each step. Pencil-in the jumpers on the jumper worksheet as you select each option. If you get lost, refer to the table of contents at the beginning of the chapter.
4. Go on to Chapter 8, "CPU Subsystem Configuration."

You may also want to look at:

- the programming information for each of the devices covered in this chapter
- the I/O configuration question and answers on page 7-88

Default Configuration

The default I/O configuration is as follows:

- SBX address swap disabled
- 82C54A PIT counter 0 to PIC
- 82C54A PIT counter 1 to 8274 serial channel B
- 82C54A PIT counter 2 to 8274 serial channel A
- 82C55A PPI port A configured for output
- SCSI ID of 7

Default I/O Addresses

Table 7-1 gives the iSBC 486/12 board I/O port addresses. No I/O resources are accessible from the Multibus interface.

Table 7-1. I/O Port Addresses

Address	Device	Description
0H-7FH	Multibus	
80H-0BFH	Multibus	If SBX 1 and SBX 2 are not installed
80H-8EH (even)	SBX 1 [1] (byte)	To 8-bit or 16-bit SBX, activates MCS0
80H-8EH (even)	SBX 1 [1] (word)	To 16-bit SBX, activates MCS0 and MCS1
81H-8FH (odd)	SBX 1 [1] (byte)	To 16-bit SBX, activates MCS1
90H-9EH (even)	SBX 1 [1] (byte)	To 8-bit SBX, activates MCS1
0A0H-0AEH (even)	SBX 2 [2] (byte)	To 8-bit or 16-bit SBX, activates MCS0
0A0H-0AEH (even)	SBX 2 [2] (word)	To 16-bit SBX, activates MCS0 and MCS1
0A1H-0AFH (odd)	SBX 2 [2] (byte)	To 16-bit SBX, activates MCS1
0B0H-0BEH (even)	SBX 2 [2] (byte)	To 8-bit SBX, activates MCS1
0C0H	Master 82C59A	Status, ICW1 (byte)
0C2H	Master 82C59A	Mask (byte)
0C4H	Slave 82C59A	Status, ICW1 (byte)
0C6H	Slave 82C59A	Mask (byte)
0C8H	82C55A PPI, Port A	Input/Output (byte)
0CAH	82C55A PPI, Port B	Input (byte)
0CAH	Mode Bit (see 00E4H)	Switch to Protected Mode with word write
0CCH	82C55A PPI, Port C	Output (byte)
0CEH	82C55A PPI	Control (byte)
0D0H	82C54 PIT	Counter 0 (byte)
0D2H	82C54 PIT	Counter 1 (byte)
0D4H	82C54 PIT	Counter 2 (byte)
0D6H	82C54 PIT	Control (byte)
0D8H	8274 MPSC	Channel A data (byte)
0DAH	8274 MPSC	Channel B data (byte)
0DCH	8274 MPSC	Channel A control (byte)
0DEH	8274 MPSC	Channel B control (byte)
0E0H	SW Reset	Byte write of data 01H
0E0H	Reset Flag	Byte read, D0 = 1 is SW reset Byte read, D0 = 0 is HW reset
0E2H	Clear Reset Flag	Byte write, Data = 00H, clears flag
0E4H	Mode Bit	Byte read, D0 = 0, Real Mode Byte read, D0 = 1, Protected Mode Byte write, D0 = 0, Set Real Mode Byte write, D0 = 1, Set Protected Mode
0E6H	Clear Serial Break Detect	Byte write, Data = 0H
0E8H	Clear Parity Error	Byte write, Data = XXH
0E8H	Set Parity Error	Byte read, Data = XXH. Next DRAM access causes parity error.
0EAH	SBX	DMA source or destination for SBX port

Notes:
X means "does not matter"
[1] Connector J5 if SBX address swap is disabled, J6 if SBX address swap is enabled. Refer to page 7-74.
[2] Connector J6 if SBX address swap is disabled, J5 if SBX address swap is enabled. Refer to page 7-74.

Table 7-1. I/O Port Addresses (continued)

Address	Device	Description
0ECH	Board ID Register	Word read, board configuration
0EEH	Dual-Port 16 Mbyte Page	Byte write
0F0H-0FFH	Reserved	Reserved
100H-17FH	Multibus interface	
180H-37FH	Multibus	If ADMA is not present
181H-19FH (odd)	SPC Registers	See Table 7-19, page 7-50
1A1H-1AFH (odd)	SCSI Control Registers	See Table 7-20
1B0H-1BFH	Reserved	Reserved
1C0H-1FFH	Multibus interface	
200H-2FFH	82258 (ADMA)	See Chapter 8
300H-360H	DMA Address Generator	See Chapter 8
361H-37FH	Reserved	
380H-0 FFFFH	Multibus	

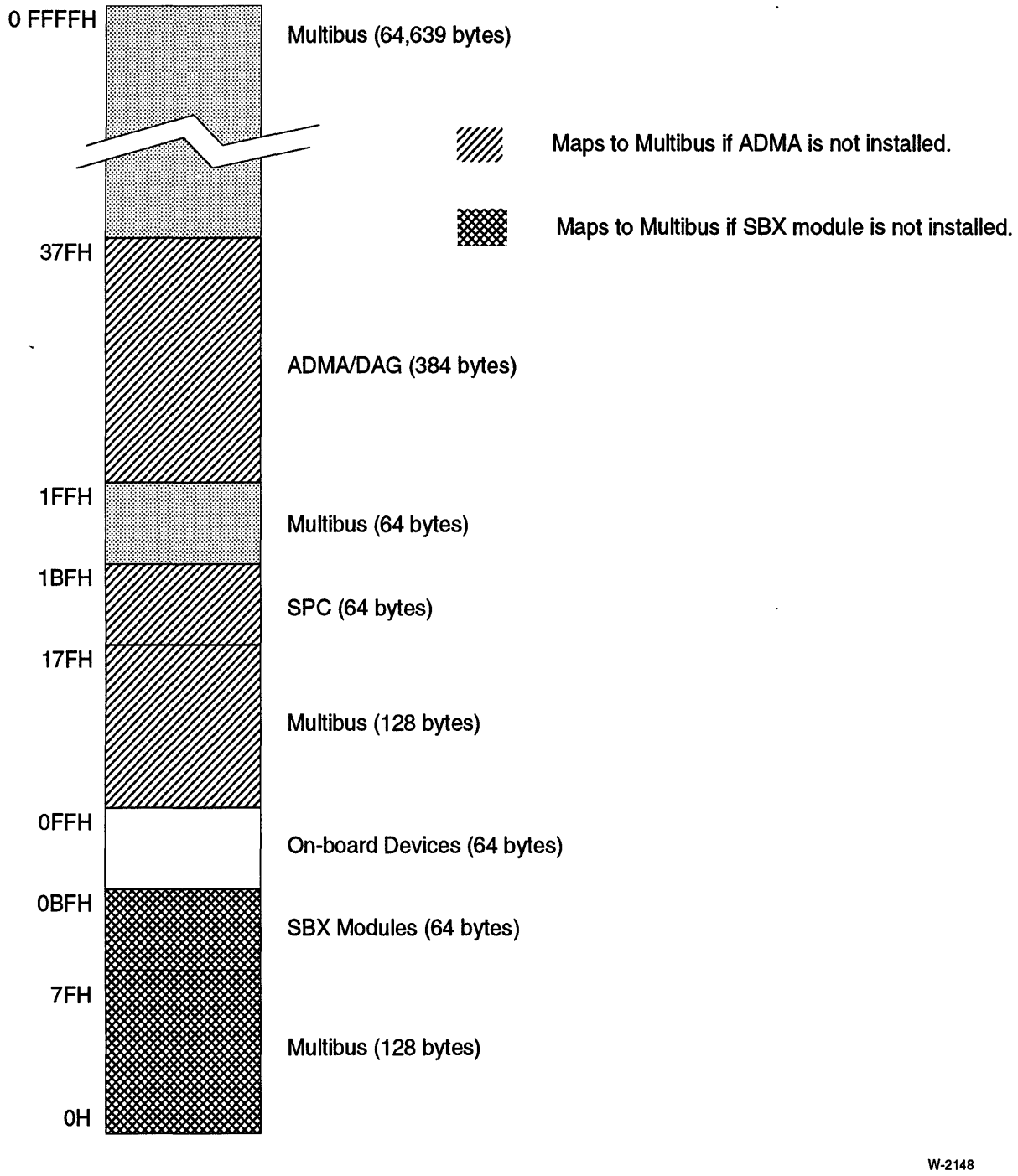


Figure 7-1. Default I/O Address Map

Special I/O Addresses

Table 7-2 summarizes the special I/O addresses that control various iSBC 486/12 board functions.

Table 7-2. Special I/O Addresses

I/O Address	Name	Operation	Result
0CAH	Mode Bit	Word Write (data = don't care)	Switch board to protected mode.
0E0H	SW Reset	Byte Write of D0=1	Assert software reset
0E0H	Reset Flag	Byte Read	D0=0 means last reset was a hardware reset D0=1 means last reset was a software reset
0E2H	Clear Reset Flag	Byte Write, D<7:0>=0	Clears reset flag to 0 (hardware reset).
0E4H	Mode Bit	Byte Read	D0=0 means board is in real mode D0=1 means board is in protected mode
0E4H	Mode Bit	Byte Write	D<7:0>=0 sets board to real mode D<7:0>=1 sets board to protected mode
0E6H	Clear serial break detect	Byte Write, D0=0	Clears serial break detect interrupt
0E8H	Clear parity error	Byte Write, D<7:0>=x	Clears on-board DRAM parity error detect interrupt
0E8H	Set parity error	Byte Read, D<7:0>=x	Sets the parity error interrupt after the **next** DRAM access.
0EAH	SBX DMA	Byte or Word, Read or Write	SBX source or destination for DMA operations.
0ECH	Board ID Register	Word Read	Board configuration information. Refer to Chapter 8.

Table 7-2. Special I/O Addresses (Continued)

I/O Address	Name	Operation	Result
0EEH	16 Mbyte Page	Byte Write	Selects 16-Mbyte Page for dual-port access. Refer to Chapter 6.
1A1H to 1AFH	SCSI Control Registers	Byte R/W	Refer to Table 7-20, page 7-52

I/O Configuration

The iSBC 486/12 board can access the full 64 Kbyte I/O address space of the Intel486 microprocessor. The I/O address space is used to access the following devices:

- 82C54 Programmable Interval Timer (PIT)
- 82C55A Programmable Peripheral Interface (PPI)
- MB87033B SCSI Peripheral Controller (SPC), if the ADMA is installed
- 8274 Multiple-Protocol Serial Controller (MPSC)
- SBX modules, if installed
- 82258 Advanced DMA controller, if installed
- Multibus I/O address space
- DMA Address Generator (DAG), if the ADMA is installed
- special I/O registers that are used to control the board

The I/O addresses used by the ADMA, DAG, and SPC are mapped to the Multibus I/O address space if the ADMA is not installed.

None of the on-board I/O address space is accessible from the Multibus interface.

82C54 Programmable Interval Timer (PIT)

The Intel 82C54 Programmable Interval Timer (PIT) supplies three 16-bit programmable counter outputs. In the default configuration, counter 0 is used as a time-tick generator (the output goes to the master PIC INTR0); counters 1 and 2 are used as baud rate clocks for the two serial channels. Table 7-3 lists the configuration options.

Table 7-3. 82C54 Configuration Options

Input	Counter	Output
1.229 MHz, or 4.0 MHz clock	Counter 0	Interrupt Matrix, or Counter 1 input
1.229 MHz clock, or Counter 0 output	Counter 1	Interrupt Matrix, or Serial Channel B
1.229 MHz, or 4.0 MHz clock	Counter 2	Interrupt Matrix, or Serial Channel A

The iSBC 486/12 board features a choice of two clock frequencies and the option to cascade counters 0 and 1. The two clock frequencies allow maximum flexibility in programming the counters. The option to cascade the counters allows you to connect the output of counter 0 to the clock input on counter 1. By cascading counters 0 and 1, you can form a 32-bit counter with a maximum count delay for this configuration of about 58 minutes.

82C54 PIT Default Configuration

The 82C54 PIT default configuration is as follows:

- counter 0 clock = 1.229 MHz
- counter 1 clock = 1.229 MHz
- counter 2 clock = 1.229 MHz
- counter 0 output connected to master PIC interrupt request level 0
- counter 1 output connected to 8274 MPSC channel B transmit clock
- counter 2 output connected to 8274 MPSC channel A transmit clock

Figure 7-2 shows the default configuration for the PIT.

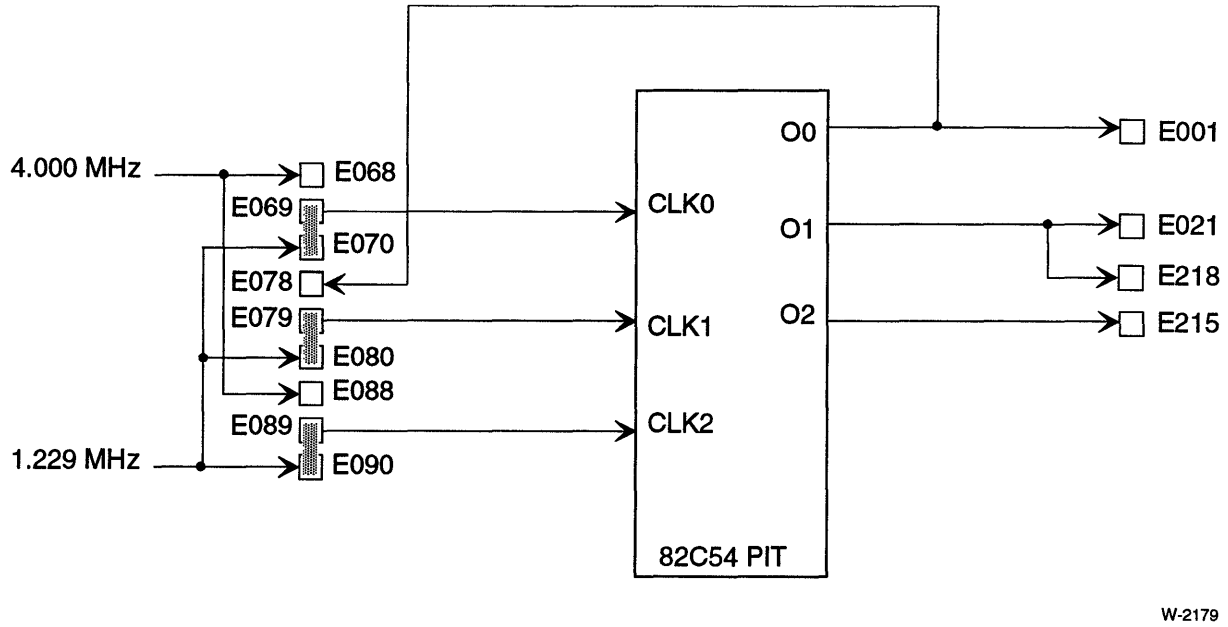


Figure 7-2. 82C54 PIT Default Configuration

82C54 PIT Jumpers

Table 7-4 lists the clock input jumpers for the 82C54 PIT. For information on the clock output jumpers, refer to the sections on the 8274 MPSC (in this chapter) or interrupt jumpering (in Chapter 8).

Table 7-4. 82C54 PIT Jumpers

Counter	Clock Input	Jumpers
0	1.229 MHz	E069-E070§ IN
0	4.000 MHz	E069-E070 OUT E068-E069 IN
1	1.229 MHz	E079-E080§ IN
1	Counter 0 output	E079-E080 OUT E078-E079 IN
2	1.229 MHz	E089-E090§ IN
2	4.000 MHz	E089-E090 OUT E088-E089 IN

Note:
§ default

82C54 PIT Programming

This section includes information that is specific to the iSBC 486/12 board. For complete 82C54 PIT programming information, refer to the *Peripheral Components* Handbook (Intel order number 296467). This section includes:

- 82C54 modes of operation supported by the iSBC 486/12 board (page 7-13)
- a brief discussion of initialization (page 7-13)
- I/O addresses for the 82C54 internal registers (page 7-13)
- a simple programming example (page 7-15)
- count values for standard baud rates and typical time intervals (page 7-16)

82C54 PIT Modes of Operation

The 82C54 PIT has six modes of operation. The iSBC 486/12 board ties the GATE input to each counter high so it can only support four of the 82C54 modes of operation:

- Mode 0: Interrupt on terminal count
- Mode 2: Rate generator
- Mode 3: Square wave generator
- Mode 4: Software triggered strobe

Modes 1 and 5 are not supported. These modes use the rising edge of the GATE input to trigger the count and are not supported because the GATE inputs are tied high.

82C54 PIT Initialization

After power-up, the state of the 82C54 is undefined. The initialization routine must program each output channel for the proper mode of operation.

To program the 82C54 PIT, your code must perform these steps in order:

1. initialize the timer for the desired count
2. set the value of the counter
3. start the timer

82C54 I/O Port Addresses

Table 7-5 gives the 82C54 I/O port addresses.

NOTE

All I/O read and write operations are byte-wide. The 82C54 PIT does not support 16-bit read or write operations.

Table 7-5. 82C54 PIT I/O Port Addresses

I/O Port Address	Register	Operation	Function
0D0H	Counter 0	Byte read[1] Byte write	count value count value
0D2H	Counter 1	Byte read[1] Byte write	count value count value
0D4H	Counter 2	Byte read[1] Byte write	count value count value
0D6H	Control	Byte read Byte write	Illegal control register

Note:
[1] The count value is read or written in two byte-read or byte-write operations. The byte order is set by the control word for the counter.

82C54 Programming Example

82C54 Programming Example. Figure 7-3 shows a PL/M programming example of the 82C54 counter 1 being programmed for a 9600 baud rate. This example uses a count value of 0080H for 9600 baud and a Control Word value of 76H. The Control Word specifies the following:

- selects counter 1
- specifies that the next operation will write the least significant byte first, then the most significant byte
- sets counter 1 in mode 3 (square wave mode)
- selects 16-bit binary counter mode

The count value is loaded into the 82C54 by two byte-write operations.

After writing a Control Word and initial count, the counter will be loaded on the next input clock pulse. The count will be decremented by two on succeeding clock pulses. When the count expires, the output changes value and the counter is reloaded with the initial count. This process is repeated indefinitely.

```
/*Intel 82C54 Programmable Interval Timer programming example
  Copyright Intel Corp., 1989, 1990
  This example is intended for your benefit in developing applications/
  systems using the Intel iSBC 486/12 Single Board Computers.  Intel
  hereby grants you permission to modify and incorporate as needed.*/


/*Declarations*/

Timer$Counter1          Lit '0D2H',    /* I/O location of counter 1 */
Timer$Control           Lit '0D6H',    /* I/O location of control register */
BAUD9600                Lit '076H';    /* Counter 1 in Mode 3 */


/*Program*/

output(Timer$Control)=BAUD9600;   /* Initialize Counter 1 */
output(Timer$Counter1)=80H;       /* Set the value of counter, least */
                                  /* significant byte */
output(Timer$Counter1)=00H;       /* Set the value of counter, most
                                  /* significant byte */
```

Figure 7-3. 82C54 Programming Example

82C54 PIT Count Values

Table 7-6 gives the programming word for baud rate generation. The 1.229 MHz clock should be used to generate the standard baud rates. The programming word must be sent as two byte-writes.

The value of the programming word depends on the clock multiplier in the 8274 MPSC. For a 1x multiplier, the value is calculated as follows:

```
COUNT VALUE = CLOCK FREQUENCY × BAUD RATE
```

where CLOCK FREQUENCY = 1,228,800 Hz or 4,000,000 Hz

For example, the count value for 9600 baud is:

count value = 1228800 × 9600 = 128 = 80H

This programming word is loaded as two bytes: 0H and 80H.

Table 7-6. Baud Rate Generation

Programming Word[1]	Counter Output[2,3] (kHz)	Synchronous Baud Rate	Asynchronous Baud Rate (8274 MPSC Clock Multiplier) x1	x16	x32	x64
00 02H	615.0	615,000	–	–	19,200	9,600
00 04H	307.5	307,500	–	19,200	9,600	4,800
00 08H	153.8	153,800	–	9,600	4,800	2,400
00 10H	76.9	76,900	–	4,800	2,400	1,200
00 20H	38.4	38,400	–	2,400	1,200	600
00 40H	19.2	19,200	19,200	1,200	600	300
00 80H	9.6	9,600	9,600	600	300	150
01 00H	4.8	4,800	4,800	300	150	75
02 00H	2.4	2,400	2,400	150	75	–
04 00H	1.2	1,200	1,200	75	–	–
08 00H	0.6	600	600	–	–	–

Notes:
[1] The programming word must be sent as two byte-writes.
[2] The input clock frequency is 1.229 MHz.
[3] Control Word set for 16-bit binary counter operation.

Table 7-7 gives the count values for the 82C54 operating as a real-time clock (mode 2). For longer time intervals than the maximum shown in the table, you must cascade counters 0 and 1. The count value is calculated as follows:

```
COUNT VALUE = TIME INTERVAL ÷ CLOCK FREQUENCY
```

where CLOCK FREQUENCY = 1,228,800 Hz or 4,000,000 Hz
and TIME INTERVAL is in seconds

For example, for a 50 ms time interval:

count value = 0.050 × 1228800 = 61440 = 0 F000H

The programming word is loaded as two bytes: 0F0H and 0H.

Table 7-7. 82C54 Count Values for Real Time Clock

Count Value[1] (Programming Word)	1.2288 MHz Clock	4.0000 MHz
65,535 (0FF FFH)	53 ms	16 ms
40,000 (9C 40H)	–	10 ms
4,000 (0F A0H)	–	1 ms
61,440 (0F0 00H)	50 ms	–
1 (00 01H)	813 ns	250 ns

Notes:
[1] Control Word set for 16-bit binary counter operation, mode 2 (rate generator).

82C55A Programmable Peripheral Interface (PPI)

82C55A PPI Overview

The 82C55A Programmable Peripheral Interface (PPI) supplies three programmable 8-bit ports. In the default configuration, Port A is configured as a parallel printer port and can be accessed at connector J1.

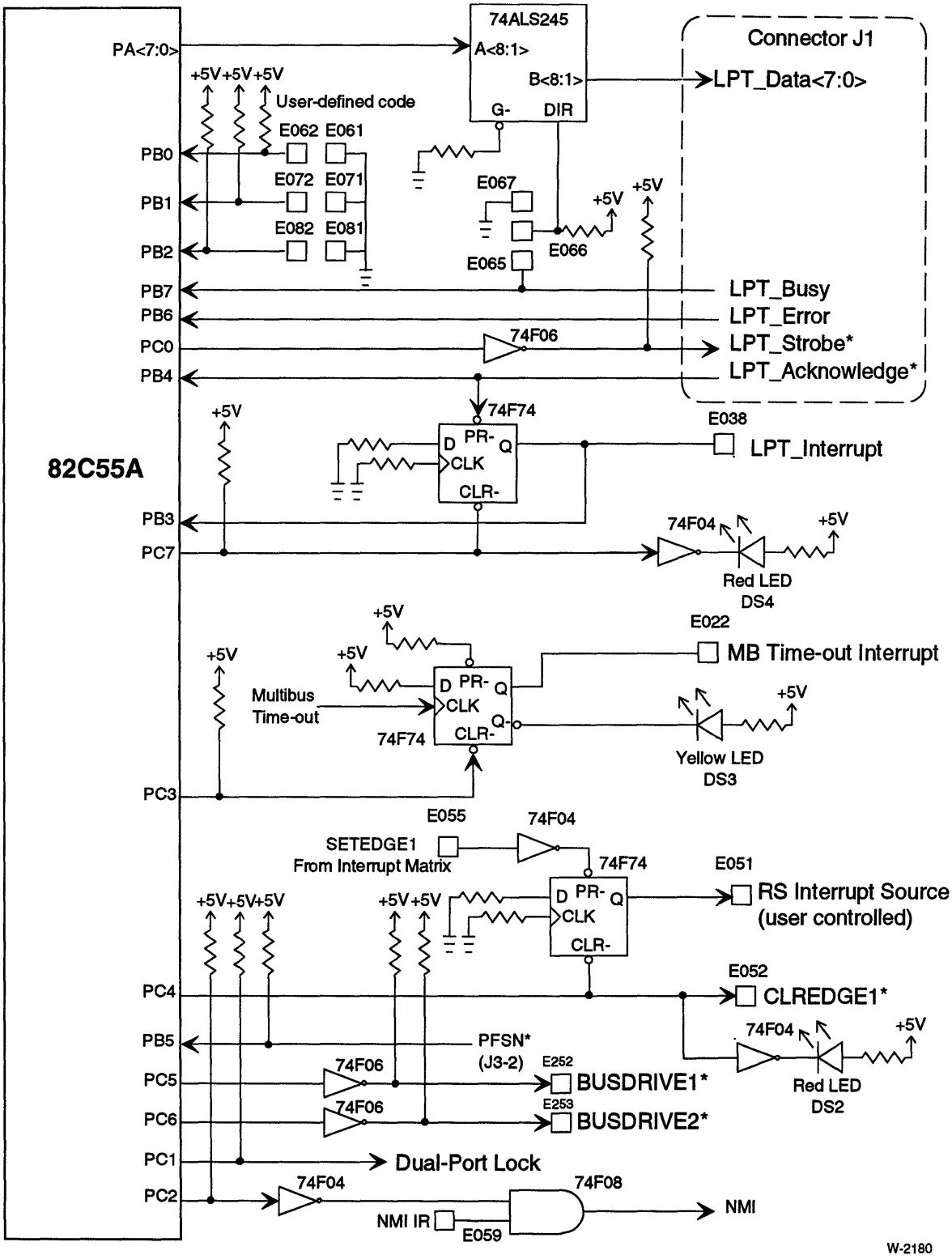


Figure 7-4. 82C55A Parallel Peripheral Interface Logic Diagram

82C55A PPI Default Configuration

On the iSBC 486/12 board, the PPI ports are configured as follows:

- Port A Input/output
- Port B Input only
- Port C Output only

Table 7-8 gives the bit assignments for the three parallel ports.

Table 7-8. Parallel Port Bit Assignments for the 82C55A PPI

Bit	Description
Port A (Input/Output)	
Port A, Bit 0	General purpose I/O or Line Printer Data 0
Port A, Bit 1	General purpose I/O or Line Printer Data 1
Port A, Bit 2	General purpose I/O or Line Printer Data 2
Port A, Bit 3	General purpose I/O or Line Printer Data 3
Port A, Bit 4	General purpose I/O or Line Printer Data 4
Port A, Bit 5	General purpose I/O or Line Printer Data 5
Port A, Bit 6	General purpose I/O or Line Printer Data 6
Port A, Bit 7	General purpose I/O or Line Printer Data 7
Port B (Input Only)	
Port B, Bit 0	User Defined bit 0 (CODE0)
Port B, Bit 1	User Defined bit 1 (CODE1)
Port B, Bit 2	User Defined bit 2 (CODE2)
Port B, Bit 3	Line Printer Interrupt (1=active)
Port B, Bit 4	General purpose input or Line Printer Acknowledge (0=active)
Port B, Bit 5	Power fail sense input (0=active)
Port B, Bit 6	General purpose input or Line Printer Error (1=active)
Port B, Bit 7	General purpose input or Line Printer Busy (1=active)
Port C (Output Only)	
Port C, Bit 0[1]	General purpose output or Line Printer Data Strobe (1=active)
Port C, Bit 1	Dual-Port Lock (0 = lock active)
Port C, Bit 2	NMI Mask (0 = NMI enabled)
Port C, Bit 3	Time-out Interrupt Clear (0 = Clear, 1 = Enable)
Port C, Bit 4	LED DS2 (1 = On) and RSINT Clear (0 = Clear)
Port C, Bit 5	Multibus interrupt drive 1 (1 = Active)
Port C, Bit 6	Multibus interrupt drive 2 (1 = Active)
Port C, Bit 7	LED DS4 (1 = On) and Line Printer Interrupt clear (0 = Clear)

Note:
[1] When initialized, Port C is set to 0H. Your software must configure Port C for your system operation.

Port A Configuration

Port A Configuration for Output

Port A can be used as a printer port, or for a general purpose parallel port. The default board is configured for Centronics-compatible printer operation. To operate Port A as an input port, jumper E066-E067 must be installed. To output a byte on Port A, write to I/O address 0C8H. If jumper E065-E066 is installed, the LPTBUSY signal controls Port A direction. If LPTBUSY is asserted (logic 1), Port A is an output port.

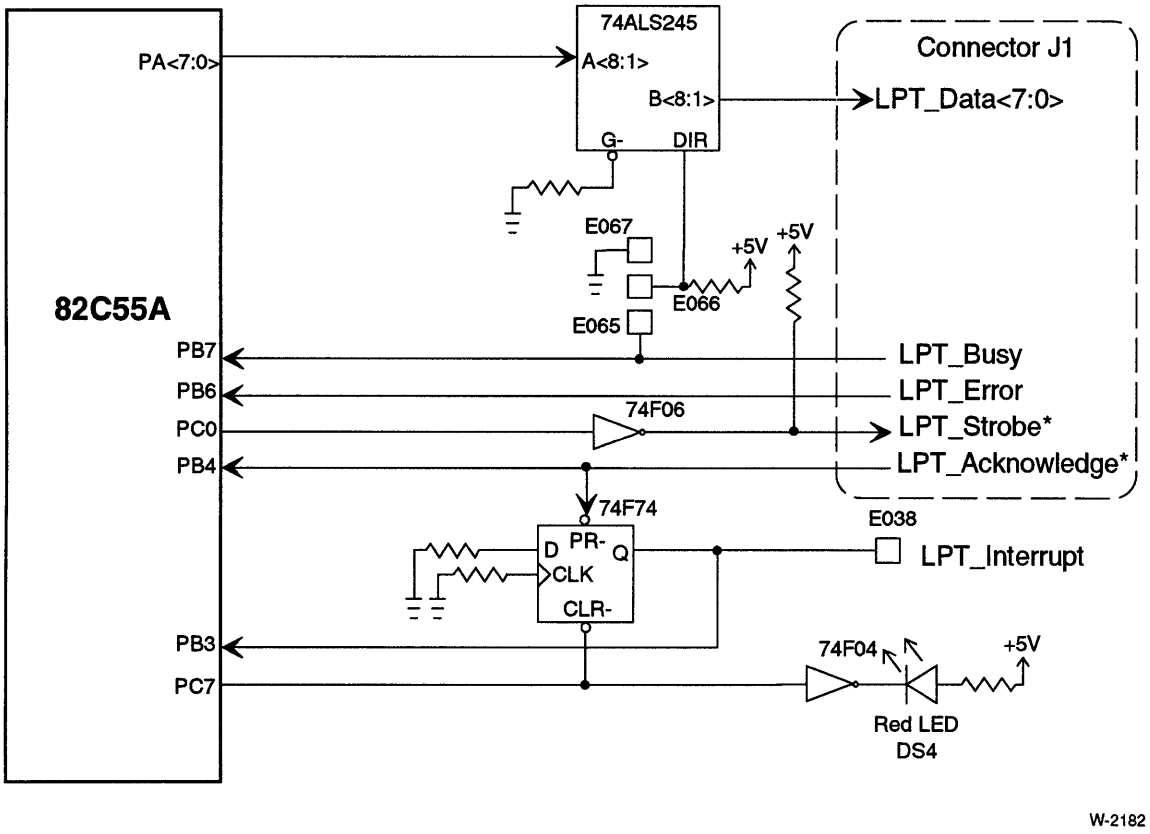


Figure 7-5. Line Printer Interface (Port A)

Port A Configuration for Input

To use Port A as an 8-bit input port, install jumper E066-E067. Read the input at I/O address 0C8H.

Port A Configuration for Bidirectional I/O

For software control of the port A transceiver direction using bit PC4, wire-wrap stake pins E052 to E066. For external control of port A, insert jumper E065-E066, this allows the LPT_BUSY signal to control Port A direction. If LPT_BUSY is not asserted (logic 0), Port A is an input port.

NOTE

Stake pin E066 only controls the direction of the port A transceiver. You must program the 82C55A PPI to correspond to the direction set for the port A transceiver.

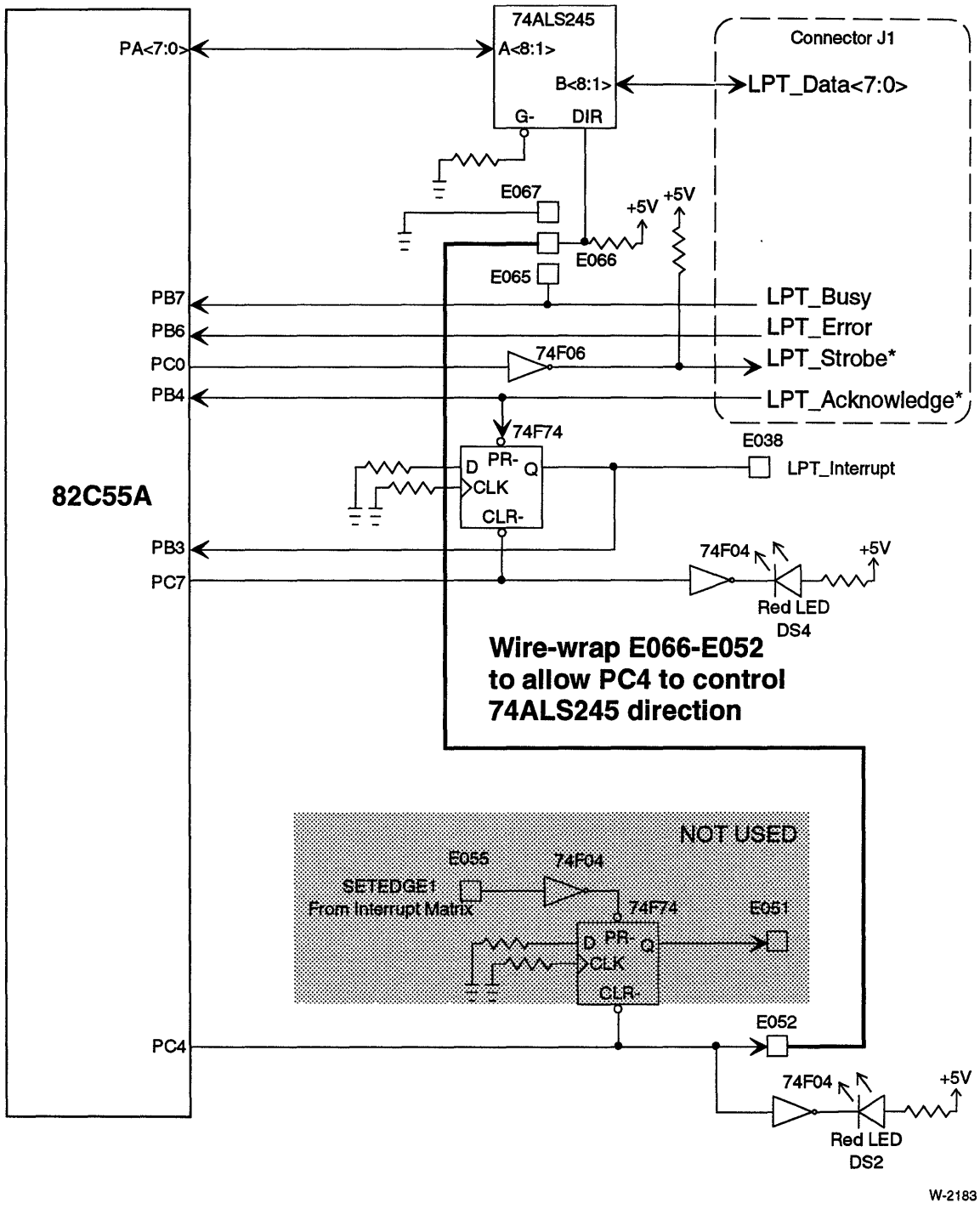


Figure 7-6. Using PC4 to Control Port A Transceiver Direction

Port B Configuration

With Port B, you can read the user-defined code, receive the printer handshake signals, and detect a power-fail interrupt. To input a byte on Port B, read I/O address 0CAH.

PB<2:0> User-Defined Code

The three-bit user-defined code can be accessed by reading bits PB<2:0>. The three bit code is set using jumpers. These bits can also be used as general purpose inputs by wire-wrapping the input to the associated stake-pin.

NOTE

On the iSBC 286/10/12/14/16 and 386/12 boards, these three bits were used to identify the type of board. This function has been replaced on the iSBC 486/12 board with the Board ID Register at I/O address 0ECH. This register provides more information than the old 3-bit code.

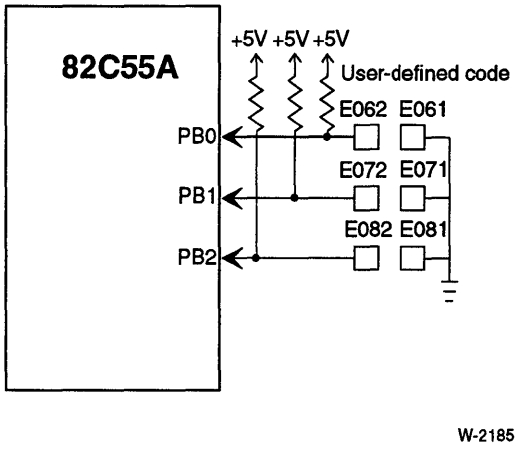


Figure 7-7. User-Defined Code

Table 7-9. User-Defined Code Jumpers

Code PB<2:0>	E082-E081	E072-E071	E062-E061
000	IN	IN	IN
001	IN	IN	OUT
010	IN	OUT	IN
011	IN	OUT	OUT
100	OUT	IN	IN
101	OUT	IN	OUT
110§	OUT	OUT	IN
111	OUT	OUT	OUT

Note:
§ default

PB3 LPT Interrupt Input

This signal is the output of the 74F74 flipflop. Figure 7-8 shows the LPT interrupt latch. When the 82C55A PPI is in mode 0, the inputs to port B are not latched. The LPT Acknowledge* input, therefore, will not be latched. PB3 is used to determine the state of the LPT interrupt.

The latch is cleared using PC7 (see page 7-32).

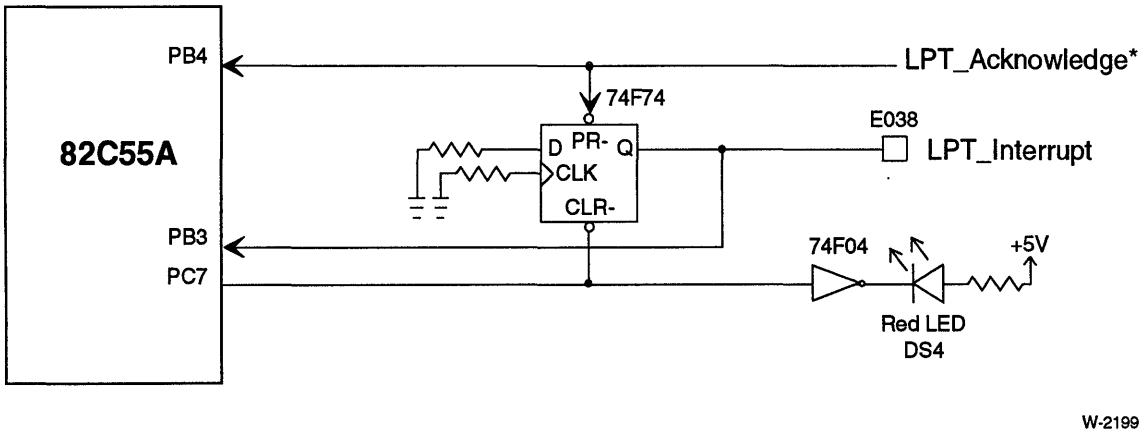


Figure 7-8. LPT Interrupt Latch

The LPT interrupt goes to stake pin E038. In the default configuration, this stake pin is tied to IR7 on the slave PIC. Chapter 8 discusses the interrupt configuration and programming.

LED DS4 only indicates the state of bit PC7. It does not indicate the state of the LPT interrupt or LPT Acknowledge*. If the LPT interrupt is not used, bit PC7 can be used to control DS4. The function of this LED must be defined by the user.

PB4 LPT Acknowledge* Input

This signal goes low to indicate that the data has been received by the printer. This is an active-low signal. The flipflop shown in Figure 7-8 latches the signal to allow you to use it as an interrupt source.

PB5 Power-Fail Sense (PFSN*) Signal

The PFSN* signal is an off-board signal that indicates an impending power-failure. This is an active-low signal. The PFSN* signal is latched by an off-board circuit powered by the standby power source. Figure 7-9 shows the PFSN* signal and bit PB5. The PB5 input is not latched in the 82C55A PPI.

For more information on the power failure timing sequence, refer to the Multibus specification (IEEE 796).

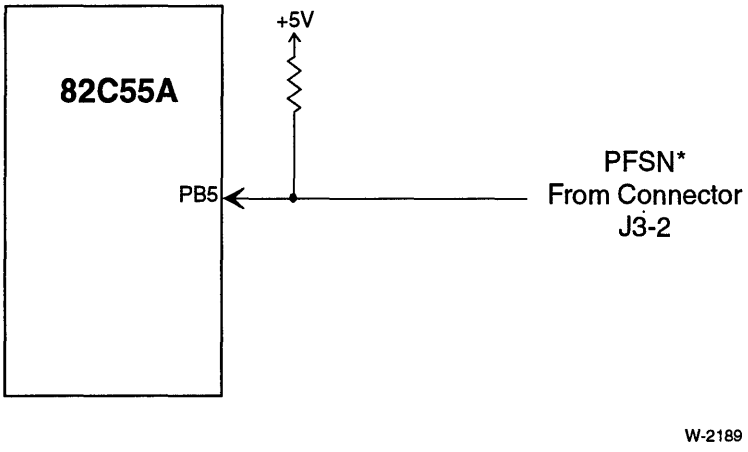


Figure 7-9. Power-Fail Sense PFSN* Signal

NOTE

The front panel interface also has the Power Failure Sense Interrupt PFSINT* signal. The PFSINT* signal is available at the interrupt jumper matrix.

PB6 LPT Error Input

Using the standard ribbon cable for Connector J1, the LPT Error signal is connected to the PE (paper empty) signal. This signal is high when the printer requires operator intervention. Some printers also use this signal as a general fault signal.

The PB6 input is not latched.

PB7 Used To Control Direction of 74ALS245 (Optional)

You can use the signal from Connector J1, pin 6 (LPT Busy) to control the direction of the 74ALS245 transceiver. The direction can be determined by reading bit PB7. This allows you to use port A as a bidirectional parallel I/O port.

NOTE

The LPT Busy signal does not change the direction of port A. Because the 82C55A can only support mode 0 operation (basic I/O), you must change the direction of port A by programming the control port. Mode 2 (bidirectional I/O) is not supported because it uses port C bits that are dedicated to other board functions.

Table 7-10. Jumper Configuration for Control of Port A Transceiver

Port A Transceiver Direction	Jumper E066-E067	Jumper E065-E066
INPUT[1]	IN	OUT
OUTPUT§[2]	OUT	OUT
Controlled by LPT Error	OUT	IN

Notes:
§ default
[1] INPUT means from Connector J1 to 82C55A port A
[2] OUTPUT means from 82C55A port A to Connector J1

PB7 LPT Busy Input

When used as a Centronics-compatible printer port, the LPT Busy signal goes high after the computer pulses the LPT Strobe* signal. The LPT Busy signal is low when the printer is not busy. Figure 7-19 shows the timing relationship of the LPT Busy signal and the other handshaking signals.

The LPT Busy input on PB7 is not latched. The pulse width of the LPT Busy signal varies from printer to printer.

Port C Configuration

Port C controls the Multibus interface interrupts (BUSDRIVE 1 and 2), two of the LEDs, and a number of other board functions. To output a byte on Port C, write to I/O address 0CCH.

PC0 LPT Strobe* Output

The PC0 output is inverted to form the active low LPT Strobe signal. The LPT Strobe signal indicates that the data on the port A data lines is valid. Figure 7-19 shows the timing relationship of the LPT Strobe* signal and the other handshaking signals.

The PC0 output is latched.

PC1 Dual-Port Lock Output

The dual port lock is used to lock the dual-port memory to the local bus. When PC1 is high, the lock is inactive. When PC1 is low, the lock is active. Figure 7-10 shows the dual-port lock.

The PC1 output is latched when the 82C55A is in mode 0.

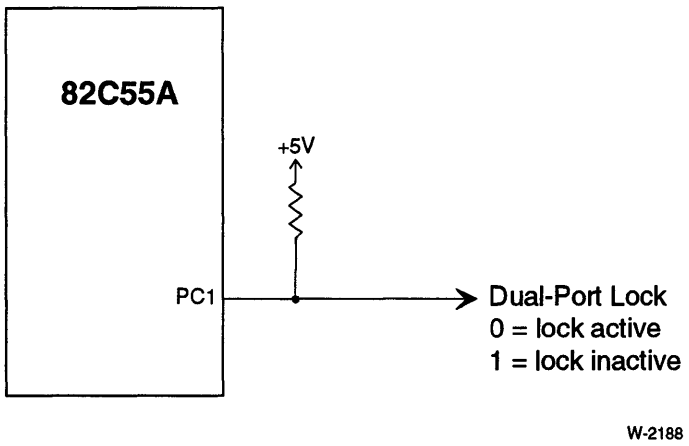


Figure 7-10. Dual-Port Lock

NOTE

To prevent a deadlock situation, the dual-port memory will be granted to a Multibus agent, regardless of the dual-port lock, if the agent's dual-port request is received at the same time the CPU on the iSBC 486/12 board is starting a Multibus access.

PC2 NMI Mask Output

Bit PC2 is used to control the NMI mask. When PC2 is high, the NMI interrupt request source (NMIR) is masked. When PC2 is low, the NMIR signal is not masked. Figure 7-11 shows the NMI mask logic.

The PC2 output is latched by the 82C55A when programmed for mode 0.

The main application of the NMI mask is to mask the parity interrupt after power-up. This application is discussed later in the section on applications of the 82C55A (starting on page 7-39).

The configuration of the NMI sources is discussed in Chapter 8 in the section on the 82C59A programmable interrupt controller.

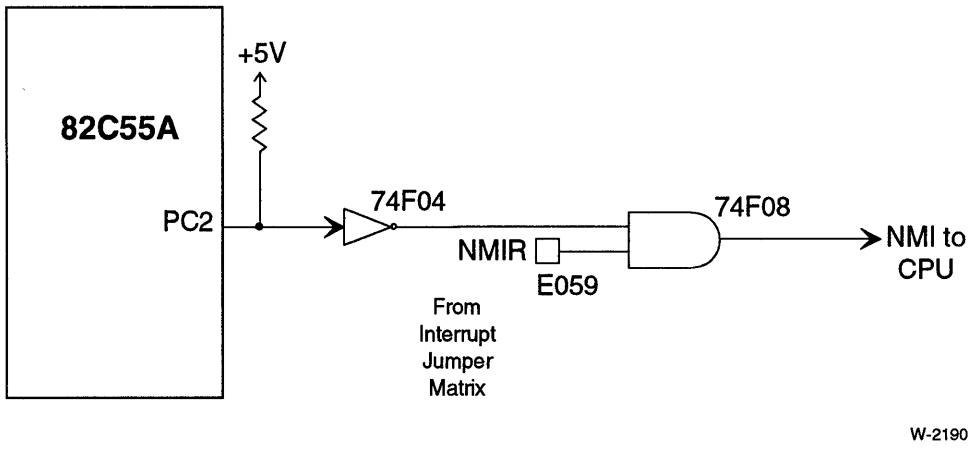
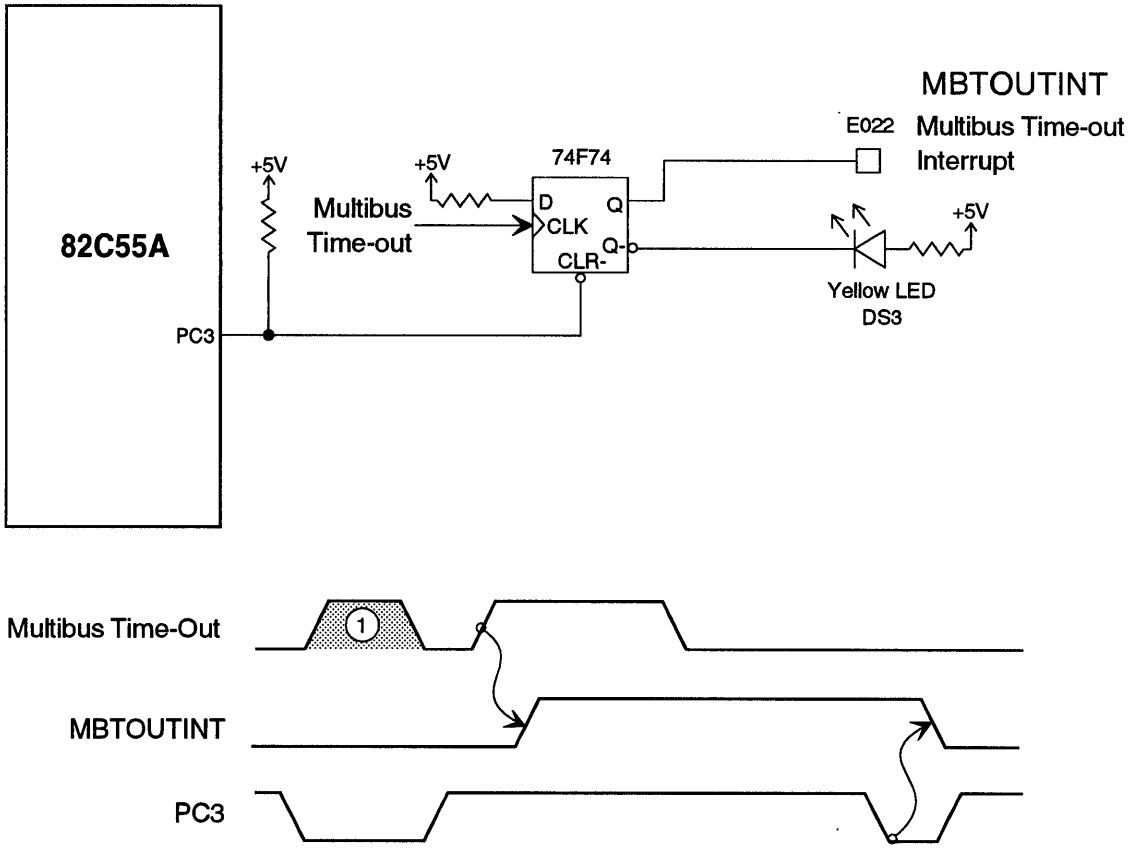


Figure 7-11. NMI Mask Logic

PC3 Multibus Time-Out Clear Output

Bit PC3 is used to clear the Multibus time-out latch. Figure 7-12 shows the Multibus Time-out logic and the PC3 signal.



Note:
1. Multibus Time-Out signal has no effect while PC3 is low.

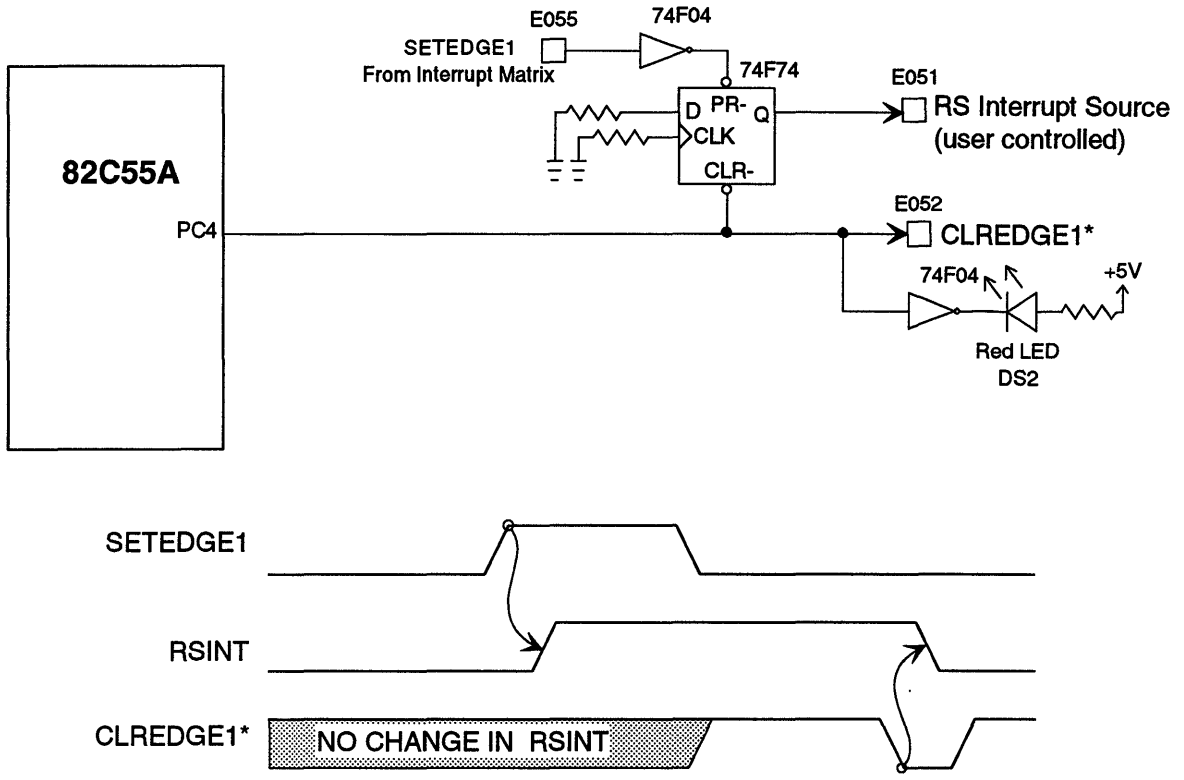
MB time-out	PC3	MBTOUTINT	LED DS3
Low-to-high	1	1	ON
Don't care	0	0	OFF
0	1	No Change	No Change
1	1	No Change	No Change

W-2184

Figure 7-12. Multibus Time-Out Clear

PC4 CLREDGE1* Output (Optional)

The CLREDGE1* signal is used to clear the optional 74F74 flipflop and control the DS2 LED. This flipflop can be used to latch interrupt sources. The output of the flipflop (RSINT) is available at stake pin E051 in the interrupt matrix. Figure 7-13 shows the CLREDGE1* signal. The signal is active low. The 82C55A operating in mode 0 will latch the PC4 output.



SETEDGE1	CLREDGE1*	RSINT
0	1	No Change
0	0	0
1	1	1
1	0	Not Stable

W-2186

Figure 7-13. RSINT User-Defined Interrupt

PC<6:5> Multibus Interrupt Drivers

Bits PC5 and PC6 are available to drive Multibus interrupts. The Multibus interrupts are active low signals. The 82C55A operating in mode 0 will latch the outputs. Figure 7-14 shows the PC5 and PC6 outputs.

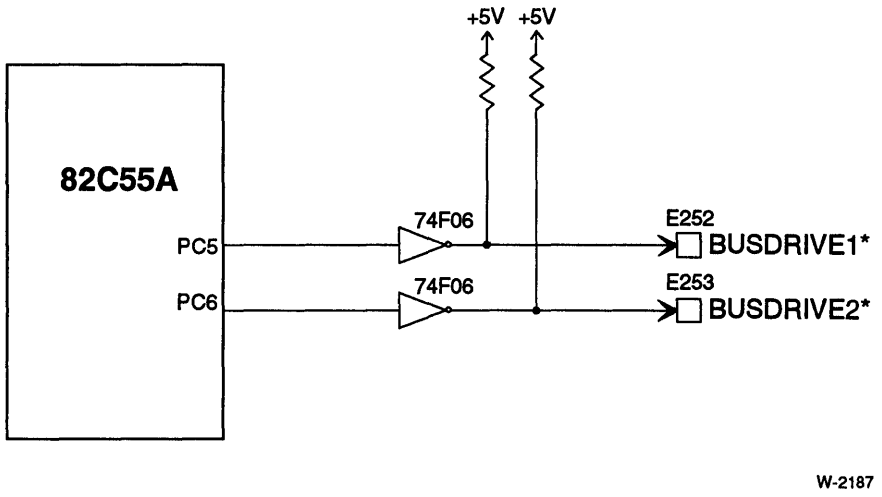
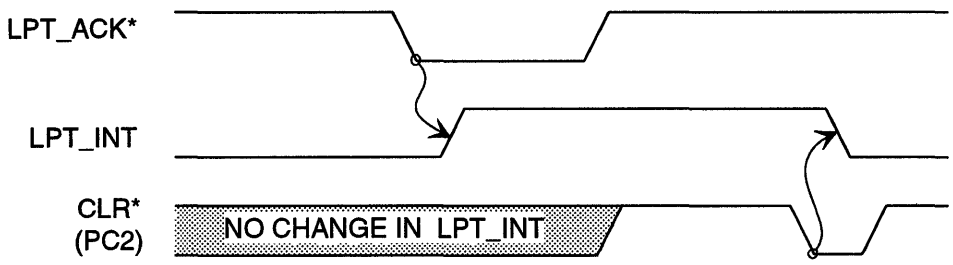


Figure 7-14. Multibus Interrupt Drivers (BUSDRIVE1*, BUSDRIVE2*)

PC7 LPT Interrupt Latch Clear Output

Figure 7-8 shows the LPT interrupt latch. Bit PC7 is an active low signal that clears the interrupt latch. Figure 7-15 shows the timing relationship between LPT Acknowledge*, LPT_INT, and CLR*.



LPT_ACK*	CLR* (PC2)	LPT_INT
1	1	No Change
1	0	0
0	1	1
0	0	Not Stable

W-2195

Figure 7-15. LPT Interrupt Latch Clear

The CLR* signal is an active-low signal: CLR* goes low to clear the latch.

PC7 LED DS4 Control Output (Optional). PC7 LED DS4 Control Output (Optional). Bit PC7 can be used to control LED DS4 if the LPT interrupt is not used. LED DS4 is on when PC7 is high. The function of DS4 can be defined by the user.

82C55A PPI Stake Pins

Table 7-11 lists the stake pins that affect parallel port operation.

Table 7-11. 82C55A PPI Stake Pins

Stake Pin	Description
E022	MBTOINT
E028	PFINT, inverted Power Fail Sense PFSN* signal
E038	LPT_INT
E051	RSINT, user-defined interrupt source from 74F74 latch
E052	CLREDGE1*, clears user-defined interrupt source
E055	SETEDGE1*, sets user-defined interrupt source
E061	PB0, User-defined Code 0
E062	Ground
E065	LPT_Busy
E066	DIR input to 74ALS245 transceiver (port.A)
E067	Ground
E071	PB1, User-defined Code 1
E072	Ground
E081	PB2, User-defined Code 2
E082	Ground
E252	PC5, BUSDRIVE1*
E253	PC5, BUSDRIVE2*

Note:
§ E061-E062 IN is the default configuration.

82C55A Programming

82C55A PPI I/O Addresses

Table 7-12 lists the I/O addresses for the 82C55A PPI control and data ports.

Table 7-12. 82C55A I/O Addresses

I/O Address	Size	R/W	Description
0C8H	Byte	R/W[1]	Port A data (input or output)[1]
0CAH	Byte	R	Port B data (input)
0CCH	Byte	W	Port C data (output)
0CEH	Byte	W	Control Word or Port C bit set/reset
0CEH	Byte	R	Invalid.

Note:
[1] Port A direction depends on configuration set by Control Word.

Supported Modes of Operation

The iSBC 486/12 board only supports the 82C55A Mode 0 (basic input/output). Modes 1 and 2 use bits from port C that are used for other functions on the iSBC 486/12 board.

Mode 0 has the following characteristics:

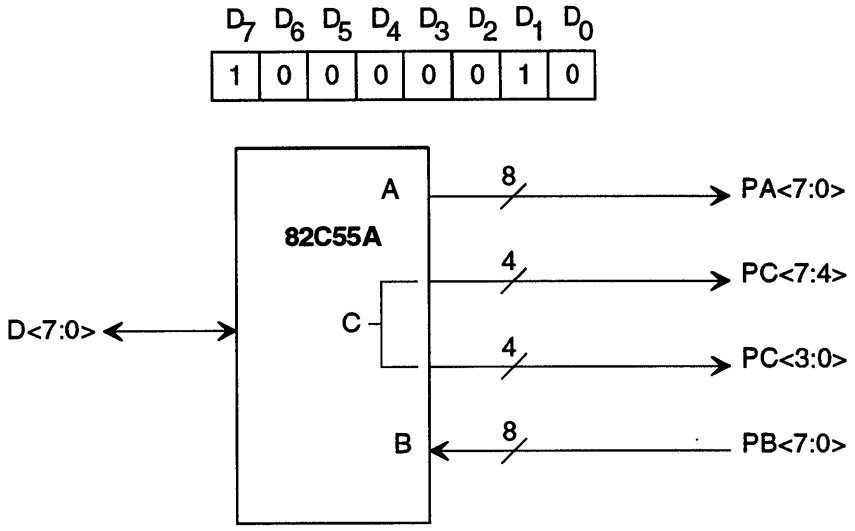
- two 8-bit ports (ports A and B)
- two 4-bit ports (ports C<7:4> and C<3:0>)
- outputs are latched
- inputs are not latched

On the iSBC 486/12 board, the direction of the 82C55A ports can be programmed as follows:

- Port A output or input
- Port B input only
- Port C output only

Figure 7-16 shows the 82C55A Control Words for the two mode 0 configurations.

Mode 0 Configuration, Port A Output
Control Word = 82H



Mode 0 Configuration, Port A Input
Control Word = 92H

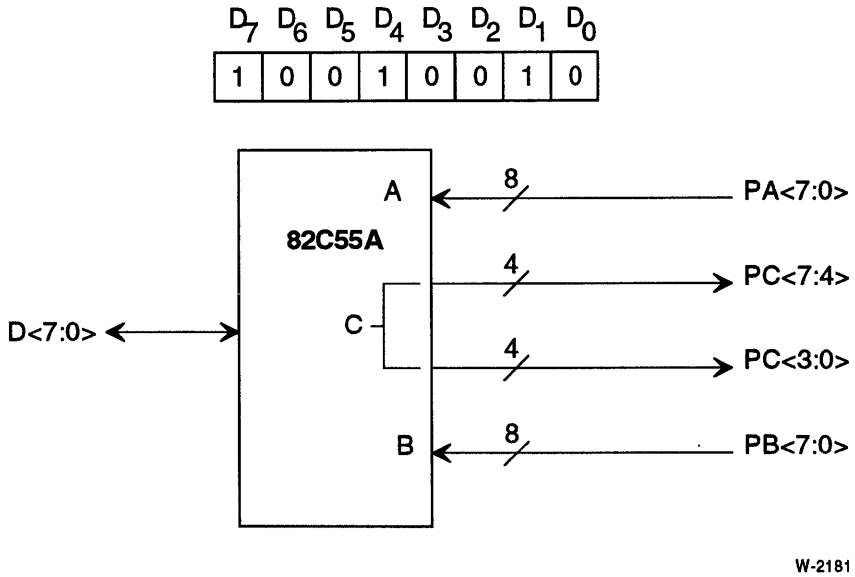


Figure 7-16. 82C55A Control Words

82C55A PPI Initialization

After power-up or a reset, the 82C55A is configured for mode 0 operation (basic I/O) with all ports as inputs. Your initialization routine must do the following:

1. Program the 82C55A Control Word for 82H (Mode 0, Port A direction configured for output, B for input, and Port C for output) or 92H (Mode 0, Port A direction configured for input, B for input, and Port C for output).
2. Configure the Dual-Port Lock (port PC1) to allow other boards to access the dual-port memory. After power up, the dual-port memory is locked.
3. Configure the NMI Mask (port PC2). After power-up, the NMI is enabled.

NOTE

Set bit PC2 to mask the interrupt sources (parity error and Multibus interface interrupt 0 in the default configuration). Using the NMI mask prevents a NMI before initialization is complete.

4. Configure the Multibus Time-Out (port PC3). After power up, the Multibus Time-out Interrupt is cleared and remains cleared until bit PC3 is set.
5. You may also want to read the user-defined code if your software uses this information (port PB<2:0>).

To initialize the 82C55A PPI, write the Control Word to control port 0CEH. Figure 7-17 is an assembly language programming example of 82C55A initialization. Refer to the *Peripheral Components* Handbook for complete 82C55A programming information.

```
; Intel 82C55 Programmable Peripheral Interface programming example
; Copyright Intel Corp., 1989, 1990
; This example is intended for your benefit in developing applications/
; systems using the Intel iSBC 486/12 Single Board Computers.  Intel
; hereby grants you permission to modify and incorporate as needed.

;Declarations:

PPI_CONTROL    EQU      0CEH      ;PPI control 3register
PPI_DEFAULT    EQU      082H      ;Default Port A & C Out, Port B In
PARITY         EQU      0E8H      ;Write to E8H clears parity error latch
MASK           EQU      00EH      ;Bit 3 =  enable TIME-OUT interrupt,
                                  ;Bit 2 =  disable NMI
                                  ;Bit 1 =  disable DUAL-PORT LOCK
PORT_C         EQU      0CCH      ;Port C on the PPI
CLEAR          EQU      00H


;Program:


MOV            AL, CLEAR         ;Clear parity error before programming PPI.
OUT            PARITY, AL        ;If parity is set, an NMI will occur after
                                 ;PPI is initialized.

MOV            AL, PPI_DEFAULT
OUT            PPI_CONTROL, AL   ;Initialize PPI so LED can be turned on
IN             AL, PORT_C        ;Read in port_C so NMI can be masked,
                                 ;Time-out enabled, and SW LOCK masked.

OR             AL, MASK          ;NMI masked,
OUT            PORT_C, AL        ;dual-port lock inactive,
                                 ;(enable time-out
```

Figure 7-17. 82C55A Programming Example

Bit Set/Reset Control of Port C

Table 7-13 shows the board functions that can be controlled using the bit set/reset feature of the PPI port C.

Table 7-13. Port C Bit Set/Reset Control Word

Control Word[1] (Binary)	Port C Bit Number	Operation (Set/Reset)	Function
0000 0000	PC0	Reset	LPT Strobe* goes high
0000 0001	PC0	Set	LPT Strobe* goes low
0000 0010	PC1	Reset	Dual-port lock active
0000 0011	PC1	Set	Dual-port lock inactive
0000 0100	PC2	Reset	NMI interrupts enabled
0000 0101	PC2	Set	NMI interrupts masked
0000 0110	PC3	Reset	Multibus time-out interrupt clear
0000 0111	PC3	Set	Multibus time-out interrupt enabled
0000 1000	PC4	Reset	RSINT clear and LED DS2 off
0000 1001	PC4	Set	RSINT enabled and LED DS2 on
0000 1010	PC5	Reset	Multibus Drive 1 high
0000 1011	PC5	Set	Multibus Drive 1 low (asserted)
0000 1100	PC6	Reset	Multibus Drive 2 high
0000 1101	PC6	Set	Multibus Drive 2 low (asserted)
0000 1110	PC7	Reset	Line Printer Interrupt clear and LED DS4 off
0000 1111	PC7	Set	Line Printer Interrupt enable and LED DS4 on

Note:
[1] I/O address 0CEH

```
; Copyright Intel Corp., 1989, 1990
; This example is intended for your benefit in developing applications/
; systems using the Intel iSBC 486/12 Single Board Computers.  Intel
; hereby grants you permission to modify and incorporate as needed.

CWR       EQU       0CEH          ;ADDRESS OF 82C55 CONTROL REGISTER

;*****
;         SET BIT 3 -- ENABLE MULTIBUS TIME-OUT INTERRUPT
;*****

          MVI       A, 00000111B        ;LOAD CONTROL WORD TO SET BIT 3
          OUT       CWR                 ;OUTPUT TO 82C55
```

Figure 7-18. Port C Bit Set/Reset Example

82C55A PPI Applications

Centronics-Compatible Printer

Connector J1 can be used as a Centronics-compatible printer port. To do this, you need a printer driver that transfers the data and controls the handshaking. This section discusses the Centronics-compatible signals on connector J1 and the basic handshaking protocol.

Table 7-14 describes the Centronics-compatible signals used with the iSBC 486/12 board.

Table 7-14. Centronics-Compatible Signals

Signal	Direction	Description
LPT_Data	Out	Data lines. The interface uses 8 data lines.
LPT_Strobe*	Out	Strobe*. This line is normally high and set low after the data is valid.
LPT_Busy	In	Busy. The printer uses this line to indicate when it is ready for data: Busy is set low when the printer is ready for data; high when the printer is not ready for data.
LPT_Ack*	In	Acknowledge*. This signal is active low. The printer sets this signal low to indicate it has received the data.
LPT_Error	In	PE (or paper empty). This signal goes high to indicate the printer requires operator attention (for example, the paper is out).

Figure 7-19 shows the general timing relationships for the signals on the Centronics interface.

The general sequence of events is as follows:

1. The computer places data on the data lines.
2. The computer asserts the active-low Strobe* signal to indicate that the data is valid.
3. The printer asserts the Busy signal to indicate that it is now receiving data (busy).
4. The printer pulses the Acknowledge signal to indicate it has received the data.
5. The printer deasserts the Busy signal to indicate it is ready again.
6. The computer waits for the Busy signal to go low and then places the next byte on the data lines.

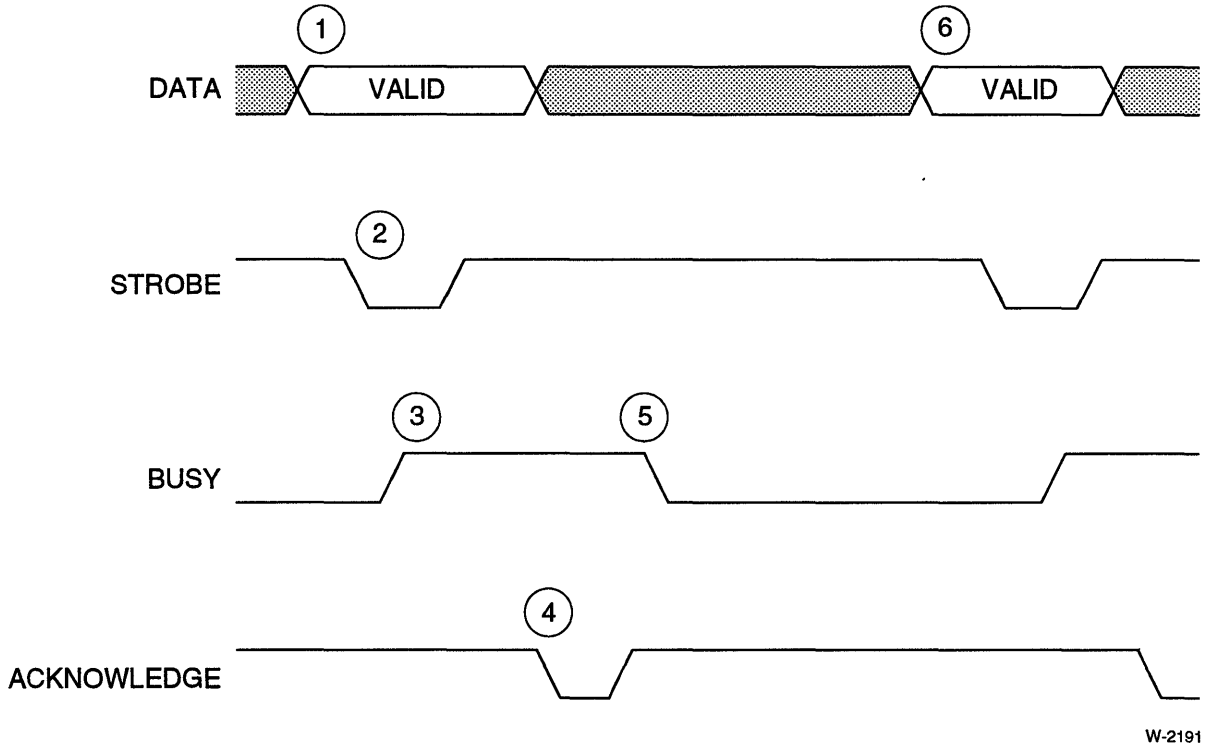


Figure 7-19. Centronics-Compatible Printer Timing

The timing requirements vary from printer to printer. Your printer driver must provide the correct timing delays to meet the specifications for your printer. Table 7-15 list the common timing parameters.

Table 7-15. Timing Parameters for Centronics-Compatible Printers

Parameter	Description
t_1	Data to Strobe* Low set-up time
t_2	Strobe* pulse width
t_3	Strobe* high to Data invalid hold off time
t_4	Strobe* low to Busy high
t_5	Busy high duration
t_6	Acknowledge* low pulse width
t_7	Acknowledge* high to Busy low
t_8	Busy low to next data

Figure 7-20 shows the general timing parameters for a Centronics-compatible printer.

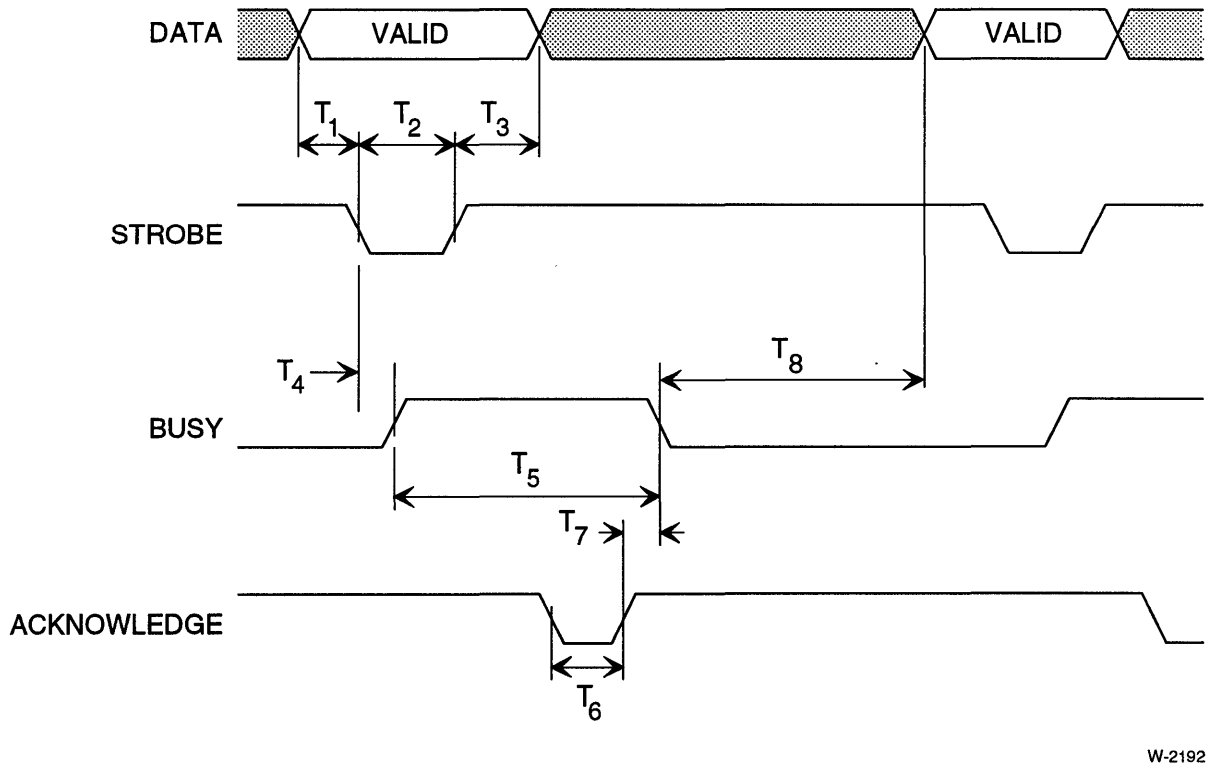


Figure 7-20. General Timing Parameters for Centronics-Compatible Printer

The general interrupt-driven procedure for sending data to the printer port is as follows:

1. When interrupted, clear interrupt by setting PC7 = 0.
2. Wait for ACK*-to-Busy-low plus Busy-low-to-next-cycle time.
3. Read Port B, check LPT_Busy and LPT_ACK*, wait for Busy = 0 and LPT_ACK* = 1.
4. Write data to port A.
5. Wait for data-valid-to-Strobe*-low-set-up time.
6. Set LPT_Strobe* low (bit PC0 = 1).
7. Wait for minimum Strobe*-on-pulse-width time.
8. Clear LPT_Strobe* (bit PC0 = 0).
9. Enable interrupts by setting PC7 = 1. When the printer has accepted the data it will pulse the Acknowledge signal. Asserting the LPT_ACK* signal will set the 74F74 flipflop high. The LPT_INT output from the flipflop results in an interrupt to the slave PIC level 7 (this is the default configuration).

You will also need a special procedure to wait after power-up for the printer to complete initialization. Some printers pulse the Acknowledge* signal once after power-up to indicate the printer is ready. You should then check the Error or Fault signal to verify that the printer is functioning properly.

NMI Mask Control

Bit 2, port C controls the NMI mask. The mask is used to mask the NMI input to the Intel486 CPU. (Chapter 8 discusses which interrupt sources can be connected to the NMI request line.)

The most important use of this bit is to disable NMI after a power-up reset to prevent spurious NMIs from parity errors while reading uninitialized DRAM. By default, the parity error interrupt is an input to the NMI. Before reading uninitialized DRAM, do the following:

1. Write a byte of any data to 0E8H to clear parity error interrupt.
2. Set bit PC2 to 1 to mask NMI.
3. Read or write the DRAM.
4. Write a byte of any data to 0E8H to clear parity error interrupt again.
5. Reset bit PC2 to 0 to enable NMI.

Switching the Board Between Real and Protected Modes

The iSBC 486/12 board, like the Intel486 CPU, has two addressing modes of operation: real address mode, and protected virtual address mode. Both the iSBC 486/12 board and the Intel486 CPU should operate in the same mode. After power-up, the iSBC 486/12 board and the Intel486 CPU operate in real mode.

To switch from real mode to protected mode:

1. Create any protected mode data structures needed for your application (this is discussed in the *Intel486™ Microprocessor Family Programmer's Reference Manual*).

2a. **Method 1.** Perform a **word** write to 0CAH using any data value to switch the board into protected mode. This method is compatible with the iSBC 286/12 Single Board Computer.

NOTE

I/O address 0CAH is also used as the address for 82C55A Port B data. Performing a word write to this address has no effect on the 82C55A because port B is initialized as an input port.

2b. **Method 2 (Preferred).** Perform a **byte** write to 0E4H using data D0 = 1 to switch the board into protected mode.

3. Switch the Intel486 CPU into protected mode (this is discussed in the *Intel486™ Microprocessor Family Programmer's Reference Manual*).

To switch from protected mode to real mode, you can reset the board using the software reset, or use the following method:

1. Create any real mode data structures needed for your application (this is discussed in the *Intel486™ Microprocessor Family Programmer's Reference Manual*).

2. Perform a **byte** write to 0E4H using data bit D0 = 0 to switch the board into protected mode.

3. Switch the Intel486 CPU into real mode (this is discussed in the *Intel486™ Microprocessor Family Programmer's Reference Manual*).

To determine which mode the board is in, do a byte read to I/O address 0E4H. If D0 = 0, the board is in real mode. If D0 = 1, the board is in protected mode.

MB87033B SCSI Peripheral Controller (SPC)

This section discusses the following topics:

- SCSI overview
- SCSI ID jumpers
- SCSI termination resistors
- SCSI termination power
- SCSI interrupt
- Programming the Fujitsu MB87033B SCSI Protocol Controller (SPC)
- Programming the SCSI Control Register

Refer to Chapter 8 for information on DMA operations using the SCSI interface.

SCSI Interface Overview

iSBC® 486/12S SCSI Block Diagram

The I/O interface between the CPU and SPC allows programming and status checking by the CPU. The SPC also has a separate DMA bus which is buffered to the ADMA data bus by 256 x 9 FIFO buffers. This arrangement decouples the I/O bus from the slower SCSI bus.

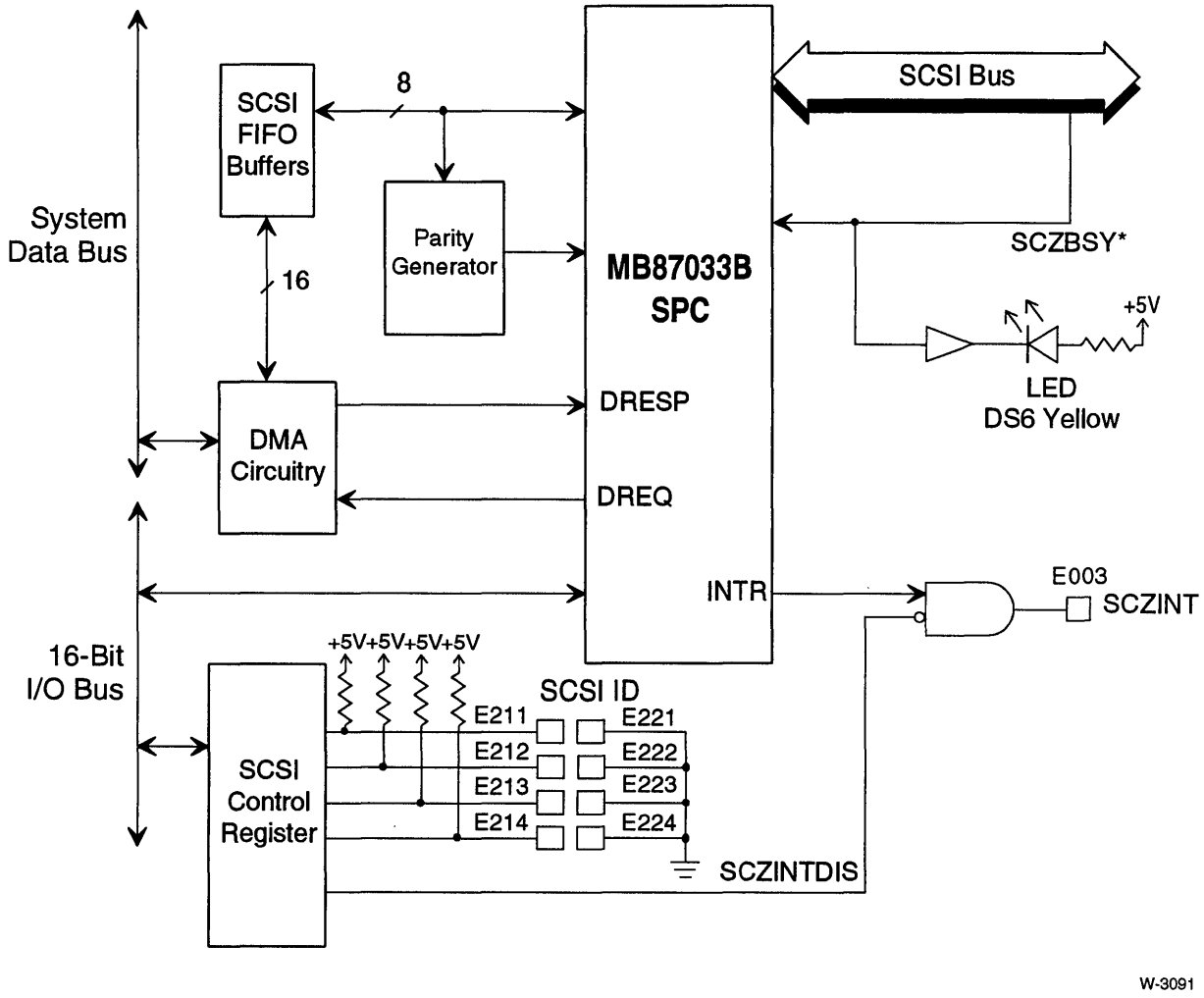


Figure 7-21. SCSI Interface Block Diagram

SCSI Signals

Table 7-16 lists the signals on the SCSI interface.

Table 7-16. SCSI Signals

Signal	Description
DB<7:0>, DBP	8-bit data bus, one parity bit. The optional parity bit is always odd. Bit ordering is from right to left, low to high (DB7 is the most significant data bit).
BSY*	Busy. This signal is driven low to indicate that the SCSI bus is busy.
SEL*	Select. An initiator asserts SEL* signal to select a target. A target asserts SEL* to select an initiator.
C/D*	Control/Data. This signal indicates the type of data on the bus. Control data is on the bus when C/D* is asserted.
I/O*	Input/Output. Data direction on the SCSI bus. I/O* is asserted indicates the data direction is from the target to the initiator.
MSG*	Message. This signal is asserted during the message phase.
REQ*	Request. This signal is the first part of the REQ*/ACK* handshake.
ACK*	Acknowledge. This signal is the second part of the REQ*/ACK* handshake.
ATN*	Attention. This signal is driven to indicate the attention condition.
RST*	Reset. This signal indicates a reset condition.

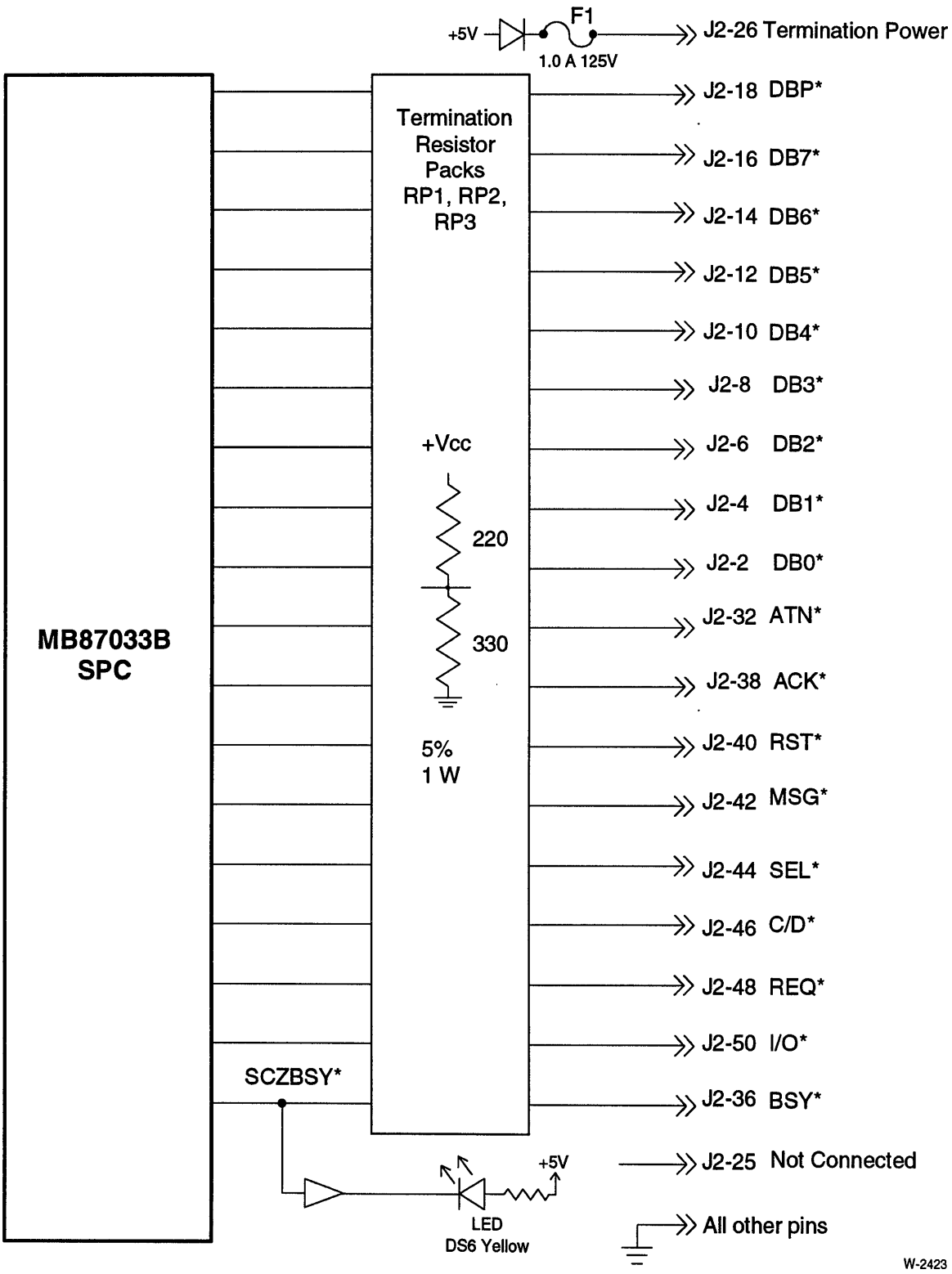


Figure 7-22. SCSI Connector J2 Pin-Out

SCSI ID Jumpers and GP3 Bit Jumper

There are three jumpers that configure the SCZID register. The SCSI ID jumpers establish a three-bit binary code that determines the SCSI ID number (0-7) assigned to the iSBC 486/12S. A jumper installed across two stake pins establishes a binary 0 (see Table 7-17). When no jumper is installed, a binary 1 is established.

Table 7-17 shows the SCSI ID jumpers. Table 7-18 shows the General Purpose GP3 Bit jumper.

Table 7-17. SCSI ID Jumpers

SCSI ID	E212-E222	E213-E223	E214-E224
0	IN	IN	IN
1	IN	IN	OUT
2	IN	OUT	IN
3	IN	OUT	OUT
4	OUT	IN	IN
5	OUT	IN	OUT
6	OUT	OUT	IN
7§	OUT	OUT	OUT

Note:
§ default SCSI ID

Table 7-18. SCZID General Purpose Bit GP3 Jumpers

GP3 Bit	E084-E211
0	IN
1§	OUT

Note:
§ default

SCSI Termination Resistors

The SCSI devices located on both ends of the SCSI bus must have termination resistors to pull the signal lines up. The iSBC 486/12S provides three sockets for the termination resistors (RP1, RP2 and RP3 shown in Figure 7-23). Chapter 3 provides installation instructions for the termination resistors.

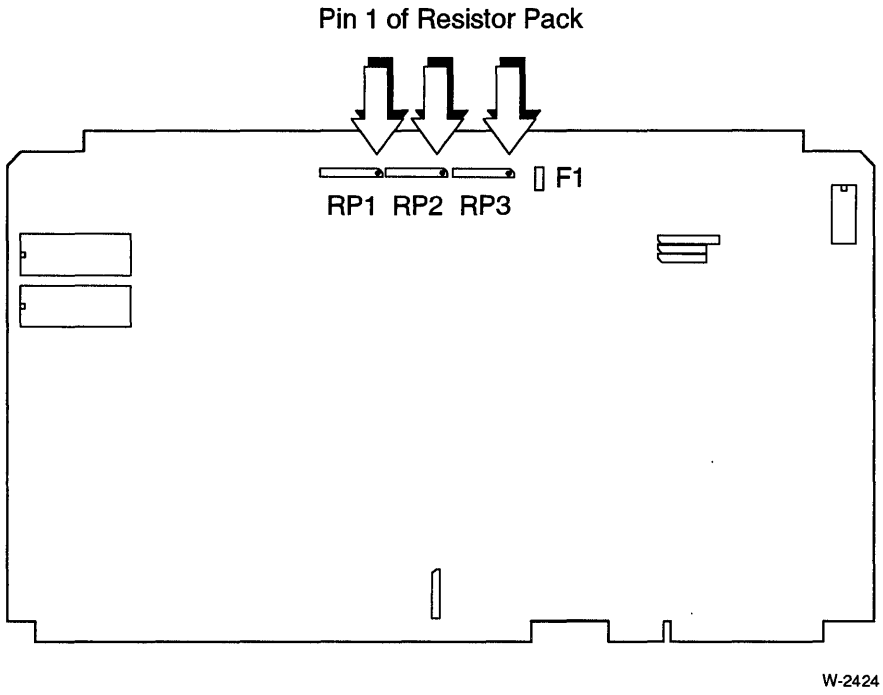


Figure 7-23. SCSI Termination Resistor Packs RP1, RP2, RP3

In the default configuration, RP1, RP2, and RP3 are installed. If you are connecting the iSBC 486/12S to the end of the SCSI cable, leave the terminations installed. If you do not connect the board to the end of the SCSI cable, remove RP1, RP2, and RP3.

SCSI Termination Power

The iSBC 486/12S board can power the termination resistors on both ends of the SCSI bus. This is useful when one or both devices on the end of the cable is powered off. A socketed fuse allows the user to provide or remove termination power to the SCSI bus. Figure 7-23 shows the location of fuse F1. Power is supplied to the SCSI bus when the fuse is installed (default).

Vendor information for the fuse is listed in Chapter 3.

MB87033B SPC Registers

The iSBC 486/12S board has a Fujitsu MB87033B SCSI Protocol Controller (SPC) that supports both synchronous and asynchronous data transfers. The following discussion is an overview of SPC operation on the iSBC 486/12. For further information, refer to the *Fast Track To SCSI* manual published by Fujitsu.

MB87033B SPC Register I/O Addresses.

Table 7-19 lists the board I/O addresses for the internal SPC registers. See page 7-51 for the board-level SCSI control registers.

Table 7-19. SPC I/O Register Addresses

I/O Address	Operation	Register Name
181H	Byte R/W	Bus Device ID
183H	Byte R/W	SPC Control
185H	Byte R/W	Command
187H	Byte R/W	Transfer Mode
189H	Byte R	Interrupt Sense
189H	Byte W	Reset Interrupt
18BH	Byte R	Phase Sense
18BH	Byte W	SPC Diagnostic Control
18DH	Byte R	SPC Status
18DH	Byte W	--
18FH	Byte R	SPC Error Status
18FH	Byte W	--
191H	Byte R/W	Phase Control
193H	Byte R	Modified Byte Counter
193H	Byte W	Extended Transfer Count[1]
195H	Byte R/W	Data Register
197H	Byte R/W	Temporary Register
199H	Byte R/W	Transfer Counter High
19BH	Byte R/W	Transfer Counter Middle
19DH	Byte R/W	Transfer Counter Low
19FH	Byte R/W	External Buffer

Note:
[1] Upper 4 bits of 28-bit transfer counter. Consult the latest Fujitsu MB87033B data sheet for details.

MB87033B SPC Programming

There are two basic steps for data transfers via the SCSI interface:

1. The SPC is programmed for a specific phase.
2. The ADMA and SCSI I/O control registers are programmed to transfer the data.

The SPC generates an interrupt on the completion of a transfer, an error condition, or a phase change. If the SCZINTDIS register is not set (0), the interrupt goes to the 82C59A PIC. The SPC INTS register can be read to determine the cause of the interrupt.

In the default configuration, the SPC interrupt (SCZINT) is jumpered through E003-E013 to the master PIC at interrupt level 2 (MIR2).

CAUTION

Do not mask the SPC interrupt using the SPC internal interrupt register. The SPC interrupt is used during data transfers to synchronize the FIFO and, if masked, will result in incomplete data transfers. The SCZINTDIS register is provided to prevent the SPC interrupt from reaching the CPU.

SCSI Control Registers

The SCSI Control Registers are implemented in discrete logic on the iSBC 486/12S board. They control different aspects of the SCSI interface. After a power-up or hardware reset, all programmable registers are cleared to 0H. The SCZID register is set by hardware jumpers. Table 7-20 lists the I/O addresses for the SCSI Control registers.

NOTE

Except for the SCSI ID register, each register uses only bit 0; all other bits should be zero.

Table 7-20. SCSI Control Register Addresses

I/O Address	Operation	Register Name
1A1H	W	SCZINTDIS. SCSI interrupt disable.
1A1H	R	SCZID. SCSI ID
1A3H	R/W	BLAST. DMA Blast Mode
1A5H	R/W	HIN. Data direction for DMA transfers.
1A7H	R/W	BYTXFER. DMA transfer data width.
1A9H	R/W	TURBO. SCSI DMA transfer mode.
1ABH	W	SCZRST. Reset SCSI FIFOs and support circuitry.
1ADH	–	Reserved
1AEH	R/W	Address of FIFO data for two-cycle DMA.
1AFH	–	Reserved

SCZID Register (1A1H - Read)

This is a general purpose four-bit register that indicates the SCSI ID value. The SCSI ID value is set by jumpers shown in Tables 7-17 and 7-18 (page 7-48). This value is used when programming the MB87033B SCSI ID register. Table 7-21 shows the bit definitions for the SCSI ID register bits<3:0> (bits<7:4> are zero).

Table 7-21. SCSI ID Register Bits<3:0>

Bit 3	Bit 2	Bit 1	Bit 0
GP3	GP2/SID2	GP1/SID1	GP0/SID0

Note:
default = 0FH

Bits<7:4> Not used.

Bit 3 GP3 is a general-purpose bit (user-defined by jumper E084-E211).

Bits 2-0 SID<2:0> form a three-bit binary code that indicates the SCSI ID assigned to the SCSI interface on the iSBC 486/12S board. SID2 is the most significant bit, and SID0 is the least significant bit. These bits can also be used with GP3 to form a general purpose register.

See Tables 7-17 and 7-18 (page 7-48) for jumper information.

SCZINTDIS Register (1A1H - Write)

This register masks the SPC interrupt signal.

If bit 0 is cleared (0), the routing of the SPC interrupt to the 82C59A PIC is enabled.

If bit 0 is set (1), the SPC interrupt is disabled. If the SPC interrupt is disabled, the CPU must then poll the SPC for phase changes.

CAUTION

Do not mask the SPC interrupt using the SPC internal interrupt register. The SPC interrupt is used during data transfers to synchronize the FIFO and, if masked, will result in incomplete data transfers. The SCZINTDIS register is provided to prevent the SPC interrupt from reaching the CPU.

BLAST Register (1A3H)

This register controls the DMA blast mode of the DAG/ADMA/FIFO logic. This register is only valid when the TURBO register bit 0 is set (1).

Table 7-22 defines the blast register functions.

Table 7-22. Blast Register Functions

BLAST Bit 0	TURBO Bit 0	Result
X	0	Blast register has no effect.
0	1	The FIFO expects one-cycle or two-cycle 8 or 16-bit DMA transfers.
1	1	The ADMA/DAG will transfer 16 bytes of data during each ADMA cycle.

Note:
"X" means don't care

To operate the ADMA/DAG in blast mode, do the following:

1. Program the ADMA for one-cycle, 16-bit operation.
2. Set the ADMA byte count to 1/16 the actual value used in the SPC.
3. Set the DAG to burst mode.
4. Set the TURBO bit 0 to 1.
5. Set the BLAST bit 0 to 1.

HIN Register (1A5H)

This register controls the direction of data transfer during DMA operations to and from the SPC. The register should be set to the proper value before starting the DMA operation.

If bit 0 is cleared (0), the direction is as follows:

Memory → FIFO → SPC → SCSI interface.

If bit 0 is set (1), the direction is as follows:

Memory ← FIFO ← SPC ← SCSI interface.

BYTXFER Register (1A7H)

This register sets the FIFO data width (8- or 16-bit) during DMA operations. The register should be set before initiating the DMA operation.

If bit 0 is cleared (0), SCSI DMA is word transfer.

If bit 0 is set (1), SCSI DMA is byte transfer.

TURBO Register (1A9H)

This register sets the DMA request mode for the SCSI data FIFO.

If bit 0 is cleared (0), DMA request is asserted whenever data is available in, or required by the FIFO.

When bit 0 is set (1), data is transferred in 16-byte bursts whenever the FIFO reach a 128-byte threshold. This mode maximizes effective use of the system bus.

SCZRST Register (1ABH - Write Only)

Writing a 1 to bit 0 in this register resets the SCSI FIFO data pointer to 0 and initializes the SCSI support circuitry. Data contained in the FIFO is lost after this register is set. The FIFO should be reset before each transfer is started.

FIFO Data (1AEH)

This I/O address is used to access the FIFO data in non-burst DMA transfers.

8274 Multiple-Protocol Serial Controller (MPSC)

The 8274 Multiple-Protocol Serial Controller is a two-channel, multiple protocol serial controller. It supports RS-232C and RS-422A asynchronous and synchronous operation. You can configure Channel A for DCE or DTE operation. Channel B is configured for DCE operation. Figures 7-24, 7-25, and 7-26 illustrate the various configurations that are available.

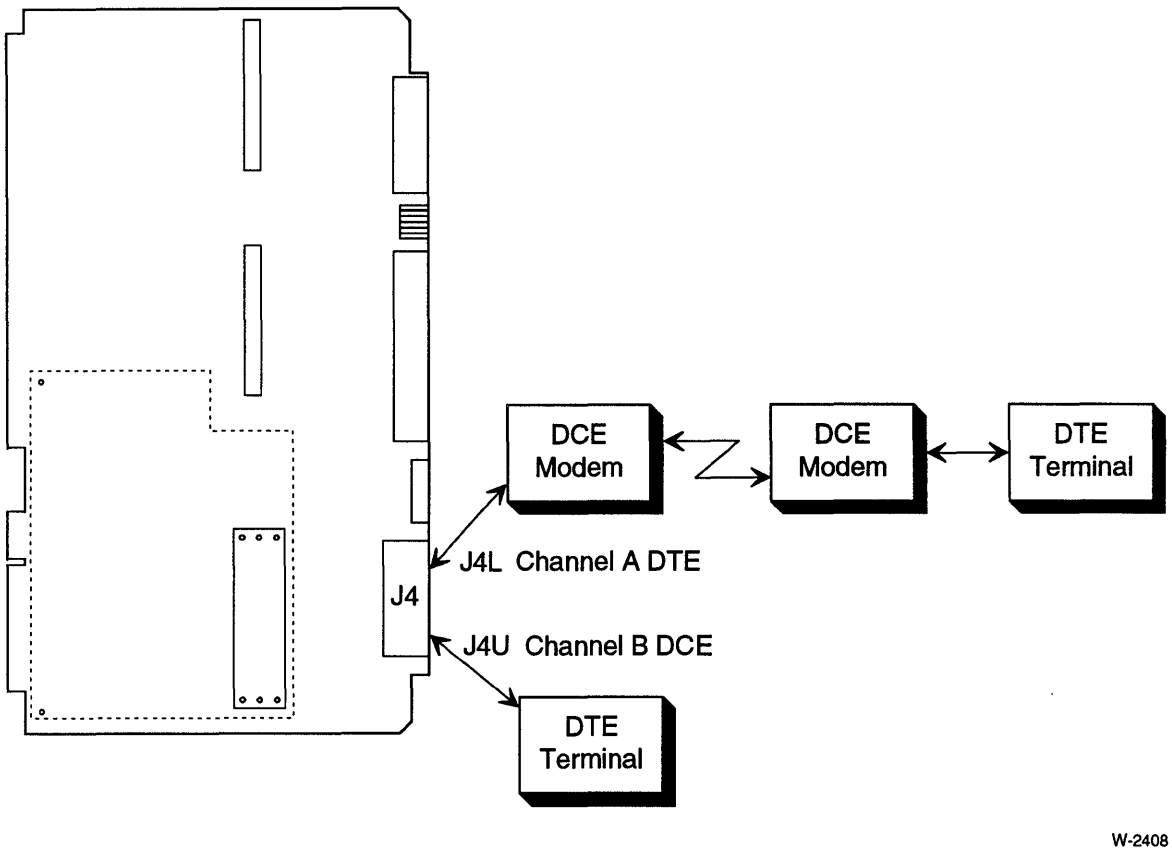


Figure 7-24. Channel A Configured for DTE Operation and Channel B Configured for DCE Operation

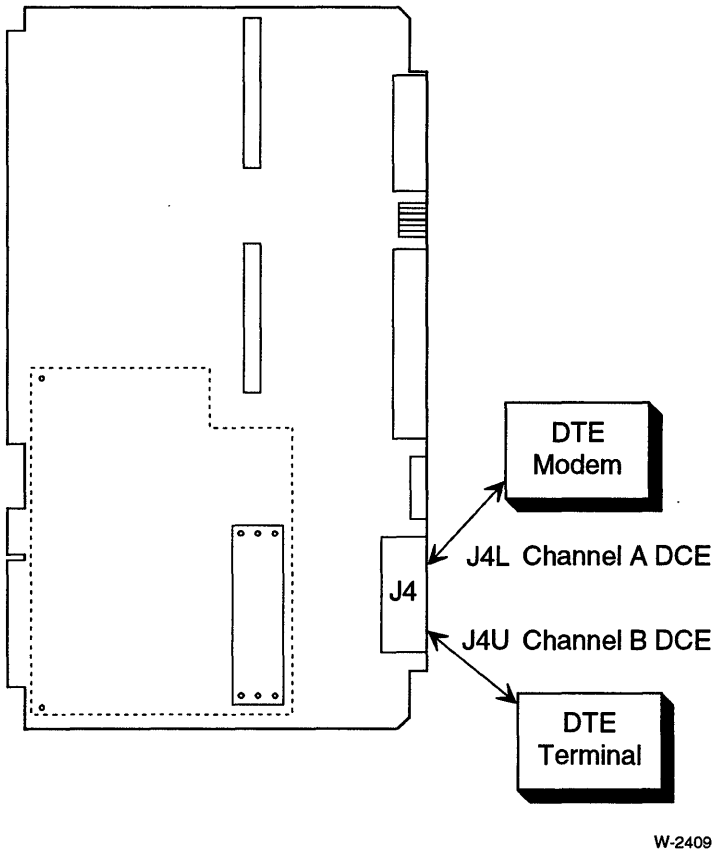


Figure 7-25. Channel A and B Configured for DCE Operation

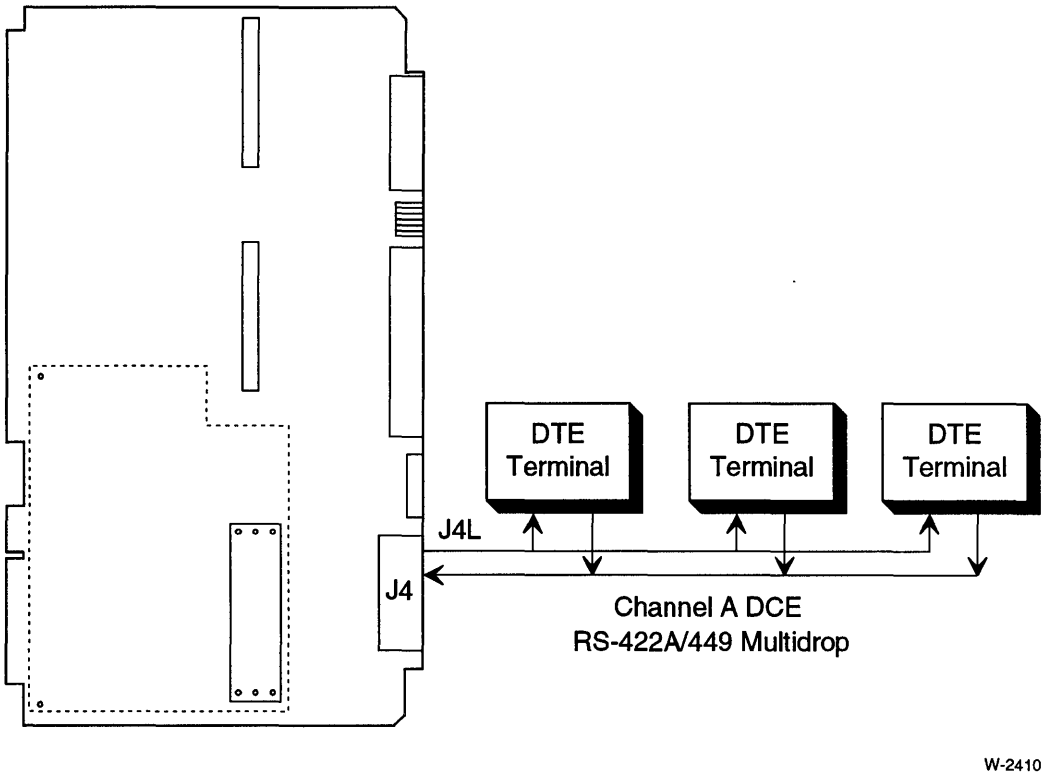


Figure 7-26. RS-422A Multidrop Operation

Serial Channel Configurations

The iSBC 486/12 board provides two serial channels at connector J4 using the Intel 8274 Multiple-Protocol Serial Controller (MPSC). Channel A (connector J4L) can be configured for RS-232C and RS-422A operation.

Table 7-23. Serial Channel Configurations

Mode	8274 MPSC Channel A (J4L)	8274 MPSC Channel B (J4U)	Figure
RS-232C DCE	X	X	Figure 7-27 Figure 7-28
RS-232C DTE	X		Figure 7-29
RS-422A DCE	X		Figure 7-30

The Intel *Microcommunications* Handbook provides the pin descriptions for the 8274 MPSC.

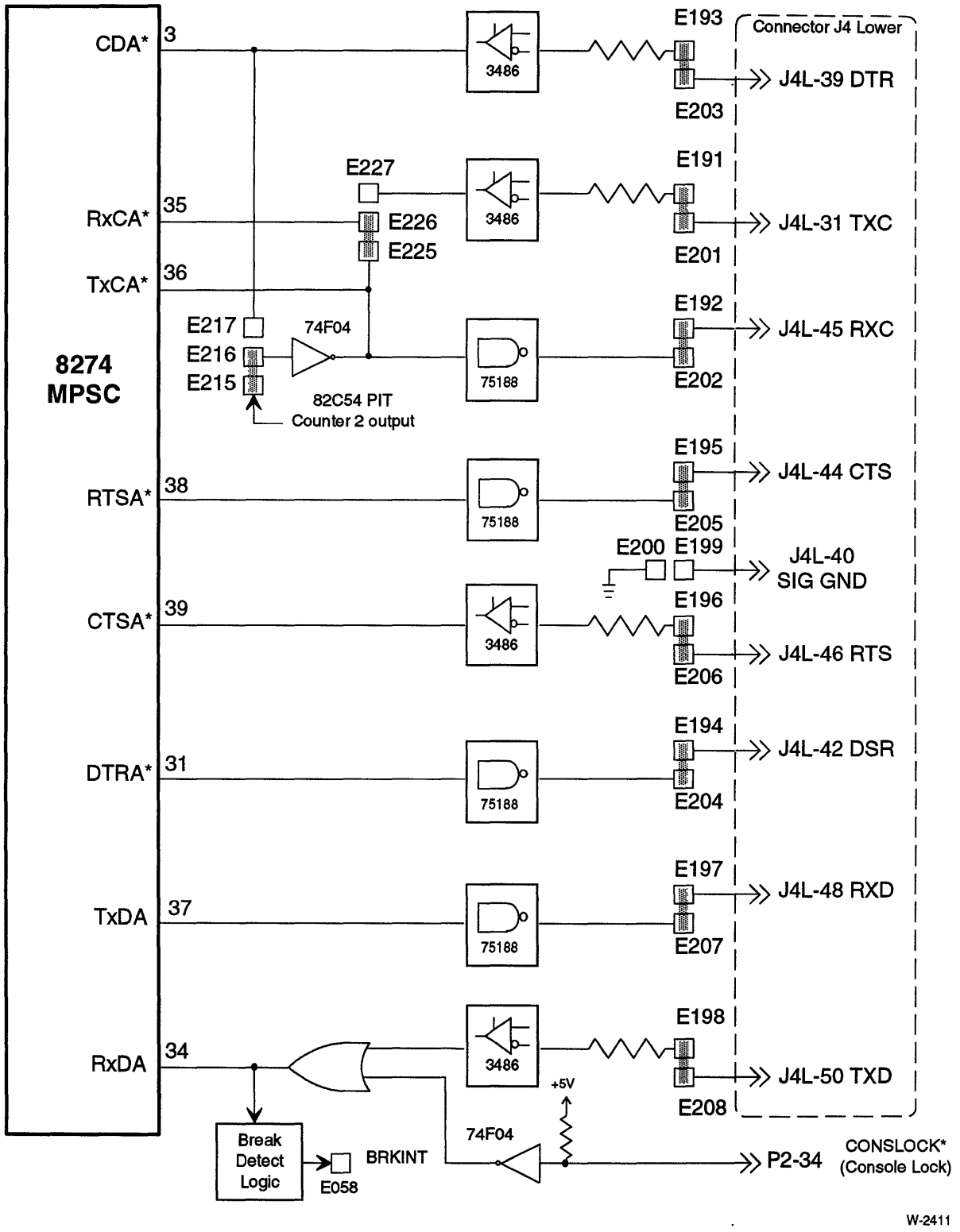


Figure 7-27. Channel A RS-232C DCE

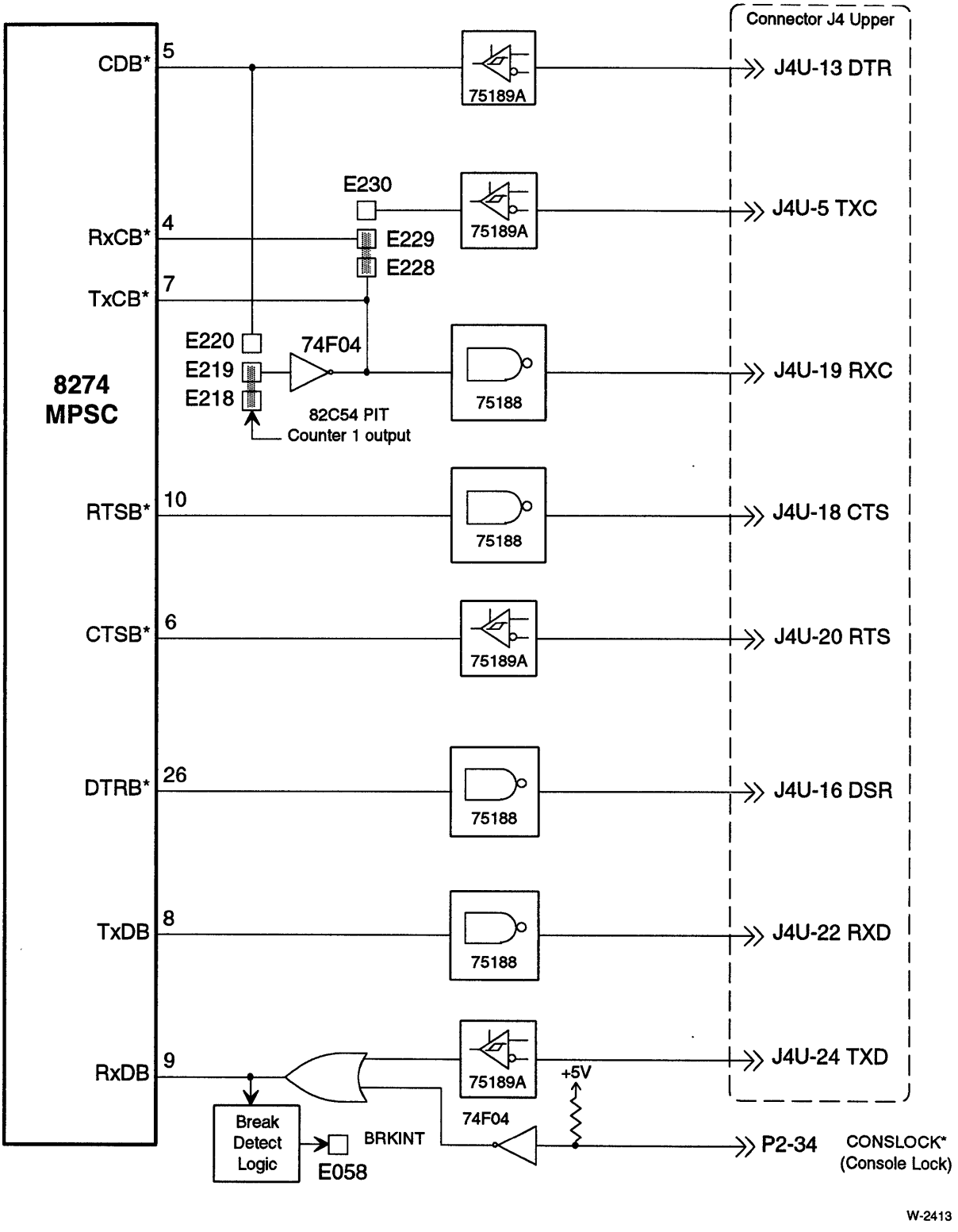


Figure 7-28. Channel B RS-232C DCE

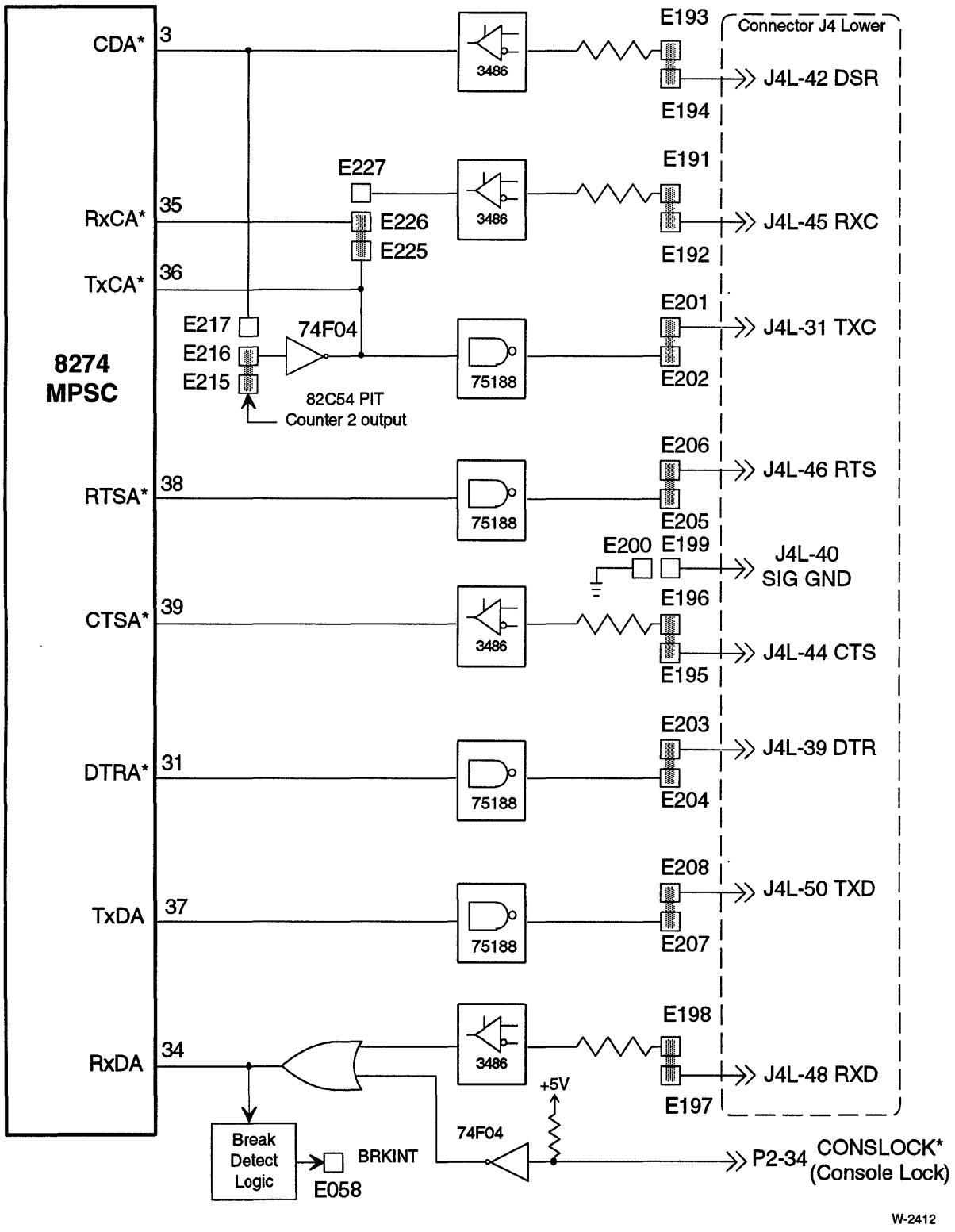


Figure 7-29. Channel A RS-232C DTE

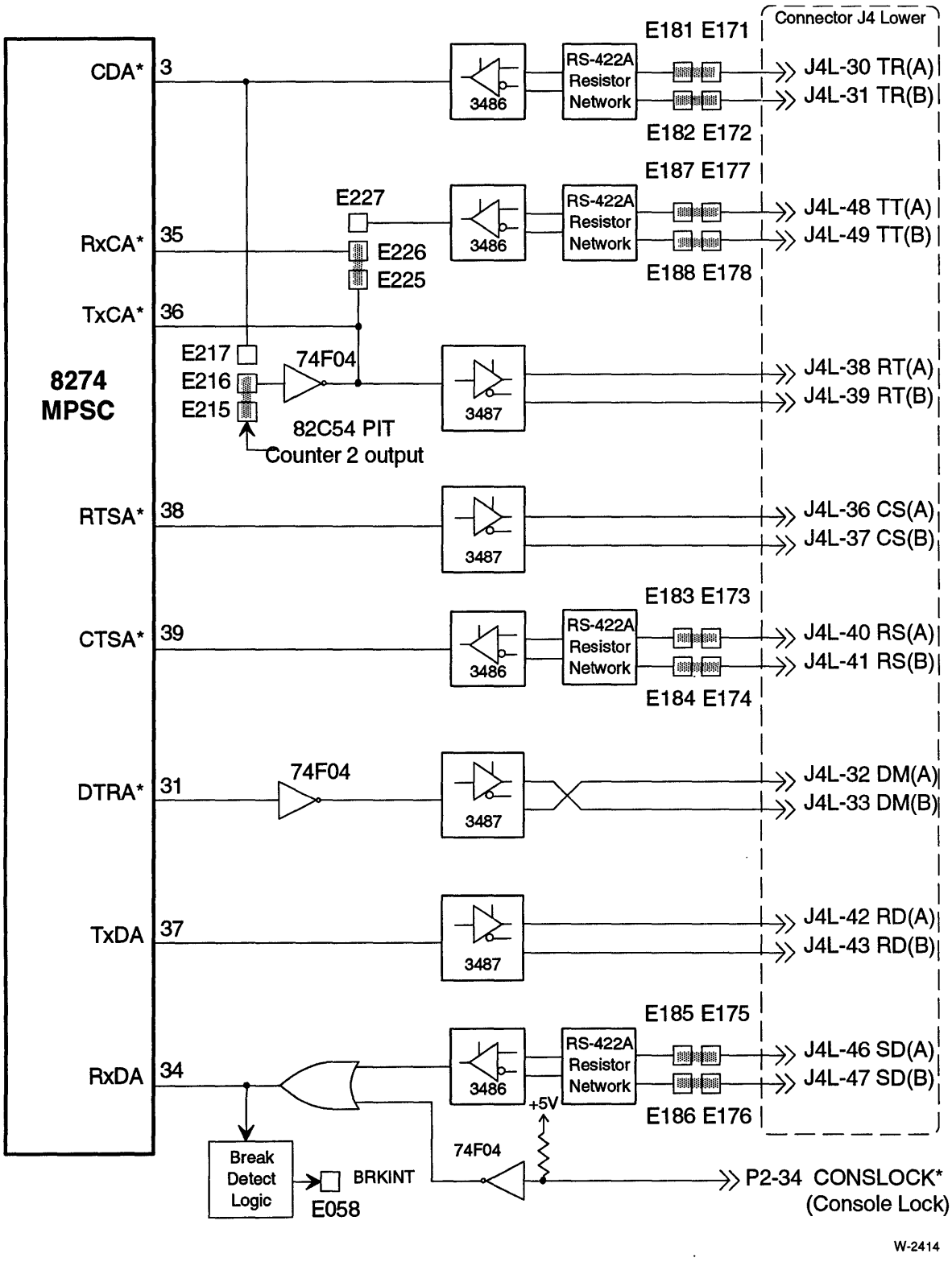


Figure 7-30. Channel A RS-422A DCE

Serial Connectors

Table 7-24. Serial Interface Connectors (J4)

8274 MPSC Channel	Connector	Connector Type	Pin Numbers
A	J4 lower	26-pin shrouded header	27 through 52
B	J4 upper	26-pin shrouded header	1 through 26

Chapter 9, "Connectors and Cables," shows the pin numbering for connector J4 and describes the various serial cables.

MPSC I/O Ports

Table 7-25. 8274 MPSC I/O Addresses

I/O Address	Read/Write	Size	Function
0D8H	R/W	Byte	Channel A data
0DAH	R/W	Byte	Channel B data
0DCH	R/W	Byte	Channel A control
0DEH	R/W	Byte	Channel B control

MPSC Configuration

Default Serial Configuration

The default configuration is configured as follows:

- Channel A RS-232C DCE (Connector J4L)
  - –Channel A RxC connected to Channel A TxC
  - –Channel A TxC connected to 82C54A PIT counter 2 output
- Channel B RS-232C DCE (Connector J4U)
  - –Channel B TxC connected to Channel B TxC
  - –Channel B RxC connected to 82C54A PIT counter 1 output

Notes:
TxC is the transmit clock
RxC is the receive clock

Table 7-26 lists the default serial jumpers.

Table 7-26. Default Serial Jumpers

Function	Jumper	Default Configuration
RS-422A/449	E171-E181 E172-E182 E173-E183 E174-E184 E175-E185 E176-E186 E177-E187 E178-E188 E179-E189	OUT OUT OUT OUT OUT OUT OUT OUT OUT
RS-422A/449 Multidrop	E180-E190	OUT
RS-232C DCE	E191-E201 E192-E202 E193-E203 E194-E204 E195-E205 E196-E206 E197-E207 E198-E208 E199-E209	IN IN IN IN IN IN IN IN IN
RS-232C DTE	E191-E192 E193-E194 E195-E196 E197-E198 E201-E202 E203-E204 E205-E206 E207-E208	OUT OUT OUT OUT OUT OUT OUT OUT
Shield	E200-E210	IN
Pit Counter 2 to TxCA CDA to TxCA PIT Counter 1 to TxCB CDB to TxCA TxCA to RxCA TxC to RxCA TxCB to RxCB TxC to RxCB	E215-E216 E216-E217 E218-E219 E219-E220 E225-E226 E226-E227 E228-E229 E229-E230	IN OUT IN OUT IN OUT IN OUT

Table 7-27. Default Serial Components

Component	Default Configuration
RP4	OUT
RP5	IN, pin 1 of RP5 in pin 1 of socket
RP6	IN, pin 1 of RP6 in pin 1 of socket
U14 (3487 driver)	OUT

Channel A RS-232C DCE Configuration

This is the default configuration. Refer to the following:

- Figure 7-27 (page 7-58)
- Table 7-26 (page 7-63) lists the jumpers
- Table 7-28 lists the configuration options

Table 7-28. Channel A RS-232C DCE Configuration Options (Connector J4L)

Configuration Option	Jumper	Configuration
Receive clock input connected to transmit clock	E225-E226 E226-E227	IN§ OUT§
RxCA* clock input connected to timing signal from J4L-31 TxC	E225-E226 E226-E227	OUT IN
TxCA* clock connected to Counter 2 output	E215-E216 E216-E217	IN§ OUT§
TxCA* clock connected to J4L-39 DTR timing signal	E215-E216 E216-E217	OUT IN

Note:
§ default

Channel A RS-232C DTE Configuration

To configure channel A for DTE operation, refer to the following:

- Figure 7-29 (page 7-60)
- Table 7-29 for jumpers
- Table 7-30 for jumper options
- Table 7-31 (page 7-66) for configuration of the resistor packs

Table 7-29. Channel A RS-232C DTE Jumpers (Connector J4L)

Jumper	Jumper Configuration
E171-E181 E172-E182 E173-E183 E174-E184 E175-E185 E176-E186 E177-E187 E178-E188 E179-E189 E180-E190	OUT OUT OUT OUT OUT OUT OUT OUT OUT OUT
E191-E201 E192-E202 E193-E203 E194-E204 E195-E205 E196-E206 E197-E207 E198-E208	OUT OUT OUT OUT OUT OUT OUT OUT
E191-E192 E193-E194 E195-E196 E197-E198 E201-E202 E203-E204 E205-E206 E207-E208	IN IN IN IN IN IN IN IN

Table 7-30. RS-232C DTE Configuration Options (Connector J4L)

Configuration Option	Jumper	Configuration
Receive clock input connected to transmit clock	E225-E226 E226-E227	IN§ OUT§
RxCA* clock input connected to timing signal from J4L-45 RxC	E225-E226 E226-E227	OUT IN
TxCA* clock connected to Counter 2 output	E215-E216 E216-E217	IN§ OUT§
TxCA* clock connected to J4L-42 DSR timing signal	E215-E216 E216-E217	OUT IN

Note:
§ default

Table 7-31. Configuration of RS-232C DTE Serial Components (Connector J4L)

Component	Configuration
RP4	OUT
RP5	IN, pin 1 of RP5 in pin 1 of socket
RP6	IN, pin 1 of RP6 in pin 1 of socket
U14 (3487 driver)	OUT

Channel A RS-422A/449 DCE Configuration

To configure channel A for RS-422A/449 DCE operation, refer to the following:

- Figure 7-30 (page 7-61)
- Table 7-32 for jumpers
- Table 7-33 for jumper options
- Table 7-34 (page 7-67) for resistor and driver configuration

Table 7-32. Channel A RS-422A DCE Jumpers (Connector J4L)

Jumper	Jumper Configuration
E171-E181 E172-E182 E173-E183 E174-E184 E175-E185 E176-E186 E177-E187 E178-E188 E179-E189	IN IN IN IN IN IN IN IN IN
E191-E201 E192-E202 E193-E203 E194-E204 E195-E205 E196-E206 E197-E207 E198-E208	OUT OUT OUT OUT OUT OUT OUT OUT
E191-E192 E193-E194 E195-E196 E197-E198 E199-E209 E201-E202 E203-E204 E205-E206 E207-E208	OUT OUT OUT OUT OUT OUT OUT OUT OUT

Table 7-33. RS-422A Configuration Options

Configuration Option	Jumper	Configuration
Cable shield connected to ground	E200-E210	IN
Cable shield not connected to ground	E200-E210	OUT§
Receive clock input connected to transmit clock	E225-E226 E226-E227	IN§ OUT§
RxCA* clock input connected to Transmit Timing (TT) signal from J4L	E225-E226 E226-E227	OUT IN
TxCA* clock connected to Counter 2 output	E215-E216 E216-E217	IN§ OUT§
TxCA* clock connected to J4-30/J4-31 TR timing signal	E215-E216 E216-E217	OUT IN

Note:
§ default

Table 7-34. Configuration of RS-422A Serial Components

Component	Configuration
RP4	OUT[1] §
RP5	IN[1], pin 1 of RP5 in pin 1 of socket §
RP6	IN[1], pin 1 of RP6 in **pin 6** of socket
U14 (3487 driver)	IN

Note:
[1] Insert RP4 and remove RP5 and RP6 if iSBC is last slave in multidrop line (refer to Appendix C, "Multidrop Information").

CAUTION

To convert from RS-232C to RS-422A DCE, you must add a 3487 RS-422A Signal Line Driver in socket U14 and rotate resistor pack RP6, so that RP6 pin 1 is installed in pin 6 of the socket.

Chapter 3 gives installation procedures for each of these parts.

Channel A RS-422A Multidrop Configuration:

Table 7-35 shows the jumpers for RS-422A/449 multidrop operation on channel A. Appendix C, "Multidrop Information," gives information on how to compute the values for bias resistor packs RP4, RP5, and RP6.

Table 7-35. RS-422A Multidrop Configuration Jumpers

Configuration Option	Jumper	Configuration
Enable multidrop operation	E180-E190	IN
Disable multidrop operation	E180-E190	OUT§

Note:
§ default

Serial Interrupt Configuration

The interrupt from the MPSC is available at stake pin E007. In the default configuration, the serial interrupt is jumpered to master PIC interrupt request level IR6 (jumper E007-E017). Refer to Chapter 8, "CPU Subsystem Configuration," for interrupt jumper information.

Serial Break Detect Circuit

The receive data from each channel is monitored by the iSBC 486/12 board break-detect logic for a break condition (a series of space bits). The BRKINT is asserted when a break condition is detected on either channel. This circuit operates independently of the 8274 MPSC's internal break detect logic. The BRKINT signal is latched. A byte write of 0H to I/O address 0E6H clears the BRKINT signal.

The break detect logic can be used to remotely interrupt the CPU. The BRKINT signal can be used even when the 8274 MPSC is not initialized or functioning.

The output of the break detect circuit is an interrupt signal at stake pin E058. Refer to Chapter 8, "CPU Subsystem Configuration," for interrupt jumper information.

8274 MPSC Programming

The 8274 MPSC supports the following modes of operation:

- polled mode
- interrupt mode (vectored/non-vectored)

Figure 7-31 (page 7-70) is a PL/M programming example for initializing the 8274, Channel B, for an interrupt (non-vectored) on Receive Data and wait on Transmit Data. The following subsections discuss the modes of operation. Refer to the *Microcommunications* Handbook for complete 8274 programming information.

Polled Mode. Polled operation is accomplished by repetitively reading the status of the 8274 and making decisions on that status. The 8274 can be read at any time.

Read the status at the following I/O port addresses:

- 0DCH (Channel A)
- 0DEH (Channel B)

Interrupt Mode. If the 8274 requires service in interrupt mode, the device sends an interrupt request signal to the 82C59A. If the 8274 is in vectored operation, the 8274 places the interrupt vector on the data bus during the second interrupt acknowledge cycle. If the 8274 is in non-vectored operation, the 82C59A places the vector on the data bus, and the 8274 outputs are tri-stated.

NOTE

If the 8274 MPSC is in vectored interrupt mode, its interrupt signal must be connected to the master 82C59A PIC's interrupt request 6 input (by installing jumper E007-E017). The 82C59A must then be programmed for slave operation at interrupt 6.

```
/*Declarations:*/

Declare Serial$Init () STRUCTURE (
                    S$Port    byte,
                    S$Data    byte)
          data(
               ØDEH,ØØØH,     /* Channel B Point to WRØ */
               ØDEH,Ø19H,     /* Channel B reset send abort */
               ØDEH,ØØ4H,     /* Channel B Point to WR4B */
               ØDEH,Ø4CH,     /* Channel B 16x clock, 8-bit sync, 2 stop bits, */
                              /* odd parity, parity disabled */
               ØDEH,ØØ1H,     /* Channel B Point to WR1B */
               ØDEH,Ø1CH,     /* Channel B Disable wait, wait on transmit, */
                              /* interrupt on all receive, parity does not */
                              /* affect vector, status affects vector, no */
                              /* transmit interrupt, no external interrupt */
               ØDEH,ØØ2H,     /* Channel B point to WR2B */
               ØDEH,Ø1ØH,     /* Channel B Pin 1Ø is RTS, non-vectored */


/*Program:*/


index=Ø;

Do while index <= (last(Serial$Init));
     output(Serial$Init(index).S$Port)=Serial$Init(index).S$Data;
     index=index+1;
end;

/* Data may now be written to or read from Channel B. */
/* 8274 causes an interrupt if RECEIVE DATA is received. */
```

Figure 7-31. 8274 MPSC Programming Example

SBX Interface

To increase the iSBC 486/12 board's I/O capabilities, you can add up to two SBX boards, using SBX connectors J5 and J6. Both connectors support the IEEE Specification 959. Refer to Chapter 1 for the SBX compliance levels. Both connectors support DMA operations when the 82258 ADMA is installed.

Connector J6 supports single-wide and double-wide SBX boards. Because of the physical interference with the memory module on connector P3, connector J5 supports only single-wide SBX boards. If you install a double-wide board on connector J6, you cannot install a single-wide board on connector J5 because of the physical interference between the double-wide board and connector J5.

NOTE

If your application requires a double-wide SBX module on SBX1, use the SBX I/O address swap option (page 7-74) to swap address mapping of SBX1 and SBX2 between connectors J5 and J6.

Chapter 3, "Installation," discusses how to install the SBX modules.

Each connector supports both 8-bit and 16-bit data transfers, depending on the size of the SBX module installed.

SBX I/O Addresses

Each connector supports both 8-bit and 16-bit data transfers. The SBX board input signals, MCS0 and MCS1, are activated by accesses to I/O addresses 80H through 0BFH.

Default SBX I/O Addresses. Tables 7-36 and 7-37 list the I/O addresses assigned to the SBX interfaces with the addresses swap disabled (default).

Table 7-36. Default SBX I/O Addresses (Connector J5)

I/O Address	Size	Module Data Path	MCS0[1]	MCS1[1]
80H 82H 84H 86H 88H 8AH 8CH 8EH	Byte	8-bit, 16-bit	1	0
80H 82H 84H 86H 88H 8AH 8CH 8EH	Word	16-bit	1	1
81H 83H 85H 87H 89H 8BH 8DH 8FH	Byte	16-bit	0	1
90H 92H 94H 96H 98H 9AH 9CH 9EH	Byte	8-bit	0	1

Note:
[1] 1 = active, 0 = inactive

Table 7-37. Default SBX I/O Addresses (Connector J6)

I/O Address	Size	Module Data Path	MCS0[1]	MCS1[1]
0A0H 0A2H 0A4H 0A6H 0A8H 0AAH 0ACH 0AEH	Byte	8-bit, 16-bit	1	0
0A0H 0A2H 0A4H 0A6H 0A8H 0AAH 0ACH 0AEH	Word	16-bit	1	1
0A1H 0A3H 0A5H 0A7H 0A9H 0ABH 0ADH 0AFH	Byte	16-bit	0	1
0B0H 0B2H 0B4H 0B6H 0B8H 0BAH 0BCH 0BEH	Byte	8-bit	0	1

Note:
[1] 1 = active, 0 = inactive

SBX Address Swap Option. If your software requires a double-wide SBX module with I/O addresses 80H to 9EH, you can swap the I/O addresses for connectors J5 and J6. This gives you addresses 80H to 9EH on Connector J6.

NOTE

The SBX address swap option only swaps the SBX I/O addresses. It does not swap any of the SBX signal lines. The SBX1 interrupt, DMA, and option lines go to connector J5. The SBX 2 interrupt, DMA, and option lines go to connector J6.

Tables 7-39 and 7-40 list the I/O addresses assigned to the SBX interfaces with the SBX address swap option.

Table 7-38 shows the jumpers for the SBX address swap option.

Table 7-38. SBX Address Swap Jumpers

Configuration	E073-E074
No SBX address swap 80H to 9FH on J5 0A0H to 0BFH on J6	OUT §
Swap SBX addresses 0A0H to 0BFH on J5 80H to 9FH on J6	IN

Note:
§ default

Table 7-39. SBX I/O Addresses Swap (Connector J6)

I/O Address	Size	Module Data Path	MCS0[1]	MCS1[1]
80H 82H 84H 86H 88H 8AH 8CH 8EH	Byte	8-bit, 16-bit	1	0
80H 82H 84H 86H 88H 8AH 8CH 8EH	Word	16-bit	1	1
81H 83H 85H 87H 89H 8BH 8DH 8FH	Byte	16-bit	0	1
90H 92H 94H 96H 98H 9AH 9CH 9EH	Byte	8-bit	0	1

Note:
[1] 1 = active, 0 = inactive

Table 7-40. SBX I/O Address Swap (Connector J5)

I/O Address	Size	Module Data Path	MCS0[1]	MCS1[1]
0A0H 0A2H 0A4H 0A6H 0A8H 0AAH 0ACH 0AEH	Byte	8-bit, 16-bit	1	0
0A0H 0A2H 0A4H 0A6H 0A8H 0AAH 0ACH 0AEH	Word	16-bit	1	1
0A1H 0A3H 0A5H 0A7H 0A9H 0ABH 0ADH 0AFH	Byte	16-bit	0	1
0B0H 0B2H 0B4H 0B6H 0B8H 0BAH 0BCH 0BEH	Byte	8-bit	0	1

Note:
[1] 1 = active, 0 = inactive

SBX I/O Command Recovery Time

The iSBC 486/12 board's SBX I/O command recovery time is fixed at 400 ns. For longer command recovery times, insert one or more **non-cached** Intel486 instructions between the SBX I/O commands.

CAUTION

Check the I/O recovery time for the module you plan to use with the iSBC 486/12 board. If your SBX module has an I/O command recovery time greater than 400 ns, make sure you do not violate this specification when your program accesses the SBX module.

SBX DMA Operations

The 82258 ADMA (iSBC 486/12S series only) is configured as follows:

- SBX 1 on ADMA channel 1
- SBX 2 on ADMA channel 3
- SBX 1 or 2 can be configured on ADMA channel 2

Figure 7-32 shows the ADMA configuration. Chapter 8 discusses the configuration of the 82258 ADMA.

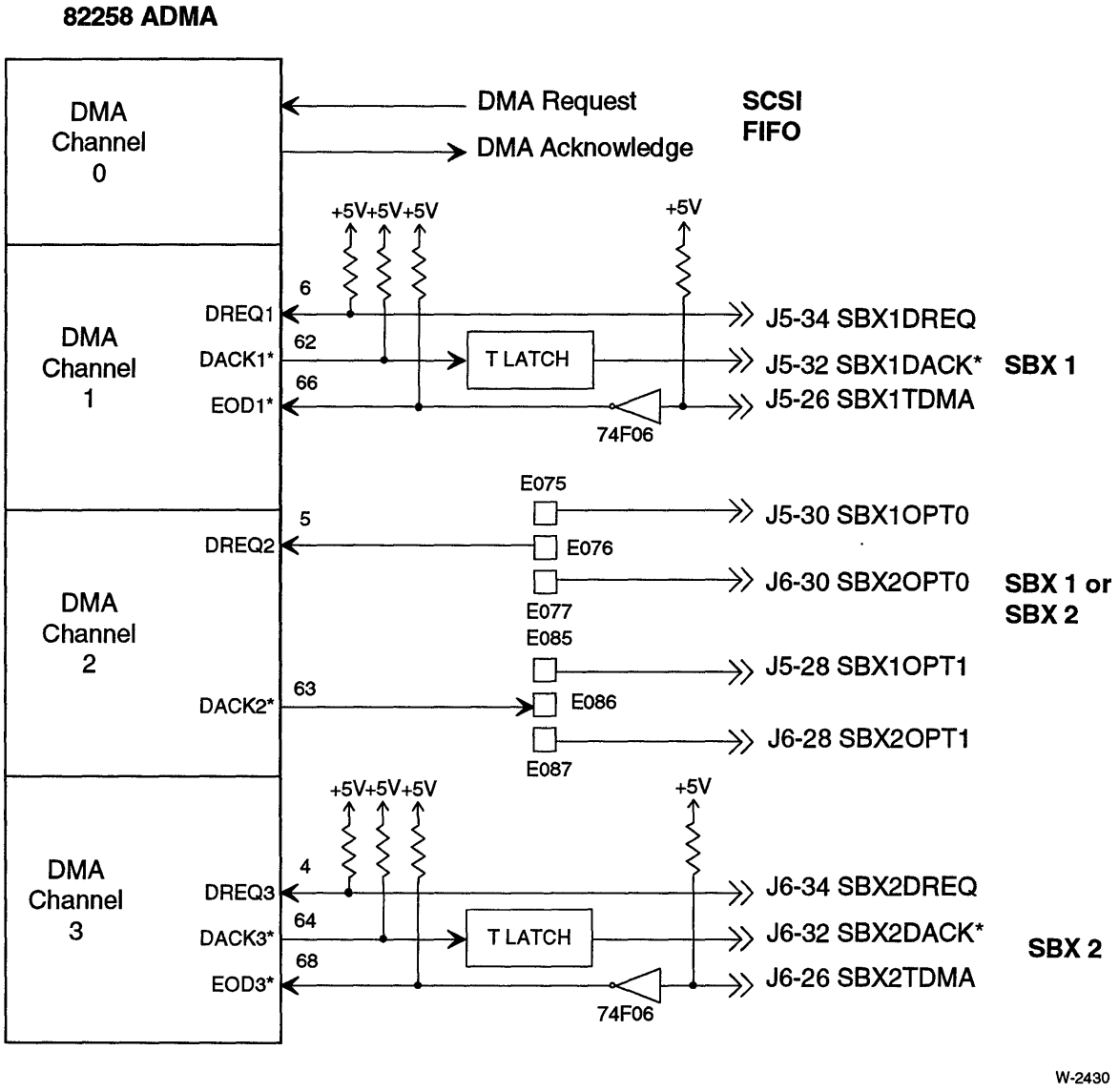


Figure 7-32. SBX DMA Configuration

Typically, the chip select signals (MCS0* and MCS1*), are not used for SBX DMA transfers. Use port address 0EAH as the source or destination address for DMA access. Do not increment or decrement the I/O port address after each DMA transfer. Port address 0EAH does not activate the chip select signals; however, it does provide a Ready signal to the 82258 ADMA.

SBX Option Lines

Each SBX port has two option lines that are available at stake pins on the iSBC 486/12 board. These option lines can be used as DMA request and DMA acknowledge signals for channel 2 of the 82258 ADMA.

Table 7-41. SBX Option Lines Stake Pins

Stake Pin	Function
E075	SBX1 [1], Option 0
E077	SBX2 [2], Option 0
E085	SBX1 [1], Option 1
E087	SBX2 [2], Option 1

Notes:
[1] Connector J5
[2] Connector J6

SBX Interrupts

Each SBX interface has two interrupt request lines, IR0 and IR1. These interrupt request lines are available at stake pins in the interrupt jumper matrix. Chapter 8 discusses the configuration of the interrupt sources to the 82C59 PICs.

Table 7-42. SBX Interrupt Stake Pins

Stake Pin	Function
E034	SBX1 [1] IR0
E035	SBX1 [1] IR1
E036	SBX2 [2] IR0
E037	SBX2 [2] IR1

Notes:
[1] Connector J5
[2] Connector J6

Multibus Interface (Connector P1)

The Multibus interface (Connector P1) complies with the IEEE 796 specification. Chapter 1 lists the IEEE 796 compliance level. In this section, you have the following configuration options:

- driving the BCLK* signal, page 7-80
- driving the CCLK* signal, page 7-80
- driving the INIT* signal , page 7-81
- configuring the software reset signal to drive the INIT* signal, page 7-82
- Multibus address line termination resistor, page 7-83
- driving or using the Multibus interrupts, page 7-84
- configuring the Multibus LOCK* signal, page 7-85
- configuring the Multibus buffered I/O Write option, page 7-85
- configuring the Multibus arbitration, page 7-85

BCLK* Signal Configuration

The Multibus Bus Clock (BCLK*) signal is used to synchronize the bus exchange logic. The iSBC 486/12 board can drive the BCLK* signal with a 9.83 MHz clock signal.

CAUTION

Only one board in the system should drive the BCLK* signal.

Table 7-43. BCLK* Signal Jumpers

Configuration	E267-E272
BCLK* not driven by 486/12 board	OUT
BCLK* driven by 486/12 board	IN §

Note:
§ default

CCLK* Signal Configuration

The Constant Clock (CCLK*) signal provides an optional timing source for any module on the bus. This signal should only be driven by one module on the bus. You can configure the iSBC 486/12 board to drive the CCLK* line with a 9.83 MHz clock signal.

CAUTION

Only one board in the system should drive the CCLK* signal.

Table 7-44. CCLK* Signal Jumpers

Configuration	E266-E271
CCLK* not driven by 486/12 board	OUT
CCLK* driven by 486/12 board	IN §

Note:
§ default

INIT* Signal Configuration

The Multibus INIT* signal always resets the iSBC 486/12 board's Multibus interface circuitry. As an option, the INIT* signal can also be connected to the iSBC 486/12 board's hardware reset signal. The hardware reset signal is asserted during power-up. The auxiliary reset signal from the front panel interface, if used, also asserts the hardware reset signal. Chapter 8, "CPU Subsystem Configuration," describes the reset actions performed by the various reset signals.

Figure 7-33 shows how the INIT* signal is connected to the HWRESET* signal. Notice that when jumper E264-E269 is IN, not only can the INIT* signal drive the HWRESET* signal, but the HWRESET* signal can also drive the Multibus INIT* signal.

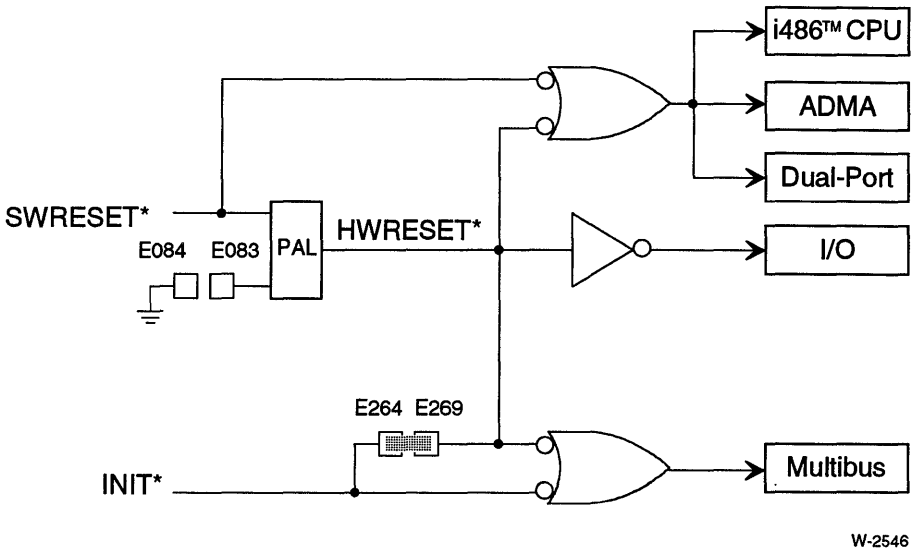


Figure 7-33. INIT* and HWRESET* Signal

Table 7-45. INIT* Signal Jumpers

Configuration	E264-E269
INIT* signal only resets the Multibus interface (it does not drive HWRESET* signal)	OUT
INIT* signal is tied to HWRESET* signal	IN §

Note:
§ default

Software Reset Configuration

The software reset is asserted by performing a byte-write of 1H to I/O address 0E0H. The software reset signal can be configured to reset just the CPU, dual-port memory, and ADMA, or it can be configured to also reset the I/O and Multibus circuits. Figure 7-33 shows the SWRESET* signal. Jumper E084-E083 allows the SWRESET* signal to also reset the I/O and Multibus circuits.

The SWRESET* signal can also be configured to drive the Multibus INIT* signal. To have the software reset signal drive the INIT* signal, you must also configure the HWRESET* signal to drive the INIT* signal (jumper E264-E269). Table 7-46 lists both jumper options.

Table 7-46. Software Reset Jumpers

Configuration	E083-E084	E264-E269
Software reset signal only resets the CPU, ADMA, and dual-port	OUT§	Don't care[1]
Software reset signal resets entire board	IN	OUT
Software reset signal resets entire board and drives the INIT* signal	IN	IN§

Notes:
§ default
[1] See "INIT* Signal Configuration," page 7-82.

Multibus Address Line Termination Resistors

Multibus address lines MA<23:20>, if used, must be terminated on one board in the system. In the default configuration, RP7 termination resistor is installed. Figure 7-34 shows the location of RP7.

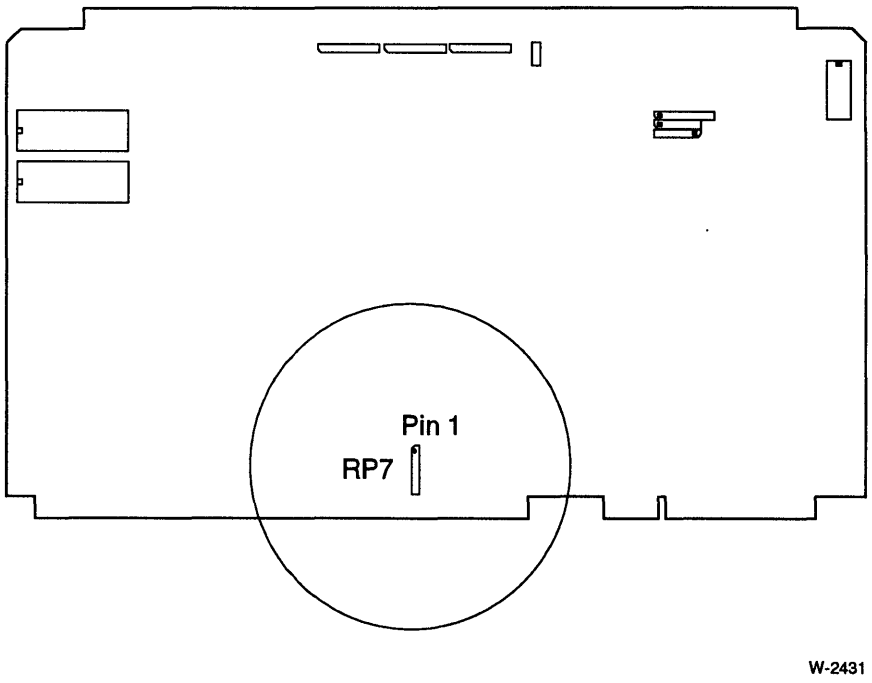


Figure 7-34. Location of Multibus Termination Resistor Pack RP7

Table 7-47. Multibus Address Line Termination Resistors

Configuration	Resistor Pack RP7
The 486/12 board supplies termination resistors	IN §
The 486/12 board does not supply termination resistors	OUT

Note:
§ default

Multibus Interrupts

The Multibus interrupt signals can be interrupt sources for the 82C59A PICs. In the default configuration, Multibus INT0* is used as one of two NMI interrupt sources. Refer to Chapter 8 for information on how to configure the on-board interrupts.

The iSBC 486/12 board has three signals that can be used to drive the Multibus interrupts: BUSDRIVE1*, BUSDRIVE2*, and PER*. Table 7-48 lists the Multibus interrupt stake pins.

Table 7-48. Multibus Interrupt Stake Pins

Stake Pin	Function[1]
E246 E247 E248 E249 E254 E259 E258 E257	Multibus INT0* Multibus INT1* Multibus INT2* Multibus INT3* Multibus INT4* Multibus INT5* Multibus INT6* Multibus INT7*
E252 E253	BUSDRIVE1* (output) BUSDRIVE2* (output)
E251	PER* DRAM parity error (output)

Notes:
[1] Multibus INT<7:0>* stake pins are outputs from the iSBC 486/12 board to the Multibus. See Chapter 8, "CPU Subsystem Configuration," for Multibus interrupt input stake pins. The interrupt input stake pins are inverted to allow them to drive the PIC.

Multibus LOCK* Signal

The iSBC 486/12 board can assert the Multibus LOCK* signal by performing a Multibus cycle using an instruction with the LOCK prefix.

Other bus masters can assert the Multibus LOCK* signal to lock the dual-port memory.

Refer to Chapter 6 for information on how to configure dual-port memory and the dual-port lock.

Multibus Buffered I/O Write Option

The iSBC 486/12 board has an option to buffer I/O write operations to the Multibus interface. After a Multibus I/O write operation, if Multibus I/O writes are not buffered, the CPU can access its internal cache, but not the EPROM, DRAM, I/O, or the Multibus until the Multibus cycle is completed. If buffering is enabled, the CPU can access its internal cache, EPROM, DRAM, and I/O, but cannot access the Multibus. With buffering enabled, the recovery time for back-to-back I/O write operations is approximately 150 ns. Buffering should be disabled if your application requires a longer I/O recovery time.

In the default configuration, Multibus I/O write operations are not buffered.

Table 7-49. Multibus Buffered I/O Write Jumper

Configuration	E030-E040
Buffered Multibus writes enabled	OUT
Buffered Multibus writes disabled	IN §

Note:
§ default

Multibus Arbitration

You have two configuration options for the Multibus arbitration:

- the arbitration priority (serial or parallel)
- the bus release mode

Multibus Arbitration Priority. The Multibus arbitration priority scheme is implemented in logic on the backplane. Configure the iSBC 486/12 board to match the arbitration method your backplane uses. Figure 7-35 shows the two arbitration methods.

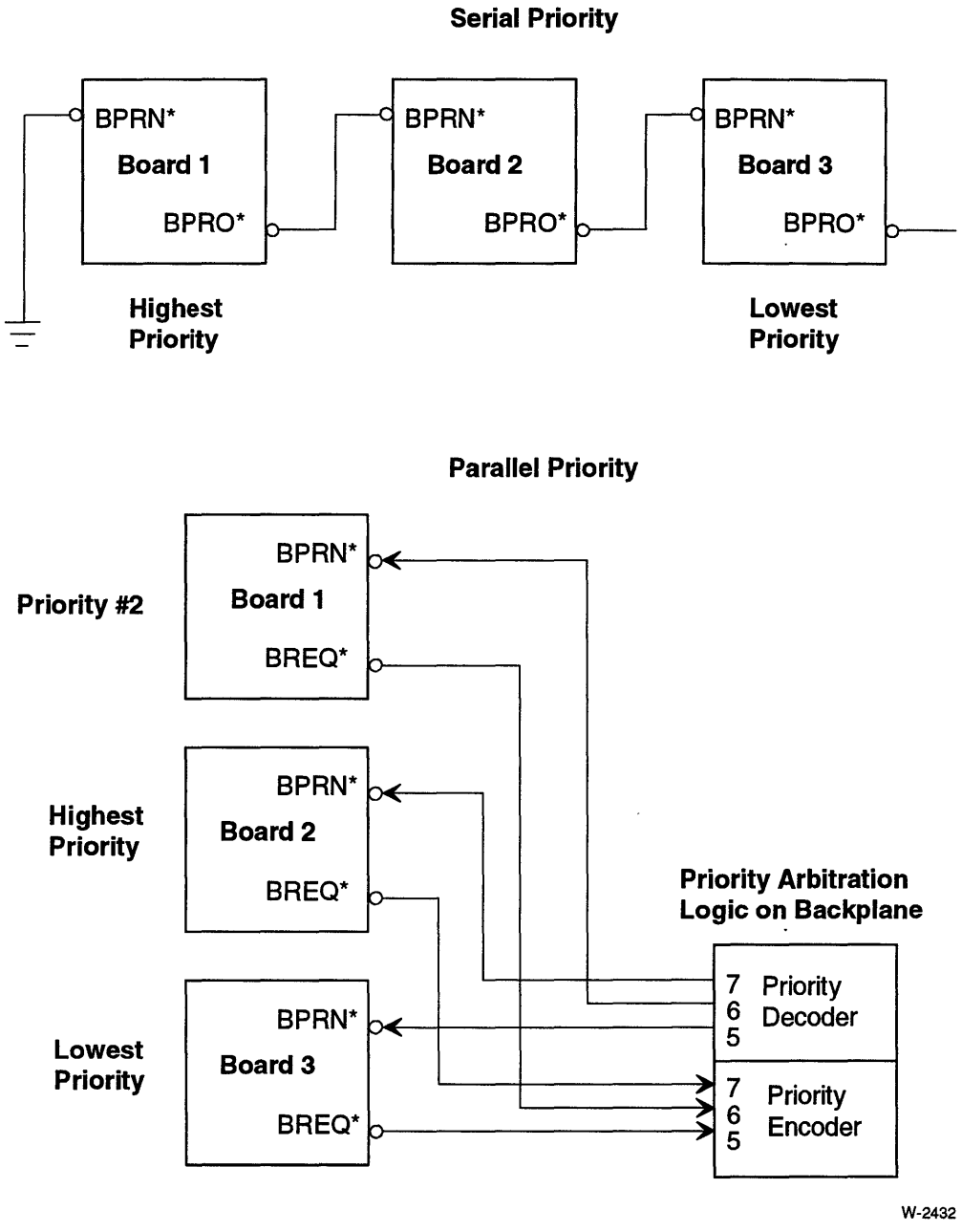


Figure 7-35. Multibus Serial and Parallel Arbitration

Table 7-50. Multibus Arbitration Priority Jumpers

Configuration	E263-E268
Parallel priority backplane	OUT
Serial priority backplane	IN §

Note:
§ default

Multibus Bus Release Mode. The bus release mode determines when the iSBC 486/12 board will release the Multibus when it controls the bus.

Table 7-51. Multibus Arbitration Mode Jumpers

Configuration	E256-E261	E261-E262
Releases the bus only to higher priority bus masters.	IN	OUT
Releases bus at the end of the current bus cycle when another bus master requests the bus.	OUT §	OUT §
Always releases the bus after completing a bus cycle.	OUT	IN
Not Allowed	IN	IN

Note:
§ default

Configuration Questions and Answers

Q. Does the iSBC 486/12S board support DMA operations to the serial ports?

A. No. The ADMA is used to support the SBX interfaces and the SCSI interface.

Q. If I set the SCSI ID using the jumpers for the SCSI ID Jumpers, do I still need to program the SCSI ID in the SPC?

A. Yes. You need to program the SCSI Bus Device ID Register (BDID) in the SPC.

Q. When should I remove the SCSI termination resistors?

A. The SCSI bus is daisy-chained between devices. Only the devices on the ends of the cable should have the termination resistors installed. Remove RP1, RP2, and RP3 if the iSBC 486/12S board is not at one end or the other of the SCSI cable.

Q. Does the SBX address swap feature swap the other signals too?

A. No. Only the addresses that are mapped to connectors J5 and J6 are swapped. The SBX interrupt, DMA, and option lines are not swapped.

***

CPU Subsystem Configuration 8

Chapter Contents

Introduction 8-3
Related Topics 8-3
New Terminology in This Chapter 8-3
Configuration Procedure 8-3
Intel486 ™ Microprocessor 8-4
Switching Between Real and Protected Mode 8-4
Interrupts 8-4
Board ID Register 8-5
Resets 8-7
Multibus Time-Out Action 8-9
Multibus Request Error 8-10
CPU Self-Test 8-11
Cache 8-11
Floating Point Unit 8-12
82258 ADMA 8-12
ADMA Configuration 8-12
ADMA Default Configuration 8-13
ADMA Configuration Options 8-14
ADMA Programming 8-14
General Mode Register Initialization 8-15
One- and Two-Cycle Modes 8-16
DMA Start/Termination Conditions 8-17
DMA Address Generator (DAG) 8-17
DAG Block Diagram 8-17
DAG Registers 8-18
DAG Register I/O Addresses 8-20
DAG Register Programming 8-22
DAG Control Register (Channel 0) 8-23
One-Cycle Page Register 8-24
Memory Address Page Registers 8-24
Test Register 8-25
Maximum DMA Transfer Rates 8-26
DMA Transfer Specifications 8-27
One-Cycle Transfer Modes 8-28
Paged/Non-Paged Mode 8-29
Burst/Non-Burst Mode 8-29
Compatibility/Extended Mode 8-29
One-Cycle Transfers 8-30

Paged, Non-Burst Transfer 8-30
Paged, Burst Transfer 8-30
Non-Paged, Non-Burst Transfer 8-31
Non-Paged, Burst, Compatibility Mode Transfer 8-31
Non-Paged, Burst, Extended Mode Transfer 8-32
Two-Cycle Transfers 8-33
Memory-to-Memory Transfers 8-34
Memory-to-I/O Transfers 8-34
I/O-to-Memory Transfers 8-35
I/O-to-I/O Transfers 8-35
82C59A Programmable Interrupt Controller (PIC) 8-36
Interrupt Configuration 8-38
Interrupt Jumper Matrix 8-38
Default PIC Configuration 8-41
Default NMI Assignments 8-41
Interrupt Destinations 8-42
Master PIC 8-42
Slave PIC 8-43
Non-Maskable Interrupt Request (NMIR) 8-43
Spare Gates 8-44
Multibus Interrupt Destinations 8-44
Interrupt Sources 8-45
82C54A PIT Counter 0, Counter 1 Interrupts 8-46
Multibus Interrupts 8-46
Multibus Time-Out Interrupt 8-48
MB87033B SPC Interrupt (SCZINT) 8-49
External Interrupt (EXTINT) 8-49
Serial Interrupt 8-49
DMA Interrupt 8-49
SBX Interrupts 8-49
Multibus Request Error Interrupt 8-50
Line Printer Interrupt 8-50
Break Detect Interrupt 8-51
Parity Error Interrupt 8-51
Spare Inverter 8-51
Edge-Triggered Latch (RSINT) 8-52
OR-Gate 8-52
Power-Fail Interrupt (PFINT) 8-52
Multibus Interrupt Drivers 8-53
Bus Drive (BUSDRIVE1*, BUSDRIVE2*) 8-53
Parity Error (PER*) 8-53
Interrupt Programming 8-54
82C59A PIC Polled Mode 8-54
Direct-Vectored Mode 8-54
Light-Emitting Diodes (LEDs) 8-56
Questions and Answers 8-58

Introduction

Related Topics

The following topics are related to CPU configuration:

- for I/O configuration, see Chapter 7, "I/O Subsystem Configuration"

New Terminology in This Chapter

82C59A PIC — 82C59A Programmable Interrupt Controller (PIC)

DAG — DMA Address Generator (a custom gate array circuit)

82258 ADMA — 82258 Advanced DMA Controller (ADMA)

one-cycle DMA mode — In the one-cycle DMA transfer mode (also called single cycle), the data is transferred directly from the source to the destination without going through the internal registers in the ADMA. On the iSBC 486/12S board, only ADMA channel 0 (SCSI) supports one-cycle transfers.

two-cycle DMA mode — In the two-cycle DMA transfer mode, the data is transferred from the source to ADMA, then from the ADMA to the destination. On the iSBC 486/12S board, all four channels support two-cycle transfers.

paging — In this chapter, *paging* refers to the DAG's address paging capability. During paging, the value in the appropriate page register is combined with the ADMA address to produce the physical address.

compatibility mode — A mode used by the DAG. This mode makes the board compatible with software written for Intel386™ CPU-based Multibus II products such as the iSBC 386/100 and 386/116/120 boards.

extended mode — A mode used by the DAG. *Extended mode* is similar to *compatibility mode* except that the block size is increased to 128 Mbytes for non-paged burst transfers.

Configuration Procedure

To configure your board for protected mode, we recommend you do the following:

1. Copy the CPU Configuration Worksheets in Appendix A.
2. Use the copy of the Jumper Configuration Worksheet that you used in previous chapters.
3. Follow the worksheet and check-off each step. Pencil-in the jumpers on the jumper worksheet as you select each option. If you get lost, refer to the table of contents at the beginning of the chapter.
4. Save your worksheets as a record of your board configuration.

You may also want to look at:

- the programming information for the ADMA, DAG, and PIC
- the configuration question and answers on page 8-57

Intel486™ Microprocessor

Switching Between Real and Protected Mode

After power-up or reset, the processor and the board operate in Real Mode. To switch the board to Protected Mode, perform a byte-write to Port 0E4H, setting data bit D0 to a 1. To switch to Real Mode, perform a byte-write to Port 0E4H, setting data bit D0 to a 0. Also, for compatibility with the iSBC 286/12 board, place the iSBC 486/12 board in Protected Mode with a word-write to I/O port address 0CAH.

Interrupts

The Intel486 microprocessor has two interrupt inputs: NMIR (non-maskable interrupt request) and INTR (maskable interrupt request). The configuration of the 82C59A PIC later in this chapter discusses how to configure the INTR and NMIR inputs.

Board ID Register

This is a 16-bit I/O register that is accessed through I/O port 0ECH. This register is read-only.

NOTE

The Board ID Register replaces the iSBC 386/12 board's 3-bit Board ID Code read through the 82C55A Port B. Bits B<2:0> are now user-defined bits.

Table 8-1 defines the 16-bit Board ID Register.

Table 8-1. Board ID Register

Bits	Value	Description
<15:12>	0011	Intel486 microprocessor
<11:9>	001	Board type: iSBC 486/12 or 486/12S
<8:7>	00 01	CPU clock: 25 MHz 33 MHz
6	0 1	ADMA present[1] ADMA not present
5	0 1	2nd memory module present 2nd memory module not present
4	0 1	SPC present SPC not present
3	0 1	SBX module(s) present SBX module(s) not present
<2:0>	000 001 010 011 100 101 110 111	Size of Bottom Memory Module: Reserved Reserved Reserved Reserved 8 Mbytes (MM308 module) Reserved 4 Mbytes (MM304 module) 2 Mbytes (MM302 module)

Note:
[1] The presence of the 82258 ADMA is detected after power-up reset. An iSBC 486/12S series board with the 82258 ADMA chip removed will have a 1 in this bit. If the ADMA is removed, the I/O address space for the ADMA, DAG, and SPC are mapped to the Multibus interface.

Resets

The following signals or conditions trigger a reset of the iSBC 486/12 board:

- INIT* signal from the Multibus interface
- AUXRST* signal from connector J4 (or P2), if used
- power-up (HWRESET*) reset logic
- software reset

Both the INIT* and the software reset signals can be tied to the HWRESET* signal. Chapter 7, "I/O Subsystem Configuration," describes how to configure the INIT* and software reset signals.

Table 8-2 shows what actions the iSBC 486/12 board performs following a reset.

CAUTION

The data in the DRAMs may be invalid because the DRAM refresh cycles are suspended during resets.

The following components remain unchanged after a reset (these components are only reset when the power is cycled):

- 82C59A Programmable Interrupt Controllers
- 82C54A Programmable Interval Timer
- DRAM (data may be invalid)

The software reset is asserted by writing 1H to I/O address 0E0H. Table 8-2 shows what actions the iSBC 486/12 board performs following a software reset.

To determine if the last reset was a hardware or software reset, read I/O port 0E0H. Bit 0 is the reset type flag:

0 = hardware reset (power-up, INIT*, or AUXRST*)
1 = software reset (SWRESET* only)

A byte-write of 0H to 0E2H clears the reset type flag.

Table 8-2. iSBC® 486/12 Board Reset Actions

Action	INIT*[1]	AUXRST*	Power-Up (HWRESET*)	Software Reset[2]
Resets the Intel486 CPU		X	X	X
Flushes the Intel486 cache		X	X	X
Flushes the SCSI FIFOs		X	X	
Resets the Multibus interface and time-out logic	X	X	X	
Resets the 82C55A PPI and the board functions controlled by port C		X	X	
Asserts the RESET signal on SBX1 and SBX2		X	X	
Resets board addressing mode to Real Mode		X	X	X
Resets the dual-port memory control logic		X	X	X
Resets the 8274 MPSC		X	X	
Resets the MB37088B SPC		X	X	
Resets the 82258 ADMA and DAG		X	X	X
Resets the SCSI Control Register		X	X	
Resets the DRAM refresh logic		X	X	X
Flushes buffered writes to Multibus and I/O		X	X	
Changes the Reset Flag Register to reflect the type of the last reset		X	X	X
Clears serial break detect interrupt, parity error interrupt, and the 16-Megabyte Page Register		X	X	

Note:
[1] Jumper E264-E269 OUT. If E264-E269 is IN, the INIT* reset actions are the same as the power-up reset.
[2] Jumper E083-E084 OUT. If E083-E084 is IN, the software reset actions are the same as the power-up reset.

Multibus Time-Out Action

A Multibus time-out condition exists if a Multibus cycle does not finish within approximately 8 ms. You can configure the iSBC 486/12 board to respond to a Multibus time-out in two ways:

- Force the completion of the cycle. The cycle is completed by providing the Ready signal to the Intel486 microprocessor and the 82258 ADMA, so they can resume processing. If this happens, the yellow LED (DS3) lights. In this case, the data is usually invalid.
- Wait until an XACK signal is received from the Multibus interface (LED DS3 is disabled in this case).

Table 8-3. Multibus Time-Out Response Jumper

Configuration	E010-E020
Force completion of cycle	IN§
Wait for XACK signal	OUT

Note:
§ default

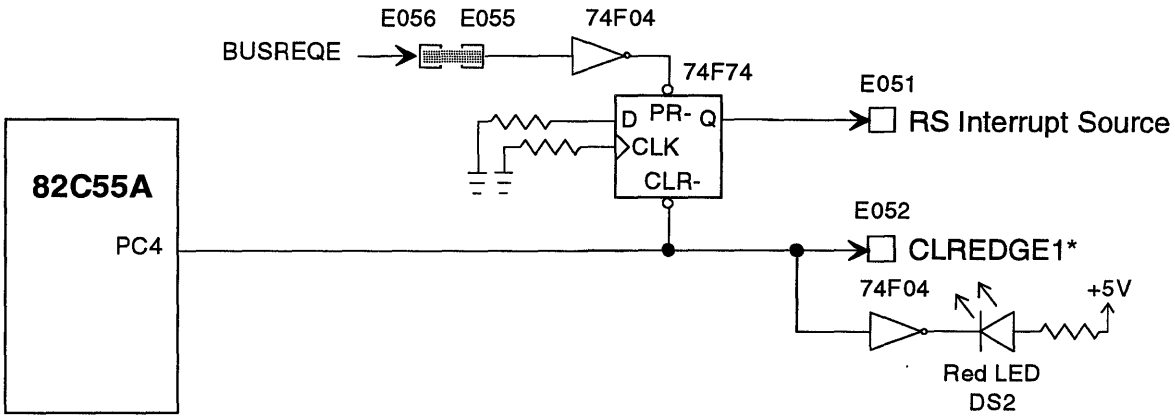
If you need to know if a time-out has occurred, configure the time-out interrupt (TOUTINT) as an input to either interrupt controller.

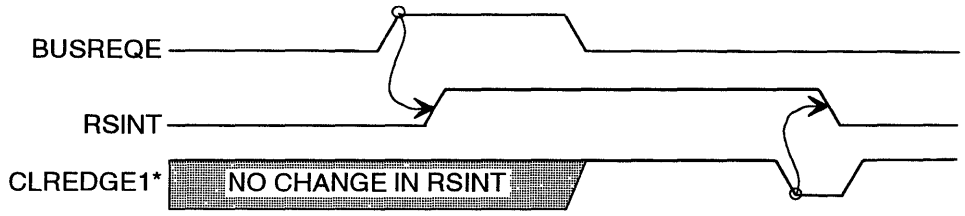
NOTE

The Multibus time-out feature is not affected by enabling or disabling the buffered write operations. Chapter 7, "I/O Subsystem Configuration," discusses the buffered write option.

Multibus Request Error

If the Multibus is not obtained within the timeout period, the BUSREQE signal will be asserted. This signal is available at stake pin E056. This signal can be latched using the extra flipflop and used as an interrupt source. Figure 8-1 shows how the BUSREQE signal can be latched.





BUSREQE	CLREDGE1*	RSINT
0	1	No Change
0	0	0
1	1	1
1	0	Not Stable

W-2547

Figure 8-1. Using BUSREQE as an Interrupt Source

CPU Self-Test

The Intel486 microprocessor has a Built-in Self Test (BIST) that is always activated when the CPU is reset. The BIST tests the non-random logic, control ROM, translation lookaside buffer, and on-chip cache memory.

The BIST is initiated by a CPU reset. The BIST takes approximately 31 ms with a 33 MHz Intel486 or 66 MHz Intel486 DX2 microprocessor. The result of the BIST is stored in the EAX register. A value of zero in the EAX register indicates the BIST was successfully passed. A non-zero value indicates that the BIST detected a flaw in the microprocessor.

After the BIST is complete, the Intel486 microprocessor completes the reset operation. The DX register will contain a component identifier at the conclusion of the reset. The upper byte of DX will contain 04 and the lower byte will contain a stepping identifier.

Cache

The cache is disabled after the CPU is reset.

NOTE

The cache is disabled after reset. The CD (cache disable) bit in the CR0 Register is set to **0** after reset; this **disables** the cache.

When the cache is enabled, only accesses to DRAM are cached. EPROM and Multibus memory cycles are not cached.

NOTE

The cache write policy is set to write-through by default and should not be changed.

Floating Point Unit

The operation of the on-chip floating point unit is exactly the same as the Intel387™ math coprocessor. Software written for the Intel387 math coprocessor will run on the on-chip floating point unit without any modifications.

82258 ADMA

The 82258 Advanced Direct Memory Access (ADMA) controller provides four independently programmable channels for high-performance DMA operations. The ADMA supports memory-to-memory, memory-to-I/O, and I/O-to-I/O transfers. The ADMA works in conjunction with the DMA Address Generator (DAG). The DAG extends the addressing capabilities of the ADMA (to 32 bits) and provides a "burst" mode for high-speed SCSI DMA.

ADMA Configuration

Figure 8-2 shows a block diagram of the ADMA configuration.

3/25/94 DJL
ADMA Removed from GTAMS Radian 486-125 Because its 200H - 2FFH I/O
Space conflicts w/ 2nd 270 - 22FH I/O range
See Note P 8-6: 7 bits is allowed
Moved 2nd ⇒ 1F0 → 1FFH

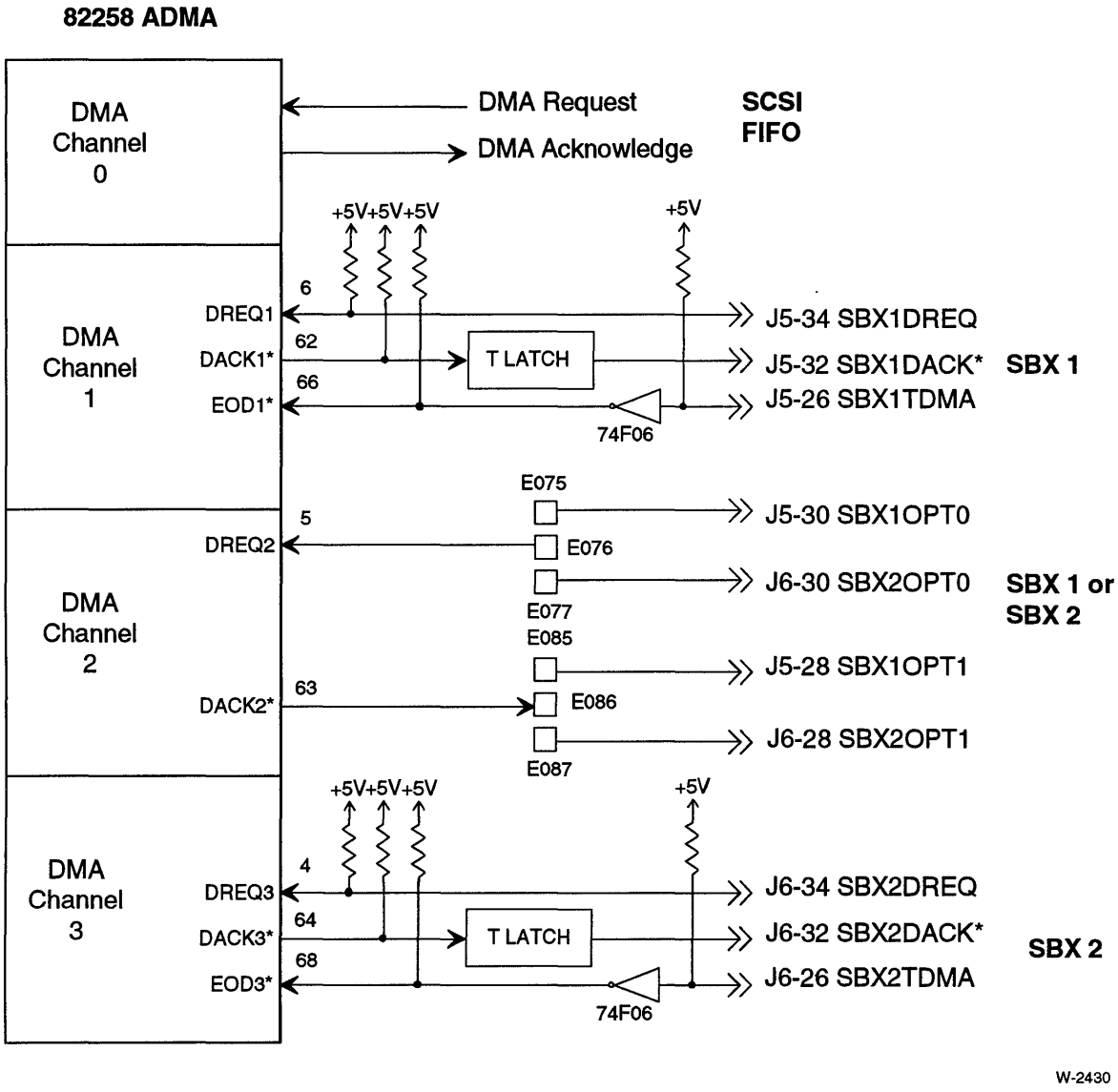


Figure 8-2. ADMA Configuration

ADMA Default Configuration. On the default board, the ADMA device is configured as follows:

- Channel 0 is dedicated to the SPC (Connector J2).
- Channel 1 is dedicated to SBX1 (Connector J5)
- Channel 2 can be configured to support either SBX1 or SBX2
- Channel 3 is dedicated to SBX2 (Connector J6)

Table 8-4 lists the default jumpers for the ADMA.

Table 8-4. Default ADMA Jumpers

Jumper	Description	Default Configuration
E075-E076 E076-E077	Channel 2 DREQ input to SBX1 OPT0 Channel 2 DREQ input to SBX2 OPT0	OUT§ OUT§
E085-E086 E086-E087	Channel 2 DACK* output to SBX1 OPT1 Channel 2 DACK* output to SBX2 OPT1	OUT§ OUT§
E033-E043	DMAINT to Slave PIC IR2	IN§

Note:
§ default

ADMA Configuration Options

ADMA Configuration Options. You have two options when configuring the ADMA:

- Change the interrupt request level for the DMAINT interrupt request (this is discussed later in this chapter starting on page 8-41)
- Configure ADMA channel 2 for either SBX 1 or SBX 2 by using the SBX OPT0 line for DMA request signal, and the SBX OPT1 line for DMA acknowledge signal. The EOD input signal is not available for channel 2.

Table 8-5 lists the jumper configurations options for the ADMA channel 2.

Table 8-5. ADMA Jumper Options

Description	Jumper	Configuration
ADMA channel 2 connected to SBX 1 (OPT0 is DMA request, OPT1 is DMA Acknowledge)	E075-E076 E076-E077 E085-E086 E086-E087	IN OUT§ IN OUT§
ADMA channel 2 connected to SBX 2 (OPT0 is DMA request, OPT1 is DMA Acknowledge)	E075-E076 E076-E077 E085-E086 E086-E087	OUT§ IN OUT§ IN

Note:
§ default

ADMA Programming

The 82258 ADMA has 256 internal registers. You can access the registers at I/O port addresses 200H through 2FFH. Table 8-6 lists the ADMA registers and I/O addresses. The *82258 ADMA User's Guide* describes the ADMA's internal registers.

Table 8-6. ADMA Internal Register I/O Addresses

Register Name	Size (Bits)	Channel 0	Channel 1	Channel 2	Channel 3
General Command	8	200H			
Subchannel (Not Used)		202H			
General Status	16	204H			
Reserved		206H			
General Mode	16	208H			
General Burst	8	20AH			
General Delay	8	20CH			
Reserved		20EH			
Reserved		240H-24EH			
Reserved		280H-28EH			
Reserved		2C0H-2CEH			
Channel Status	8	210H	250H	290H	2D0H
Assembly/Disassembly	16	212H	252H	292H	2D2H
Mask	16	214H	254H	294H	2D4H
Compare	16	216H	256H	296H	2D6H
Reserved		218H	258H	298H	2D8H
Reserved		21AH	25AH	29AH	2DAH
Reserved		21CH	25CH	29CH	2DCH
Reserved		21EH	25EH	29EH	2DEH
Command Pointer Low	24	220H	260H	2A0H	2E0H
Command Pointer High		222H	262H	2A2H	2E2H
Source Pointer Low	24	224H	264H	2A4H	2E4H
Source Pointer High		226H	266H	2A6H	2E6H
Destination Pointer Low	24	228H	268H	2A8H	2E8H
Destination Pointer High		22AH	26AH	2AAH	2EAH
Translator Table Pointer Low	24	22CH	26CH	2ACH	2ECH
Translator Table Pointer High		22EH	26EH	2AEH	2EEH
List Pointer Low	24	230H	270H	2B0H	2F0H
List Pointer High		232H	272H	2B2H	2F2H
Reserved		234H	274H	2B4H	2F4H
Reserved		236H	276H	2B6H	2F6H
Byte Counter Low	24	238H	278H	2B8H	2F8H
Byte Counter High		23AH	27AH	2BAH	2FAH
Channel Command Low	24	23CH	27CH	2BCH	2FCH
Channel Command High		23EH	27EH	2BEH	2FEH

In Real Mode, the ADMA source and destination address should be restricted to the first megabyte of address space. In Protected Mode, the ADMA has 16 Mbytes of address space. The ADMA is used with the DMA Address Generator to extend the addressing range to 4 Gbytes.

General Mode Register Initialization

Table 8-7 shows the ADMA GMR initialization.

Table 8-7. ADMA General Mode Register Initialization

Bit	Value	Function
0	1	Physical memory bus width is 16 bits.
1	1	Physical I/O bus width is 16 bits.
2	0	82258 operates in local mode.
3	0	Channel 3 is DMA only.
4	X	Two-cycle or one-cycle on DMA channel 0.
5	0	Two-cycle on DMA channel 1.
6	0	Two-cycle on DMA channel 2.
7	0	Two-cycle on DMA channel 3.
8	X	User defined.
9	X	User defined.
10	X	User defined.
11	X	User defined.
12	X	User defined.
13	X	User defined.
14	X	User defined.
15	0	Must be zero.

Note:
"X" means this bit is defined by the user.

NOTES

1. Channel 3 cannot be used as a multiplexer channel; other than this exception, the iSBC 486/12 board supports all ADMA functions.

2. The 82258 ADMA does not have the memory protection features of the Intel486 microprocessor. Memory protection is the system programmer's responsibility. To accomplish memory protection, the programmer must limit access to the 82258 ADMA.

One- and Two-Cycle Modes

The ADMA has two modes of transferring data: one-cycle and two-cycle.

Two-cycle data transfer is the traditional DMA mode of transfer where an I/O device asserts DREQ and the ADMA asserts DACK for each byte or word transferred. This mode is used when the ADMA transfers data to or from the SBX ports.

In the one-cycle transfer mode, the data is transferred directly from the source to the destination without being stored in the ADMA's registers.

NOTE

One-cycle mode is only supported for data transfers to or from the SCSI FIFO (DMA channel 0).

When the DAG, SCSI Control Register, and ADMA are programmed to work in "blast" mode, 16-bytes of data can be transferred between on-board DRAM and the SCSI FIFO in each ADMA cycle.

DMA Start/Termination Conditions

The DMA operation is started by programming the ADMA, DAG, and the SPC or SBX board with the appropriate start commands.

DMA operations are terminated when one of the following conditions are met:

- byte count exceeded
- End of DMA (EOD) asserted (channels 1 and 3 only)

During an SBX transfer operation, the ADMA termination condition should be set for both "byte-count exceeded" and "external terminate."

DMA Address Generator (DAG)

The DMA address generator (DAG) works with the ADMA, allowing it to operate as a limited 32-bit DMA controller.

The DAG has the following functions:

- extends the address range of the ADMA to 4 Gbytes
- supports *paging* for both 1-cycle and 2-cycle DMA operations
- supports DMA *burst mode*

DAG Block Diagram

Figure 8-3 shows the DAG, ADMA, and the system address and byte enable buses.

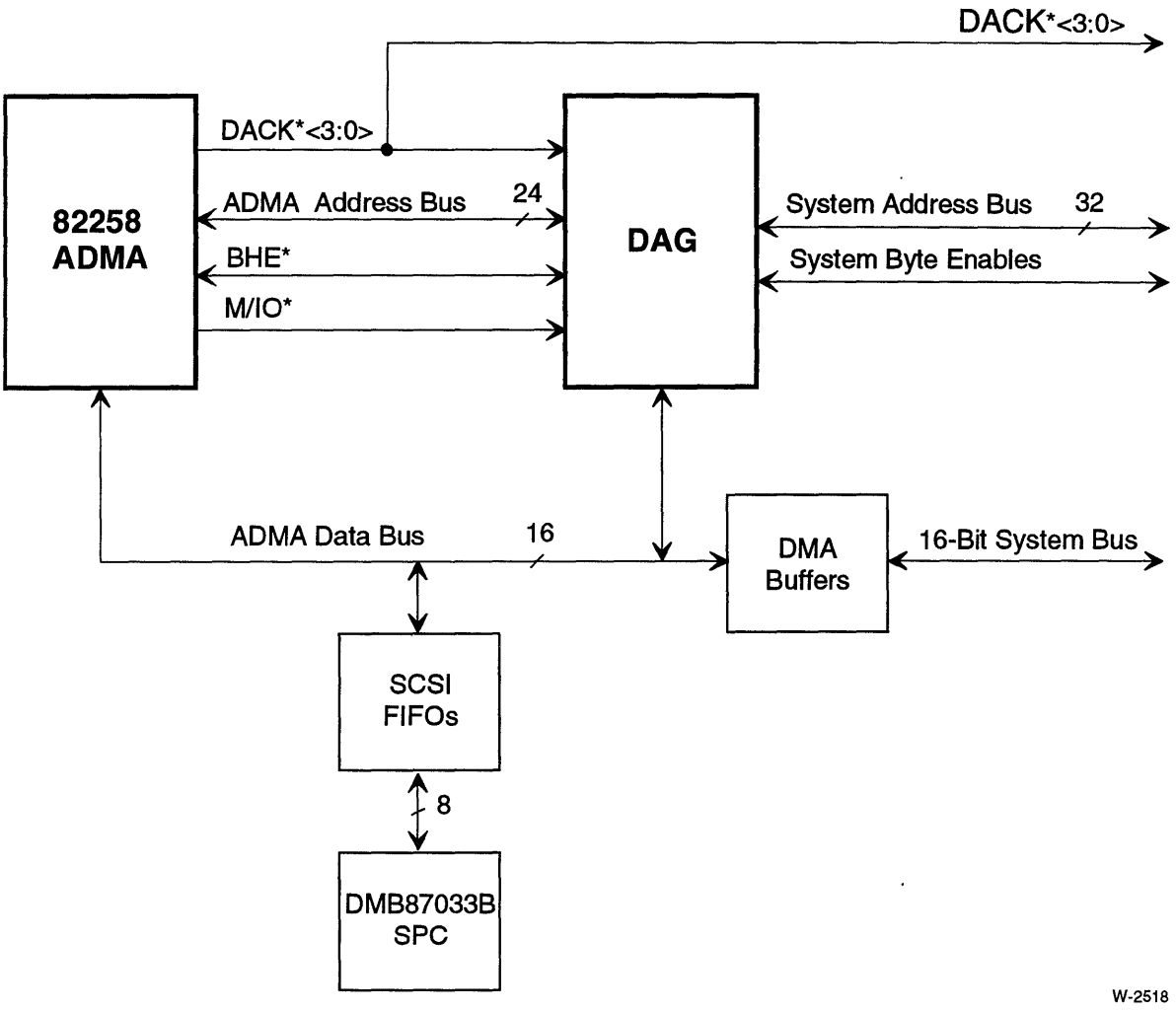


Figure8-3. Overview of ADMA and DAG

DAG Registers

This section describes the programmable registers within the DAG. These registers can be grouped into three logical groups of registers:

- DAG control registers
- One-Cycle Page Registers
- Two-Cycle Page Registers

Figure 8-4 shows the DAG registers.

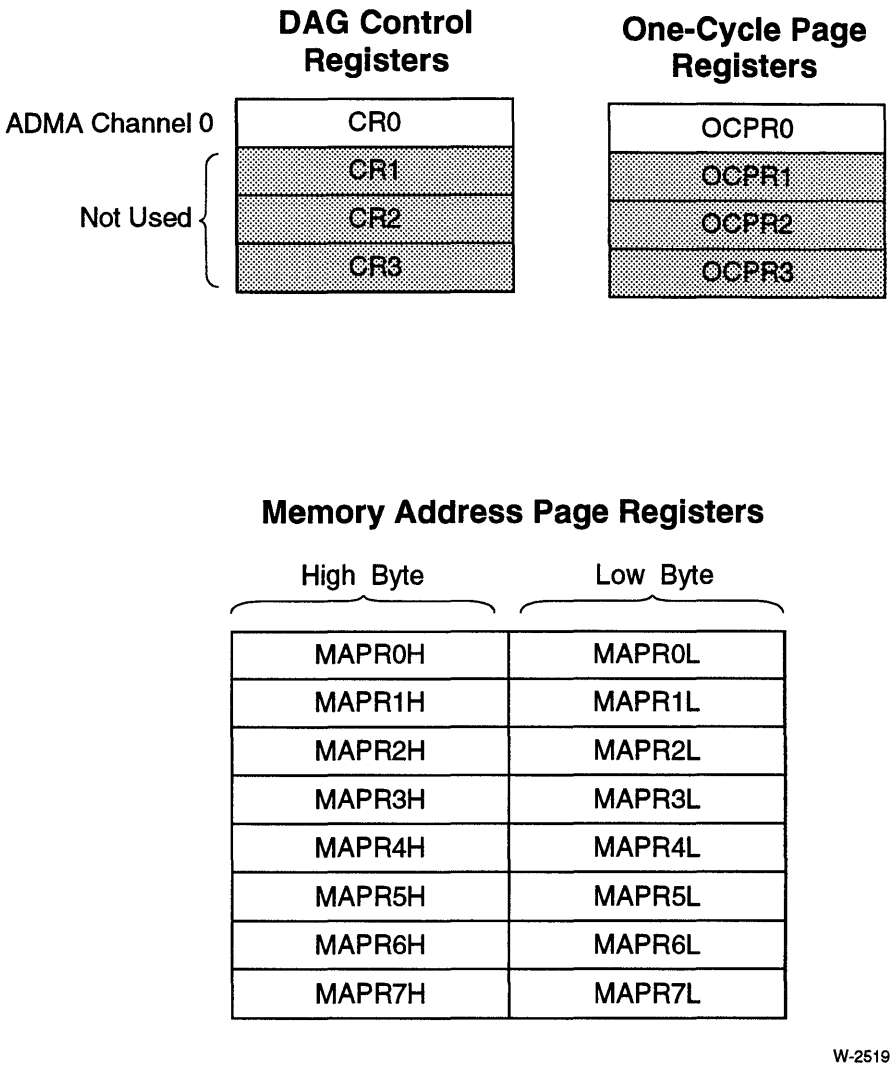


Figure 8-4. DAG Registers

How the registers shown in Figure 8-4 are used depends on the type of ADMA cycle (one- or two-cycle), and whether the DAG is programmed for paged or non-paged operation. Table 8-8 summarizes which DAG registers are used in each of the four types of cycles.

Table 8-8. Use of DAG Registers

Mode	Paged	Non-Paged
One-Cycle	DAG Control Registers, One-Cycle Page Registers	Control Registers
Two-Cycle	Memory Address Page Registers	Memory Address Page Registers

The DAG Control Register 0 and One-Cycle Page Register 0 are used for one-cycle transfers between DRAM and the SCSI FIFO's. The memory address registers are used for two-cycle transfers (memory or I/O), and channel command block accesses.

In addition to these registers, there is a DAG test register.

DAG Register I/O Addresses

All DAG registers are accessed by reading from or writing to byte-wide locations. The 16-bit DAG registers use two byte-wide locations to form the full register. The upper and lower bytes of the 16-bit registers have separate I/O addresses.

The DAG registers are aligned on even double-word (4-byte) boundaries. The DAG registers are accessed using byte 0 of the double-word.

Table 8-9 lists the I/O addresses of the DAG registers and their default values after power-up or auxiliary reset.

Table 8-9. I/O Addresses of DAG Registers

I/O Address	Register Name	Description	Default Value
300H	CR0	Control Register 0	9H
304H	CR1	Not Used	9H
308H	CR2	Not Used	9H
30CH	CR3	Not Used	9H
310H	OCPR0	One-Cycle Page Register 0	0H
314H	OCPR1	Not Used	0H
318H	OCPR2	Not Used	0H
31CH	OCPR3	Not Used	0H
320H	MAPR0L	Memory Address Page Register 0 Low Byte	0H
324H	MAPR0H	Memory Address Page Register 0 High Byte	0H
328H	MAPR1L	Memory Address Page Register 1 Low Byte	20H
32CH	MAPR1H	Memory Address Page Register 1 High Byte	0H
330H	MAPR2L	Memory Address Page Register 2 Low Byte	40H
334H	MAPR2H	Memory Address Page Register 2 High Byte	0H
338H	MAPR3L	Memory Address Page Register 3 Low Byte	60H
33CH	MAPR3H	Memory Address Page Register 3 High Byte	0H
340H	MAPR4L	Memory Address Page Register 4 Low Byte	80H
344H	MAPR4H	Memory Address Page Register 4 High Byte	0H
348H	MAPR5L	Memory Address Page Register 5 Low Byte	0A0H
34CH	MAPR5H	Memory Address Page Register 5 High Byte	0H
350H	MAPR6L	Memory Address Page Register 6 Low Byte	0C0H
354H	MAPR6H	Memory Address Page Register 6 High Byte	0H
358H	MAPR7L	Memory Address Page Register 7 Low Byte	0E0H
35CH	MAPR7H	Memory Address Page Register 7 High Byte	0H
360H	DAGTEST	DAG Test Register	1H

DAG Register Programming

The DAG's registers can be programmed by either the CPU or the ADMA.

NOTE

The ADMA can perform a memory-to-I/O write to the DAG registers. It cannot, however, read the DAG registers.

DAG Control Register (Channel 0)

The DAG Control Register 0 contains four mode bits. If one-cycle mode is detected (when DACK* is active and M/IO* = 1), the DAG will generate an address based on the control bits set in this register.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
X	X	X	X	CMP/EXT*	P/NP*	BM	32M

Default = 09H

Bits <7:4> **X** (Don't care)

Bit 3 **CMP/EXT***

0 Extended mode active
1 Compatibility mode active

Bit 2 **P/NP***

0 Non-paged mode
1 Paged mode

NOTE

When bit 2 = 1 (paged mode is active), bit 3 is a "don't care." Extended and compatibility modes are only valid with burst transfers.

Bit 1 **BM**

0 Burst mode inactive
1 Burst mode active

Bit 0 **32M**

0 8/16-bit mode
1 Not Supported

CAUTION

Bit 0 (32M) must be 0. The default value after reset is 1. This bit must be reprogrammed to 0 before using the ADMA to transfer SCSI data.

One-Cycle Page Register

The One-Cycle Page Register is eight-bits wide. The number of significant bits are determined by the DMA mode selected. The number of significant bits for the three page modes are as follows:

Table 8-10. Significant Bits of One-Cycle Page Register

Control Register	Modes Selected	Significant Bits of One-Cycle Page Register
0000 X100	Paged Mode Non-burst Mode	<7:0>
0000 X110	Paged Mode Burst Mode	<7:3>

In each case, the register should be programmed with a left-justified value. Bits <2:0> of the One-Cycle Page Register in burst mode are "don't cares."

Memory Address Page Registers

Two-cycle DRAM addresses and channel control block accesses use a separate set of page registers. Two-cycle I/O addresses do not use the page registers.

These page registers can be used with all four channels of the ADMA.

There are eight 11-bit page registers. The upper three bits of the ADMA address is used to select the Memory Address Page Registers. The contents of these registers represent system address bits <31:21>. Each register must be programmed with two byte writes.

High Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SA31	SA30	SA29	SA28	SA27	SA26	SA25	SA24

Bits 7:0 System address bits <31:24>

Low Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SA23	SA22	SA21	X	X	X	X	X

Bits 7:5 System address bits <23:21>

Bits 4:0 X (Don't care)

After reset, these registers set up eight contiguous 2 Mbyte blocks from 0H to 0FF FFFFH. If no paging is desired for memory addresses, no programming is needed. The default memory address translation is shown in Table 8-11.

Table 8-11. Default Memory Address Translation

ADMA Address	System Address Generated
000 0000H	000 0000H
020 0000H	020 0000H
040 0000H	040 0000H
060 0000H	060 0000H
080 0000H	080 0000H
0A0 0000H	0A0 0000H
0C0 0000H	0C0 0000H
0E0 0000H	0E0 0000H

If the memory address page registers are reprogrammed to non-contiguous pages, the ADMA address could roll into the next page. This is a problem only if the page registers are not contiguous. If memory address paging is desired, all ADMA transfers that cross page boundaries must use page registers that are programmed with contiguous pages. A detailed description of the steps needed for programming these registers is provided later in this chapter.

Test Register

The DAG test register allows software to read the DAG version number and determine if the DAG is functioning properly. The register data bits are scrambled. This provides a read value unique to the written value. The data bits are scrambled in this manner:

Data bits written	7 6 5 4 3 2 1 0
Data bits read	3 7 2 6 1 5 0 4

After reset, the value read from the test register will be the DAG version number. This will remain valid until a test pattern is written to the register. If the version number is to be used, it must be read and saved until the test patterns have been verified. The version number can be considered valid only after the test patterns are verified.

Maximum DMA Transfer Rates

Table 8-12 lists the maximum DMA transfer rates for one-cycle transfers. Table 8-13 lists the maximum DMA transfer rates for two-cycle transfers.

Table 8-12. Maximum DMA Transfer Rate (One-Cycle Transfers Between DRAM and SCSI FIFOs)

Mode	SCSI to DRAM, 8-bit	SCSI to DRAM, 16-bit	DRAM to SCSI, 8-bit	DRAM to SCSI, 16-bit
Burst Mode	--	16 Mbytes/s	--	13.3 Mbytes/s
Non-Burst Mode	3.33 Mbytes/s	6.67 Mbytes/s	5 Mbytes/s	10 Mbytes/s

Table 8-13. Maximum DMA Transfer Rate (Two-Cycle Transfers)

Mode	DRAM to DRAM, 8-bit	DRAM to DRAM, 16-bit	DRAM to SBX, 16-bit	SBX to DRAM 16-bit
Paged	2 Mbytes/s	4 Mbytes/s	2.0 Mbytes/s	2.22 Mbytes/s

DMA Transfer Specifications

Tables 8-14 and 8-15 list the data alignment and maximum byte counts for the various DMA transfers.

CAUTION

The maximum byte count of the SPC is 256 Mbytes. Do not program the SPC byte count greater than the maximum byte count listed for the modes you are using.

Table 8-14. DMA Transfer Specifications (One-Cycle Transfers)

Mode	Data Alignment	Maximum Byte Count	Total Address Space
Paged, Burst Mode	Even 16-byte	128 Mbytes	4 Gbytes
Non-Paged, Burst Mode, Extended	Even 16-byte	128 Mbytes	128 Mbytes
Non-Paged, Burst Mode, Compatibility	Even 16-byte	128 Mbytes	16 Mbytes
Non-burst Paged 8-bit	Byte	16 Mbytes	4 Gbytes
Non-burst Paged 16-bit	Word	16 Mbytes	4 Gbytes
Non-burst Non-paged 16-bit	Word	16 Mbytes	16 Mbytes
Non-burst Non-paged 8-bit	Byte	16 Mbytes	16 Mbytes

Table 8-15. DMA Transfer Specifications (Two-Cycle Transfers)

Source or Destination	Data Alignment	Maximum Byte Count	Total Address Space
Memory	--	2 Mbytes	4 Gbytes
I/O	--	64 Kbytes	64 Kbytes

One-Cycle Transfer Modes

One-cycle modes are used for DMA transfers between DRAM and the SCSI FIFOs. Data is transferred directly from source to destination and does not pass through the ADMA. In one ADMA cycle, either 1, 2, or 16 bytes are transferred, depending on the type of transfer selected.

One-cycle transfers offer the highest data transfer rates; however, some transfer types have data alignment requirements, which are listed in Table 8-14.

NOTE

Only ADMA channel 0 (SCSI FIFOs to DRAM, or DRAM to SCSI FIFOs) supports one-cycle DMA transfers.

Table 8-16. Supported One-Cycle Transfer Modes

DAG Control Word 0	Mode
0000 x100	Paged, Non-burst
0000 x11x	Paged, Burst
0000 x000	Non-paged, Non-burst
0000 101x	Non-paged, Burst, Compatibility
0000 001x	Non-paged, Burst, Extended

Note:
"x" means "don't care"

The DAG modes are discussed in the following sections:

- Paged/Non-Paged Mode, page 8-29
- Burst/Non-Burst Mode, page 8-29
- Compatibility/Extended Mode, page 8-29

One-cycle DMA transfers are discussed in detail starting on page 8-30.

Paged/Non-Paged Mode

In paged mode, the DAG extends the ADMA address to 32 bits by concatenating the contents of the One-Cycle Page Register to the ADMA address.

To use paged mode, you must do the following:

- Set bit 2 of DAG Control Register 0 to 1, which puts the DAG in paged mode.
- Program the DAG One-Cycle Page Register 0.

In non-paged mode, the DMA address is 24- or 27-bits wide, depending on the type of transfer. For non-paged transfers, 0s are placed in front of the upper bits of the ADMA address, which may or may not be shifted.

To use non-paged mode, set bit 2 of DAG Control Register 0 to 0.

Burst/Non-Burst Mode

In burst mode, the ADMA/DAG performs 16-byte transfers for each ADMA cycle. This mode requires that the data be aligned on 16-byte boundaries.

During burst transfers, circuitry external to the ADMA passes 16 bytes between the SCSI FIFO and DRAM, while holding the ADMA in wait-states.

To use burst mode, set bit 1 of DAG Control Register 0 to 1.

In non-burst mode, the ADMA/DAG performs 8- or 16-bit transfers, depending on the data size bit in the ADMA.

Compatibility/Extended Mode

The DAG is in compatibility mode after power-up. This mode makes the board compatible with software written for Intel386™ CPU-based Multibus II products such as the iSBC 386/100 and 386/116/120 boards. In this mode, the board drives the full 32-bit address, but forces the upper 8 bits (SA<31:24>) to zero.

To use compatibility mode, bit 3 of DAG Control Register 0 must be set to 1. The programming of this bit affects only the operation of the non-paged burst mode. All other data transfer operations ignore this bit.

Extended mode is similar to compatibility mode except that the block size is increased to 128 Mbytes for non-paged burst transfers.

To use extended mode, bit 3 of DAG Control Register 0 must be set to 0. The programming of this bit affects only the operation of the non-paged burst modes. All other data transfer operations ignore this bit.

One-Cycle Transfers

The following sections describe the one-cycle transfers:

- Paged, Non-Burst Transfer, page 8-30
- Paged, Burst Transfer, page 8-30
- Non-Paged, Non-Burst Transfer, page 8-31
- Non-Paged, Burst, Compatibility Mode Transfer, page 8-31
- Non-Paged, Burst, Extended Mode Transfer, page 8-32

Paged, Non-Burst Transfer

In this mode, a page register from the one-cycle page register set is used. The DACK* signal is used to select the appropriate page register from within the bank of four registers. The system address is generated by concatenating the unshifted ADMA address with the contents of the selected page register.

System Address Bits SA<31:2>

31 24	23 2
One-Cycle Page Register bits 7:0	ADMA Address bits 23:2

Paged, Burst Transfer

In a paged burst transfer, four 32-bit words are transferred for every ADMA address. The DAG extends the system address to a full 32 bits. This is done by shifting the ADMA address by three bits and using five bits out of the 8-bit page register to extend the 24-bit ADMA address. Bits 3:0 are always 0 in this mode. The ADMA is held in a wait-state by external circuitry while 16 bytes of data are transferred. Because 128 bits of data are transferred, the data must be aligned on 16-byte boundaries.

System Address Bits SA<31:2>

31 27	26 4	3	2
One-Cycle Page Register bits 7:3	ADMA Address bits 23:1	0	0

Non-Paged, Non-Burst Transfer

To generate the system address in this mode, the DAG keeps the ADMA address unchanged and places 0s in front of the upper eight bits. If a 16-bit one-cycle transfer is to be carried out, the address must be word-aligned. For 8-bit transfers, the address can be on any boundary.

System Address Bits SA<31:2>

31							24	23 2
0	0	0	0	0	0	0	0	ADMA Address bits 23:2

Non-Paged, Burst, Compatibility Mode Transfer

In this mode, four 32-bit words are transferred for every ADMA address. The ADMA address is shifted by three bits and 0s are placed in front of the upper eight bits to generate the system address. Bits 3:0 of the system address are always 0 in this mode. The ADMA is held in a wait-state by external circuitry while 16 bytes of data are transferred. The data must be aligned on 16-byte boundaries.

System Address Bits SA<31:2>

31							24	23 4	3	2
0	0	0	0	0	0	0	0	ADMA Address bits 20:1	0	0

Non-Paged, Burst, Extended Mode Transfer

In this mode, four 32-bit words are transferred for every ADMA address. The ADMA address is shifted by three bits and 0s are placed in front of the upper five bits to generate the system address. Bits 3:0 of the system address are always 0 in this mode. The ADMA is held in a wait-state by external circuitry while 16 bytes of data are transferred. The data must be aligned on 16-byte boundaries.

System Address Bits SA<31:2>

31				27	26 4	3	2
0	0	0	0	0	ADMA Address bits 23:1	0	0

Two-Cycle Transfers

There are four types of two-cycle transfers:

- memory-to-memory, page 8-34
- memory-to-I/O, page 8-34
- I/O-to-memory, page 8-35
- I/O-to-I/O, page 8-35

The three transfers that involve memory addresses for source or destination (or both), use the DAG's memory address page registers. The I/O to I/O transfers do not use paging.

In two-cycle transfers, the contents of a selected 11-bit memory page register are used along with the 21 bits of ADMA address to generate a 32-bit system address. The DAG selects one of the eight page registers by decoding the upper three bits (A<23:21>) of the ADMA address.

After reset or power-up, the memory page registers are preset to addresses in the range of 0 - 16 Mbytes. This makes the paging transparent until the page registers are reprogrammed. The memory page registers can be programmed to point to completely different areas of memory, or they can be programmed in groups to provide larger contiguous blocks.

All two-cycle DMA data transfers are restricted to 8 or 16 bits, since the data must pass through the ADMA.

Two-cycle transfers occur when a DACK* signal is inactive. The flow for a two-cycle transfer is as follows:

Cycle 1	Cycle 2
Source → ADMA	ADMA → Destination

Memory-to-Memory Transfers

During both cycles of the transfer, M/IO* = 1. The DAG selects one of the eight page registers by decoding the upper three bits (A<23:21>) of the ADMA address. The contents of the selected 11-bit memory page register are used along with the 21 bits of ADMA address to generate a 32-bit system address. This operation occurs during both cycles, when the source address is formed (cycle 1) and when the destination address is formed (cycle 2).

System Address Bits SA<31:2> during Cycle 1 and Cycle 2 (Memory Addresses)

31 21	20 2
Memory Address Page Register bits 10:0	ADMA Address bits 20:2

Memory-to-I/O Transfers

During cycle 1, M/IO* = 1. The DAG selects one of the eight page registers by decoding the upper three bits (A<23:21>) of the ADMA address. The contents of the selected 11-bit memory page register are used along with the 21 bits of ADMA address to generate a 32-bit system address.

During cycle 2, M/IO* = 0. The DAG does not modify the ADMA address, except to set the most significant byte (SA<31:24>) to 0.

System Address Bits SA<31:2> during Cycle 1 (Memory Address)

31 21	20 2
Memory Address Page Register bits 10:0	ADMA Address bits 20:2

System Address Bits SA<31:2> during Cycle 2 (I/O Address)

31 24								23 2
0	0	0	0	0	0	0	0	ADMA Address bits 23:2

I/O-to-Memory Transfers

During cycle 1, M/IO* = 0. The DAG does not modify the ADMA address, except to set the most significant byte (SA<31:24>) to 0.

During cycle 2, M/IO* = 1. The DAG selects one of the eight page registers by decoding the upper three bits (A<23:21>) of the ADMA address. The value of the selected 11-bit memory page register is used along with the 21 bits of ADMA address to generate a 32-bit system address.

System Address Bits SA<31:2> during Cycle 1 (I/O Address)

31							24	23 2
0	0	0	0	0	0	0	0	ADMA Address bits 23:2

System Address Bits SA<31:2> during Cycle 2 (Memory Address)

31 21	20 2
Memory Address Page Register bits 10:0	ADMA Address bits 20:2

I/O-to-I/O Transfers

During I/O to I/O transfers the DAG passes the ADMA address through unchanged.

System Address Bits SA<31:2> during Cycle 1 (I/O Address)

31							24	23 2
0	0	0	0	0	0	0	0	ADMA Address bits 23:2

System Address Bits SA<31:2> during Cycle 2 (I/O Address)

31							24	23 2
0	0	0	0	0	0	0	0	ADMA Address bits 23:2

82C59A Programmable Interrupt Controller (PIC)

The iSBC 486/12 board processes interrupt signals with two 82C59A Programmable Interrupt Controllers (PICs). Figure 8-5 shows an overview of the board interrupts. Figure 8-6 (page 8-38) shows the default configuration of the interrupt sources.

Jumper block E063 - E064 provides an interrupt filter to prevent noise from being latched as an interrupt. With the jumper in (user selected option), interrupts to the Intel486 CPU are filtered and latched when INTRI is active; however any assertion of INTRI less than 50 ns duration is not allowed to pass to the CPU. Any assertion of INTRI longer than 50 ns is latched and is released only after an interrupt acknowledge cycle has at least started and INTRI is inactive. With jumper block E063 - E064 out (default), all interrupts to the CPU are simply passed through.

The iSBC 486/12 board has an NMI mask that allows you to mask the NMI input to the CPU. The NMI interrupt can be masked by setting 82C55A PPI, bit PC4 to 1.

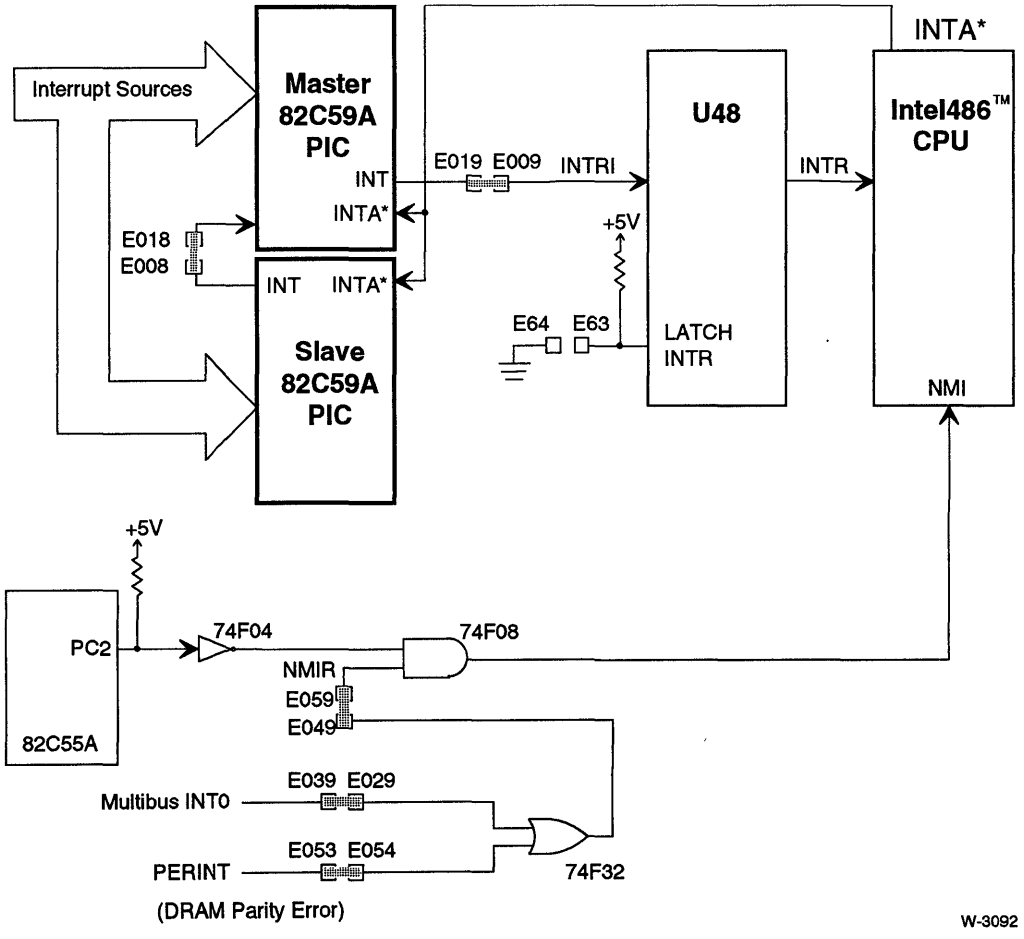


Figure 8-5. 82C59A Programmable Interrupt Controllers Overview

Interrupt Configuration

You have the following interrupt configuration options:

- interrupt sources for the 82C59A PICs
- interrupt sources for the Intel486 CPU NMI

CAUTION

Both 82C59A PICs must be programmed before any interrupts are received, even if only one PIC is used. If not, bus contention may occur between the master and slave PICs.

This section includes the following subsections:

- Interrupt Jumper Matrix, page 8-37
- Default Configuration, page 8-40
- Interrupt Destinations, page 8-41
- Interrupt Sources, page 8-44
- Multibus Interrupt Drivers, page 8-52

Interrupt Jumper Matrix

Figures 8-6 and 8-7 show the interrupt jumper matrix and the spare gates.

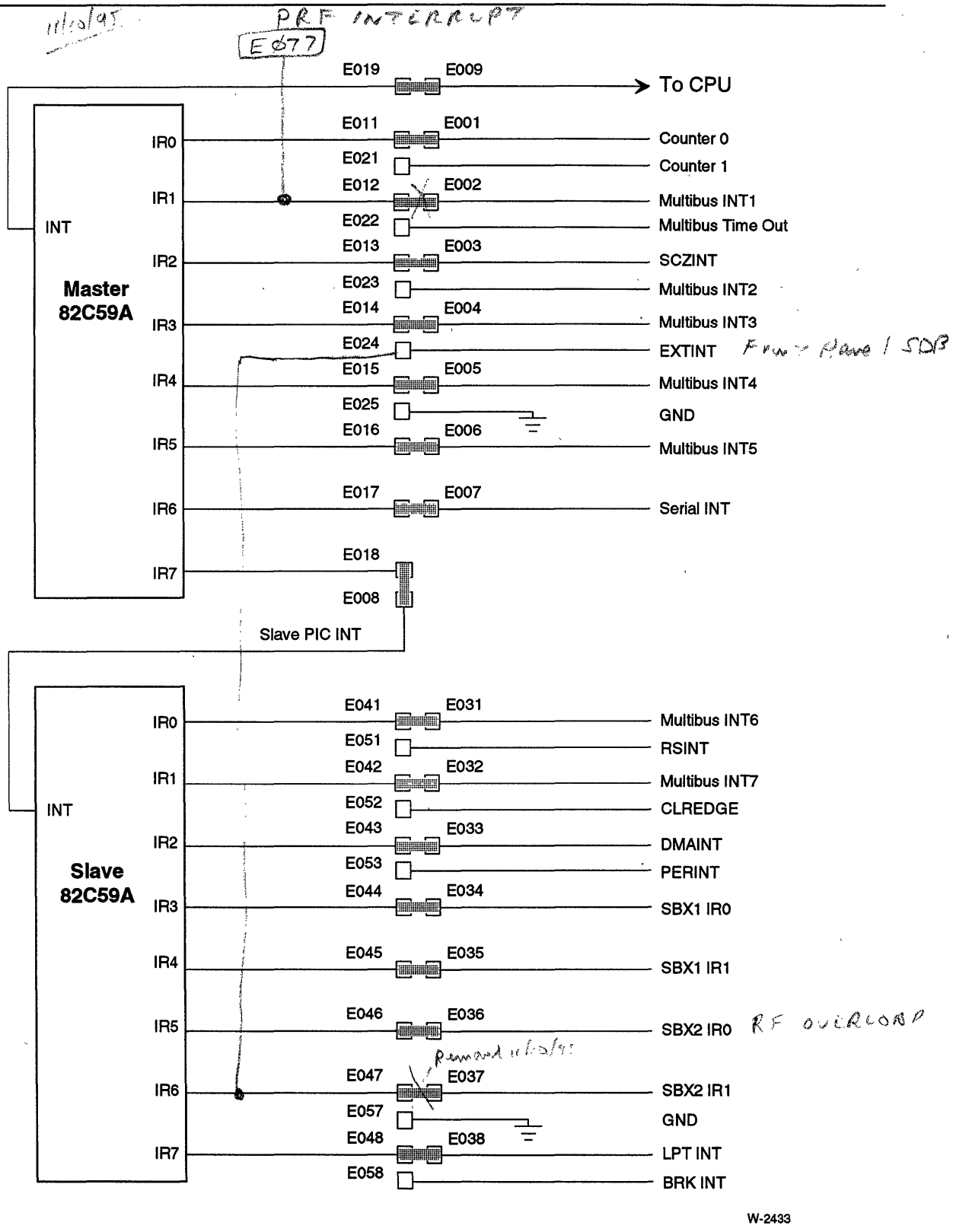


Figure 8-6. 82C59A PIC Interrupt Configuration

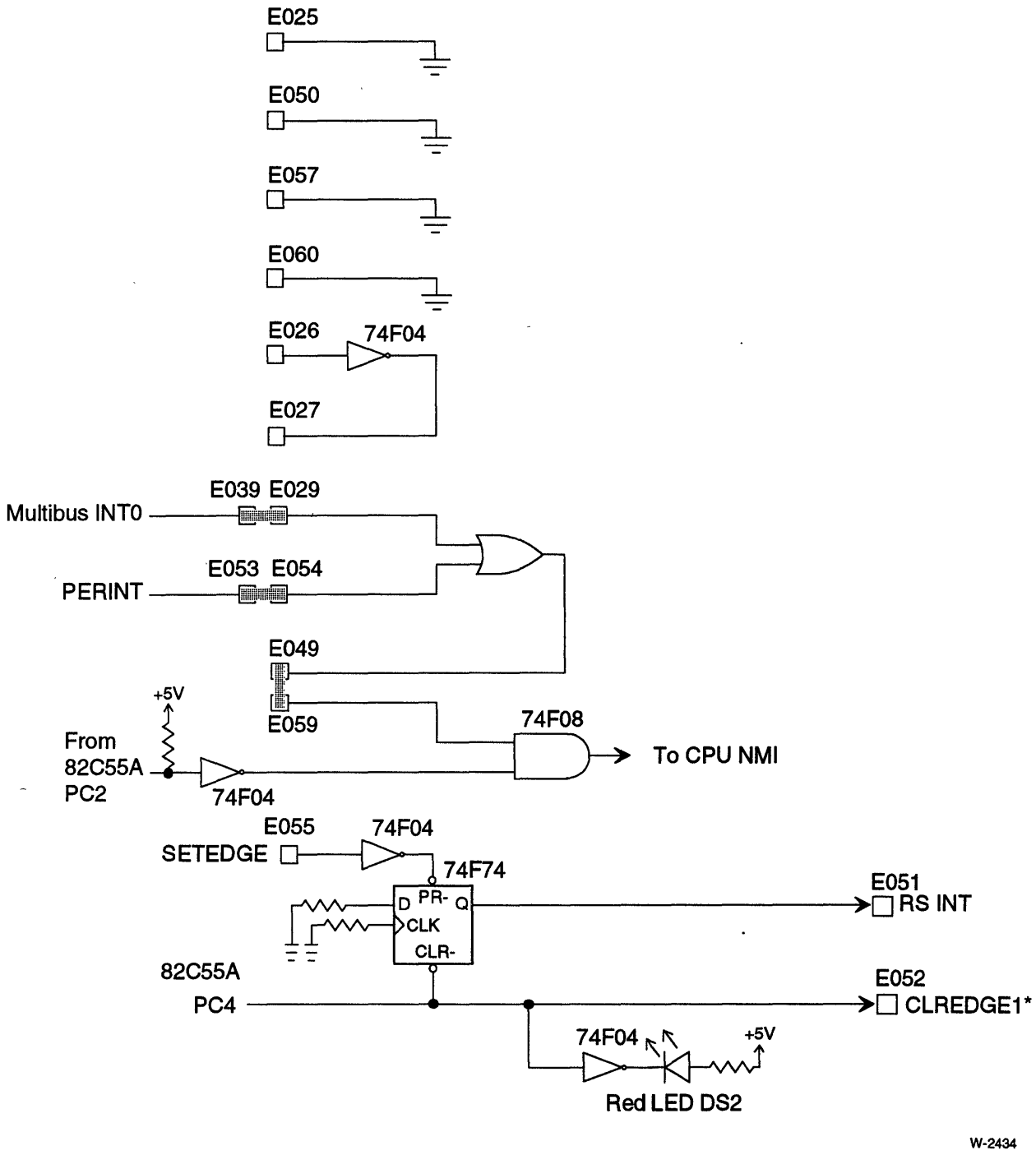


Figure 8-7. NMI Interrupt Logic and Spare Gates

Default PIC Configuration

Table 8-17 lists the default 82C59A PIC interrupt assignments.

Table 8-17. Default Interrupt Assignments

PIC	Interrupt Level	Interrupt Source
Master	IR0	Counter 0
	IR1	Multibus interface interrupt 1
	IR2	SCSI interface interrupt
	IR3	Multibus interface interrupt 3
	IR4	Multibus interface interrupt 4
	IR5	Multibus interface interrupt 5
	IR6	Serial I/O
	IR7	Slave PIC
Slave	IR0	Multibus interface interrupt 6
	IR1	Multibus interface interrupt 7
	IR2	DMA interrupt
	IR3	SBX 1 IR0
	IR4	SBX 1 IR1
	IR5	SBX 2 IR0
	IR6	SBX 2 IR1
	IR7	Line Printer interrupt

Default NMI Assignments

In the default configuration, the Intel486 CPU NMI input is tied to:

Multibus interrupt 0
OR-ed with memory parity error interrupt

Interrupt Destinations

You can configure the interrupt inputs using the stake pins listed in Table 8-7. In addition to the master and slave controller inputs, you can configure the interrupts through an inverter, edge-triggered latch, or an OR-gate. The inverter, latch, and OR-gate sections follow later in this chapter.

Table 8-18 lists the board interrupt destinations (the two PICs and the Intel486 microprocessor interrupt inputs); Table 8-21 lists the interrupt sources.

Table 8-18. Interrupt Destinations

Stake Pin	Destination	Page
E011 E012 E013 E014 E015 E016 E017 E018	Master PIC IR0 (MIR0) Master PIC IR1 (MIR1) Master PIC IR2 (MIR2) Master PIC IR3 (MIR3) Master PIC IR4 (MIR4) Master PIC IR5 (MIR5) Master PIC IR6 (MIR6) Master PIC IR7 (MIR7)	8-41
E041 E042 E043 E044 E045 E046 E047 E048	Slave PIC IR0 (SIR0) Slave PIC IR1 (SIR1) Slave PIC IR2 (SIR2) Slave PIC IR3 (SIR3) Slave PIC IR4 (SIR4) Slave PIC IR5 (SIR5) Slave PIC IR6 (SIR6) Slave PIC IR7 (SIR7)	8-42
E059	NMI Logic Input (NMIR)	8-42
E029 E054	OR-gate input 1 OR-gate input 2	8-43
E055	Latch input (SETEDGE1)	8-43
E026	Inverter input	8-43

Master PIC. The master PIC (Figure 8-6, page 8-38) has eight interrupt request inputs. In the default configuration, interrupt request level 0 (IR0) is the highest priority, and IR7 is the lowest. In some applications, there are a number of interrupts of equal priority. In this case, the 82C59A PICs can be programmed to rotate the priority assignments of the IR levels.

The interrupts at the IR input lines are handled inside the 82C59A PIC by two cascaded registers:the Interrupt Request Register (IRR), and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service. The ISR is used to store all the interrupt levels which are being serviced.

The slave PIC is normally connected to IR7. This allows the two PICs to service up to 15 interrupt levels.

The 8274 MPSC can be programmed to operate in the vectored mode. The serial interrupt from the MPSC is normally connected to IR6. The 82C59A PIC must be programmed to support a slave on IR6. The 8274 MPSC must be programmed for vectored operation.

NOTE

An occasional spurious interrupt will result in an interrupt at IR7. Your interrupt handler will then need to poll the device at IR7 to determine if it caused the interrupt. If you have unused IR levels on the master PIC, you can move the slave PIC to that IR level and connect IR7 to ground. If IR7 is connected to ground, all interrupts at IR7 must be spurious interrupts.

Slave PIC. The slave PIC is identical to the master PIC, except that the interrupt request output goes to stake pin E008 instead of the Intel486 CPU.

NOTE

An occasional spurious interrupt will result in an interrupt at IR7. If you have unused IR levels on the slave PIC, you can connect IR7 to ground. If IR7 is connected to ground, all interrupts at IR7 must be spurious interrupts.

Non-Maskable Interrupt Request (NMIR). The Non-Maskable Interrupt Request (NMIR) at E059 goes to the NMI input of the Intel486 CPU. This signal can be masked using the NMI mask. Figure 8-7 shows the NMI input. Table 8-19 shows how the NMI Mask is used to mask the NMIR signal.

Table 8-19. NMI Mask

NMIR	82C55A PPI, Bit PC2	NMI Input to CPU
X	1	0
0	0	0
1	0	1

Spare Gates. The iSBC 486/12 board has the following spare gates:

- inverter
- OR-gate
- 74F74 flipflop configured as an R*,S* latch

Figure 8-7 (page 8-39) shows the spare gates.

In the default configuration, the OR-gate is used to combine two sources for the NMIR input.

The 74F74 flipflop can be used to convert edge-triggered interrupts to level-triggered interrupts.

Multibus Interrupt Destinations. Table 8-20 lists the Multibus interrupt destinations (Multibus INT0* through INT7*). You can use the following signals to drive the Multibus interrupts:

- BUSDRIVE1* (E252)
- BUSDRIVE2* (E253)
- PER* (E251)

These signals are described on page 8-52.

Table 8-20. Multibus Interrupt Destinations

Stake Pin	Destination
E246	Multibus INT0*
E247	Multibus INT1*
E248	Multibus INT2*
E249	Multibus INT3*
E254	Multibus INT4*
E259	Multibus INT5*
E258	Multibus INT6*
E257	Multibus INT7*

Interrupt Sources

Table 8-21. Interrupt Sources

Source	Stake Pin	Description	Page
82C54A PIT	E001 E021	PIT Counter 0 PIT Counter 1	8-45
Connector P1	E039 E002 E023 E004 E005 E006 E031 E032	Multibus MBINT0 Multibus MBINT1 Multibus MBINT2 Multibus MBINT3 Multibus MBINT4 Multibus MBINT5 Multibus MBINT6 Multibus MBINT7	8-45
Multibus control logic	E022	MultibusTime-Out Interrupt	8-47
MB87033B SPC	E003	SCZINT (SCSI interrupt)	8-48
Front panel	E024	EXTINT (External interrupt)	8-48
8274 MPSC	E007	SERINT (Serial interrupt)	8-48
82258 ADMA	E033	DMAINT	8-48
Connector J5 Connector J6	E034 E035 E036 E037	SBX1 IR0 SBX1 IR1 SBX2 IR0 SBX2 IR1	8-48
Multibus logic	E056	BUSREQE (bus request error, pulsed)	8-49
Latch	E038	LPTINT (Line printer interrupt)	8-49
Break Detect	E058	BRKINT (Break Interrupt)	8-50
DRAM	E053	PERINT DRAM Parity Error	8-50
Inverter	E027	Output from spare inverter	8-50
Flipflop	E051	RSINT (User-defined function)	8-51
82C55A bit PC4	E052	CLREDGE1* (User-defined interrupt)	8-51
OR-gate	E049	Output from spare OR-gate	8-51
P2-19 J3-1	E028	PFINT* (power fail interrupt)	8-51

82C54A PIT Counter 0, Counter 1 Interrupts. PIT counters 0 and 1 can be programmed to provide one-time or periodic interrupts to the CPU. Chapter 7, "I/O Subsystem Configuration," provides information on how to configure and program the PIT.

In the default configuration, counter 0 is connected to master PIC IR0, counter 1 is used as a baud rate generator for the 8274 MPSC.

Multibus Interrupts. The Multibus interface has eight interrupt signal lines (INT0 through INT7) that can be connected to interrupt request inputs on either PIC. Figure 8-8 shows INT0 through INT7. BUSDRIVE1*, BUSDRIVE2*, and PER* are outputs used to drive the Multibus interrupts (stake pins MBINT0* through MBINT7*).

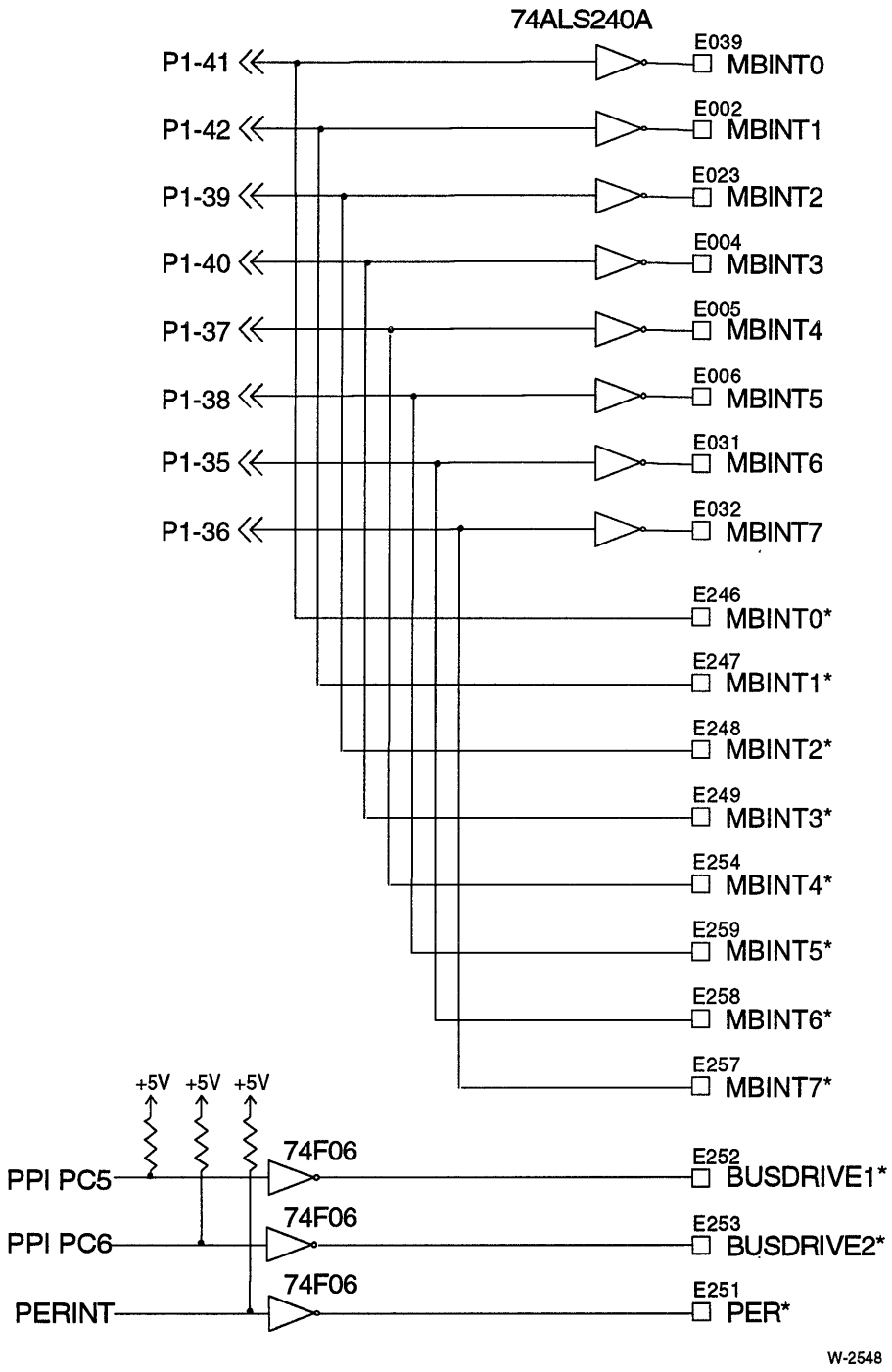
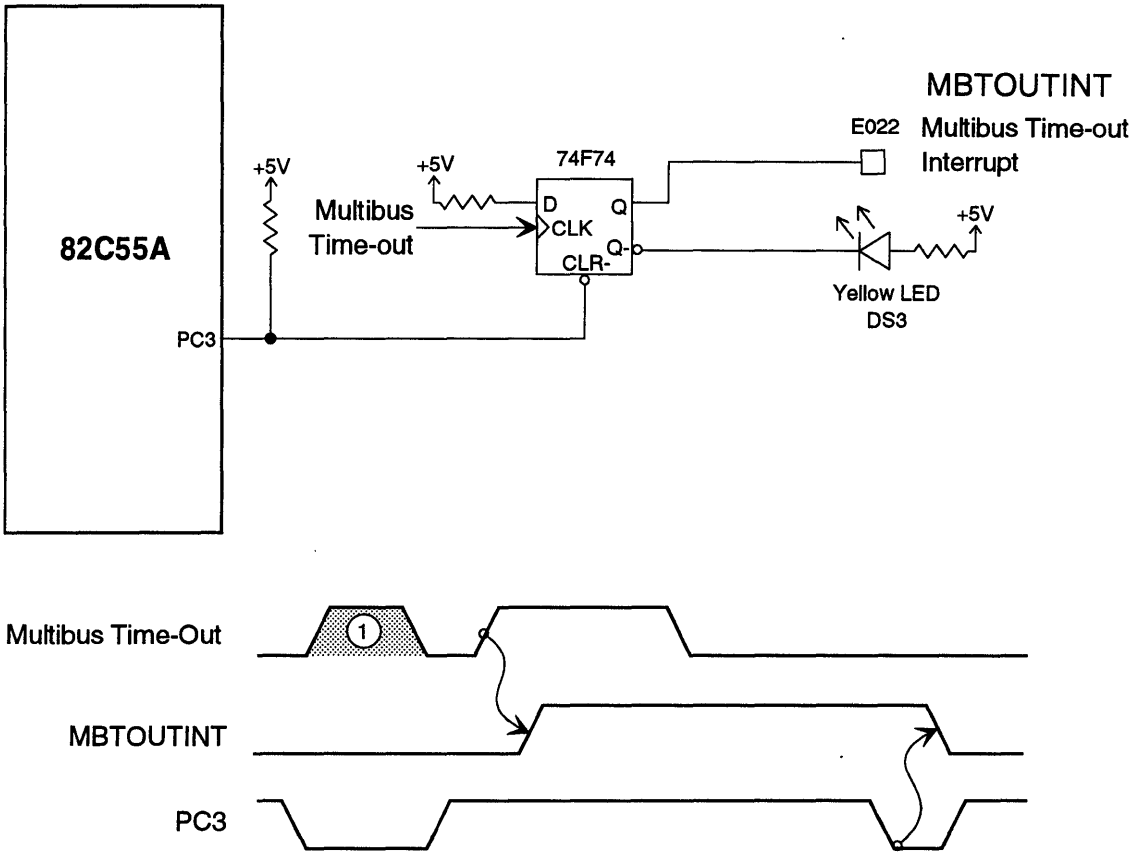


Figure 8-8. Multibus Interrupt Stake Pins

Multibus Time-Out Interrupt. The Multibus time-out interrupt is asserted if a Multibus cycle does not complete within approximately 8 ms. The interrupt is cleared by setting bit PC3 to 0. Chapter 7, "I/O Subsystem Configuration," discusses how to program the 82C55A PPI.



Note:
1. Multibus Time-Out signal has no effect while PC3 is low.

MB time-out	PC3	MBTOUTINT	LED DS3
Low-to-high	1	1	ON
Don't care	0	0	OFF
0	1	No Change	No Change
1	1	No Change	No Change

W-2184

Figure 8-9. Multibus Time-Out Interrupt

MB87033B SPC Interrupt (SCZINT). The MB87033B SPC interrupt request is indirectly connected to E003. The interrupt from the SPC can be masked using the SCZINTDIS register (described in Chapter 7).

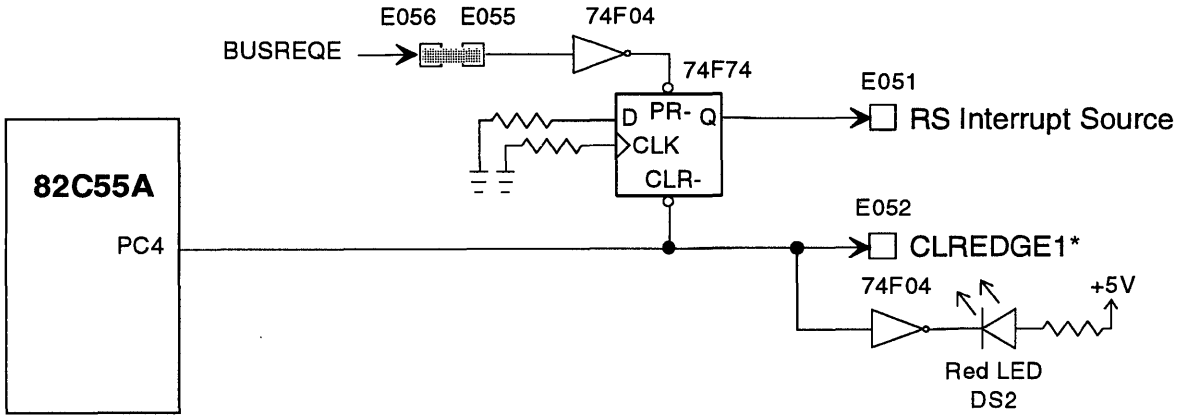
External Interrupt (EXTINT). The EXTINT* signal is an active low signal that is connected to the front panel connector (J3, pin 4). This signal is inverted to form the EXINT signal at stake pin E024. The EXINT signal can then be jumpered to a PIC interrupt request line.

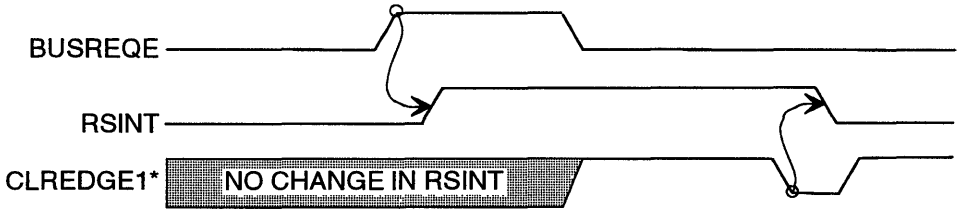
Serial Interrupt. The serial interrupt signal is from the 8274 MPSC. If you have programmed your MPSC in vectored-interrupt mode, you must also program the PIC for slave operation at the serial interrupt IR level. The MPSC in vectored mode will produce a vector indicating the source of the serial interrupt.

DMA Interrupt. The DMAINT interrupt signal is from the 82258 ADMA. Stake pin E033 is not connected if the 82258 ADMA is not installed.

SBX Interrupts. The iSBC 486/12 board has two SBX connectors. Each SBX connector has two interrupt request lines. The function of these interrupt lines depends on the type of SBX modules you have installed.

Multibus Bus Request Error Interrupt. The Multibus request error signal is momentarily asserted when the Multibus is not acquired within approximately 8 ms. (The Multibus timeout signal, on the other hand, is asserted when either the Multibus is not acquired, or the Multibus agent fails to assert XACK signal within the timeout period.) Use the spare 74F74 flipflop to latch the bus request error signal.





BUSREQE	CLREDGE1*	RSINT
0	1	No Change
0	0	0
1	1	1
1	0	Not Stable

W-2547

Figure 8-10. Latching the BUSREQE Signal

Line Printer Interrupt. The line printer interrupt is based on the LPT Acknowledge* signal from Connector J1. The LPT Acknowledge* signal is latched to produce the LPTINT signal (Figure 8-11).

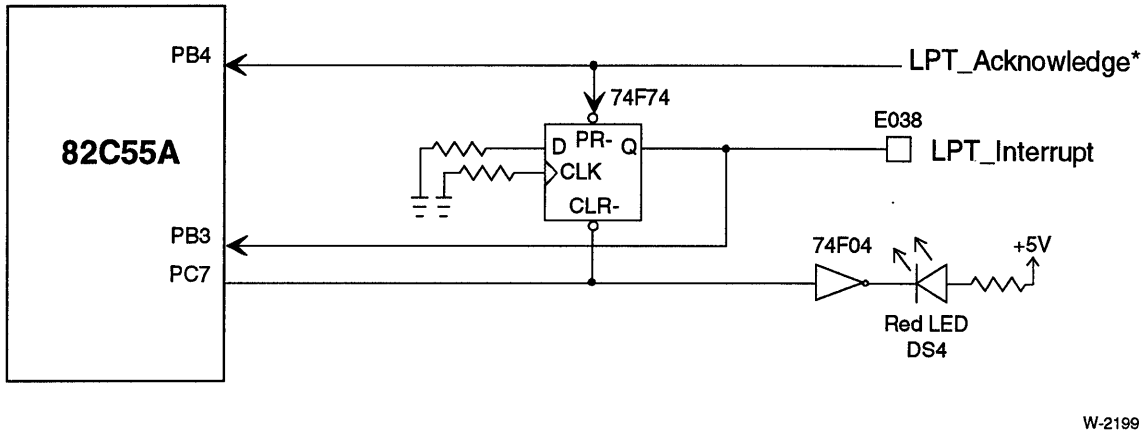


Figure 8-11. Line Printer Interrupt

The line printer interrupt is cleared by writing a 0 to the 82C55A PPI, bit PC7. Chapter 7, "I/O Subsystem Configuration," discusses how to program the 82C55A PPI.

Break Detect Interrupt. The iSBC 486/12 board has a serial break detect circuit that is separate from the break detect feature in the 8274 MPSC. Chapter 7, "I/O Subsystem Configuration," discusses how to use and reset the break detect interrupt.

Parity Error Interrupt. The board supports parity checking for all DRAM accesses. The parity error signal is latched.

The default board is configured with the parity interrupt OR-ed with the Multibus interface interrupt 0. The output from the OR-gate connects to the NMIR input (Figure 8-7).

The red LED (DS5) is lit when the parity latch is set. To turn DS5 off, clear the parity latch.

To clear the parity latch, execute a byte-write (use any data) at I/O address 0E8H.

To set the parity error latch, execute a byte-read at I/O address 0E8H. A parity error will result on the **next** DRAM access. (Note that the CPU may execute a number of instructions out of cache before accessing DRAM. The parity error interrupt will not be set until the next DRAM cycle after reading 0E8H.)

Chapter 7, "I/O Subsystem Configuration," discusses how to use the NMI mask to mask the parity error interrupt during power-up initialization..

Spare Inverter. The iSBC 486/12 board has a spare inverter to allow you to convert an active-low interrupt signal to an active-high signal for the PIC. Figure 8-7 shows the spare inverter.

Edge-Triggered Latch (RSINT). You can use the spare 74F74 flipflop to convert an edge-triggered interrupt signal to a level-triggered signal. The output of the flipflop is the RSINT signal. Figure 8-7 shows the spare flipflop.

Writing a "0" to the 82C55A PPI, Port C, bit 4 (port address 0CCH) clears the edge-triggered latch and turns red LED (DS2) off.

The latch is not configured on the default board.

OR-Gate. You can combine two interrupt inputs using the spare OR-gate.

In the iSBC 486/12 board's default configuration, Multibus Interrupt 0 and Parity Error are the OR-gate inputs. The OR-gate output is configured as the Intel486 NMIR input.

Power-Fail Interrupt (PFINT). The power-fail interrupt (PFINT*) signal is an active-low signal from either the front panel (connector J3, pin 1) or the auxiliary interface (connector P2, pin 19). The PFINT* signal is inverted to produce the PFINT signal at stake pin E028. If more than one source is driving the PFINT* signal, each source should have open collector drivers.

Multibus Interrupt Drivers

The following table lists the three Multibus interrupt drivers.

Table 8-22. Multibus Interrupt Drivers

Stake Pin	Source	Description
E252 E253	82C55A PPI PC5 PC6	BUSDRIVE1* BUSDRIVE2*
E251	DRAM	PER* (Parity Error*)

Bus Drive (BUSDRIVE1*, BUSDRIVE2*). The BUSDRIVE1* and BUSDRIVE2* signals are from the 82C55A PPI. Figure 8-12 shows the BUSDRIVEx* signals. Figure 8-8 (page 8-46) shows the INT<7:0>* signals. Open collector drivers are used to drive these signals.

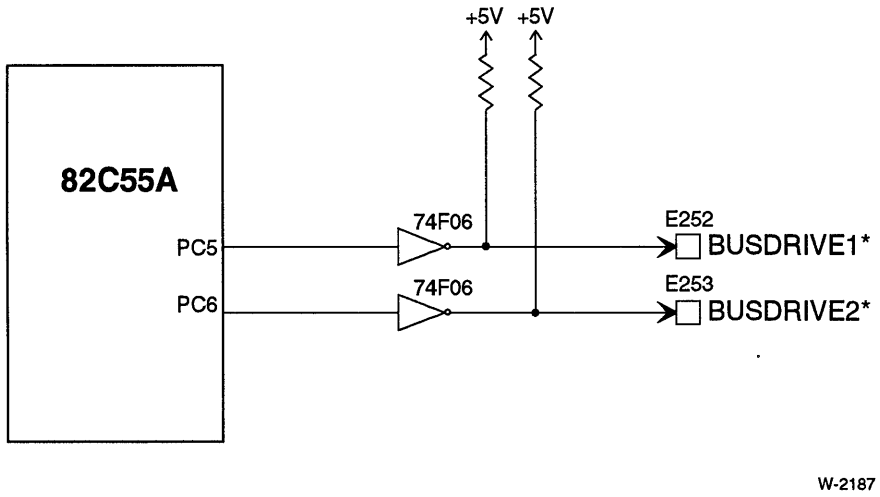


Figure 8-12. BUSDRIVE1* and BUSDRIVE2* Signals

Parity Error (PER*). The memory parity error interrupt (PERINT) is inverted to form the PER* signal. This active-low signal can be used to drive one of the Multibus interrupts (INT0* through INT7*). An open collector driver is used to drive this signal.

Interrupt Programming

The board supports direct-vectored and polled modes of servicing interrupts. Bus-vectored interrupts are not supported.

Figure 8-13 is a P/LM programming example for initializing a 82C59A PIC. The following sections discuss the two modes of interrupt operation. For more 82C59A PIC programming information, see the *Peripheral Components* Handbook.

82C59A PIC Polled Mode

In polled mode, the Intel486 microprocessor does not receive an interrupt vector. The 82C59A is polled periodically to determine the highest priority level requesting service. If you use polled mode, you must program both 82C59A PICs and the 8274 MPSC for polled mode operation.

Direct-Vectored Mode

In direct-vectored mode, the master 82C59A PIC receives an interrupt request from the slave PIC, the 8274 MPSC, or a direct input. The master then passes the request to the Intel486 Microprocessor. The interrupt vector is supplied by the requesting device during the second interrupt acknowledge pulse.

For direct-vectored operation, all interrupt devices must be programmed. Note the following programming requirements:

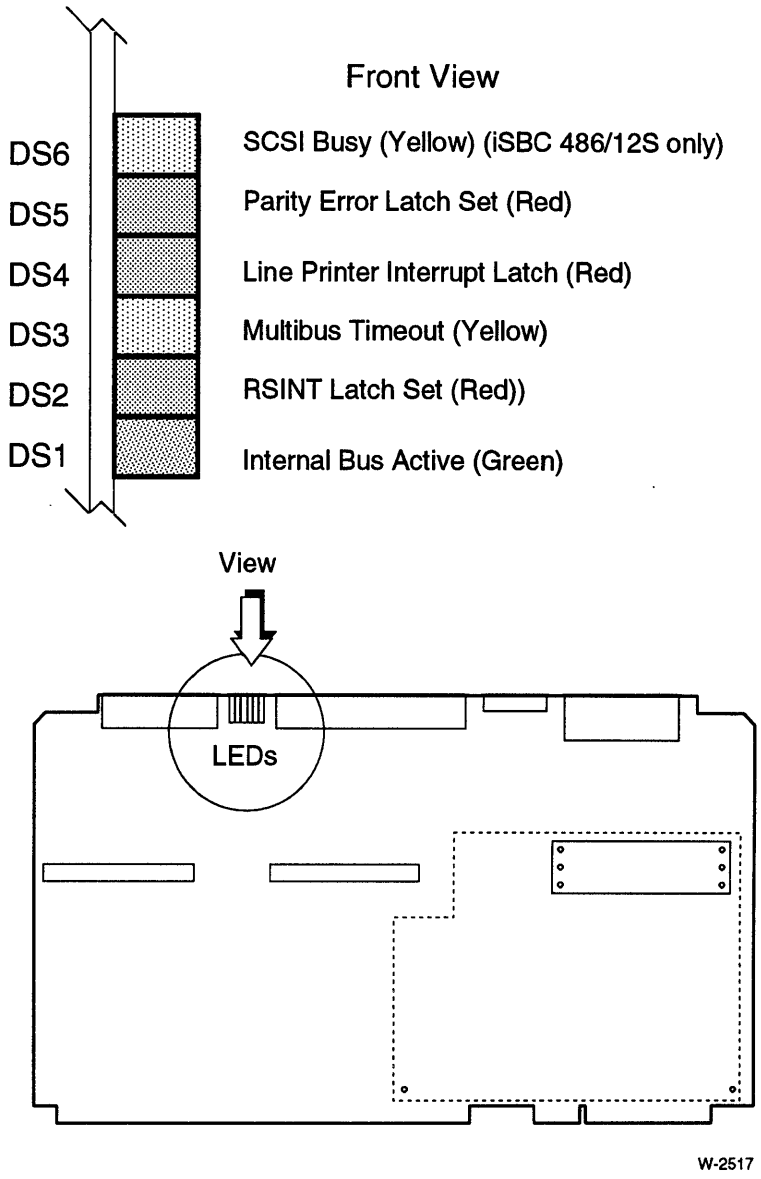
- Program both 82C59As for ICW4.
- Program both 82C59As for cascade mode (ICW1: IC4=1, SNGL=0).
- Program both 82C59As with the vector base, using ICW2.
- Program the master 82C59A for the slave levels connected to it, using ICW3.
  - In the default configuration, the slave 82C59A level is IR7.
  - In the default configuration, the 8274 level is IR6.
- Both 82C59As must be programmed for buffered mode, using ICW4.
- Program the 8274 to meet your system requirements.

```
/*Declarations*/

declare
     m_pic_lo       lit '0C0H',           /* ICW1,OCW2,OCW3 register address */
     s_pic_lo       lit '0C4H',           /* Edge-triggered, SNGL = 0 */
     m_pic_hi       lit '0C2H',           /* ICW2-4, OCW1 register address */
     s_pic_hi       lit '0C6H',           /* ICW2-4, OCW1 register address */
     pics_icw1      lit '0001$0001B',     /* Local interrupts, NO ICW5. */
     m_pic_icw2     lit '0010$1000B',     /* Vector base of 28H (40D) */
     s_pic_icw2     lit '0011$0000B',     /* Vector base of 30H (48D) */
     m_pic_icw3     lit '1000$0000B',     /* IR 7 has slave attached */
     s_pic_icw3     lit '0000$0111B',     /* Slave ID 7 */
     m_pic_icw4     lit '0000$1101B',     /* Master, Buffered 8086 MODE */
     s_pic_icw4     lit '0000$1001B',     /* Slave, Buffered 8086 MODE */
     m_pic_mask     lit '1011$1111B',     /* Serial interrupt on 6, */
                                          /* Slave on 7 */
     s_pic_mask     lit '1111$1111B',     /* Mask all int on slave */
     m_pic_eoi      lit '011$00$110B';    /* Master interrupt 6 serial, */
                                          /* Slave interrupts not used */


/*Program*/

 Initialize Master and Slave PICS

     output(m_pic_lo)=pics_icw1;
     output(s_pic_lo)=pics_icw1;

     output(m_pic_hi)=m_pic_icw2;         /* vector base master */
     output(s_pic_hi)=s_pic_icw2;         /* vector base slave */

     output(m_pic_hi)=m_pic_icw3;
     output(s_pic_hi)=s_pic_icw3;

     output(m_pic_hi)=m_pic_icw4;
     output(s_pic_hi)=s_pic_icw4;

     output(m_pic_hi)=m_pic_mask;      /* On master level 6 Serial Int */
     output(s_pic_hi)=s_pic_mask;      /* Also on master Level 7 Slave PIC*/

     enable;

End Hardware_Init;
```

Figure 8-13. 82C59A PIC Initialization Programming Example

Light-Emitting Diodes (LEDs)

Figure 8-14 shows the LEDs as viewed from the front of the system card cage.



Note:
LED DS6 (SCSI Busy) is only on iSBC® 486/12S series boards.

Figure 8-14. Light-Emitting Diodes (LEDs)

Table 8-23 defines the function of the LEDs.

Table 8-23. LED Functions

LED	Color	Function	Refer To
DS1	Green	Run LED. DS1 is on when the CPU, ADMA, or dual-port memory is performing a bus cycle. OFF = no bus activity ON = system running or hung Flashing = system running	--
DS2	Red	RSINT Latch. DS2 is on if the RS Latch is set. (The latch is set when the 82C55A PPI, bit PC4 is set to 1.)	Chapter 7
DS3	Yellow	Multibus Time-Out. DS3 is on if a time-out has occurred.	Chapter 7
DS4	Red	Line Printer Interrupt Latch. DS4 is on when the interrupt is asserted. The interrupt is asserted when the printer is ready for another byte.	Chapter 7
DS5	Red	DRAM Parity Error Latch. DS5 is on when the parity error latch is set.[1]	Chapter 7
DS6	Yellow	SCSI Busy. DS6 is on when the SCSI BUSY* signal is asserted. (iSBC 486/12S series boards only.)	--

Note:
[1] The parity error latch may be set by writing to I/O address 0E8H. The parity error latch will be set at the next DRAM access.

Questions and Answers

Q. How is the ADMA configuration on the iSBC 486/12 board different from the iSBC 386/12S?

A. The ADMA on the iSBC 486/12S series does not support the serial channels of the 8274 MPSC. On the iSBC 486/12S series, DMA channel 0 is dedicated to the SPC; the remaining channels are dedicated to the iSBX interfaces.

Q. How is the 82C59 PIC configuration on the iSBC 486/12 board different from the iSBC 386/12S?

A. All 16 interrupt request levels on the two 82C59A PICs are configurable on the iSBC 486/12 board. In addition, serial break detect interrupt has been added.

Q. Does the iSBC 486/12 board support bus-vectored interrupts?

A. No. The iSBC 486/12 board only supports direct-vectored interrupts from the slave PIC or 8274 MPSC, or polled mode.

***

Connectors and Cables 9

Chapter Contents

Introduction 9-2
User-Supplied Connectors 9-5
User-Supplied Cables 9-6
Multibus Interface (Connector P1) 9-7
Auxiliary Interface (Connector P2) 9-12
Memory Interface (Connector P3) 9-14
Parallel Interface (Connector J1) 9-15
SCSI Interface (Connector J2) 9-18
Front Panel Interface (Connector J3) 9-21
Serial Interfaces (Connectors J4L and J4U) 9-22
SBX Interfaces (Connectors J5 and J6) 9-29

Introduction

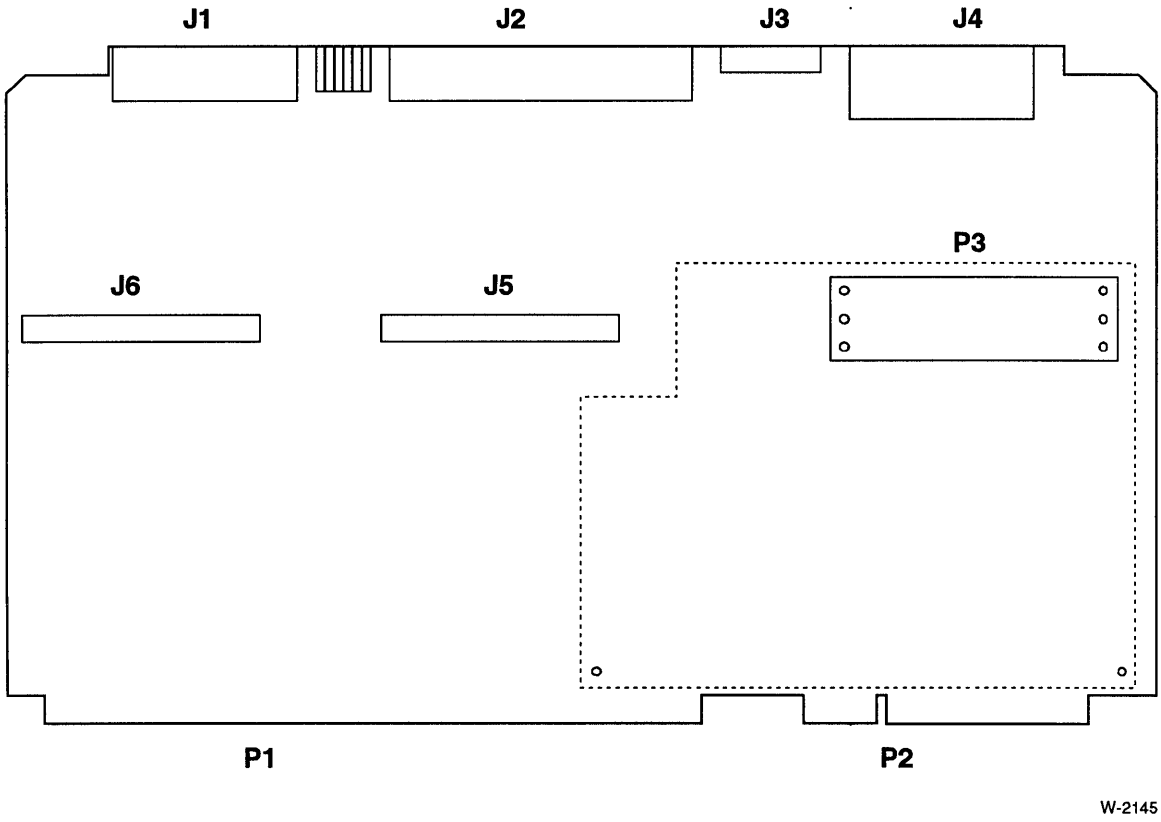
This chapter discusses all the on-board connectors for the iSBC 486/12 series boards. Table 9-1 lists the connectors.

Table 9-1. Connectors

Connector	Description	Page
P1 P2 P3	Multibus interface Auxiliary interface MM3 memory module	9-7 9-12 9-14
J1	Parallel printer	9-15
J2	Single-ended SCSI[1]	9-18
J3	Front panel	9-21
J4U J4L	RS-232C DCE interface RS-232C DCE, DTE, or RS-422A/449 DCE interface	9-22 9-22
J5 J6	SBX 1 SBX 2	9-29 9-29

Note:
[1] iSBC 486/12S series only

Figure 9-1 shows the location of the respective connectors.



Note:
Connector J2 (SCSI interface) is only on iSBC® 486/12S series boards.

Figure 9-1. Location of Connectors

Connectors P1 and P2 are edge connectors that plug into the Multibus backplane. Connector P3 connects to Intel MM3 memory modules. Connectors J1, J2, J3, and J4 connect to cables which typically lead to the chassis back panel. Figure 9-2 shows the following connectors:

J1	printer cable	see page 9-16
J2	SCSI cable	see page 9-19 (iSBC 486/12S series only)
J3	front panel cable	depends on application
J4 Upper	RS-232C cable	see page 9-25
J4 Lower	RS-232C cable or RS-442A/449 cable	see page 9-25 see page 9-27

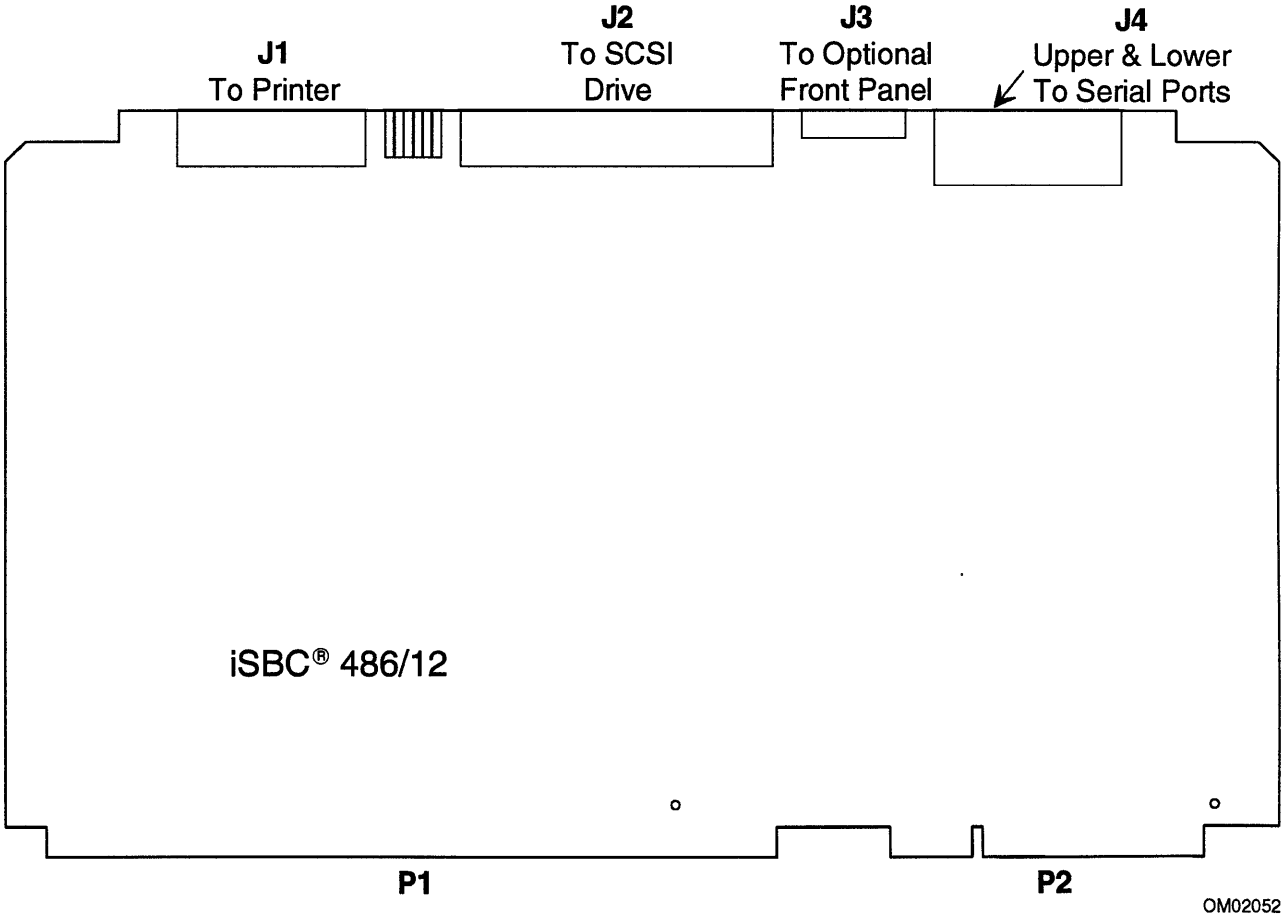


Figure 9-2. Connectors and Cables

User-Supplied Connectors

Table 9-2. User-Supplied Connectors

On-Board Connector	Number of Pins	Centers (Inches)	Connector Type	Vendor	Vendor Part Number
P1 (Multibus)	86	0.156	Soldered[1] PC board mount	Viking ELFAB	2KH43/9AMK12 BS1562D43PBB
			Wirewrap[2] (without mounting ears)	EDAC ELFAB	337086540201 BW1562D43PBB
			Wirewrap[2] (with mounting ears and 0.128 mounting holes)	EDAC ELFAB	337086540202 BW1562A43PBB
P2 (Auxiliary)	60	0.1	Soldered Soldered	KELAM KELAM Soldered	RF30-2803-5 110-10-001-37 T&B Ansley A3020
			Soldered Soldered	EDAC ELFAB	345060500202 97169001
			Wirewrap (without mounting ears)	EDAC ELFAB	345060540201 BW1020D30PBB
			Wirewrap (with mounting ears and 0.128 mounting holes)	EDAC ELFAB	345060524202 BW1020A30PBB
			Wirewrap (with mounting ears and 0.128 mounting holes)	TI Viking	H421121-30 3KH30/9JNK
J3 (front panel)	14	0.5	Un-shrouded Flat Crimp Socket	3M	3385-6014
J1 (parallel port)	26	0.1	Flat Crimp Socket	3M	3399-6026
	25	0.1	D-shell M/F Filter	Positronic Ind. Inc.	FDF25F35200X
J2 (SCSI port)	50	0.1	Flat Crimp Socket	3M	3433-1033
J4 (serial ports)	26	0.1	Flat Crimp Socket	3M	3399-6026
	25	0.1	D-shell M/F Filter	Positronic Ind. Inc.	FDF25F35200X
J5/J6 (8-bit) (16-bit SBX)	36 36	0.1 0.1	Soldered Soldered	Viking Viking	VSP01VT18A01 (male) VSP01VT22A01 (male)

Notes:
[1] Connector heights are not guaranteed to conform to Intel packaging equipment standards.
[2] Wirewrap pin lengths are not guaranteed to conform to Intel packaging equipment standards.

User-Supplied Cables

NOTE

The 25-pin D-shell Male/Female **filter** connectors listed in the previous table are recommended for use with the serial cables to minimize electromagnetic emissions from a system enclosure.

Table 9-3 lists the cable wire and connectors for the user-supplied cables shown in this chapter.

Table 9-3. User-Supplied Cable Parts

Interface Type	Serial Mode	Board Connector	Cable Wire Ribbon	Back Panel Connector
Auxiliary			3M-3365/60	
Parallel		26-pin[1] 3M-3399-6026	3M-3365/25	36-pin 3M-3367-1000
SCSI		50-pin 3M-3433-1003	3M-3365/50	50-pin 3M-3425-6000
RS-232C	 DTE DCE	26-pin[1] 3M-3399-6026 3M-3399-6026	3M-3365/25	25-pin[2] 3M-3634-1000[3] 3M-3635-1000[4]
RS-422/ RS-449A	DCE	26-pin[1] 3M-3399-6026	3M-3365/256	37-pin[5] 3M-3637-1000

Notes:
[1] Pin 1 of the edge connector is not connected to the flat cable.
[2] Connector may be used with cable housing 3M-3485-2300.
[3] DTE (Data Terminal Equipment) is a male connector.
[4] DCE (Data Communication Equipment) is a female connector.
[5] Connector may be used with cable housing 3M-3485-2400.

Multibus Interface (Connector P1)

Connector P1 is an 86-pin edge connector that serves as the primary Multibus interface. Connector P1 supports the IEEE 796 Specification.

Table 9-4 lists the Multibus signals on Connector P1. The iSBC 486/12 board does not support the INH2* and INTA* signals.

Table 9-5 (page 9-10) lists the pin assignments for the P1 connector in numerical order.

Table 9-4. Multibus Interface Signals (Connector P1)

Signal Name	Pin Number	Description
ADR0* ADR1* ADR2* ADR3* ADR4* ADR5* ADR6* ADR7* ADR8* ADR9* ADR10* ADR11* ADR12* ADR13* ADR14* ADR15* ADR16* ADR17* ADR18* ADR19*	57 58 55 56 53 54 51 52 49 50 47 48 45 46 43 44 28 30 32 34	*Address.* These 20 lines transmit the address of the memory or I/O port to be accessed. For memory access, ADR0* (when active) enables the even byte bank (DATA<7:0>* on the Multibus. ADR19* is the most significant address bit for 20-bit addressing. Address lines ADR<23:20> are on connector P2.
BCLK*	13	*Bus Clock.* Used to synchronize the bus arbitration logic on all bus masters. The iSBC 486/12 board can be configured to drive the BCLK* signal. Configuration of the BLCK* signal is discussed in Chapter 7.
BHEN*	27	*Byte High Enable.* When active low, enables the odd byte bank (DATA<15:8>*) onto the Multibus data lines.
BPRO*	16	*Bus Priority Out.* In serial (daisy chain) priority resolution schemes, BPRO* must be connected to the BPRN* input of the bus master with the next lower bus priority. Chapter 7 discusses how to configure the BPRO* and BPRN* signals.
BREQ*	18	*Bus Request.* In parallel priority resolution schemes, BREQ* indicates that a particular bus master requires control of the bus for one or more data transfers. BREQ* is synchronized with BCLK*. Chapter 7 discusses how to configure the BPREQ* and BCLK* signals.

Table 9-4. Multibus Interface Signals (Connector P1 Continued)

Signal Name	Pin Number	Description
BUSY*	17	*Bus Busy.* Indicates that the bus is in use and prevents all other bus masters from gaining control of the bus. BUSY* is synchronized with BCLK*.
CBRQ*	29	*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.
CCLK*	31	Constant Clock. Provides a clock signal of constant frequency for use by other system modules. The iSBC 486/12 board can be configured to drive the CCLK* signal. Configuration of the CCLK* signal is discussed in Chapter 7.
DATA0* DATA1* DATA2* DATA3* DATA4* DATA5* DATA6* DATA7* DATA8* DATA9* DATA10* DATA11* DATA12* DATA13* DATA14* DATA15*	73 74 71 72 69 70 67 68 65 66 63 64 61 62 59 60	*Data.* These 16 bidirectional data lines transmit and receive data to and from the addressed memory location or I/O port. DATA15* is the most significant bit. For data byte operations, DATA<7:0>* is the even byte, and DATA<15:8>* is the odd byte.
INH1*	24	*Inhibit RAM.* For system applications, allows dual port memory addresses to be overlaid by ROM/PROM or memory mapped I/O devices. The iSBC 486/12 board does not allow Multibus access to its own EPROM.
INH2*	N.C.	*Inhibit ROM.* Not connected.
INIT*	14	Initialize. Reset the entire system to a known internal state. Chapter 8 discusses how to configure the INIT* signal.
INTA*	N.C.	*Interrupt Acknowledge.* This signal is not connected.
INT0* INT1* INT2* INT3* INT4* INT5* INT6* INT7*	41 42 39 40 37 38 35 36	*Interrupt Request.* These eight lines transmit interrupt requests to the appropriate interrupt handler. INT0* has the highest priority. The iSBC 486/12 board can be configured to drive three of the interrupt request lines. Chapter 8 discusses how to configure interrupts to the iSBC 486/12.

Table 9-4. Multibus Interface Signals (Connector P1 Continued)

Signal Name	Pin Number	Description
IORC*	21	*I/O Read Command.* Indicates that the address of an I/O port is on the Multibus address lines and that the output of that port is to be read (placed) onto the Multibus data lines.
IOWC*	22	*I/O Write Command.* Indicates that the address of an I/O port is on the Multibus address lines and that the contents of the Multibus data lines are to be accepted by the addressed port.
LOCK*	25	*Lock.* Indicates a locked access is being performed on the Multibus. The iSBC 486/12 dual port memory can be locked to the Multibus by another agent using this signal. The iSBC 486/12 board can assert this signal to lock dual port memory on another board in the system. If the LOCK* signal is asserted, the current bus master retains control of the bus even when a higher priority bus master is requesting the bus.
MRDC*	19	*Memory Read Command.* Indicates that the address of a memory location is on the Multibus address lines and that the contents of that location are to be read (placed) on the Multibus data lines.
MWTC*	20	*Memory Write Command.* Indicates that the address of a memory location is on the Multibus address lines and that the contents of the Multibus data lines are to be written into that location.
XACK*	23	*Transfer Acknowledge.* Indicates that the address memory location has completed the specified read or write operation. That is, data has been placed onto or accepted from the Multibus data lines.

Table 9-5. Connector P1 Pin Assignments

Component Side		Solder Side	
1	GND	2	GND
3	+5 V	4	+5 V
5	+5 V	6	+5 V
7	+12 V	8	+12 V
9	Not Connected	10	Not Connected
11	GND	12	GND
13	BCLK*	14	INIT*
15	BPRN*	16	BPRO*
17	BUSY*	18	BREQ*
19	MRDC*	20	MWTC*
21	IORC*	22	IOWC*
23	XACK	24	INH1*
25	LOCK*	26	Not Connected
27	BHEN*	28	AD16*
29	CBRQ*	30	AD17*
31	CCLK*	32	AD18*
33	Not Connected	34	AD19*
35	INT6	36	INT7
37	INT4	38	INT5
39	INT2	40	INT3
41	INT0	42	INT1
43	ADR14*	44	ADR15*
45	ADR12*	46	ADR13*
47	ADR10*	48	ADR11*
49	ADR8*	50	ADR9*
51	ADR6*	52	ADR7*
53	ADR4*	54	ADR5*
55	ADR2*	56	ADR3*
57	ADR0*	58	ADR1*
59	DATA14*	60	DATA15*
61	DATA12*	62	DATA13*
63	DATA10*	64	DATB*
65	DATA8*	66	DATA9*
67	DATA6*	68	DATA7*
69	DATA4*	70	DATA5*
71	DATA2*	72	DATA3*
73	DATA0*	74	DATA1*
75	GND	76	GND
77	Not Connected	78	Not Connected
79	-12 V	80	-12 V
81	+5 V	82	+5 V
83	+5 V	84	+5 V
85	GND	86	GND

Figure 9-3 shows the pin numbering of connector P1.

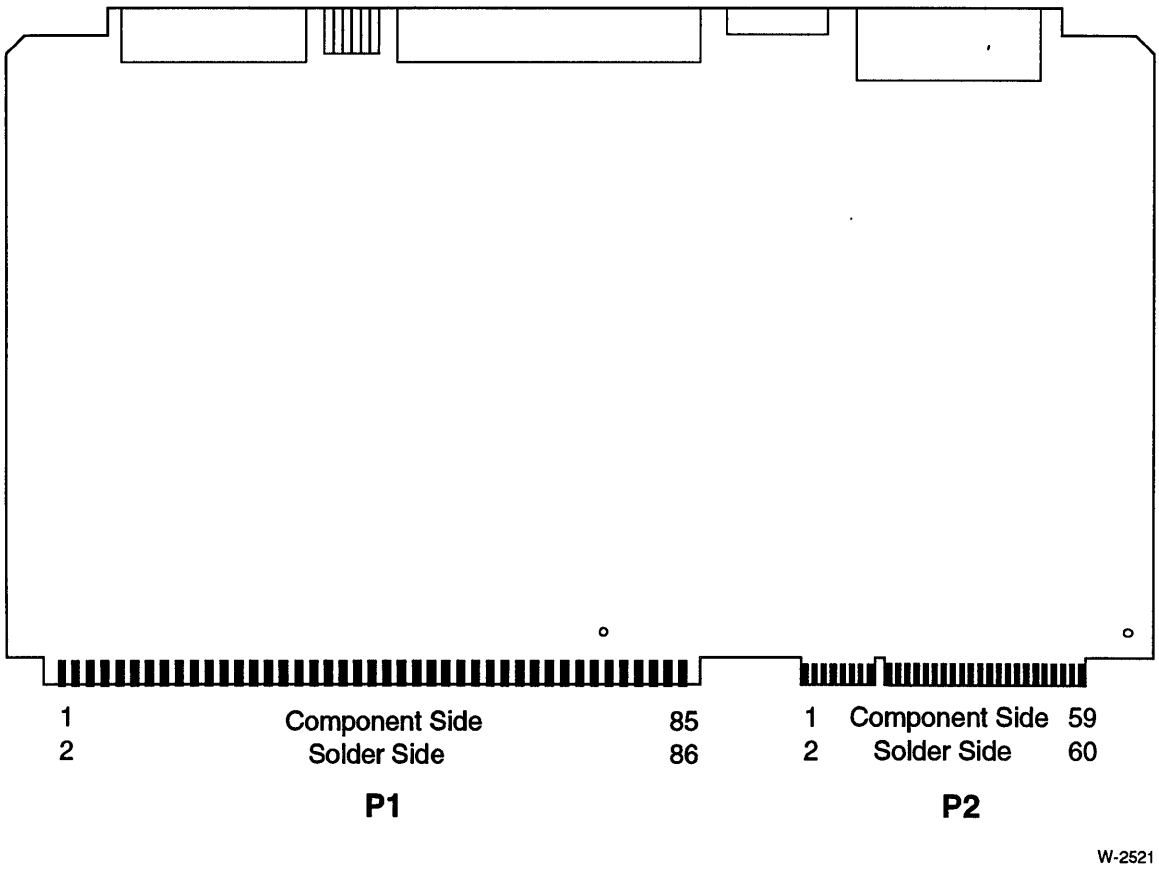


Figure 9-3. Pin Numbering on Connectors P1 and P2.

Auxiliary Interface (Connector P2)

Connector P2 is a 60-pin edge-connector and serves as the auxiliary interface. Table 9-6 describes the signals on connector P2. Some of the signals on connector P2 are also available on the front panel interface (connector J3). Table 9-7 (page 9-14) lists the pin assignments for connector P2.

NOTE

Multibus address lines ADR<23:20>* are on connector P2 and are terminated with a removable resistor pack. The resistor pack should be removed if the backplane or another board terminates the upper address lines. Refer to Chapter 8 for information on the Multibus termination resistor pack.

CAUTIONS

1. **CYCLE and SCZBSY are outputs and must not be driven by other boards. Only the Multibus address lines should be daisy chained with other boards in the system.**

2. **Connector P2 does not support the Intel Local Bus Extension (iLBX) interface or the Intel synchronous memory interface.**

Table 9-6. Auxiliary Interface Signals (Connectors P2 and J3)

Signal Name	Connector P2 Pin Number	Connector J3 Pin Number	Description
ACLO	Not Connected	Not Connected	*AC Low.* Not connected.
ADR20* ADR21* ADR22* ADR23*	57 58 55 56	Not Connected	Address. These 4 lines are the 4 most significant address bits in 24-bit addressing. Termination of ADR<23:20>* is discussed in Chapter 8.
CYCLE	32	8	*Internal Bus Cycle.* CYCLE is asserted when the CPU, ADMA, or dual-port memory is performing a bus cycle.
HALT*	Not Connected	Not Connected	*Halt.* Not connected.
MPRO*	Not Connected	Not Connected	*Memory Protect.* Not connected.
PFSN*	Not Connected	2	*Power Fail Sense.* Provides a latch for power failure event. Chapter 7 discusses how to use the 82C55 to sense the state of this signal.
PFINT*	19	1	*Power Fail Interrupt.* This signal from the power supply interrupts the processor when a power failure occurs. Chapter 8 discusses how to configure the interrupt.
PFSR*	Not Connected	Not Connected	*Power Fail Reset.* Not connected.
AUXRST*	38	6	*Auxiliary Reset.* This externally generated signal initiates a power-up reset. Chapter 8 describes the effect of the reset.
WAIT*	Not Connected	Not Connected	*Bus Master Wait State.* Not connected.
CONSLOCK*	34	Not Connected	*Console Lock.* This signal is used to inhibit or lock both channels of the 8274 MPSC from receiving characters. When asserted, this signal also blocks the serial signal to the break-detect logic. Chapter 7 discusses how to use this signal.
SCZBSY	10	3	*SCSI Busy.* This signal indicates when the SCSI interface is busy. This signal also drives LED DS6.
EXINT*	Not Connected	3	*External Interrupt.* This signal is driven externally and provides an interrupt to the CPU. Chapter 8 discusses how to configure this signal.

Table 9-7. Connector P2 Pin Assignments

Component Side		Solder Side	
1		2	
3		4	
5		6	
7		8	
9		10	SCZBSY
11		12	
13		14	
15		16	
17		18	
19	PFINT*	20	
21		22	
23		24	
25		26	
27		28	
29		30	
31		32	CYCLE
33		34	CONSLOCK*
35		36	
37		38	AUXRST*
39		40	
41		42	
43		44	
45		46	
47		48	
49		50	
51		52	
53		54	
55	ADR22*	56	ADR23*
57	ADR20*	58	ADR21*
59		60	

Note:
All pin numbers without signals listed above are not connected.

Figure 9-3 (page 9-11) shows the pin numbering on connector P2.

Memory Interface (Connector P3)

Connector P3 is the Intel MM3 memory module interface. You can install two memory modules stacked on connector P3.

Refer to Chapter 3 for memory module installation instructions.

Parallel Interface (Connector J1)

Connector J1 is a 26-pin parallel interface. Table 9-8 lists the pin assignments.

Refer to Chapter 7, "I/O Subsystem Configuration," for a discussion of the parallel interface signals and the configuration options for connector J1.

Table 9-8. Connector J1 Pin Assignments

Connector J1 Pin Number	Centronics Pin Number	Centronics Function	82C55A PPI Port
1	31	Not Connected	
2	13	Not Connected	
3	30	Not Connected	
4	12	Error	Port B, Bit 6
5	29	GND	
6	11	Busy	Port B, Bit 7
7	28	GND	
8	10	Acknowledge*	Port B, Bit 4
9	27	GND	
10	9	Data Bit 7	Port A, Bit 7
11	26	GND	
12	8	Data Bit 6	Port A, Bit 6
13	25	GND	
14	7	Data Bit 5	Port A, Bit 5
15	24	GND	
16	6	Data Bit 4	Port A, Bit 4
17	23	GND	
18	5	Data Bit 3	Port A, Bit 3
19	22	GND	
20	4	Data Bit 2	Port A, Bit 2
21	21	GND	
22	3	Data Bit 1	Port A, Bit 1
23	20	GND	
24	2	Data Bit 0	Port A, Bit 0
25	19	GND	
26	1	Data Strobe	Port C, Bit 0

Figure 9-4 shows two cables: a ribbon cable from connector J1 to the back panel of the chassis, and a ribbon cable from the back panel to the printer. The pin assignments and connectors of the second cable are compatible with standard Centronics printer cables. Figure 9-5 shows the construction of the printer cables. Notice that some pins in the 36-pin connector are not connected.

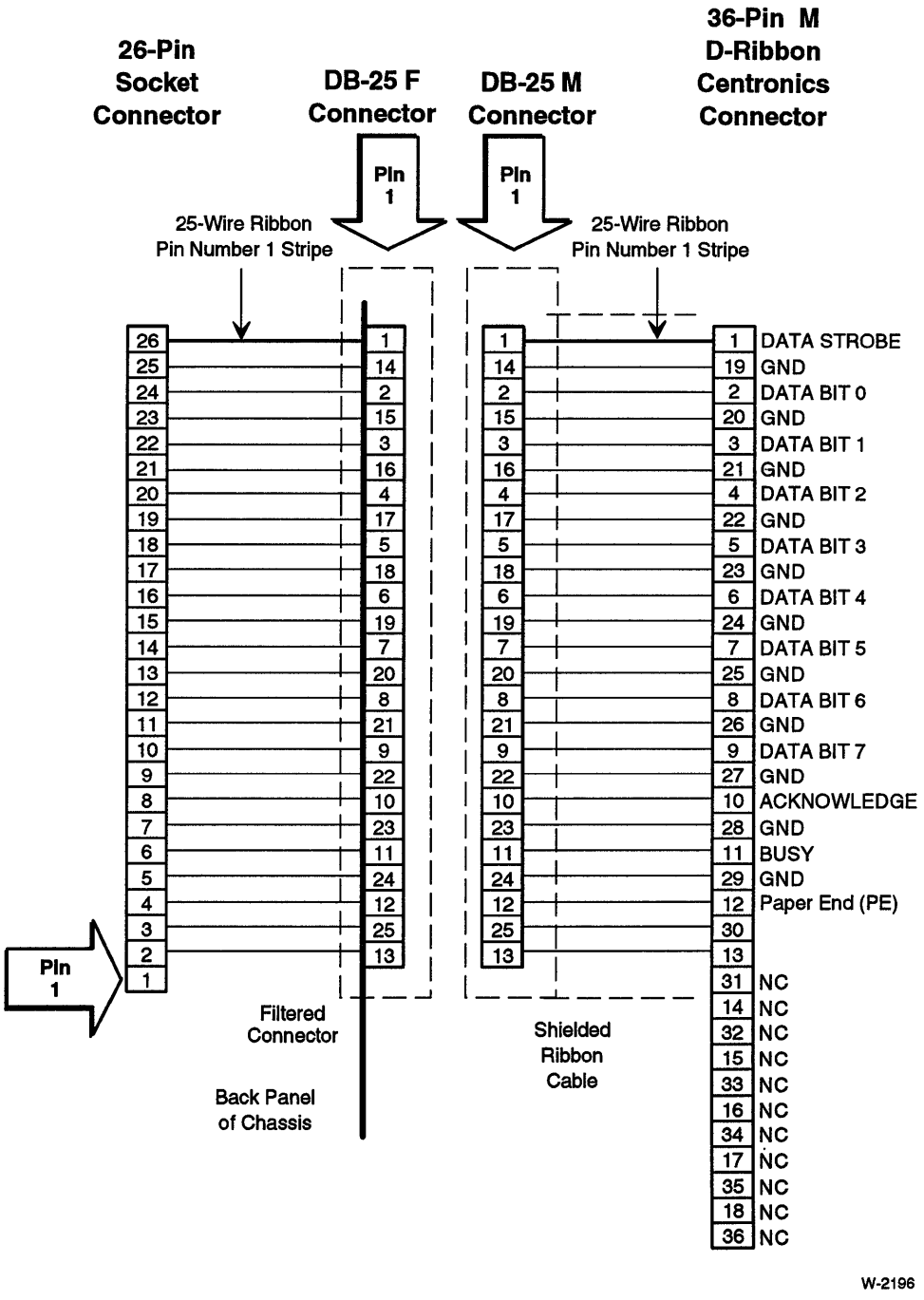


Figure 9-4. Cable From J1 to Back Chassis Panel

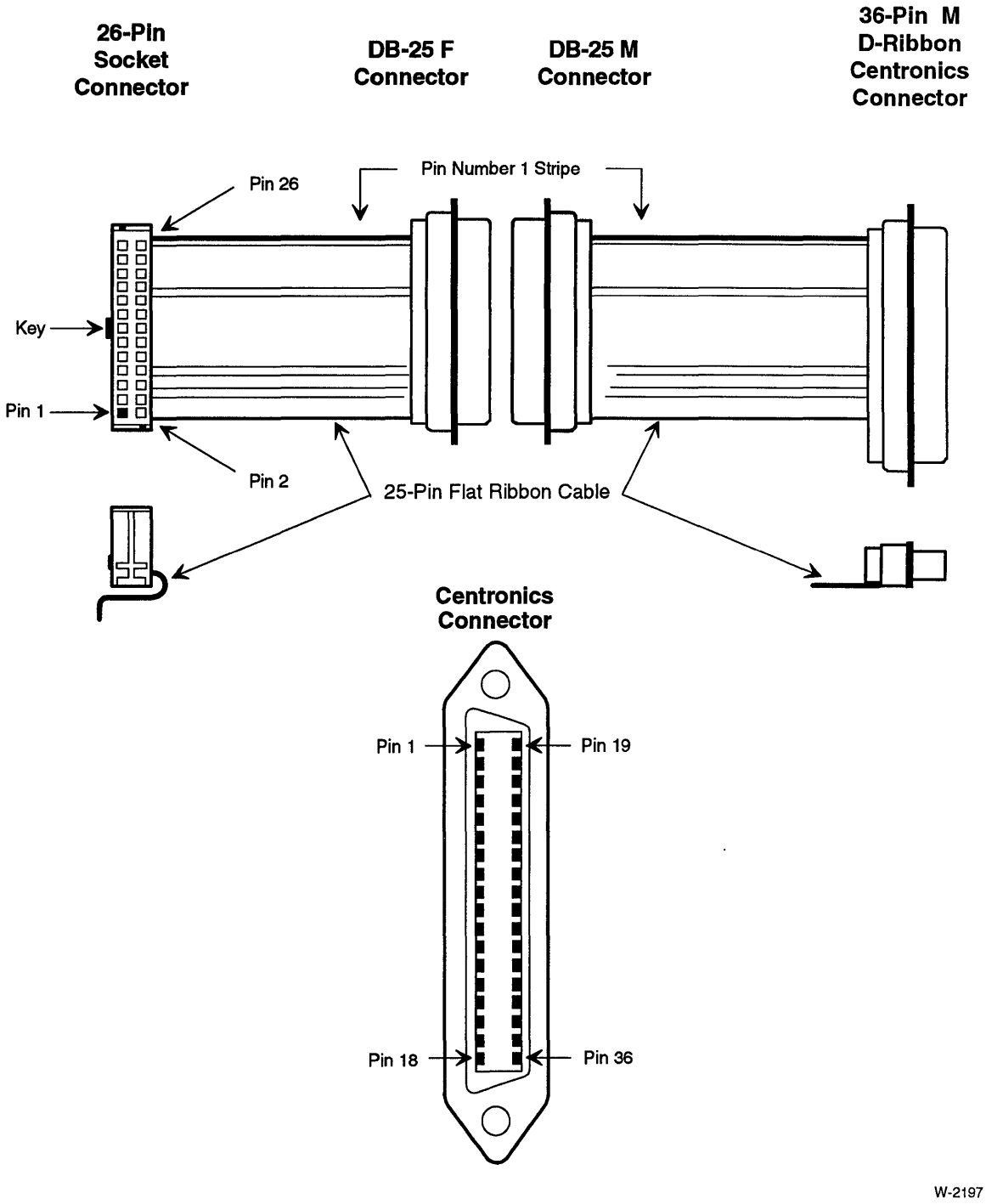


Figure 9-5. Printer Cable Construction

SCSI Interface (Connector J2)

Connector J2 is a 50-pin connector that provides the single-ended SCSI interface for the iSBC 486/12S series boards. This connector is not installed on the non-SCSI iSBC 486/12 board.

Table 9-9 lists the SCSI pin assignments for connector J2. Figure 9-6 shows the SCSI cable. Figure 9-7 shows the SCSI cable construction.

Table 9-9. Connector J2 Pin Assignments

Pin	Function	Pin	Function
1	Ground	27	Ground
2	Data Bit 0	28	Ground
3	Ground	29	Ground
4	Data Bit 1	30	Ground
5	Ground	31	Ground
6	Data Bit 2	32	Attention (ATN*)
7	Ground	33	Ground
8	Data Bit 3	34	Ground
9	Ground	35	Ground
10	Data Bit 4	36	Busy (BSY*)
11	Ground	37	Ground
12	Data Bit 5	38	Acknowledge (ACK*)
13	Ground	39	Ground
14	Data Bit 6	40	Reset (RST*)
15	Ground	41	Ground
16	Data Bit 7	42	Message (MSG*)
17	Ground	43	Ground
18	Parity Bit	44	Select (SEL*)
19	Ground	45	Ground
20	Ground	46	Control/Data (C/D*)
21	Ground	47	Ground
22	Ground	48	Request (REQ*)
23	Ground	49	Ground
24	Ground	50	Input/Output (I/O*)
25	Not Connected		
26	Termination Power (Termpwr)		

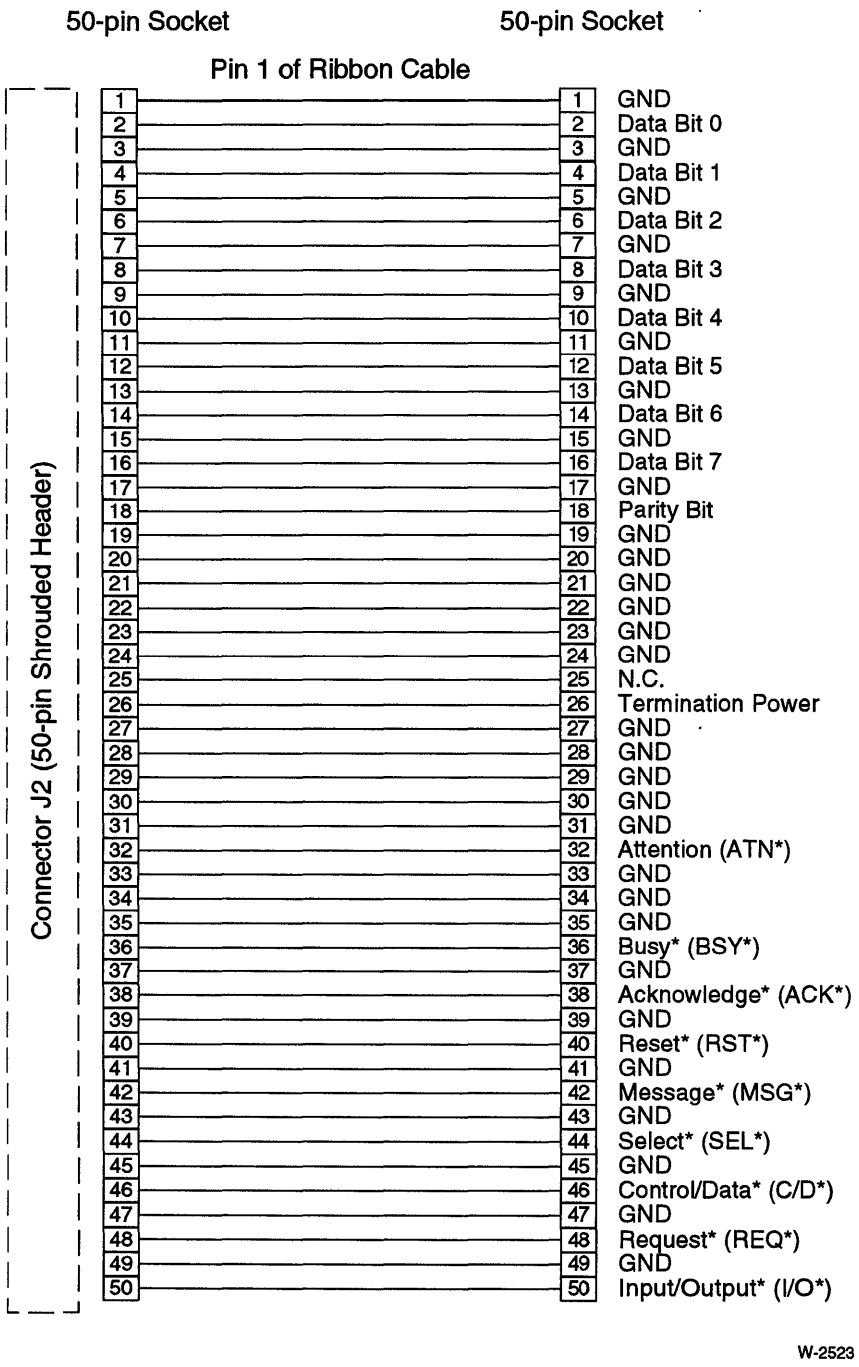


Figure 9-6. SCSI Cable

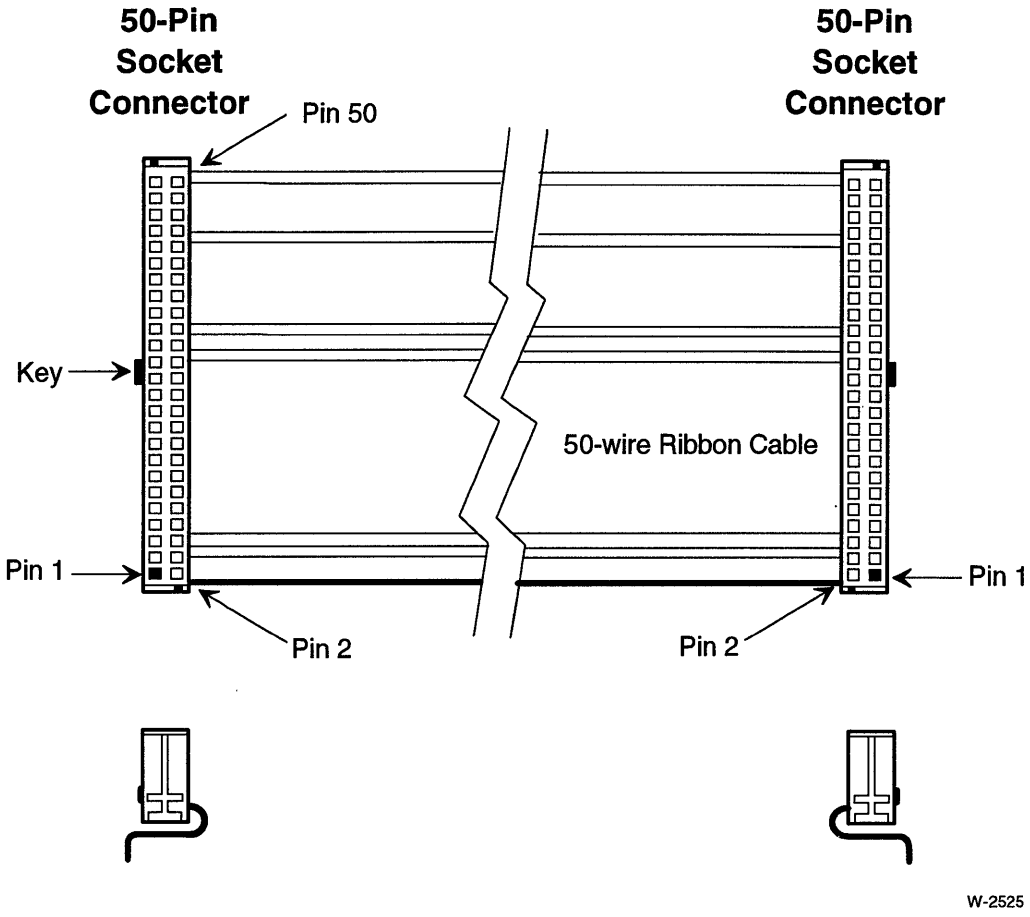


Figure 9-7. SCSI Cable Construction

Front Panel Interface (Connector J3)

Connector J3 is a 14-pin connector that provides the front panel interface for the board. Some of the front panel signals are also available on connector P2.

Table 9-10 lists the connector J3 pin assignments. Figure 9-8 shows the pin numbering sequence for connector J3. Table 9-6 (page 9-13) describes the front panel signals.

Table 9-10. Connector J3 Pin Assignments

Top Row		Bottom Row	
1	PFINT*	2	PFSN*
3	SCZBSY	4	EXINT*
5	GND	6	AUXRST*
7	GND	8	CYCLE
9	Not Connected	10	Not Connected
11	GND	12	+5 V
13	GND	14	+5 V

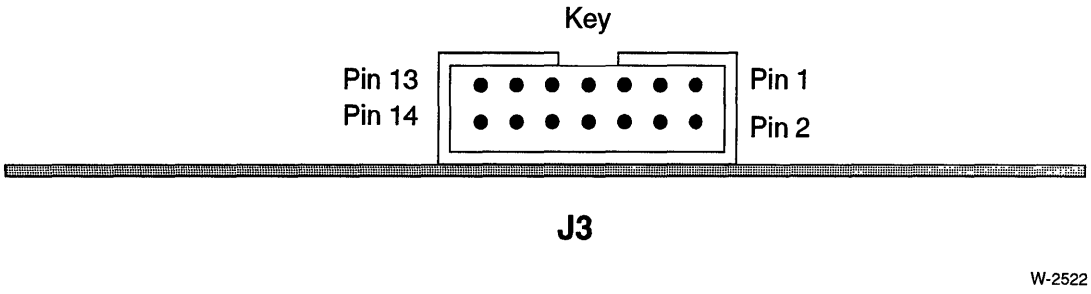


Figure 9-8. Connector J3 Pin Locations

Serial Interfaces (Connectors J4L and J4U)

Connector J4 consists of two stacked 26-pin connectors. Figure 9-9 shows the pin numbering of connector J4.

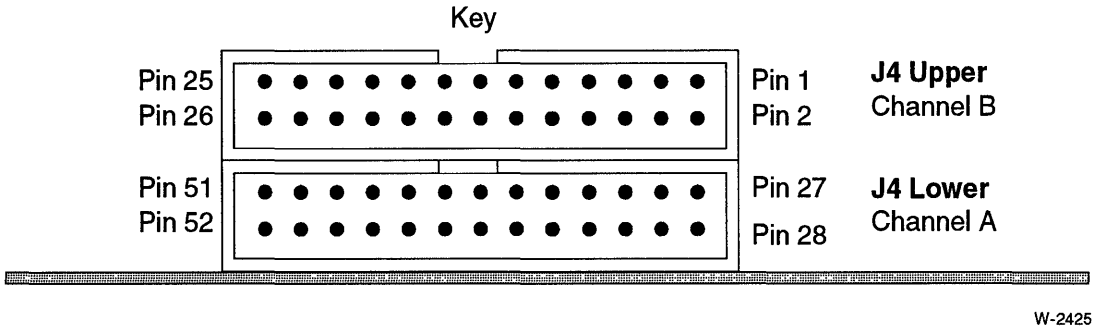


Figure 9-9. Connector J4 Pin Locations

Table 9-11. Serial Connector J4 Cross-Reference

Connector	Configuration	Pin-Out	Cable
J4U	RS-232C DCE	Table 9-12	Figures 9-10, 9-11
J4L	RS-232C DCE	Table 9-12	Figures 9-10, 9-11
J4L	RS-232C DTE	Table 9-12	Figures 9-10, 9-11
J4L	RS-422A/449 DCE	Table 9-13	Figures 9-12, 9-13

Refer to Chapter 7, "I/O Subsystem Configuration," for information on configuring the serial interface.

Table 9-12. Connectors J4L and J4U RS-232C Pin Assignments

Connector Pin Numbers J4U	J4L	RS-232C Pin Number	RS-232C DCE Name	RS-232C DCE Function	RS-232C DTE Name	RS-232C DTE Function
1	27	N.C.				
2	28	13				
3	29	25				
4	30	12				
5	31	24	TxC	Transmit Clock	RxC	Receive Clock
6	32	11				
7	33	23				
8	34	10				
9	35	22				
10	36	9				
11	37	21				
12	38	8				
13	39	20	DTR	Data Terminal Ready	DSR	Data Set Ready
14	40	7	SG	Signal Ground	SG	Signal Ground
15	41	19				
16	42	6	DSR	Data Set Ready	DTR	Data Terminal Ready
17	43	18				
18	44	5	CTS	Clear To Send	RTS	Request To Send
19	45	17	RxC	Receive Clock	TxC	Transmit Clock
20	46	4	RTS	Request To Send	CTS	Clear To Send
21	47	16				
22	48	3	RxD	Receive Data	TxD	Transmit Data
23	49	15				
24	50	2	TxD	Transmit Data	RxD	Receive Data
25	51	14				
26	52	1				

Note:

N.C. means "not connected." Pin numbers without signal names are not used.

Table 9-13. Connector J4L RS-442A/449 Pin Assignments

Connector J4L Pin Number	RS-422A/449 Pin Number	Signal Name[1]	RS-442A/449 Function
27	N.C.		
28	13		
29	31		
30	12	TR (A)	Terminal Ready
31	30	TR (B)	Terminal Ready
32	11	DM (A)	Data Mode
33	29	DM (B)	Data Mode
34	10		
35	28		
36	9	CS (A)	Clear To Send
37	27	CS (B)	Clear To Send
38	8	RT (A)	Receive Timing
39	26	RT (B)	Receive Timing
40	7	RS (A)	Request To Send
41	25	RS (B)	Request To Send
42	6	RD (A)	Receive Data
43	24	RD (B)	Receive Data
44	5		
45	23		
46	4	SD (A)	Send Data
47	22	SD (B)	Send Data
48	3	TT (A)	Terminal Timing
49	21	TT (B)	Terminal Timing
50	2		
51	20	RC	Receive Common
52	1	SHIELD	Shield

Notes:

N.C. means "not connected." Pin numbers without signal names are not used.
[1] (A) is the negative signal; (B) is the positive signal.

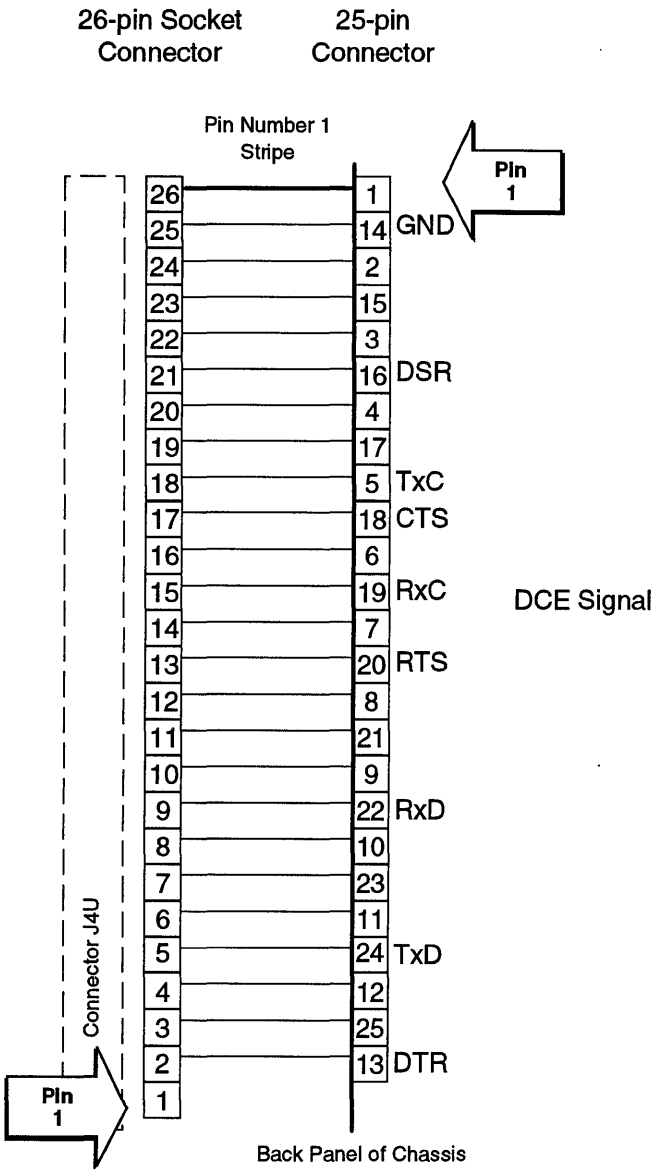


Figure 9-10. Connector J4U and J4L RS-232C Cable to Back Panel

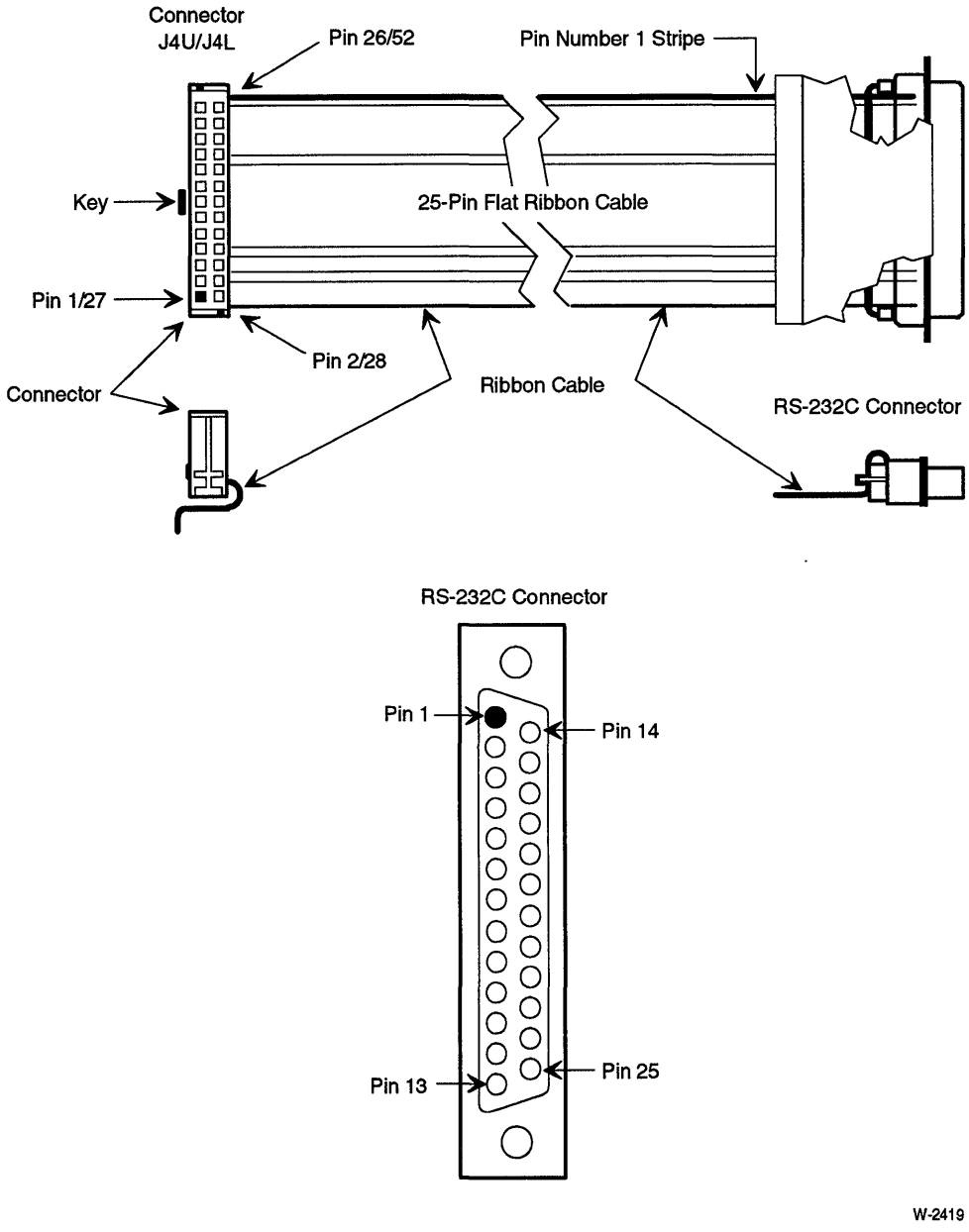


Figure 9-11. RS-232C Cable Construction

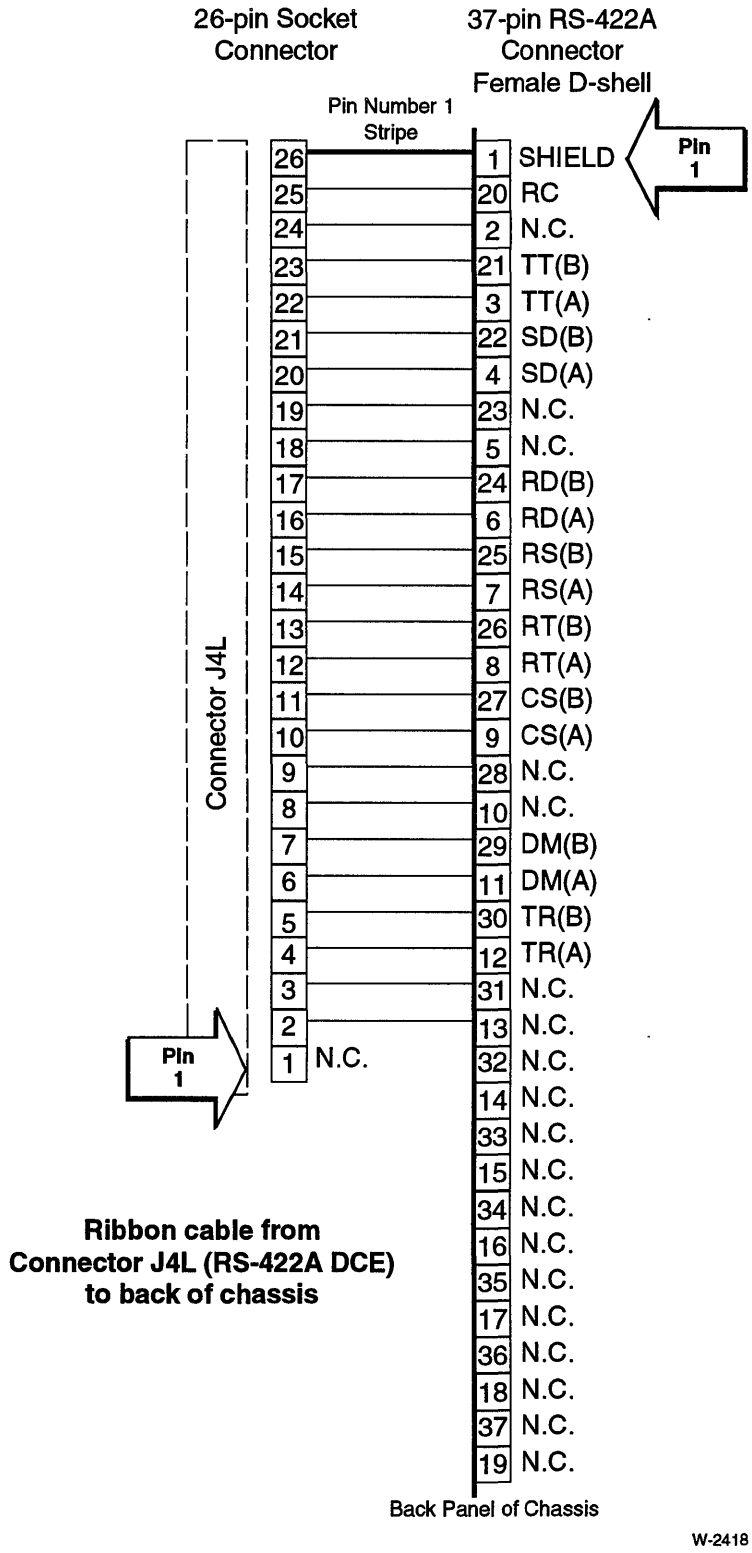


Figure 9-12. Connector J4L RS-442A/449 Cable to Back Chassis Panel

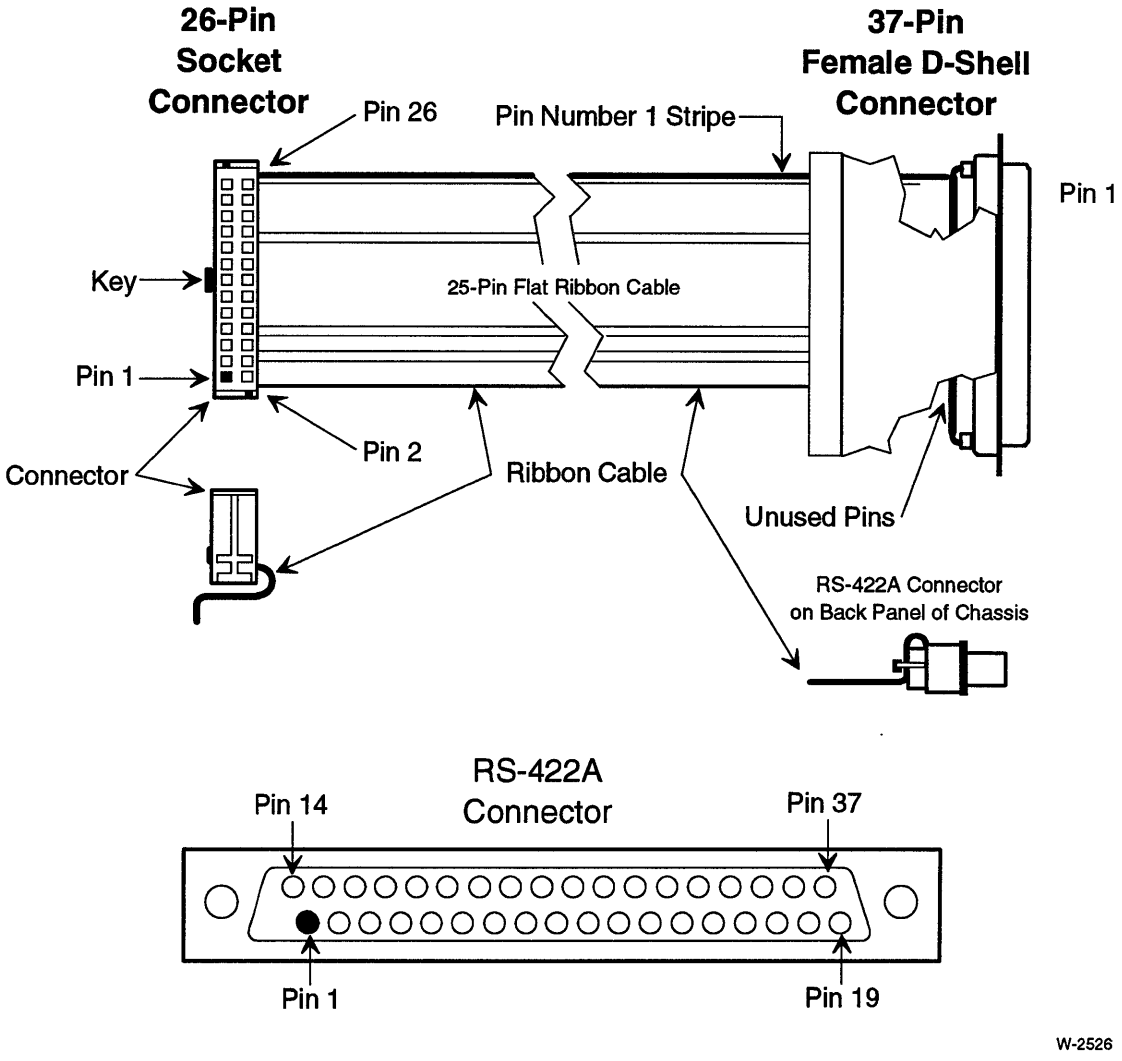


Figure 9-13. RS-442A/449 Cable Construction

SBX Interfaces (Connectors J5 and J6)

Connectors J5 and J6 are the SBX interfaces. The SBX interface is defined in the IEEE 959 Specification. Table 9-14 describes the SBX signals. Table 9-15 (page 9-32) lists the pin assignments for connectors J5 and J6. Figure 9-14 (page 9-33) shows the location and pin numbering of J5 and J6.

Refer to Chapter 7, "I/O Subsystem Configuration," for configuration details and I/O addresses. Refer to Chapter 8, "CPU Subsystem Configuration," for information on DMA and interrupt configuration of the SBX interfaces. Chapter 3, "Installation," describes how to install your SBX modules.

Table 9-14. SBX Interface Signals (Connectors J5 and J6)

iSBC 486/12 Signal Name (SBX Name)	Pin Number (J5 or J6)	Description
I_SBXRD* (IORD*)	15	*I/O Read.* An active low signal which indicates that the address lines are valid and that the SBX module should place the data on the data lines. This signal is conditioned by the chip select signals.
I_SBXWR* (IOWRT*)	13	*I/O Write.* An active low signal which indicates that the address lines and data lines are valid and that the SBX module should accept the data and write to the specified location. This signal is conditioned by the chip select signals.
I_IOA1 (MA0) I_IOA2 (MA1) I_IOA3 (MA2)	11 9 7	*Module Address.* These three positive true signals, in conjunction with the chip select signals, establish the I/O port address being addresses.
M_CLK9_8 (MCLK)	6	*Module Clock.* This signal is a 9.83 MHz clock signal from the baseboard to the module.
I_SBX*x*CS0* (MCS0) I_SBX*x*CS1* (MCS1)	22 20	*Module Chip Select.* The chip selects are active low signals that condition the I/O command signals. Refer to Chapter 7 for SBX I/O port addresses.
I_IOD0 (MD0) I_IOD1 (MD1) I_IOD2 (MD2) I_IOD3 (MD3) I_IOD4 (MD4) I_IOD5 (MD5) I_IOD6 (MD6) I_IOD7 (MD7) I_IOD8 (MD8) I_IOD9 (MD9) I_IOD10 (MDA) I_IOD11 (MDB) I_IOD12 (MDC) I_IOD13 (MDD) I_IOD14 (MDE) I_IOD15 (MDF)	33 31 29 27 25 23 21 19 43 44 41 42 39 40 37 38	*Module Data.* 8 or 16 active-high data lines. Connectors J5 and J6 support both 8-bit and 16-bit modules. I_IOD15 (MDF) is the most significant bit.

Table 9-14. SBX Interface Signals (Connectors J5 and J6 Continued)

iSBC 486/12 Signal Name (SBX Name)	Pin Number (J5 or J6)	Description
I_SBX*x*DACK* (MDACK*)	32	*Module DMA Acknowledge.* This is an active-low signal from the 82258 ADMA that acknowledges the DMA request.
I_SBX*x*DREQ (MDRQT)	34	*Module DMA Request.* This is an active-high signal from the module to the 82258 ADMA on the baseboard. This signal is used to request a DMA cycle.
I_SBX*x*IR0 (MINTR0) I_SBX*x*IR1 (MINTR1)	14 12	*Module Interrupt Request.* These two active high signals are used to make interrupt requests to the baseboard. Chapter 8 discusses how to configure the SBX interrupts.
I_SBXPRSNT* (MPST)	8	*Module Present.* This signal indicates when a module is present. This signal is read by Board ID Register (described in Chapter 8).
I_MWAIT*x** (MWAIT)	16	*Module Wait.* This active-low signal will put the baseboard into a wait state providing additional time for the SBX module to perform the requested operation.
I_SBX*x*OPT0 (OPT0) I_SBX*x*OPT1 (OPT0)	30 28	*Option Lines.* These two signals are connected to stake pins on the iSBC 486/12 board. For an optional DMA channel, OPT0 and OPT1 can be used for DMA request and DMA acknowledge signals for ADMA channel 2. Chapter 8 describes how to configure the ADMA.
I_RESET (RESET)	5	*Reset.* A reset signal to the module. This signal is activated by the power-up or auxiliary reset signal on the baseboard.
I_SBX*x*TDMA (TDMA*)	26	*Terminate DMA.* This active-high signal is used by the SBX module to terminate DMA activity. The TDMA signal is available for ADMA channels 1 and 3. Chapter 8 describes how to configure the ADMA.

Note
The letter *'x'* in signal names is replaced with 1 for SBX1, 2 for SBX2

Table 9-15. Connector J5 and J6 Pin Assignments

Row 1		Row 2	
1	+12 V	2	-12 V
3	GND	4	+5 V
5	I_RESET	6	M_CLK9_8
7	I_IOA3	8	I_SBXPRSNT*
9	I_IOA2	10	Not Connected
11	I_IOA1	12	I_SBX*x*IR1
13	I_SBXWR*	14	I_SBX*x*IR0
15	I_SBXRD*	16	I_MWAIT*x**
17	GND	18	+5 V
19	I_IOD7	20	I_SBX*x*CS1*
21	I_IOD6	22	I_SBX*x*CS0*
23	I_IOD5	24	Not Connected
25	I_IOD4	26	I_SBX*x*TDMA
27	I_IOD3	28	I_SBX*x*OPT1
29	I_IOD2	30	I_SBX*x*OPT0
31	I_IOD1	32	I_SBX*x*DACK*
33	I_IOD0	34	I_SBX*x*DREQ
35	GND	36	+5 V
37	I_IOD14	38	I_IOD15
39	I_IOD12	40	I_IOD13
41	I_IOD10	42	I_IOD11
43	I_IOD8	44	I_IOD9

Note
The letter *'x'* in signal names is replaced with 1 for SBX1, 2 for SBX2

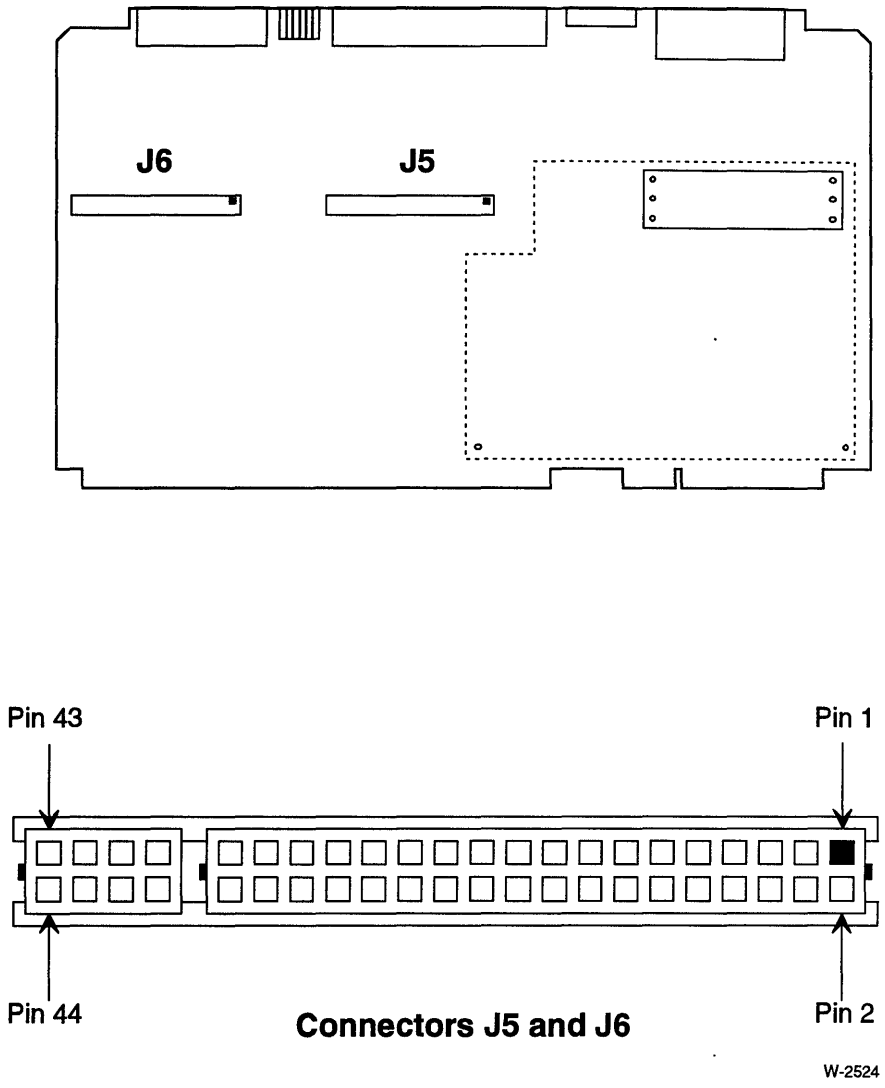


Figure 9-14. SBX Pin Numbering

***

Service Information 10

Chapter Contents

Services Offered 10-2
System Engineering Consulting Services 10-2
Customer Training 10-2
Hardware On-Site Support 10-2
Mail-In Service 10-2
How to Get Service 10-3
Call This Number for Service 10-3
Package Your Board for Shipment 10-4
Send Your Board to This Address 10-4

Services Offered

Intel provides a wide variety of worldwide support, including consulting, customer training, 24-hour on-site support, and mail-in repair services. Intel will customize a program to meet your individual needs.

System Engineering Consulting Services

Intel's system engineers have the experience and expertise that can save valuable development time. Their knowledge of Intel boards, systems, software and tools can get your product on the market sooner, boost your productivity, and ensure compatibility with future Intel projects. Contact your local Intel sales office for more information on how you can benefit from Intel's consulting services.

Customer Training

Intel provides in-depth training workshops on hardware and software products. Intel routinely conducts workshops on everything from microcontrollers to operating systems at its world-wide training centers. On request, Intel can customize a workshop and present it at your site. Contact your local sales office for more information on how you can benefit from Intel's customer training.

Hardware On-Site Support

An Intel customer engineer will repair your equipment at your site during normal business hours. An extended coverage of up to 24 hours a day, seven days a week is also available. In addition, Intel offers the option of including several non-Intel products under its on-site support. Contact your local sales office or call Intel's customer support for more information about on-site support.

Mail-In Service

Intel's customer support provides the following mail-in services:

- **Direct Return Authorization (DRA)**: Under this service, Intel repairs, tests, and updates the product with all mandatory engineering change orders. The product serial number does not change with this procedure. Normal turn-around time for DRA service is 10 to 14 days.
- **Return Replacement Authorization (RRA)**: Under this service, Intel replaces the defective product. The RRA service is not offered on all products, is subject to availability, and is available only to customers in certain locations. Intel tries to ship 90% of replacement products within 48 hours of receiving a defective product.

How to Get Service

If you need repair service, contact Intel Customer Support for either a Direct Return Authorization (DRA) or a Return Replacement Authorization (RRA). (The phone number is listed below.) Before calling, please have the following information readily available:

- Warranty customers:
  - proof of purchase
  - name and serial number of the product
  - your company name
  - your shipping address
  - contact name and telephone number at your site
- Service contract customers:
  - service contract number
  - name and serial number of the product
  - your company name
  - your shipping number
  - contact name and telephone number at your site
- All other customers requesting product service:
  - purchase order number (or billing information)
  - name and serial number of the product
  - your company name
  - your shipping and billing addresses
  - contact name and telephone number at your site

Call This Number for Service

- In the United States and Canada, call 1-800-INTEL-4-U (1-800-468-3548).
- In all other areas, call your local Intel sales office or distributor. A list of Intel's International Sales Offices is provided at the end of this manual.

Package Your Board for Shipment

1. Write the DRA or RRA number on the packing slip, the purchase order, and other related documents.
2. Remove all modifications you have made to the product.

CAUTION

Remove your custom-programmed devices and re-install the original, factory-supplied devices. Do NOT send your custom devices in with the board; they are not required for board service and their return is not guaranteed.

3. Place boards in antistatic bags and then in appropriate shipping containers.
4. Protect the product with protective padding, such as flow pack or foam.
5. Write the DRA or RRA authorization number on the outside of the box and label the box "FRAGILE."

NOTE

Damage resulting from improper packaging of return items can result in extra repair charges to you.

Send Your Board to This Address

For customers in the U.S.A.: send your board with the paperwork requested by Intel customer support, freight prepaid, to this address:

Intel USA Repair Center
DV2-44
2402 W. Beardsley Road
Phoenix, AZ 85027-3301

For customers outside the U.S.A.: contact your local Intel sales office for shipping instructions.

***

Configuration Worksheets A

Contents

About the Worksheets A-1
Installation Checklist A-2
Real Mode Memory Worksheet A-3
Real Mode Memory Map Worksheet A-4
Protected Mode Memory Map Worksheet A-5
Protected Mode Memory Worksheet A-6
System Memory Worksheet A-7
System Memory Map Worksheet A-8
I/O Configuration Worksheet A-9
CPU Configuration Worksheet A-12
Jumper Configuration Worksheet A-13
Programming Checklist A-14
Hexadecimal Conversion Table A-15

About the Worksheets

The worksheets provided in this appendix are provided to help you configure your iSBC 486/12 series board. Copy the entire appendix and use the worksheets as you read the chapters in this manual that correspond to the worksheet.

Installation Checklist

Step 1. Unpack and inspect board.

__ Unpack board
__ Inspect board for defects

Step 2. Configure jumpers.

__ Complete Configuration and Jumper Worksheets
__ Change jumpers
__ Wire-wrap the stake pins, if required

Step 3. Install EPROMs.

__ Program EPROMs
__ Install EPROMs

Step 4. Install or remove resistor packs and RS-422A/449 line driver.

__ Multibus Address Lines, RP7 __ Install __ Remove
__ RS-422/449, RP4 __ Install __ Remove
__ RS-422/449, RP5 __ Install __ Remove
__ RS-422/449, RP6 __ Install __ Remove
__ SCSI termination resistors __ Install __ Remove
RP1, RP2, RP3
__ SCSI fuse, F1 __ Install __ Remove
__ RS-422/449 Line Driver, U14 __ Install __ Remove

Step 5. Install SBX module

__ Install SBX 1 module _____
__ Install SBX 2 module _____

Step 6. Install second MM3 memory module. (Optional)

__ Install module with higher DRAM capacity on the bottom
__ Install top module

Step 7. Install board in backplane.

__ Check backplane arbitration __ Serial __ Parallel
__ Connect auxiliary signals on P2, if used
__ Turn power off
__ Install board

Step 8. Install cables.

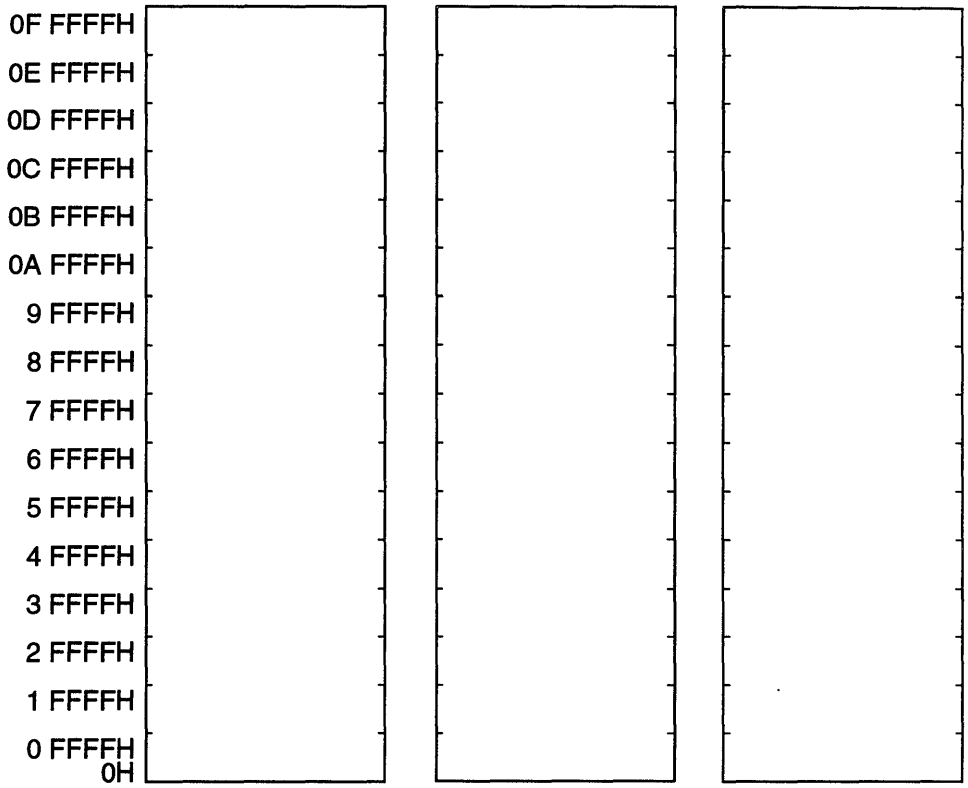
__ Serial cables
__ Parallel printer cable
__ SCSI cable
__ Front panel cable

Step 9. Check SCSI interface.

__ Check SCSI IDs in system (each ID must be unique)
__ Check SCSI termination resistors at both ends of SCSI bus
__ Check for other SCSI devices providing power on SCSI bus

W-2178

Real Mode Memory Map Worksheet



NOTES:

W-2150

Real Mode Memory Configuration Worksheet

Step 1. Configure the Multibus window

__ Starting address = _____ H
__ Ending address = _____ H
__ Size of Multibus window = _____ Kbytes
__ Record the Multibus window on the Real Mode Memory Map
__ Record the Multibus window jumpers on your jumper Worksheet

Step 2. Configure DRAM

__ DRAM ending address = _____ H
__ Size of DRAM = _____ Kbytes
__ Record DRAM ending address on the Real Mode Memory Map
__ Record DRAM ending address jumpers your Jumper Worksheet

Step 3. Configure EPROM

__ Total EPROM size = _____ Kbytes
__ EPROM starting address = _____ H
__ Record the EPROM memory spac on tthe Real Mode Memory Map
__ Record the EPROM jumpers on your Jumper Worksheet

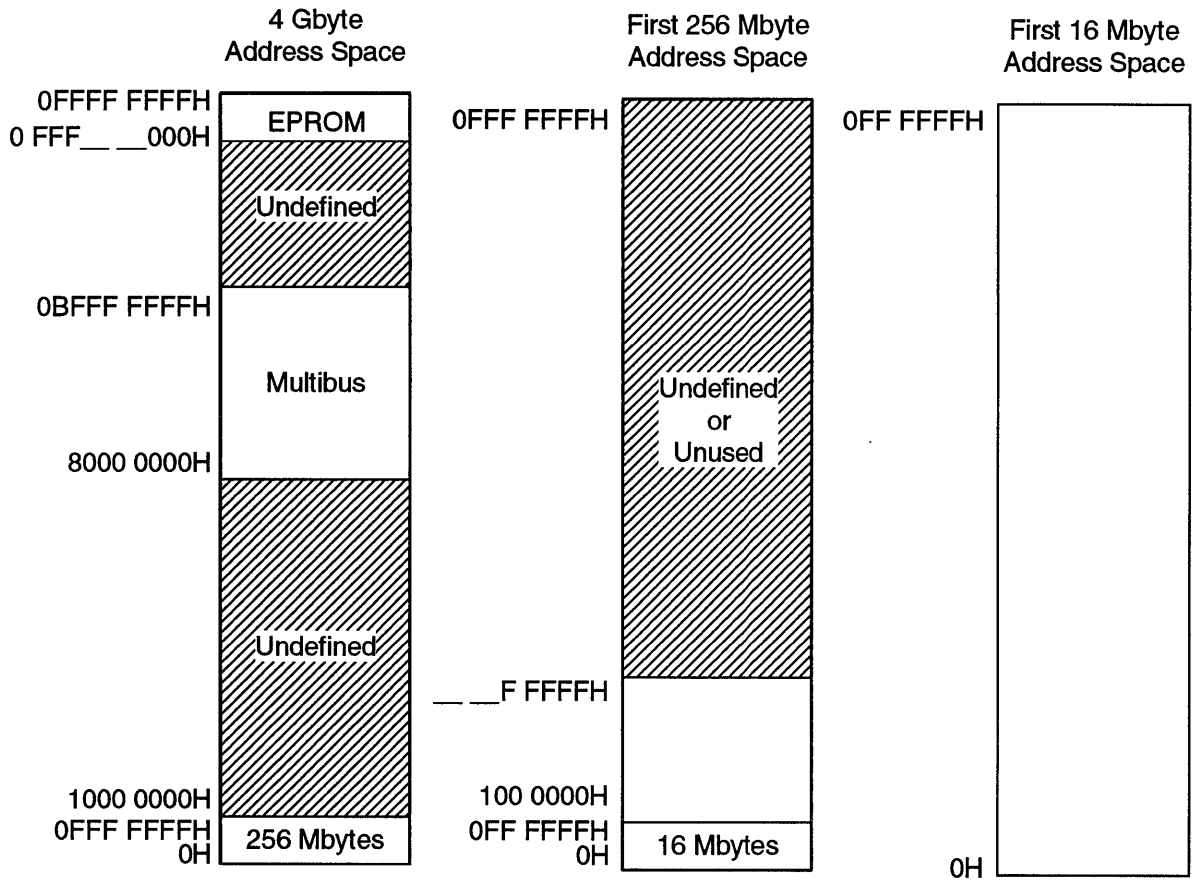
Step 4. Record the Multibus interface address space

__ Size of Multibus interface address space = _____ Kbytes
__ Record the Multibus interface address space on the Real Mode Memory Map

NOTES:

W-2152

Protected Mode Memory Map Worksheet



NOTES:

W-2157

Protected Mode Worksheet

Step 1. Select the DRAM ending address

__ Total size of DRAM = _____ Mbytes _____ H bytes
__ DRAM ending address = _____ H
__ Record DRAM on memory map
__ Record DRAM ending address jumpers on Jumper Worksheet

Step 2. EPROM configuration

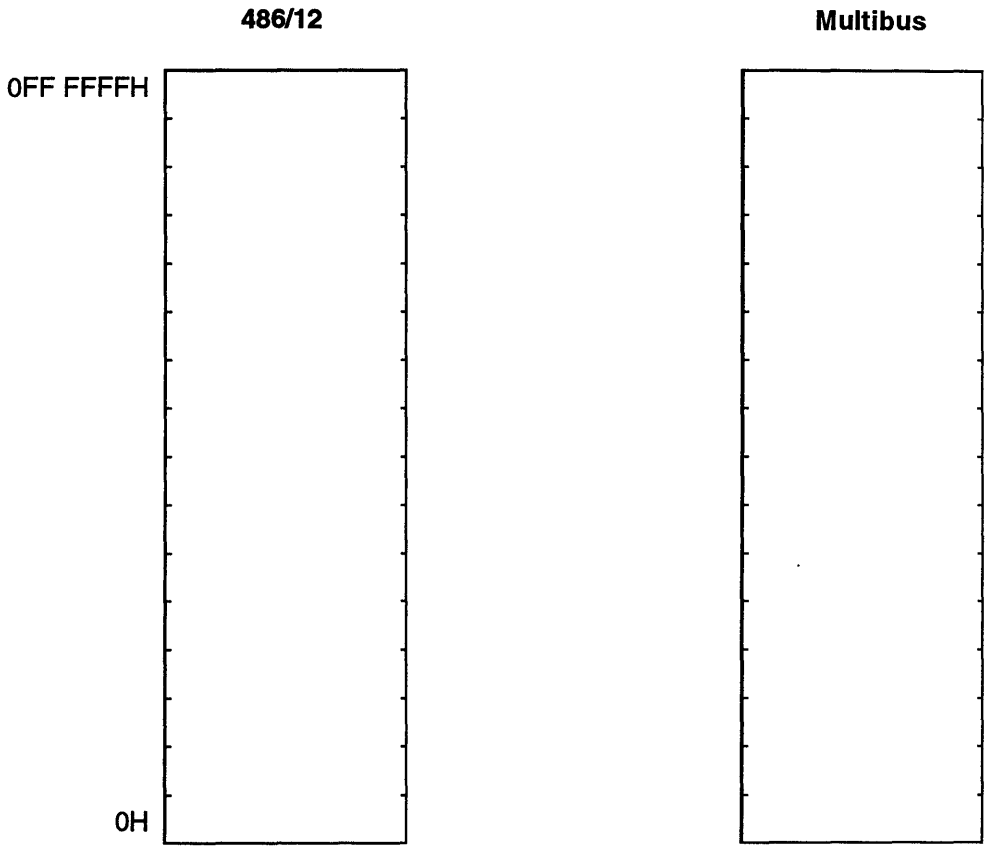
__ EPROM size = _____ Kbytes _____ H bytes
__ EPROM starting address in 4th Gbyte = _____ H
__ EPROM enabled at top of 16th Mbyte ( ) Yes ( ) No
__ EPROM starting address in 16th Mbyte = _____ ( ) Not Enabled
__ Record EPROM address space on memory map worksheet
__ Record EPROM jumpers on your Jumper Worksheet

Step 3. Record Multibus address space

__ Record the Multibus address space on the memory map worksheet

W-2159

System Memory Map Worksheet



NOTES:

W-2166

System Memory Worksheet

Step 1. Dual-port starting and ending addresses

__ Dual port starting address = _____ H
__ Dual port ending address = _____ H
__ Record dual port starting and ending addresses on the System Memory Map Worksheet
__ Record starting and ending address jumpers on the Jumper Worksheet

Step 2. Dual-port alias

__ Dual port alias = _____ H
__ Record dual port alias on the System Memory Map Worksheet
__ Record dual port alias jumpers on your Jumper Worksheet

W-2168

I/O Configuration Worksheet

Step 1. Select the 82C54 PIT, Counter 0 input clock

__ 1.229 MHz (default)
__ 4.0 MHz

Step 2. Select the 82C54 PIT, Counter 1 input clock

__ 1.229 MHz (default)
__ Output from counter 0

Step 3. Select the 82C54 PIT, Counter 2 input clock

__ 1.229 MHz (default)
__ 4.0 MHz

Step 4. Select the 82C54 PIT, Counter 0 output

__ Output to interrupt controller (default)
__ Cascade output from counter 0 to counter 1 clock input

Step 5. Select the 82C54 PIT, Counter 1 output

__ Output to 8274 Serial Channel B (default)
__ Output to interrupt controller

Step 6. Select the 82C54 PIT, Counter 2 output

__ Output to 8274 Serial Channel A (default)
__ No connection

Step 7. Configure the User-Defined Code Jumpers

__ USER DEFINED CODE (3 bits) = __ __ __ (default = 001)

Step 8. Configure Direction of PPI Port A

__ Output only (default)
__ Input only
__ Direction controlled by LPTBUSY bit

Step 9. Configure Multibus BCLK* and CCLK*

__ i486/12S board drives BCLK* __ YES (default) __ NO
__ i486/12S board drives CCLK* __ YES (default) __ NO

W-2174

I/O Configuration Worksheet (Continued)

Step 10. Configure Multibus Initialize Signal (INIT*)

__ iSBC 486/12 drives and monitors INIT* signal (default)
__ iSBC 486/12 monitors INIT* signal

Step 11. Configure Software Reset

__ Software Reset signal: Reset entire board and system)
__ Software Reset signal: Reset CPU only (default

Step 12. Configure Multibus Interrupt Outputs

__ PPI Port C5 output (any Multibus interrupt from 0 to 7): ___ (N/C is default)
__ PPI Port C6 output (any Multibus interrupt from 0 to 7): ___ (N/C is default)
__ DRAM Parity Error (PER*) output (any Multibus interrupt from 0 to 7): ___ (N/C is default)

Step 13. Configure Multibus Arbitration Priority

__ Parallel Priority Backplane
__ Serial Priority Backplane

Step 14. Configure Multibus Arbitration Mode

__ Bus arbiter releases bus at the end of each Multibus cycle. (default)
__ Bus arbiter retains Multibus until any other agent requests the bus.
__ Bus arbiter retains Multibus until an agent with a higher priority requests the bus.

Step 15. Configure Multibus I/O Buffered Write

__ Multibus I/O Buffered Writes: __ Enabled (default) __ Disabled

Step 16. SBX Address Swap

__ J5 at I/O Address range 080H-09FH (default)
J6 at I/O Address range 0A0H-0BFH
__ Address Swap Enabled:
J6 at I/O Address range 080H-09FH
J 5 at I/O Address range 0A0H-0BFH

W-2175

I/O Configuration Worksheet (Continued)

Step 17. 8274 Serial Clocks

__ Channel A Transmit Clock: __ PIT Counter 2 (default) __ External
__ Channel A Receive Clock: __ Ch A Transmit Clock __ External
__ Channel B Transmit Clock: __ PIT Counter 1 (default) __ External
__ Channel B Receive Clock: __ Ch B Transmit Clock __ External

Step 18. Configure 8274 Channel A

__ RS-232 DCE (default)
__ RS-232 DTE
__ RS-422/449 DCE

Step 19. Configure 8274 Channel A for Multidrop Operation

__ RS-422/449 Multidrop: __ Enabled __ Disabled (default)
__ Termination Resistors: RP6= _____kΩ RP7= _____kΩ RP9= _____kΩ
(Refer to Appendix B.)

Step 20. Select the Multibus Interrupt Sources

__ Multibus Interrupt 0 = _____
__ Multibus Interrupt 1 = _____
__ Multibus Interrupt 2 = _____
__ Multibus Interrupt 3 = _____
__ Multibus Interrupt 4 = _____
__ Multibus Interrupt 5 = _____
__ Multibus Interrupt 6 = _____
__ Multibus Interrupt 7 = _____

Step 21. Select the SCSI ID

__ SCSI ID = ___ (7H default)

Step 22. Select SCSI Termination Resistors and Power

__ SCSI Termination Resistors: __ In (default) __ Out
__ Power to SCSI bus from i486/12S: __ Yes (default) __ No

W-2176

CPU Configuration Worksheet

Step 1. Configure 82258 ADMA Channel 2

__ none (default)
__ SBX 1 (J5)
__ SBX 2 (J6)

Step 2. Configure Multibus Timeout

__ Multibus Timeout Circuit returns READY signal to CPU if Multibus operation is not complete within 8 ms (default)
__ Disabled

Step 3. Select the Master PIC interrupt sources

__ Master PIC IR0 = _____
__ Master PIC IR1 = _____
__ Master PIC IR2 = _____
__ Master PIC IR3 = _____
__ Master PIC IR4 = _____
__ Master PIC IR5 = _____
__ Master PIC IR6 = _____
__ Master PIC IR7 = _____

Step 4. Select the Slave PIC interrupt sources

__ Slave PIC IR0 = _____
__ Slave PIC IR1 = _____
__ Slave PIC IR2 = _____
__ Slave PIC IR3 = _____
__ Slave PIC IR4 = _____
__ Slave PIC IR5 = _____
__ Slave PIC IR6 = _____
__ Slave PIC IR7 = _____

Step 5. Select the Non-Maskable Interrupt Sources

__ i486 CPU NMI = _____ (OR) _____
__ default

W-2177

Jumper Configuration Worksheet

Jumper Matrix A

E001 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E010
E011 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E020
E021 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E030
E031 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E040
E041 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E050
E051 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E060

Jumper Matrix B

E061 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E070
E071 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E080
E081 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E090

Jumper Matrix C

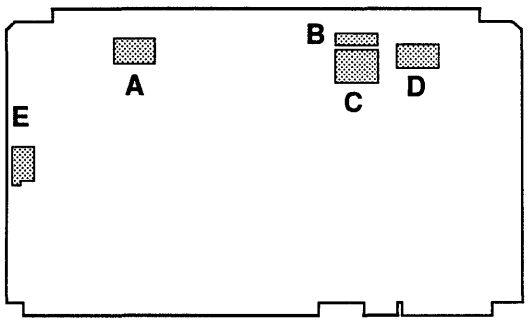
E091 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E100
E101 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E110
E111 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E120
E121 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E130
E131 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E140
E141 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E150
E151 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E160
E161 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E170

Jumper Matrix D

E171 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E180
E181 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E190
E191 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E200
E201 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E210
E211 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E220
E221 ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ E230

Jumper Matrix E

E231 ☐ ☐ ☐ ☐ ☐ E235
E236 ☐ ☐ ☐ ☐ ☐ E240
E241 ☐ ☐ ☐ ☐ ☐ E245
E246 ☐ ☐ ☐ ☐ ☐ E250
E251 ☐ ☐ ☐ ☐ ☐ E255
E256 ☐ ☐ ☐ ☐ ☐ E260
E261 ☐ ☐ ☐ ☐ ☐ E265
E266 ☐ ☐ ☐ ☐ ☐ E270
E271 ☐ ☐ E272



W-2404

Programming Checklist

__ Interrupts
__ 82C59 PIC
__ 87033 SPC
__ 8274 MPSC
__ 82C55A PPI
__ 82C54 PIT
__ DAG
__ 82258 ADMA
__ Board Registers

W-2440

Hexadecimal Conversion Table

0H	0		
0 FFFFH	64K	0F FFFFH	1M
1 FFFFH	128K	1F FFFFH	2M
2 FFFFH	192K	2F FFFFH	3M
3 FFFFH	256K	3F FFFFH	4M
4 FFFFH	320K	4F FFFFH	5M
5 FFFFH	384K	5F FFFFH	6M
6 FFFFH	448K	6F FFFFH	7M
7 FFFFH	512K	7F FFFFH	8M
8 FFFFH	576K	8F FFFFH	9M
9 FFFFH	640K	9F FFFFH	10M
0A FFFFH	704K	0AF FFFFH	11M
0B FFFFH	768K	0BF FFFFH	12M
0C FFFFH	832K	0CF FFFFH	13M
0D FFFFH	896K	0DF FFFFH	14M
0E FFFFH	960K	0EF FFFFH	15M
0F FFFFH	1024K	0FF FFFFH	16M
3FF FFFFH	64M		
0FFF FFFFH	256M	K = 1024	
3FFF FFFFH	1G	M = 1024K	
7FFF FFFFH	2G	G = 1024M	
0 BFFF FFFFH	3G		
0 FFFF FFFFH	4G		
20-bit address	1M		
24-bit address	16M		
32-bit address	4G		

***

Jumper Information B

Default Jumper Settings

This appendix provides information on the default components and stake pin jumpers for the iSBC 486/12 series boards. This appendix includes:

- a diagram showing the default jumpers for:
  - the iSBC 486/12 series with 2 Mbytes of DRAM, page B-2
  - the iSBC 486/12 series with 4 Mbytes of DRAM default jumpers, page B-3
  - the iSBC 486/12 series with 8 Mbytes of DRAM, page B-5
  - the iSBC 486/12 series with 16 Mbytes of DRAM, page B-5
  - the iSBC 486/12 series with 32 Mbytes of DRAM, page B-5
- stake pins listed in numerical order, page B-6
- stake pins listed in alphabetical order by description, page B-12
- stake pin matrix diagram with names, page B-18
- a list of the default jumpers, page B-22
- a list of the default resistor packs and components, page B-24

NOTE

To find the jumper configuration information for a particular stake pin, refer to the stake pin jumpers listed in the Index.

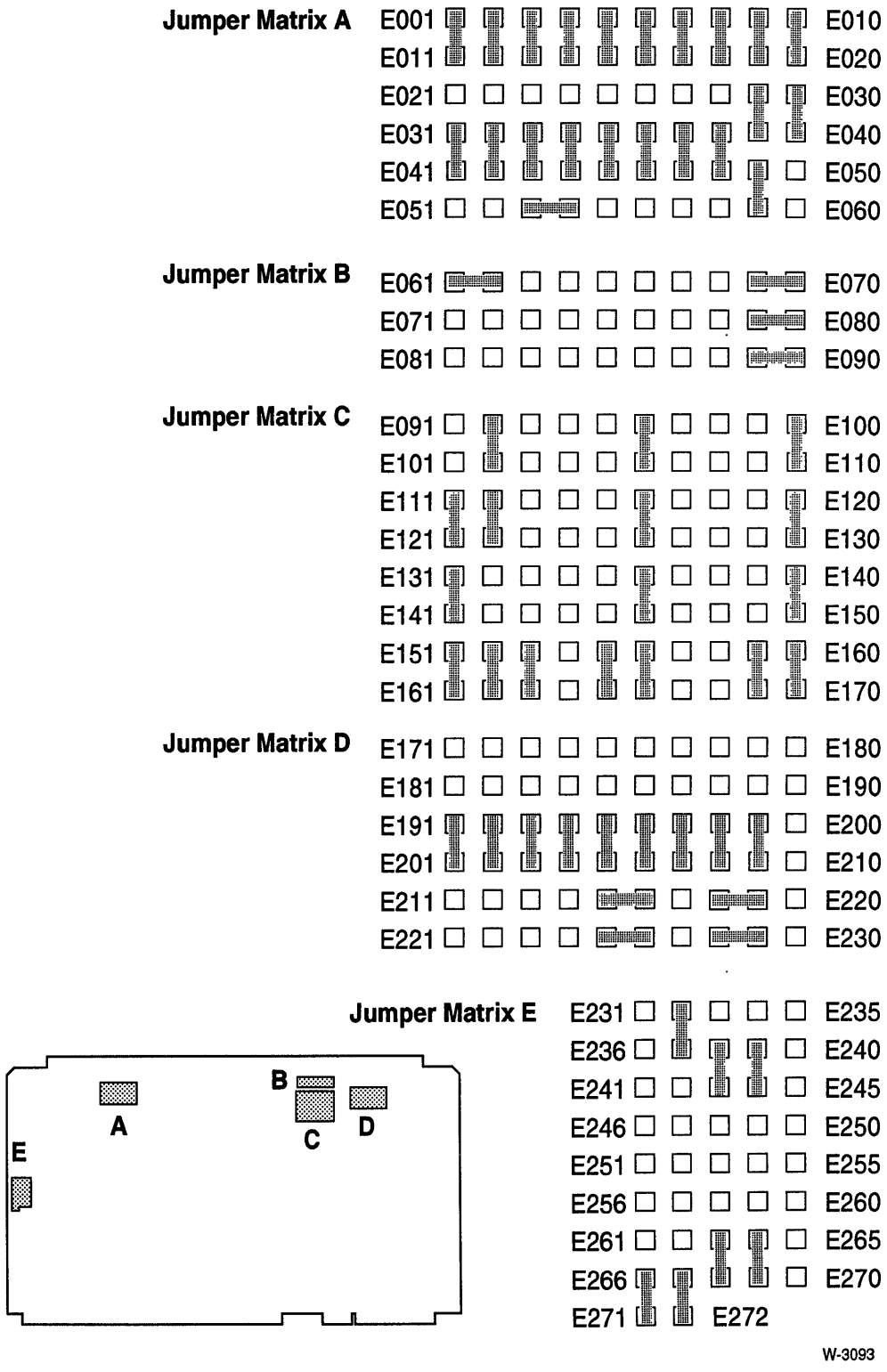


Figure B-1. Default Jumpers (iSBC® 486/12 Series with 2 Mbytes of DRAM)

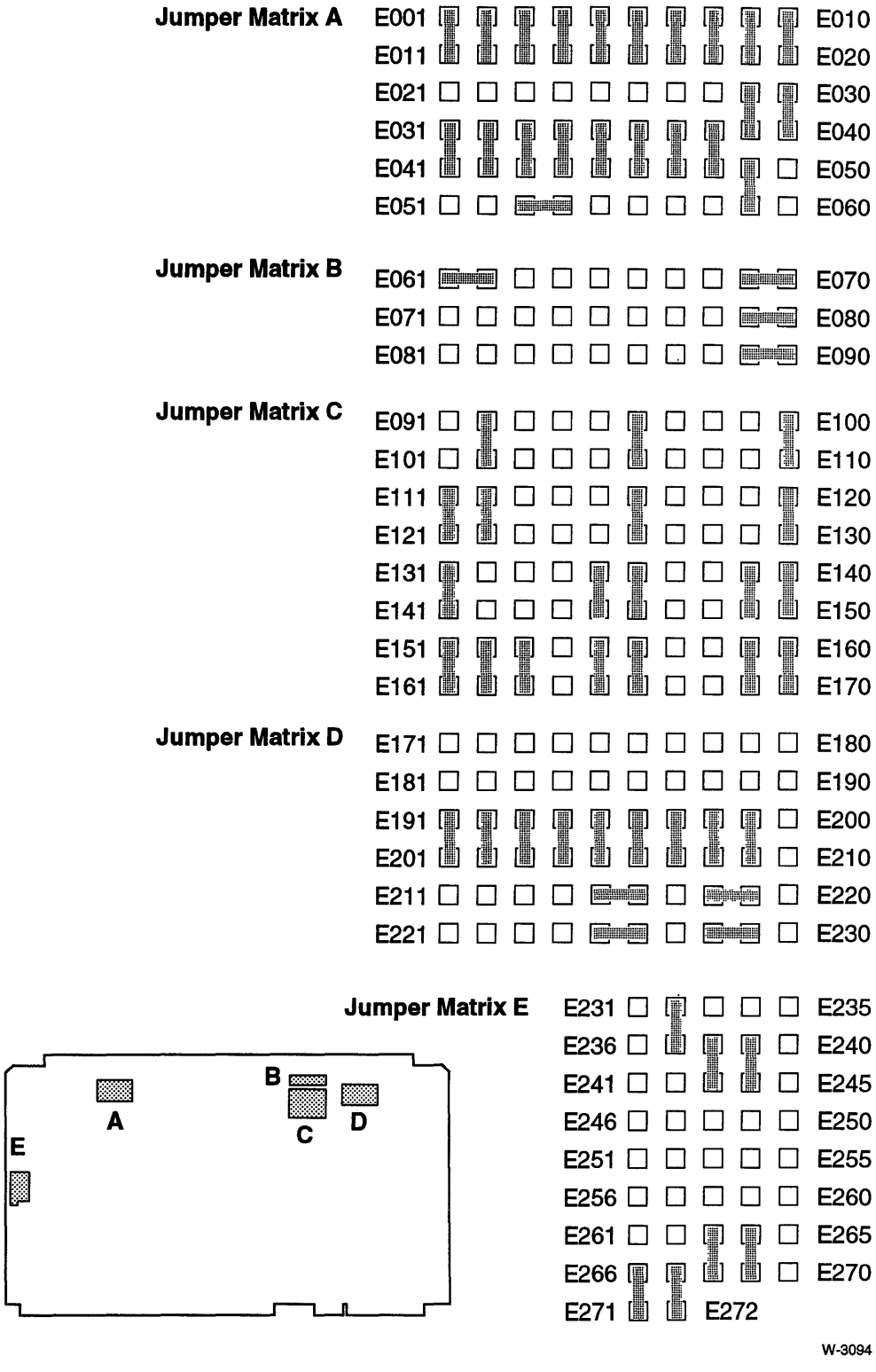


Figure B-2. Default Jumpers (iSBC® 486/12 Series with 4 Mbytes of DRAM)

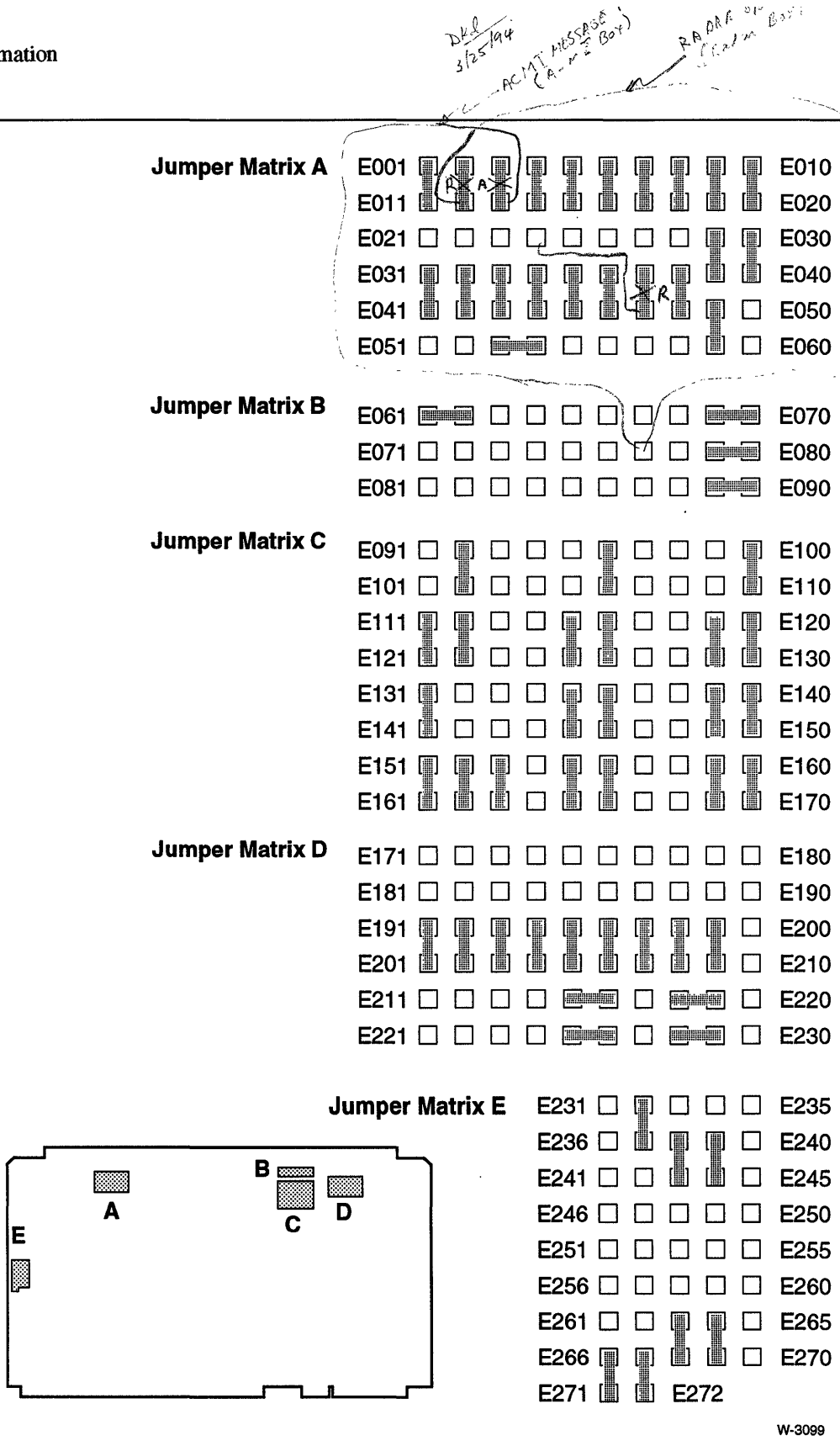


Figure B-3. Default Jumpers (iSBC® 486/12 Series with 8 Mbytes of DRAM)

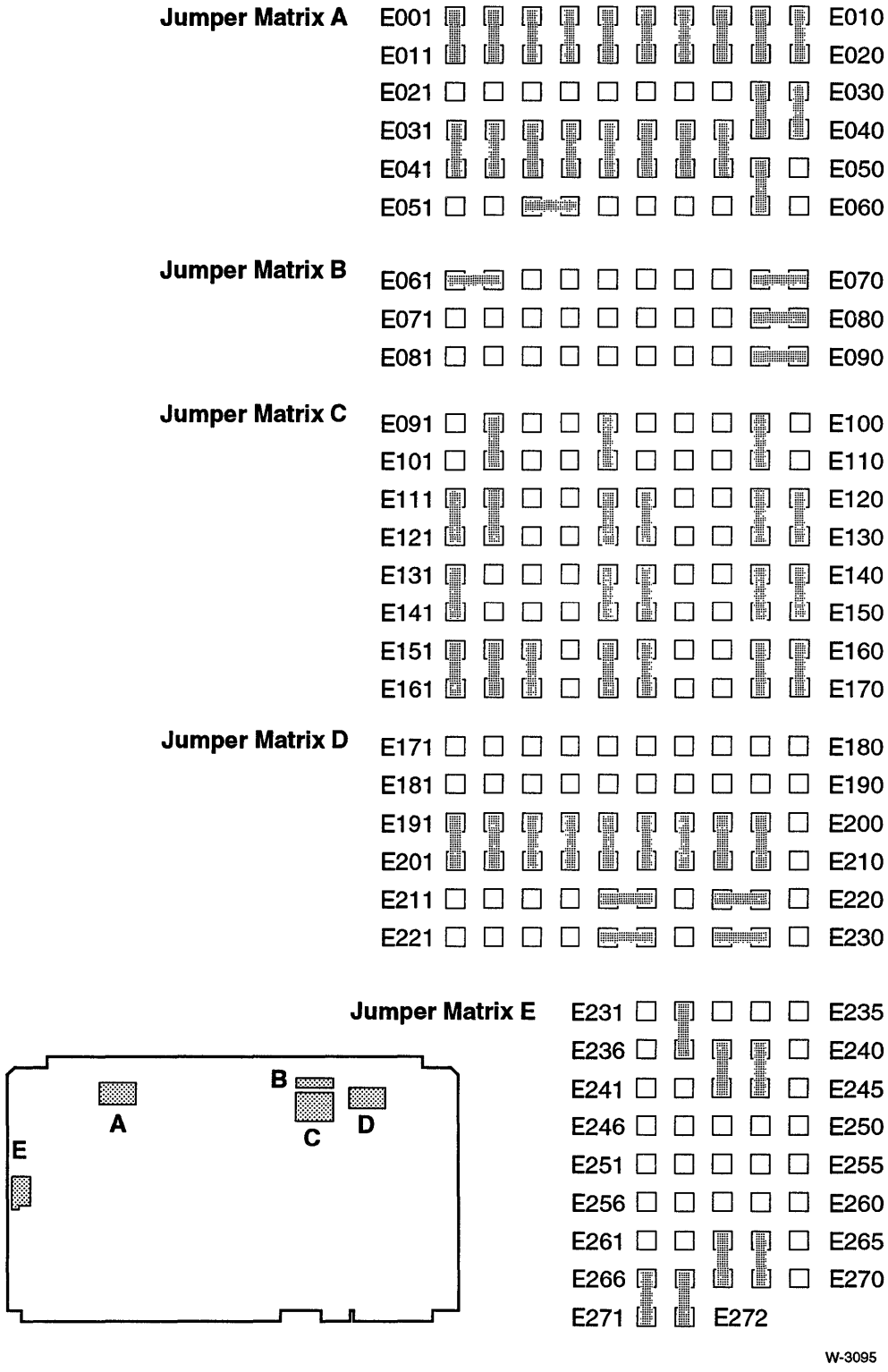


Figure B-4. Default Jumpers (iSBC® 486/12 Series with 16 or 32 Mbytes of DRAM)

Table B-1. Stake Pin and Jumper Listing (Numerical Order)

Stake Pin	Mnemonic	Description
E001	IOCNT0	PIT counter 0 output
E002	MBINT1	Multibus MBINT0 interrupt
E003	SCZINT	SCSI interrupt
E004	MBINT3	Multibus MBINT3 interrupt
E005	MBINT4	Multibus MBINT4 interrupt
E006	MBINT5	Multibus MBINT5 interrupt
E007	SERINT	serial interrupt (MPSC)
E008	SPICINT	slave PIC INT output
E009	INTRI	Intel486 CPU INTR input
E010	TOUT	Multibus time-out (input)
E011	MIR0	master PIC interrupt request level 0
E012	MIR1	master PIC interrupt request level 1
E013	MIR2	master PIC interrupt request level 2
E014	MIR3	master PIC interrupt request level 3
E015	MIR4	master PIC interrupt request level 4
E016	MIR5	master PIC interrupt request level 5
E017	MIR6	master PIC interrupt request level 6
E018	MIR7	master PIC interrupt request level 7
E019	MPICINT	master PIC INT output
E020	TOUTO	Multibus time-out (output)
E021	IOCNT1	PIT counter 1 output
E022	MBTOUT	Multibus timeout interrupt
E023	MBINT2	Multibus MBINT2 interrupt
E024	EXTINT	external interrupt from front panel interface
E025	GND	ground
E026	INVIN	spare inverter input
E027	INVOUT	spare inverter output
E028	PFINT	power failure interrupt
E029	ORIN1	spare OR input 1
E030	BWDIS	disable Multibus buffered I/O writes
E031	MBINT6	Multibus MBINT6 interrupt
E032	MBINT7	Multibus MBINT7 interrupt
E033	DMAINT	ADMA common interrupt
E034	SBX1I0	SBX1 INT0 (J5)
E035	SBX1I1	SBX1 INT1 (J5)
E036	SBX2I0	SBX2 INT0 (J6)
E037	SBX2I1	SBX2 INT1 (J6)
E038	LPTINT	line printer interrupt
E039	MBINT0	Multibus MBINT0 interrupt
E040	GND	ground
E041	SIR0	slave interrupt request level 0
E042	SIR1	slave interrupt request level 1
E043	SIR2	slave interrupt request level 2
E044	SIR3	slave interrupt request level 3
E045	SIR4	slave interrupt request level 4
E046	SIR5	slave interrupt request level 5
E047	SIR6	slave interrupt request level 6
E048	SIR7	slave interrupt request level 7
E049	OROUT	OR gate output
E050	GND	ground
E051	RSINT	R*,S* flipflop interrupt (output)
E052	CLREDG	PPI output PC4

Table B-1. Stake Pin and Jumper Listing (Numerical Order, Cont.)

Stake Pin	Mnemonic	Description
E053	PERINT	DRAM parity error interrupt
E054	ORIN2	OR gate input 2
E055	SETEDG	R*,S* flipflop set input
E056	BUSRQE	Multibus bus request error interrupt
E057	GND	ground
E058	BRKINT	serial break detect interrupt
E059	NMIR	non-maskable interrupt request (Intel486 CPU)
E060	GND	ground
E061	CODE0	user-defined code bit 0
E062	GND	ground
E063	LATCHINTR	latch interrupt enable
E064	GND	ground
E065	LTPBSY	line printer busy
E066	PPIDIR	PPI port A transceiver direction
E067	GND	ground
E068	IOCLK4	4.0 MHz clock for PIT
E069	TCLK0	PIT counter 0 clock
E070	IOC1_2	1.288 MHz clock for PIT
E071	CODE1	user-defined code bit 1
E072	GND	ground
E073	SBXSWP	SBX address swap
E074	GND	ground
E075	SBX1O0	SBX1 OPT0
E076	DREQ2	ADMA DMA request channel 2
E077	SBX2O0	SBX2 OPT0
E078	IOCNT0	PIT counter 0 output
E079	TCLK1	PIT counter 1 clock input
E080	IOC1_2	1.288 MHz clock for PIT
E081	CODE2	user-defined code bit 2
E082	GND	ground
E083	SW-HWR	software reset connected to hardware reset enable
E084	GND	ground
E085	SBX1O1	SBX1 OPT1
E086	DACK2	ADMA DMA acknowledge channel 2
E087	SBX2O1	SBX2 OPT1
E088	IOCLK4	4.0 MHz clock for PIT
E089	TCLK2	PIT counter 2 clock input
E090	IOC1_2	1.288 MHz clock for PIT
E091	GND	EPROM capacity bit 2
E092	RMDE19	real mode DRAM end address 19
E093	MBWS19	Multibus window start address 19
E094	MBWE19	Multibus window end address 19
E095	PMDE23	PVAM DRAM end address 23
E096	PMDE19	PVAM DRAM end address 19
E097	DPS23	dual-port start address 23
E098	DPS19	dual-port start address 19
E099	DPE23	dual-port end address 23
E100	DPE19	dual-port end address 19
E101	ESZE2	ground
E102	GND	ground
E103	GND	ground
E104	GND	ground

Table B-1. Stake Pin and Jumper Listing (Numerical Order, Cont.)

Stake Pin	Mnemonic	Description
E105	GND	ground
E106	GND	ground
E107	GND	ground
E108	GND	ground
E109	GND	ground
E110	GND	ground
E111	GND	EPROM capacity bit 1
E112	RMDE18	real mode DRAM end address 18
E113	MBWS18	Multibus window start address 18
E114	MBWE18	Multibus window end address 18
E115	PMDE22	PVAM DRAM end address 22
E116	PMDE18	PVAM DRAM end address 18
E117	DPS22	dual-port start address 22
E118	DPS18	dual-port start address 18
E119	DPE22	dual-port end address 22
E120	DPE18	dual-port end address 18
E121	ESZE1	ground
E122	GND	ground
E123	GND	ground
E124	GND	ground
E125	GND	ground
E126	GND	ground
E127	GND	ground
E128	GND	ground
E129	GND	ground
E130	GND	ground
E131	ESZE0	EPROM capacity bit 0
E132	RMDE17	real mode DRAM end address 17
E133	MBWS17	Multibus window start address 17
E134	MBWE17	Multibus window end address 17
E135	PMDE21	PVAM DRAM end address 21
E136	PMDE17	PVAM DRAM end address 17
E137	DPS21	dual-port start address 21
E138	DPS17	dual-port start address 17
E139	DPE21	dual-port end address 21
E140	DPE17	dual-port end address 17
E141	GND	ground
E142	GND	ground
E143	GND	ground
E144	GND	ground
E145	GND	ground
E146	GND	ground
E147	GND	ground
E148	GND	ground
E149	GND	ground
E150	GND	ground
E151	MEG	EPROM at top of 16th Mbyte in PVAM
E152	RMDE16	real mode DRAM end address 16
E153	MBWS16	Multibus window start address 16
E154	MBWE16	Multibus window end address 16
E155	PMDE20	PVAM DRAM end address 20
E156	PMDE16	PVAM DRAM end address 16

Table B-1. Stake Pin and Jumper Listing (Numerical Order, Cont.)

Stake Pin	Mnemonic	Description
E157	DPS20	dual-port start address 20
E158	DPS16	dual-port start address 16
E159	DPE20	dual-port end address 20
E160	DPE16	dual-port end address 16
E161	GND	ground
E162	GND	ground
E163	GND	ground
E164	GND	ground
E165	GND	ground
E166	GND	ground
E167	GND	ground
E168	GND	ground
E169	GND	ground
E170	GND	ground
E171	TR_A	RS-422A/449 terminal ready (negative)
E172	TR_B	RS-422A/449 terminal ready (positive)
E173	RS_A	RS-422A/449 request to send (negative)
E174	RS_B	RS-422A/449 request to send (positive)
E175	SD_A	RS-422A/449 send data (negative)
E176	SD_B	RS-422A/449 send data (positive)
E177	TT_A	RS-422A/449 terminal timing (negative)
E178	TT_B	RS-422A/449 terminal timing (positive)
E179	RC	RS-422A/449 receive common
E180	ENABLE	RS-422A/449 multidrop enable
E181	TR_A	RS-422A/449 terminal ready (negative)
E182	TR_B	RS-422A/449 terminal ready (positive)
E183	RS_A	RS-422A/449 request to send (negative)
E184	RS_B	RS-422A/449 request to send (positive)
E185	SD_A	RS-422A/449 send data (negative)
E186	SD_B	RS-422A/449 send data (positive)
E187	TT_A	RS-422A/449 terminal timing (negative)
E188	TT_B	RS-422A/449 terminal timing (positive)
E189	GND	ground
E190	IDTRA	RS-422A/449 multidrop control
E191	RXCA	serial receive clock channel A
E192	RXC	serial receive clock channel A at J4L
E193	CDA	serial data terminal ready channel A
E194	DSR	serial data terminal ready channel A at J4L
E195	CTS	serial clear to send channel A at J4L
E196	CTSA	serial clear to send channel A
E197	RXD	receive data channel A at J4L
E198	RXDA	receive data channel A
E199	SGND	serial signal ground
E200	SHIELD	RS-422A/449 cable shield
E201	TXC	serial transmit clock channel A at J4L
E202	TXCA	serial transmit clock channel A
E203	DTR	serial data terminal ready channel A
E204	DTRA	serial data terminal ready channel A at J4L
E205	RTSA	serial request to send channel A
E206	RTS	serial request to send channel A at J4L
E207	TXDA	serial transmit data channel A
E208	TXD	serial transmit data channel A at J4L

Table B-1. Stake Pin and Jumper Listing (Numerical Order, Cont.)

Stake Pin	Mnemonic	Description
E209	GND	ground
E210	GND	ground
E211	CNFID3	SCSI ID bit 3
E212	CNFID2	SCSI ID bit 2
E213	CNFID1	SCSI ID bit 1
E214	CNFID0	SCSI ID bit 0
E215	IOCNT2	PIT counter 2 output
E216	TXCA	serial transmit clock channel A
E217	DTRA	serial data terminal ready at J4L
E218	IOCNT1	PIT counter 1 output
E219	TXCB	serial transmit clock channel B
E220	DTRB	serial data terminal ready at J4U
E221	GND	ground
E222	GND	ground
E223	GND	ground
E224	GND	ground
E225	TXCA	serial transmit clock channel A
E226	RXCA	serial receive clock channel A
E227	TXC	serial transmit clock at J4L
E228	TXCB	serial transmit clock channel B
E229	RXCB	serial receive clock channel B at J4U
E230	TXCB	serial transmit clock channel B at J4U
E231	EPGM1	EPROM program 1
E232	VCC	+5 V
E233	VCC	+5 V
E234	EPGM2	EPROM program 2
E235	DPAA23	dual-port alias bit 23
E236	EA18	EPROM address bit 18
E237	EA17	EPROM address bit 17
E238	EA15	EPROM address bit 15
E239	EA14	EPROM address bit 14
E240	GND	ground
E241	A19	I/O address 19
E242	A18	I/O address 18
E243	A16	I/O address 16
E244	A15	I/O address 15
E245	DPAA22	dual-port alias address 22
E246	MBINT0	Multibus INT0* interrupt destination
E247	MBINT1	Multibus INT1* interrupt destination
E248	MBINT2	Multibus INT2* interrupt destination
E249	MBINT3	Multibus INT3* interrupt destination
E250	GND	ground
E251	PER	dual port parity error PER*
E252	BDRV1	BUSDRIVE1*
E253	BDRV2	BUSDRIVE2*
E254	MBINT4	Multibus INT4* interrupt destination
E255	DPAA21	dual port alias address 21
E256	CBRLK	common Multibus request lock
E257	MBINT7	Multibus INT7* interrupt destination
E258	MBINT6	Multibus INT6* interrupt destination
E259	MBINT5	Multibus INT5* interrupt destination
E260	GND	ground

Table B-1. Stake Pin and Jumper Listing (Numerical Order, Cont.)

Stake Pin	Mnemonic	Description
E261	GND	ground
E262	ALWAYS	always release Multibus at end of cycle
E263	BPROO	bus priority out (input)
E264	HWRST	hardware reset
E265	DPAA20	dual port alias address 20
E266	CCLKO	Multibus constant clock (input)
E267	BCLKO	Multibus bus clock (input)
E268	BPRO	bus priority out (output)
E269	INIT	Multibus INIT*
E270	GND	ground
E271	CCLK	Multibus constant clock (output)
E272	BCLK	Multibus bus clock (output)

Table B-2. Stake Pin and Jumper Listing (Alphabetical Order)

Description	Stake Pin	Mnemonic
+5 V	E232	VCC
+5 V	E233	VCC
1.288 MHz clock for PIT	E070	IOC1_2
1.288 MHz clock for PIT	E080	IOC1_2
1.288 MHz clock for PIT	E090	IOC1_2
4.0 MHz clock for PIT	E068	IOCLK4
4.0 MHz clock for PIT	E088	IOCLK4
ADMA common interrupt	E033	DMAINT
ADMA DMA acknowledge channel 2	E086	DACK2
ADMA DMA request channel 2	E076	DREQ2
always release Multibus at end of cycle	E262	ALWAYS
bus priority out (input)	E263	BPROO
bus priority out (output)	E268	BPRO
BUSDRIVE1*	E252	BDRV1
BUSDRIVE2*	E253	BDRV2
common Multibus request lock	E256	CBRLK
disable Multibus buffered I/O writes	E030	BWDIS
DRAM parity error interrupt	E053	PERINT
dual port alias address 20	E265	DPAA20
dual port alias address 21	E255	DPAA21
dual port parity error PER*	E251	PER
dual-port alias address 22	E245	DPAA22
dual-port alias bit 23	E235	DPAA23
dual-port end address 16	E160	DPE16
dual-port end address 17	E140	DPE17
dual-port end address 18	E120	DPE18
dual-port end address 19	E100	DPE19
dual-port end address 20	E159	DPE20
dual-port end address 21	E139	DPE21
dual-port end address 22	E119	DPE22
dual-port end address 23	E099	DPE23
dual-port start address 16	E158	DPS16
dual-port start address 17	E138	DPS17
dual-port start address 18	E118	DPS18
dual-port start address 19	E098	DPS19
dual-port start address 20	E157	DPS20
dual-port start address 21	E137	DPS21
dual-port start address 22	E117	DPS22
dual-port start address 23	E097	DPS23
EPROM address bit 14	E239	EA14
EPROM address bit 15	E238	EA15
EPROM address bit 17	E237	EA17
EPROM address bit 18	E236	EA18
EPROM at top of 16th Mbyte in PVAM	E151	MEG
EPROM capacity bit 0	E131	ESZE0
EPROM capacity bit 1	E121	ESZE1
EPROM capacity bit 2	E101	ESZE2
EPROM program 1	E231	EPGM1
EPROM program 2	E234	EPGM2
external interrupt from front panel	E024	EXTINT
ground	E025	GND

Table B-2. Stake Pin and Jumper Listing (Alphabetical Order, Cont.)

Description	Stake Pin	Mnemonic
ground	E040	GND
ground	E050	GND
ground	E057	GND
ground	E060	GND
ground	E062	GND
ground	E064	GND
ground	E067	GND
ground	E072	GND
ground	E074	GND
ground	E082	GND
ground	E084	GND
ground	E091	GND
ground	E102	GND
ground	E103	GND
ground	E104	GND
ground	E105	GND
ground	E106	GND
ground	E107	GND
ground	E108	GND
ground	E109	GND
ground	E110	GND
ground	E111	GND
ground	E122	GND
ground	E123	GND
ground	E124	GND
ground	E125	GND
ground	E126	GND
ground	E127	GND
ground	E128	GND
ground	E129	GND
ground	E130	GND
ground	E141	GND
ground	E142	GND
ground	E143	GND
ground	E144	GND
ground	E145	GND
ground	E146	GND
ground	E147	GND
ground	E148	GND
ground	E149	GND
ground	E150	GND
ground	E161	GND
ground	E162	GND
ground	E163	GND
ground	E164	GND
ground	E165	GND
ground	E166	GND
ground	E167	GND
ground	E168	GND
ground	E169	GND
ground	E170	GND
ground	E189	GND

Table B-2. Stake Pin and Jumper Listing (Alphabetical Order, Cont.)

Description	Stake Pin	Mnemonic
ground	E209	GND
ground	E210	GND
ground	E221	GND
ground	E222	GND
ground	E223	GND
ground	E224	GND
ground	E240	GND
ground	E250	GND
ground	E260	GND
ground	E261	GND
ground	E270	GND
hardware reset	E264	HWRST
I/O address 15	E244	A15
I/O address 16	E243	A16
I/O address 18	E242	A18
I/O address 19	E241	A19
Intel486 CPU INTR input	E009	INTRI
line printer busy	E065	LTPBSY
line printer interrupt	E038	LPTINT
latch interrupt enable	E063	LATCHINTR
master PIC INT output	E019	MPICINT
master PIC interrupt request level 0	E011	MIR0
master PIC interrupt request level 1	E012	MIR1
master PIC interrupt request level 2	E013	MIR2
master PIC interrupt request level 3	E014	MIR3
master PIC interrupt request level 4	E015	MIR4
master PIC interrupt request level 5	E016	MIR5
master PIC interrupt request level 6	E017	MIR6
master PIC interrupt request level 7	E018	MIR7
Multibus bus clock (input)	E267	BCLKO
Multibus bus clock (output)	E272	BCLK
Multibus bus request error interrupt	E056	BUSRQE
Multibus constant clock (input)	E266	CCLKO
Multibus constant clock (output)	E271	CCLK
Multibus INT*	E269	INIT
Multibus INT0* interrupt destination	E246	MBINT0
Multibus INT1* interrupt destination	E247	MBINT1
Multibus INT2* interrupt destination	E248	MBINT2
Multibus INT3* interrupt destination	E249	MBINT3
Multibus INT4* interrupt destination	E254	MBINT4
Multibus INT5* interrupt destination	E259	MBINT5
Multibus INT6* interrupt destination	E258	MBINT6
Multibus INT7* interrupt destination	E257	MBINT7
Multibus MBINT0 interrupt	E002	MBINT1
Multibus MBINT0 interrupt	E039	MBINT0
Multibus MBINT2 interrupt	E023	MBINT2
Multibus MBINT3 interrupt	E004	MBINT3
Multibus MBINT4 interrupt	E005	MBINT4
Multibus MBINT5 interrupt	E006	MBINT5
Multibus MBINT6 interrupt	E031	MBINT6
Multibus MBINT7 interrupt	E032	MBINT7
Multibus time-out (input)	E010	TOUT

Table B-2. Stake Pin and Jumper Listing (Alphabetical Order, Cont.)

Description	Stake Pin	Mnemonic
Multibus time-out (output)	E020	TOUTO
Multibus timeout interrupt	E022	MBTOUT
Multibus window end address 16	E154	MBWE16
Multibus window end address 17	E134	MBWE17
Multibus window end address 18	E114	MBWE18
Multibus window end address 19	E094	MBWE19
Multibus window start address 16	E153	MBWS16
Multibus window start address 17	E133	MBWS17
Multibus window start address 18	E113	MBWS18
Multibus window start address 19	E093	MBWS19
non-maskable interrupt request (Intel486 CPU)	E059	NMIR
OR gate input 2	E054	ORIN2
OR gate output	E049	OROUT
PIT counter 0 clock	E069	TCLK0
PIT counter 0 output	E001	IOCNT0
PIT counter 0 output	E078	IOCNT0
PIT counter 1 clock input	E079	TCLK1
PIT counter 1 output	E021	IOCNT1
PIT counter 1 output	E218	IOCNT1
PIT counter 2 clock input	E089	TCLK2
PIT counter 2 output	E215	IOCNT2
power failure interrupt	E028	PFINT
PPI output PC4	E052	CLREDG
PPI port A transceiver direction	E066	PPIDIR
PVAM DRAM end address 16	E156	PMDE16
PVAM DRAM end address 17	E136	PMDE17
PVAM DRAM end address 18	E116	PMDE18
PVAM DRAM end address 19	E096	PMDE19
PVAM DRAM end address 20	E155	PMDE20
PVAM DRAM end address 21	E135	PMDE21
PVAM DRAM end address 22	E115	PMDE22
PVAM DRAM end address 23	E095	PMDE23
R*,S* flipflop interrupt (output)	E051	RSINT
R*,S* flipflop set input	E055	SETEDG
real mode DRAM end address 16	E152	RMDE16
real mode DRAM end address 17	E132	RMDE17
real mode DRAM end address 18	E112	RMDE18
real mode DRAM end address 19	E092	RMDE19
receive data channel A	E198	RXDA
receive data channel A at J4L	E197	RXD
RS-422A/449 cable shield	E200	SHIELD
RS-422A/449 multidrop control	E190	IDTRA
RS-422A/449 multidrop enable	E180	ENABLE
RS-422A/449 receive common	E179	RC
RS-422A/449 request to send (negative)	E173	RS_A
RS-422A/449 request to send (negative)	E183	RS_A
RS-422A/449 request to send (positive)	E174	RS_B
RS-422A/449 request to send (positive)	E184	RS_B
RS-422A/449 send data (negative)	E175	SD_A

Table B-2. Stake Pin and Jumper Listing (Alphabetical Order, Cont.)

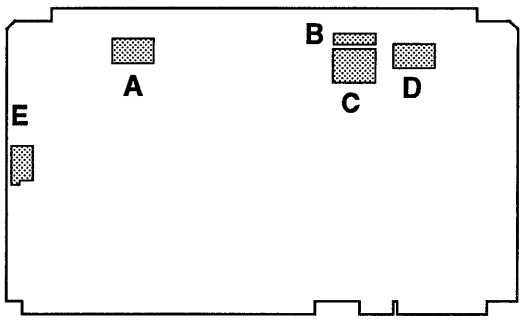
Description	Stake Pin	Mnemonic
RS-422A/449 send data (negative)	E185	SD_A
RS-422A/449 send data (positive)	E176	SD_B
RS-422A/449 send data (positive)	E186	SD_B
RS-422A/449 terminal ready (negative)	E171	TR_A
RS-422A/449 terminal ready (negative)	E181	TR_A
RS-422A/449 terminal ready (positive)	E172	TR_B
RS-422A/449 terminal ready (positive)	E182	TR_B
RS-422A/449 terminal timing (negative)	E177	TT_A
RS-422A/449 terminal timing (negative)	E187	TT_A
RS-422A/449 terminal timing (positive)	E178	TT_B
RS-422A/449 terminal timing (positive)	E188	TT_B
SBX address swap	E073	SBXSWP
SBX1 INT0 (J5)	E034	SBX1I0
SBX1 INT1 (J5)	E035	SBX1I1
SBX1 OPT0	E075	SBX1O0
SBX1 OPT1	E085	SBX1O1
SBX2 INT0 (J6)	E036	SBX2I0
SBX2 INT1 (J6)	E037	SBX2I1
SBX2 OPT0	E077	SBX2O0
SBX2 OPT1	E087	SBX2O1
SCSI ID bit 0	E214	CNFID0
SCSI ID bit 1	E213	CNFID1
SCSI ID bit 2	E212	CNFID2
SCSI ID bit 3	E211	CNFID3
SCSI interrupt	E003	SCZINT
serial break detect interrupt	E058	BRKINT
serial clear to send channel A	E196	CTSA
serial clear to send channel A at J4L	E195	CTS
serial data terminal ready at J4L	E217	DTRA
serial data terminal ready at J4U	E220	DTRB
serial data terminal ready channel A	E193	CDA
serial data terminal ready channel A	E203	DTR
serial data terminal ready channel A at J4L	E194	DSR
serial data terminal ready channel A at J4L	E204	DTRA
serial interrupt (MPSC)	E007	SERINT
serial receive clock channel A	E191	RXCA
serial receive clock channel A	E226	RXCA
serial receive clock channel A at J4L	E192	RXC
serial receive clock channel B at J4U	E229	RXCB
serial request to send channel A	E205	RTSA
serial request to send channel A at J4L	E206	RTS
serial signal ground	E199	SGND
serial transmit clock at J4L	E227	TXC
serial transmit clock channel A	E216	TXCA
serial transmit clock channel A	E225	TXCA
serial transmit clock channel A	E202	TXCA
serial transmit clock channel A at J4L	E201	TXC
serial transmit clock channel B	E219	TXCB
serial transmit clock channel B	E228	TXCB
serial transmit clock channel B at J4U	E230	TXCB
serial transmit data channel A	E207	TXDA

Table B-2. Stake Pin and Jumper Listing (Alphabetical Order, Cont.)

Description	Stake Pin	Mnemonic
serial transmit data channel A at J4L	E208	TXD
slave interrupt request level 0	E041	SIR0
slave interrupt request level 1	E042	SIR1
slave interrupt request level 2	E043	SIR2
slave interrupt request level 3	E044	SIR3
slave interrupt request level 4	E045	SIR4
slave interrupt request level 5	E046	SIR5
slave interrupt request level 6	E047	SIR6
slave interrupt request level 7	E048	SIR7
slave PIC INT output	E008	SPICINT
software reset to hardware reset enable	E083	SW-HWR
spare inverter input	E026	INVIN
spare inverter output	E027	INVOUT
spare OR input 1	E029	ORIN1
user-defined code bit 0	E061	CODE0
user-defined code bit 1	E071	CODE1
user-defined code bit 2	E081	CODE2

Stake Pin Matrix With Stake Pin Names

This section shows the five jumper matrix blocks with the stake pin mnemonic names.



The board includes five jumper blocks:

- Jumper matrix A, page B-18
- Jumper matrix B, page B-19
- Jumper matrix C, page B-19
- Jumper matrix D, page B-20
- Jumper matrix E, page B-21

Additional jumpers based on the size of the memory module are shown on page B-21.

Jumper Matrix A

IOCNTØ	MBINT1	SCZINT	MBINT3	MBINT4	MBINT5	SERINT	SPICINT	INTR	TOUT
EØØ1	EØØ2	EØØ3	EØØ4	EØØ5	EØØ6	EØØ7	EØØ8	EØØ9	EØ1Ø
|	|	|	|	|	|	|	|	|	|
MIRØ	MIR1	MIR2	MIR3	MIR4	MIR5	MIR6	MIR7	MPICINT	TOUTO
EØ11	EØ12	EØ13	EØ14	EØ15	EØ16	EØ17	EØ18	EØ19	EØ2Ø
IOCNT1	MBTOUT	MBINT2	EXTINT	GND	INVIN	INVOUT	PFINT	ORIN1	BWDIS
EØ21	EØ22	EØ23	EØ24	EØ25	EØ26	EØ27	EØ28	EØ29	EØ3Ø
								|	|
MBINT6	MBINT7	DMAINT	SBX1IØ	SBX1I1	SBX2IØ	SBX2I1	LPTINT	MBINTØ	GND
EØ31	EØ32	EØ33	EØ34	EØ35	EØ36	EØ37	EØ38	EØ39	EØ4Ø
|	|	|	|	|	|	|	|		
SIRØ	SIR1	SIR2	SIR3	SIR4	SIR5	SIR6	SIR7	OROUT	GND
EØ41	EØ42	EØ43	EØ44	EØ45	EØ46	EØ47	EØ48	EØ49	EØ5Ø
								|	
RSINT	CLREDG	PERINT	ORIN2	SETEDG	BUSRQE	GND	BRKINT	NMIR	GND
EØ51	EØ52	EØ53---EØ54		EØ55	EØ56	EØ57	EØ58	EØ59	EØ6Ø

Note: | or --- Indicates default jumper

Jumper Matrix B

CODEØ	GND	RSVD	RSVD	LTPBSY	PPIDIR	GND	IOCLK4	TCLKØ	IOC1_2
EØ61	---EØ62	EØ63	EØ64	EØ65	EØ66	EØ67	EØ68	EØ69	---EØ7Ø
CODE1	GND	SBXSWP	GND	SBX1ØØ	DREQ2	SBX2ØØ	IOCNTØ	TCLK1	IOC1_2
EØ71	EØ72	EØ73	EØ74	EØ75	EØ76	EØ77	EØ78	EØ79	---EØ8Ø
CODE2	GND	SW-HWR	GND	SBX101	DACK2	SBX201	IOCLK4	TCLK2	IOC1_2
EØ81	EØ82	EØ83	EØ84	EØ85	EØ86	EØ87	EØ88	EØ89	---EØ9Ø

Note: | or--- Indicates default jumper

Jumper Matrix C

GND	RMDE19	MBWS19	MBWE19	PMDE23	PMDE19	DPS23	DPS19	DPE23	DPE19
EØ91	EØ92	EØ93	EØ94	EØ95	EØ96	EØ97	EØ98	EØ99	E1ØØ
	|								
ESZE2	GND	GND	GND	GND	GND	GND	GND	GND	GND
E1Ø1	E1Ø2	E1Ø3	E1Ø4	E1Ø5	E1Ø6	E1Ø7	E1Ø8	E1Ø9	E11Ø
GND	RMDE18	MBWS18	MBWE18	PMDE22	PMDE18	DPS22	DPS18	DPE22	DPE18
E111	E112	E113	E114	E115	E116	E117	E118	E119	E12Ø
|	|				|				|
ESZE1	GND	GND	GND	GND	GND	GND	GND	GND	GND
E121	E122	E123	E124	E125	E126	E127	E128	E129	E13Ø
ESZEØ	RMDE17	MBWS17	MBWE17	PMDE21	PMDE17	DPS21	DPS17	DPE21	DPE17
E131	E132	E133	E134	E135	E136	E137	E138	E139	E14Ø
|					|				|
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
E141	E142	E143	E144	E145	E146	E147	E148	E149	E15Ø
MEG	RMDE16	MBWS16	MBWE16	PMDE2Ø	PMDE16	DPS2Ø	DPS16	DPE2Ø	DPE16
E151	E152	E153	E154	E155	E156	E157	E158	E159	E16Ø
|	|	|		|	|			|	|
GND	GND	GND	GND	GND	GND	GND	GND	GND	GND
E161	E162	E163	E164	E165	E166	E167	E168	E169	E17Ø

Note: | or--- Indicates default jumper

Jumper Matrix D

TR_A	TR_B	RS_A	RS_B	SD_A	SD_B	TT_A	TT_B	RC	ENABLE
E171	E172	E173	E174	E175	E176	E177	E178	E179	E18Ø
TR_A	TR_B	RS_A	RS_B	SD_A	SD_B	TT_A	TT_B	GND	IDTRA
E181	E182	E183	E184	E185	E186	E187	E188	E189	E19Ø
RXCA	RXC	CDA	DSR	CTS	CTSA	RXD	RXDA	SGND	SHIELD
E191	E192	E193	E194	E195	E196	E197	E198	E199	E2ØØ
|	|	|	|	|	|	|	|	|	
TXC	TXCA	DTR	DTRA	RTSA	RTS	TXDA	TXD	GND	GND
E2Ø1	E2Ø2	E2Ø3	E2Ø4	E2Ø5	E2Ø6	E2Ø7	E2Ø8	E2Ø9	E21Ø
CNFID3	CNFID2	CNFID1	CNFIDØ	IOCNT2	TXCA	DTR	IOCNT1	TXCB	DTRB
E211	E212	E213	E214	E215---E216		E217	E218---E219		E22Ø
GND	GND	GND	GND	TXCA	RXCA	TXC	TXCB	RXCB	TXCB
E221	E222	E223	E224	E225---E226		E227	E228---E229		E23Ø

Note: | or--- Indicates default jumper

Jumper Matrix E

```
EPGM1   VCC    VCC   EPGM2  DPAA23
 E231   E232   E233   E234   E235
         |
 EA18   EA17   EA15   EA14   GND
 E236   E237   E238   E239   E24Ø
                 |      |
  A19    A18    A16    A15  DPAA22
 E241   E242   E243   E244   E245

MBINTØ MBINT1 MBINT2 MBINT3  GND
 E246   E247   E248   E249   E25Ø

  PER  BDRV1  BDRV2  MBINT4 DPAA21
 E251   E252   E253   E254   E255

CBRLK  MBINT7 MBINT6 MBINT5  GND
 E256   E257   E258   E259   E26Ø

 GND   ALWAYS BPROO  HWRST  DPAA2Ø
 E261   E262   E263   E264   E265
                 |      |
CCLKO  BCLKO   BPRO   INIT   GND
 E266   E267   E268   E269   E27Ø
  |      |
 CCLK   BCLK
 E271   E272
```

Note: | or -- Indicates default jumper

Table B-3. Additional Jumpers due to Memory Module Type

Memory Module	Default Ending Address DRAM	Default Ending Address Dual Port	Additional Jumpers (E) 095 105	115 125	135 145	096 106	099 109	119 129	139 149	100 110
MM302 (2 Mbytes)	1F FFFFH	1F FFFFH				X				X
MM304 (4 Mbytes)	3F FFFFH	3F FFFFH			X	X			X	X
MM308 (8 Mbytes)	7F FFFFH	7F FFFFH		X	X	X		X	X	X
MM3008 (8 Mbytes)	7F FFFFH	7F FFFFH		X	X	X		X	X	X
MM3016 (16 Mbytes)	F7 FFFFH	7F FFFFH	X	X	X		X	X	X	
MM3032 (32 Mbytes)	F7 FFFFH	7F FFFFH	X	X	X		X	X	X	

Note: "X" means insert jumper.

Default Wire List

Table B-4 lists the factory installed jumpers for the iSBC 486/12 series boards. Refer to Table B-3 for jumpers that depend the size of the memory module installed.

Table B-4. Default Wire List

Jumper	Stake Pin Names
E001 E011	MIR0-CNT0
E002 E012	MIR1-MBINT1
E003 E013	MIR2-SCZINT
E004 E014	MIR3-MBINT3
E005 E015	MIR4-MBINT4
E006 E016	MIR5-MBINT5
E007 E017	MIR6-SERINT
E008 E018	MIR7-SPICINT
E009 E019	MPICINT-INTR
E010 E020	MBTOT-MBTOUT
E029 E039	MBINT0-ORIN1
E030 E040	BWDIS-GND
E031 E041	SIR0-MBINT6
E032 E042	SIR1-MBINT7
E033 E043	SIR2-DMAINT
E034 E044	SIR3-SBX1IR0
E035 E045	SIR4-SBX1IR1
E036 E046	SIR5-SBX2IR0
E037 E047	SIR5-SBX2IR1
E038 E048	SIR7-LPINT
E049 E059	OROUT-NMIR
E053 E054	PERINT-ORIN2
E061 E062	CODE0
E069 E070	TCLK0-CLK1.2
E079 E080	TCLK1-CLK1.2
E089 E090	TCLK2-CLK1.2
E111 E121	ESZE1-GND
E131 E141	ESZE0-GND
E151 E161	MEG-GND
E092 E102	RMDE19 "1"
E112 E122	RMDE18 "1"
E152 E162	RMDE16 "1"
E153 E163	MBWS16 "1"
E155 E165	PMDE20 "1"
E116 E126	PMDE18 "1"
E136 E146	PMDE17 "1"
E156 E166	PMDE16 "1"
E159 E169	DPE20 "1"
E120 E130	DPE18 "1"

Table B-4. Default Wire List, Continued

Jumper	Stake Pin Names
E140 E150	DPE17 "1"
E160 E170	DPE16 "1"
E191 E201	RXCA-TXC
E192 E202	RXC-TXCA
E193 E203	DTRA-DTR
E194 E204	DSR-DTRA
E195 E205	CTS-RTSA
E196 E206	CTSA-RTS
E197 E207	RXD-TXDA
E198 E208	RXDA-TXD
E199 E209	SGND-GND
E215 E216	CNT2-TXCA
E218 E219	CNT1-TXCB
E225 E226	TXCA-RXCA
E228 E229	TXCB-RXCB
E232 E237	VCC-EA17
E238 E243	A16-EA15
E239 E244	A15-EA14
E263 E268	BPRO-MBBPRO
E264 E269	INIT-HWRESET
E266 E271	CCLK-MBCCLK
E267 E272	BCLK-MBBCLK

Note:
Refer to Table B-3 for additional jumpers that depend on the size of the memory module installed.

Default Components

Table B-5 lists the components that are installed at the factory on the default board.

Table B-5. Default Components

Component	Description	iSBC 486/12 Series	iSBC 486/12S Series
RP1	SCSI resistor pack	OUT	IN
RP2	SCSI resistor pack	OUT	IN
RP3	SCSI resistor pack	OUT	IN
RP4	Serial resistor pack	OUT	OUT
RP5	Serial resistor pack	IN	IN
RP6	Serial resistor pack	IN	IN
RP7	Multibus resistor pack	IN	IN
U14	RS-422A/449 driver	OUT	OUT
U28	EPROM	OUT	OUT
U43	EPROM	OUT	OUT
F1	SCSI fuse	OUT	IN

***

Multidrop Information C

Introduction

The iSBC 486/12 Single Board Computer can be configured for a multidrop application. This appendix shows an example for configuring Connector J4L for operation in a full-duplex RS-422A/449 multidrop application (Figure C-1). Some of the line conditioning decisions you must make are as follows:

- You must calculate the value of the bias resistors for installation on the master unit in the system.
- You must calculate the value of the termination resistors for installation on the farthest slave unit in the system.

Figure C-1 shows a typical RS-422A/449 full-duplex multidrop configuration that assumes only one master station is attached to the system and always drives the output lines. A full-duplex system allows a slave to listen to the data line and to perform some task in parallel with a task performed on another slave, but only the selected slave may transmit to the master.

You can configure a simpler RS-422A/449 multidrop system for half-duplex operation; however, the protocol is more strict. A half-duplex configuration requires two data lines (to carry a differential signal) and a ground line (return) between the master and all slaves in the system. Remember, however, that a half-duplex system can communicate in only one direction at a time.

For all practical purposes, the half-duplex system allows no priority for masters and slaves; all units may listen to whomever is using the data line. This presents a programming constraint in that the system software protocol for half-duplex operation must be designed to allow only one unit to transmit at any given instant.

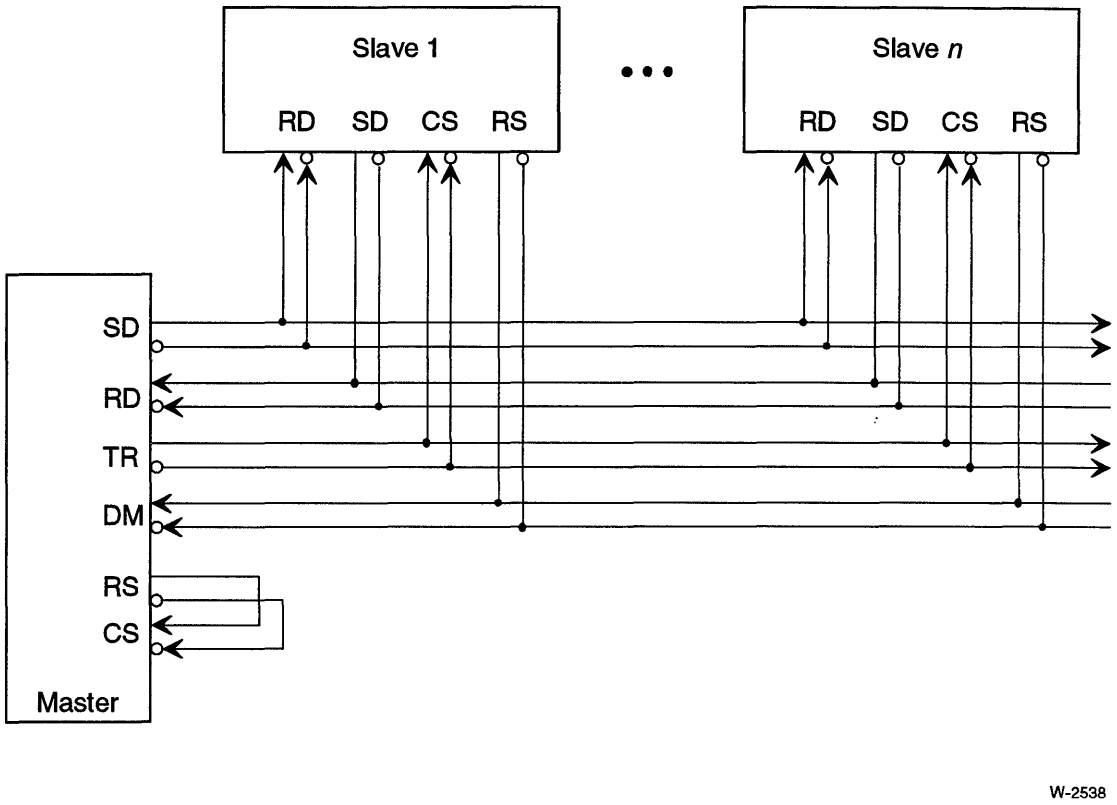


Figure C-1. Full-Duplex RS-422A/449 Multidrop Configuration Example

Multidrop Configuration Sequence

Connector J4L is factory-configured for RS-232C DCE interface operation. To reconfigure connector J4L as an RS-422A/449 interface suitable for multidrop operation, follow these steps:

1. Reconfigure connector J4L to operate as an RS-422A/449 DCE interface (refer to Chapter 7).
2. If the iSBC 486/12 board is the master in the multi-drop system, calculate values for the biasing resistor packs in sockets RP5 and RP6. If the iSBC 486/12 board is a slave, remove RP5 and RP6.
3. If the iSBC 486/12 board is the slave farthest from the master in the multidrop system, calculate value for termination resistor pack in socket RP4.
4. Install jumper E180-E190. Installing this jumper allows the 8274 MPSC to enable the output drivers by using the DM (DTR) signal. You must install E180-E190 on each master in your system.

Multidrop Bias Resistors RP5 and RP6

NOTE

The calculations given as examples in this appendix are for a system with four stations. If more than four stations are connected in a system, you must recalculate the resistance values.

In the RS-422A/449 multidrop application, the open or floating data lines in the system must be biased with user-supplied bias resistors. Without the bias resistors, the state of a floating line cannot be guaranteed.

The iSBC 486/12 board default configuration contains two 2.2 kΩ bias resistors (RP5 and RP6) that provide the pull-up/pull-down biasing for the differential signal lines. You must install RP5 and RP6 on the master and remove them from all slaves in an RS-422A/449 interface multidrop application.

The exact value of the bias resistors may be calculated only on an individual application basis since the controlling parameters vary from one application to another. The following procedure gives an example for calculating the bias resistance required to dissipate the leakage currents encountered in a typical full-duplex RS-422A/449 multidrop application. The procedure determines both the best case and the worst case resistor values. Any resistor value that satisfies both cases may be installed on the iSBC 486/12 board as a bias resistor.

Using the configuration shown in Figure C-2, the value of the bias resistors must be calculated for two conditions:

1. When the lines are tri-stated.
2. When the lines are driven to the marking (off) state.

When the lines are tri-stated, assume the following conditions:

1. $V_A - V_B \geq 0.3$ V to guarantee differential voltage for a "spacing" (on) condition.
2. All drivers are tri-stated.
3. Driver leakage current (I_{LEAK}) is ± 100 μA for each driver (refer to the EIA RS-422 specification).

4. The general equation for the input current to the receiver is

$$\text{for } I_{in(A)} \text{ (into device)} \qquad I = \frac{V + 3\ V}{4\ k\Omega}$$

$$\text{for } I_{in(B)} \qquad I = \frac{V - 3\ V}{4\ k\Omega}$$

(refer to the EIA RS-422 Specification).

5. No common mode voltage occurs between drivers and receivers.
6. Assume $V_A \geq 2.6$ V and $V_B \leq 2.3$ V.

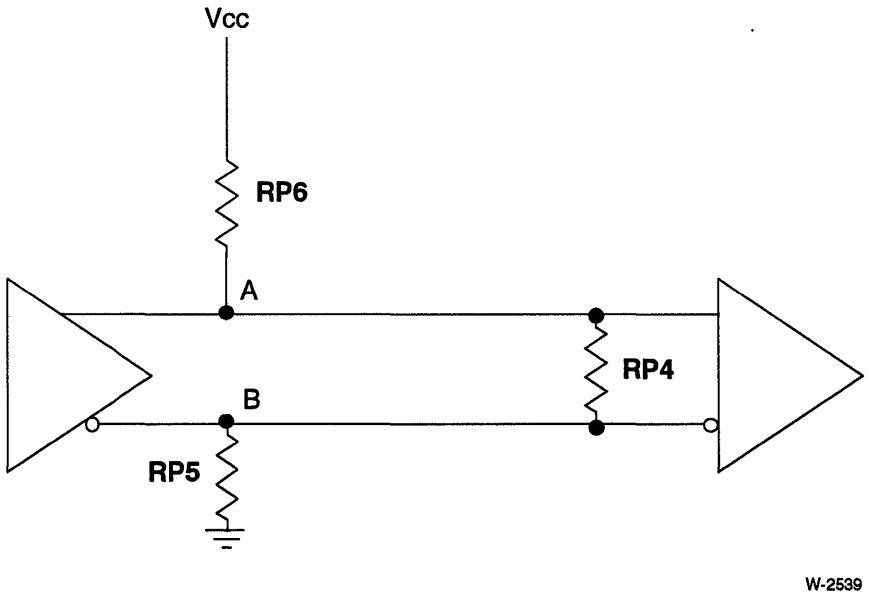


Figure C-2. Bias and Terminator Resistor Placement Example

Case 1: Lines Floating

To calculate the maximum value of RP5 and RP6, we must calculate the resistance required to guarantee at least a 0.3 V differential between the lines when they are tri-stated.

Referring to the previous general receiver input current equation:

$$I_{in}(A) = \frac{V_A + 3\ V}{4\ k\Omega} = \frac{(2.6 + 3)\ V}{4\ k\Omega} = 1.4\ mA$$

$$I_{in}(B) = \frac{V_B - 3\ V}{4\ k\Omega} = \frac{(2.2 - 3)\ V}{4\ k\Omega} = 0.18\ mA$$

Referring to Figure C-3:

$$I_T = \frac{V_A - V_B}{RP4} = \frac{(2.6 - 2.3)\ V}{RP4} = \frac{0.3\ V}{RP4}$$

with the termination resistance RP4 = 100 Ω, I_T = 3 mA.

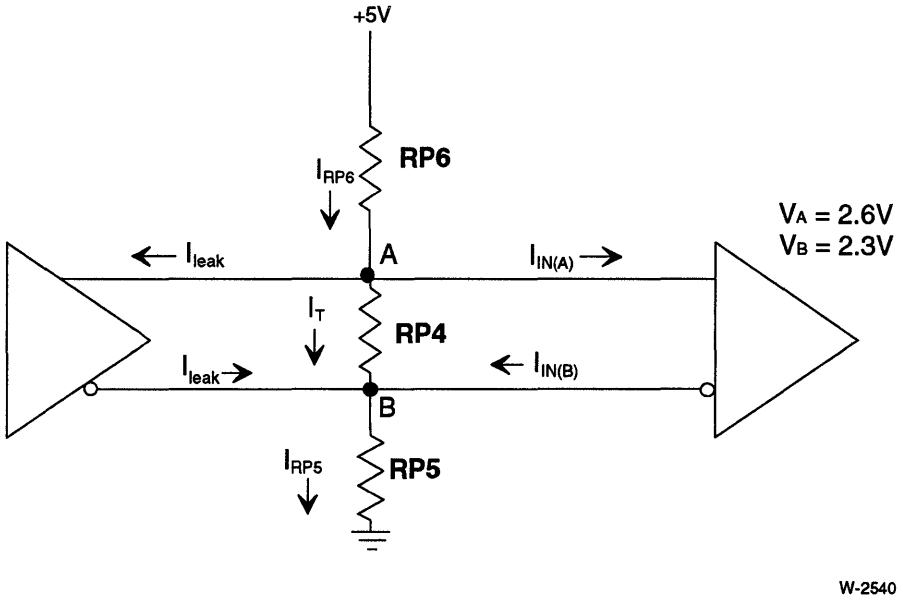


Figure C-3. Case 1 Configuration Example (Lines Floating)

Writing the node equation for Node A:

$$I_{RP6} = I_{in}(A) + I_{LEAK} + I_T = 1.4\ mA + (100\ \mu A \times 4\ devices) + 3\ mA$$

$$I_{RP6} = 4.8\ mA$$

$$\text{To keep } V_A \geq 2.6\ V,\ RP6 \leq \frac{5 - V_A}{I_{RP6}} = \frac{(5 - 2.6)\ V}{4.8\ mA}$$

$$RP6 \geq 500\ \Omega$$

Writing the node equation for Node B:

$$I_{RP5} = I_{in}(B) + I_{LEAK} + I_T = 0.18\ mA + 400\ \mu A + 3\ mA = 3.58\ mA$$

$$\text{To keep } V_B \leq 2.3\ V \text{ and } RP5 \leq \frac{V_B}{I_{RP5}} = \frac{2.3\ V}{3.58\ mA} \text{ and } RP5 \leq 642\ \Omega$$

Case 2: Lines Driven

The driver controls the lines to either a marking (off) or a spacing (on) state. The resistor values needed for the spacing state were previously calculated in Case 1. For the resistor values needed in the marking state, $V_B - V_A \geq 0.3$ V must be guaranteed as well as the following assumptions:

1. Assume $V_A \leq 2.3$ V and $V_B \geq 2.6$ V.
2. The general receiver input current equations are the same as in Case 1.
3. Driver output current is +20 mA (I_{OL}) and -20 mA (I_{OH}).
4. Driver leakage currents are negligible in relation to the drive current of the enabled driver.
5. No common mode voltage occurs between the driver and receiver.

Referring to the general receiver input current equation:

$$I_{in}(A) = \frac{V_A - 3\text{ V}}{4\text{ k}\Omega} = \frac{(2.3 - 3)\text{ V}}{4\text{ k}\Omega} = -0.18\text{ mA}$$

$$I_{in}(B) = \frac{V_B + 3\text{ V}}{4\text{ k}\Omega} = \frac{(2.6 + 3)\text{ V}}{4\text{ k}\Omega} = 1.4\text{ mA}$$

Referring to Figure C-5:

$$I_T = \frac{V_B - V_A}{RP4} = \frac{(2.6 - 2.3)\text{ V}}{100\ \Omega} = 3\text{ mA}$$

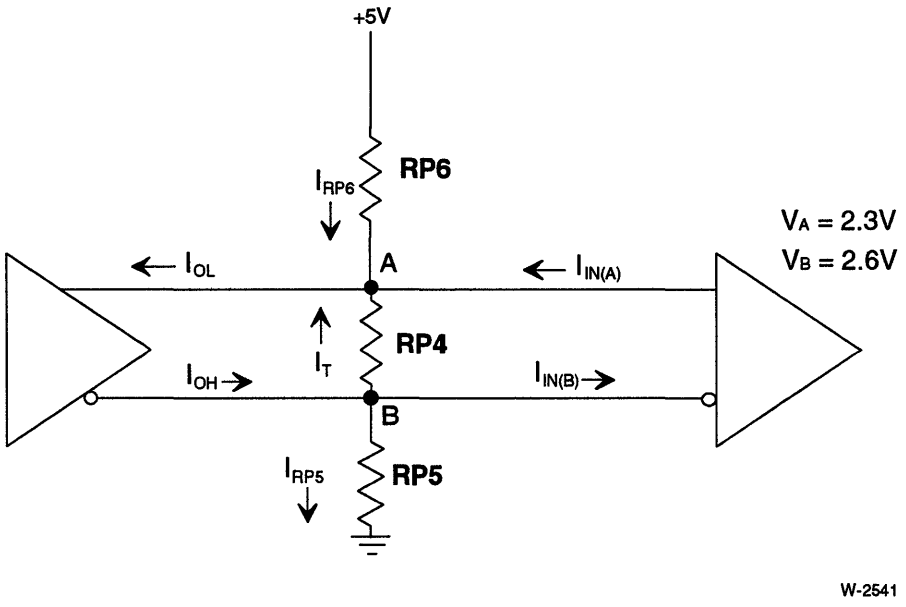


Figure C-4. Case 2 Configuration Example (Lines Driven)

Writing the node equation for Node A (refer to Figure C-4), we have:

$$I_{RP6} + I_{in}(A) + I_T \leq I_{OL}$$

$$I_{RP6} \leq I_{OL} - I_T - I_{in}(A)$$

$$I_{RP6} \leq 20 \text{ mA} - 3 \text{ mA} - 0.18 \text{ mA}$$

$$I_{RP6} \leq 16.8 \text{ mA}$$

and

$$\frac{5 \text{ V} - V_A}{RP6} \leq I_{RP6} \quad \text{so that RP6} \geq \frac{5 \text{ V} - V_A}{16.8 \text{ mA}} \quad \text{and RP6} \geq 161\ \Omega$$

Writing the node equation for Node B:

$$I_{RP5} + I_T + I_{in}(B) \leq I_{OH}$$

$$I_{RP5} \leq I_{OH} - I_T - I_{in(B)}$$

$$I_{RP5} \leq 20 \text{ mA} - 3 \text{ mA} - 1.4 \text{ mA}$$

$$I_{RP5} \leq 15.6 \text{ mA}$$

$$I_{RP5} = \frac{V_B}{RP5} \leq 15.6 \text{ mA}$$

$$RP5 \geq \frac{V_B}{I_{RP5}} \geq \frac{2.6 \text{ V}}{15.6 \text{ mA}}$$

$$RP5 \geq 167\ \Omega$$

Combining these results, we find: $167\ \Omega \leq RP5 \leq 638\ \Omega$
$161\ \Omega \leq RP6 \leq 500\ \Omega$

The values for RP5 and RP6 should be equal and near the top of the resistor range to reduce the current through the driver. Choosing a resistor value of 450 Ω ± 10% satisfies this requirement.

Multidrop Termination Resistor RP4

For applications with long transmission lines, add a termination resistor (RP4) at the receiver (slave board) farthest from the driver (master) to reduce the line signal reflection. The termination resistance value should be as close as possible to the characteristic impedance of the serial cable, approximately 100 Ω but not less than 90 Ω.

In some of the implementations with long cable stubs or drivers driving in multiple directions on the cable, the termination may need to be placed at several end-point locations on the cable. Because the RS-422 drivers do not have enough output current to drive multiple 100 Ω terminators, the resistor value for each terminator must be increased such that the total load resistance is no less than 90 Ω between the differential lines (refer to the EIA RS-422 Specification).

***

Compatibility Notes D

Table D-1 lists the major compatibility differences between the iSBC 486/12(S) series boards and the iSBC 386/12(S) board.

Table D-1. Compatibility Notes

Feature	iSBC 386/12	iSBC 486/12
EPROM access time	T_{acc} < 277 ns	T_{acc} < 250 ns
Bus-vectored interrupts	Yes	No
iLBX™ on Connector P2	Yes	No
DMA support for serial channels	Yes	No
DRAM starting address is always 0 in PVAM	No	Yes
SBX terminate DMA supported only as input to ADMA (channels 1 and 3)	No	Yes
2764 EPROM support	Yes	No
I/O addresses 300H-3FFH mapped to DAG (iSBC 486/12S only)	No	Yes
Set parity error causes parity error after next access to DRAM	No	Yes
I/O Address 0EEH	Megabyte Page Register	Dual-Port 16 Mbyte Page Register
16-bit Board ID register	No	Yes
3-bit Board ID code	Defined by board type	User-defined
DRAM address space repeated in 2nd Gbyte page	Yes	No
Slave 82C59A PIC interrupt request levels 3 through 7	Hard wired	Jumper configured
Software reset can be jumpered to Multibus system reset	No	Yes
Multibus address aliasing for dual-port access	No	Yes
Compatible with older "non-A" version of iSBC 544A and iSBC 522A	Yes	No

Table D-1. Compatibility Notes (Continued)

Feature	iSBC 386/12	iSBC 486/12
Serial break detect logic (external to MPSC)	No	Yes
Multibus Bus Request Error Interrupt	No	Yes

***

iRMX® Documentation Roadmap

E

You can develop real-time multi-tasking applications for the iSBC 486/12 series boards using the iRMX operating system. This operating system includes many features, including I/O device drivers, task scheduling, memory management, and dynamic creation and deletion of iRMX objects. This Appendix describes the iRMX operating system support for the iSBC 486/12 series boards.

- ***Read This First***

  This manual describes how to configure Multibus boards for use with iRMX. Also, this manual provides instructions on booting the iRMX operating system on Intel Multibus I microcomputers. It contains specific details for booting from diskette and installing iRMX software on a hard disk controlled by the Multibus board.

  The iRMX operating systems may be configured to include different features. The features are selected with the Interactive Configuration Utility (ICU). The ICU uses a data file called a "definition file" to record which features have been selected. The iRMX operating systems contain a set of Standard Definition Files for the iSBC 486/12 board and most other Intel Multibus I Single Board Computers which are detailed in this manual. The Standard Definition Files are the recommended starting point for developing custom iRMX application systems.

  This manual also contains jumper configuration information required to use the iSBC 486/12 with the operating systems. iRMX II.4 Update 1 and iRMX III.1 Update 3 provide SCSI boot diskettes and firmware images for the iSBC 486/12 and iSBC 486/12S boards. Currently, no Intel microcomputer system comes with an iSBC 486/12 board integrated in it.

- ***ICU User's Guide and Quick Reference***

  This manual describes the configurable parameters of the iRMX operating system.

- ***System Jobs Manual***

  This manual describes the System Jobs included with the iRMX II and III operating systems, including the PCI Server Job and its functions.

- ***iRMX Bootstrap Loader Manual***

  The first stage of the iRMX Bootstrap Loader (BSL) resides in EPROM and loads the operating systems from a diskette, hard disk or network system. This manual contains a description on how to create your own version of the BSL. It is often combined with the System Debug Monitor (iSDM II). This monitor is used with the iRMX I and II operating systems. The iRMX III operating system uses a different iSDM monitor which is provided only with the operating system and is loaded when the operating system is booted.

For more information, contact your nearest Intel sales office. A list of Intel sales offices is provided in the back of this manual.

***

Glossary

This glossary includes a *Glossary of Terms* and a *Glossary of Signal Names* (page Glossary-5).

Glossary of Terms

82258 ADMA 82258 Advanced DMA Controller (ADMA)

8274 MPSC 8274 Multiple-Protocol Serial Controller (MPSC). The MPSC has two full-duplex serial channels.

82C54 PIT 82C54 Programmable Interval Timer (PIT). The PIT has three counters/timers.

82C55A PPI 82C55A Programmable Peripheral Interface (PPI). The PPI has three 8-bit ports.

82C59A PIC 82C59A Programmable Interrupt Controller (PIC). The PIC has 8 interrupt request levels. The iSBC 486/12 board has two PIC that are cascaded to provide 15 interrupt request levels. Chapter 8, "CPU Subsystem Configuration," discusses the interrupt configuration.

87033B SPC Fujitsu 87033B SCSI Protocol Controller (SPC)

blast Refers to a DMA transfer mode or a board register that controls the DMA transfers to the SCSI FIFOs. In blast mode the ADMA/DAG will transfer 16 bytes of data during each ADMA cycle.

break detect logic The break detect logic is a circuit that is external to the 8274 MPSC. This circuit detects a serial break condition on either of the two serial channels. The output of the break detect logic is an interrupt signal that can be used as an interrupt source for the 82C59A PIC. The break detect logic can be used to generate an interrupt before the 8274 MPSC is initialized.

byte An 8-bit quantity of memory; the smallest unit of memory referenced by an address.

cache A small, fast memory which holds the active parts of a larger, slower memory. The Intel486 CPU has an on-chip cache.

cache miss A request for access to memory which requires actually reading main memory.

compatibility mode A mode used by the DAG. This mode makes the board compatible with software written for Intel386™ CPU-based Multibus II products such as the iSBC 386/100 and 386/116/120 boards.

DAG DMA Address Generator (a custom gate array circuit)

DCE Data Communication Equipment

doubleword A 32-bit quantity of memory. The Intel486™ processor allows 32-bit doublewords to begin at any byte address, but a performance penalty is taken when a doubleword crosses the boundary between two doublewords in physical memory.

DTE Data Terminal Equipment

dual-port alias An address offset that is added to the Multibus address to give the physical address of dual-port DRAM.

dual-port lock A control bit in I/O space that locks the dual-port DRAM to the local bus.

dual-port memory Memory that can be accessed both from the local CPU and from other agents on the Multibus interface.

extended mode A mode used by the DAG. *Extended mode* is similar to *compatibility mode* except that the block size is increased to 128 Mbytes for non-paged burst transfers.

FIFO Refers to the SCSI FIFO buffers.

Floating-Point Unit (FPU) The part of the Intel486 processor which contains the floating-point registers and performs the operations required by floating-point instructions.

Gbyte 1024 Mbytes (1,073,741,824 bytes). The physical address space of the Intel486 microprocessor is 4 Gbytes.

IEEE 796 IEEE 796 Multibus I interface.

IEEE 959 IEEE 959 SBX (System Bus Expansion) interface.

Kbyte 1024 bytes

local DRAM On-board DRAM that is not configured as dual-port DRAM.

Mbyte 1024 Kbytes (1,048,576 bytes)

MPSC Intel 8274 Multiple-Protocol Serial Controller (MPSC).

Multibus interface address space An address space mapped to the Multibus interface. When the iSBC 486/12 board is in real mode, the address space between the end of DRAM and the start of EPROM is mapped to the Multibus interface.

Multibus window The Multibus "window" is a contiguous block of memory addresses in real mode that is mapped to the Multibus interface memory space. The Multibus window can start and end on any 64-Kbyte boundary in the real address memory space (0H to 0F FFFFH).

multiprocessing Two or more processors working together across the Multibus interface. Data and control information can be exchanged through dual-port memory.

one-cycle DMA mode In the one-cycle DMA transfer mode (also called single cycle), the data is transferred directly from the source to the destination without going through the internal registers in the ADMA. On the iSBC 486/12S series boards, only ADMA channel 0 (SCSI) supports one-cycle transfers.

paging In this chapter, *paging* refers to the DAG's address paging capability. During paging, the value in the appropriate page register is combined with the ADMA address to produce the physical address.

PIC Intel 82C59A Programmable Interrupt Controller (PIC).

PIT Intel 82C54 Programmable Interval Timer (PIT)

PPI Intel 82C55A Programmable Peripheral Interface (PPI).

protected virtual address mode (PVAM) Refers to the mode of addressing for both the Intel486 microprocessor and the iSBC 486/12 board. The address range of the CPU and board is 4 Gbytes in protected mode.

real address mode The default addressing mode for the x86 family of microprocessors. The Intel486 microprocessor in real mode is compatible with the Intel 8086 microprocessor.

SBX IEEE 959 SBX (System Bus Expansion) interface.

SPC Fujitsu 87033B SCSI Protocol Controller.

system memory System memory is dual-port memory that is configured to be accessible from the Multibus interface.

Tbyte 1024 Gbytes (1,099,511,627,776 bytes). The virtual address space of the Intel486 microprocessor is 1 Tbyte.

turbo Refers to a DMA transfer mode or a board register that controls the DMA transfers to the SCSI FIFOs. In turbo mode, the DMA request is asserted only after the FIFOs have reached a 128-byte threshold. The turbo mode is used together with the blast mode to control the high-speed DMA transfers to and from the SCSI FIFOs.

two-cycle DMA mode In the two-cycle DMA transfer mode, the data is transferred from the source to ADMA, then from the ADMA to the destination. On the iSBC 486/12S series boards, all four channels support two-cycle transfers.

word A 16-bit quantity of memory. The Intel486 processor allows 16-bit words to begin at any byte address, but a performance penalty is taken when a word crosses the boundary between two double words in physical memory.

Glossary of Signal Names

AUXRST* signal	auxiliary reset signal from front panel interface
BRKINT signal	break detect interrupt
BUSDRIVE1*, BUSDRIVE2* signals	Multibus interrupt drivers
BUSRQE* signal	bus request error
CLREDGE1* signal	clear edge 1 (used to clear the spare flipflop)
DMAINT signal	DMA interrupt
EXTINT signal	external interrupt from front panel interface
LOCK* signal	Multibus lock
LPT_Acknowledge* signal	line printer acknowledge
LPT_Busy signal	line printer busy
LPT_Error signal	line printer error
LPT_Strobe* signal	line printer strobe
LPTINT signal	line printer interrupt
MA<23:20> signal	Multibus address lines 23..20
MBTOUTINT signal	Multibus time out
MCS0, MCS1 signals	SBX chip select signals 0 and 1
Multibus BCLK* signal	Multibus bus clock
Multibus CCLK* signal	Multibus constant clock
Multibus INT<7:0>* signals	Multibus interrupts
NMIR signal	non-maskable interrupt request to Intel486 CPU
PER* signal	Multibus interrupt driver from on-board DRAM parity error signal
PERINT signal	on-board DRAM parity error
PFSN* signal	power fail sense from front panel interface

RSINT signal	user controlled interrupt signal from spare flipflop
SBXxIRx signals	SBX interrupt signals
SERINT signal	serial interrupt (8274 MPSC)
SETEDGE1 signal	set input signal to spare flipflop
SWRESET* signal	software reset signal
TOUTINT signal	Multibus timeout interrupt

***

Index

16 Mbyte Page register, 6-19
  I/O address, 7-10
82258 ADMA, 8-12
8274 MPSC, 7-55
82C54 PIT, 7-10
82C55A PPI, 7-17
82C59A PIC, 8-36

A

access time, EPROM, 4-11
ADMA, 8-12
  see also DAG, 8-17
  channel 2 jumper options, 8-14
  default configuration, 8-13
  General Mode Register, initialization, 8-15
  I/O addresses, 8-15
  interrupt, 8-48
  jumpers, 8-14
  programming, 8-14
ADR<23:20>* signal, 9-13
arbitration, Multibus, 7-85
auxiliary connector P2, 9-12
AUXRST* signal, 8-7, 9-13

B

baud rates, 7-16
BCLK* signal, 7-80, 9-7
bias and termination resistors, 3-20
BIST, CPU, 8-11
BLAST register, 7-52, 7-53
block diagram of board, 2-3
Board ID Register, 8-5
  I/O address, 7-9
board reset actions, 8-8
boot diskettes, SCSI, E-1
bootstrap loader, iRMX, E-2
break detect circuit, 7-68
break detect interrupt, 8-50
BRKINT signal, 7-68, 8-44, 8-50
BSL, iRMX, E-2
buffered Multibus I/O write operations, 7-85
burst/non-burst modes, 8-29
bus release mode, 7-87
bus request error interrupt, 8-49
bus request error, 8-10
BUSDRIVE1* signal, 7-30, 7-84
BUSDRIVE1*, BUSDRIVE2* signals, 8-43, 8-52
BUSDRIVE2* signal, 7-30, 7-84
BUSREQE signal, 8-10, 8-44, 8-49
BYTXFER register, 7-52, 7-54

C

cable
  printer, 9-15
  RS-232C, 9-25
  RS-422A/449, 9-27
  SCSI, 9-19
cache disable bit, CD, 8-11
cache write-through policy, 8-11
cache, CPU, 8-11
CCLK* signal, 7-80, 9-8
CD bit, CR0 Register, 8-11
clear reset flag, I/O address, 7-9
CLREDGE1* signal, 7-29, 8-44
compatibility mode, DAG, 8-29
compatibility notes, D-1
configuration priority, 4-3
connector
  J1, 9-15
  J2, 9-18
  J3, 9-21
  J4, 9-22
  J5, J6, 9-29
  P1, 9-7
  P2, 9-12
  P3, 9-14
connector, location diagram, 9-2
connector, vendor data, 9-5
CONSLOCK* signal, 9-13
control registers, SCSI, 7-51

CPU
  cache, 8-11
  resets, 8-7
  self-test, 8-11
CYCLE signal, 9-13

D

DAG, 8-17
  Control Register, 8-23
  I/O addresses, 8-20
  one-cycle modes, 8-28
  programming, 8-22
  registers, 8-18
  Test Register, 8-25
data alignment, DMA, 8-27
default
  components, B-24
  configuration, real mode, 4-3
  I/O address map, 7-6
  I/O configuration, 7-5
  system memory, 6-4
  wire list, B-22
definition file, iRMX, E-1
DMA
  data alignment, 8-27
  interrupt, 8-48
  maximum byte count, 8-27
  transfer rates, 8-26
DMA Address Generator, DAG, 8-17
DMA to SBX, I/O address, 7-9
DMA, to and from SBX, 7-77
DMAINT signal, 8-44, 8-48
DRAM
  configuration (real mode), 4-8
  ending address jumpers, real mode, 4-9
  ending address, protected mode, 5-7, 5-8
  parity error signal, PER*, 7-84
  starting and ending address, real mode, 4-8
DS1, 8-55
DS2, 8-51, 8-55
DS3, 8-9, 8-55
DS4, 7-25, 8-55
DS5, 8-50, 8-55
dual port, disabling, 6-7
Dual-Port 16 Mbyte Page Register, 6-5
dual-port alias, 6-8
dual-port lock, 6-20, 7-27
dual-port memory lock, LOCK* signal, 7-85
dual-port starting and ending address, 6-7

E

EPROM
  access time, 4-11
  configuration, real mode, 4-10
  firmware, E-1
  installation, 3-16
  jumpers, 4-12
  size, 4-10
  starting address, protected mode, 5-9
extended mode, DAG, 8-29
external interrupt, 8-48
EXTINT signal, 8-44, 8-48, 9-13

F

F1, 7-49
FIFO data, 7-54
firmware, E-1
flipflop
  optional, clearing, 7-29
  spare, 8-43, 8-51
floating point unit, 8-12
force parity error, I/O address, 7-9
front panel
  cable, parts, 9-6
  connector J3, 9-21
fuse, SCSI, 3-2

G

GP3 bit jumper, 7-48

H

hardware reset, 8-7
HIN register, 7-52, 7-54
human interface, LEDs, 8-55
HWRESET* signal driving INIT* signal, 7-81

I

I/O addresses
  ADMA, 8-15
  DAG, 8-20
  MPSC, 7-62
  PPI, 7-34
  SBX, 7-71
  SBX, swap, 7-74
  SCSI FIFO, 7-52
  special board functions, 7-9
I/O addresses map, 7-6
I/O command recovery time, SBX, 7-77

I/O, default configuration, 7-5
ID Register, Board, 8-5
INIT* signal, 7-81, 8-7, 9-8
initialization
  ADMA GMR, 8-15
  PPI, 7-36
installation, iSBC 486/12 board, 3-27
INT<7:0> signals, 8-43, 8-45
INT<7:0>* signals, 9-8
Intel services, 10-2
interrupt
  ADMA, 8-48
  break detect, 8-50
  BRKINT, 7-68
  bus request error, 8-49
  configuration, 8-37
  drivers, Multibus, 8-52
  drivers, PER*, 8-52
  external, 8-48
  EXTINT, 9-13
  jumper matrix, 8-37
  line printer, 8-49
  LPT latch clear, 7-31
  masking the SPC, 7-53
  Multibus, 7-84, 8-45
  parity error, 8-50
  PFINT*, 9-13
  PIT, 8-45
  power-fail, 8-51
  SBX signals, 9-31
  SBX, 7-79, 8-48
  serial, 8-48
  SPC, 8-48
  TOUTINT, 8-9
inverter, spare, 8-43, 8-50

J

J1 connector, 9-5, 9-15
J2 connector, 9-5, 9-18
J3 connector, 9-5, 9-21
J4 connector, 9-5, 9-22
J5 connector, 9-5, 9-29
J6 connector, 9-5, 9-29
jumpers
  E001, 8-44
  E002, 8-44
  E003, 8-44
  E004, 8-44
  E005, 8-44
jumpers (continued)
  E006, 8-44
  E007, 8-44
  E008, 8-38
  E009, 8-38
  E010, 8-9
  E011, 8-41
  E012, 8-41
  E013, 8-41
  E014, 8-41
  E015, 8-41
  E016, 8-41
  E017, 8-41
  E018, 8-41
  E019, 8-38
  E020, 8-9
  E021, 8-44
  E022, 7-33, 8-44
  E023, 8-44
  E024, 8-44
  E025, 8-39
  E026, 8-41
  E027, 8-44
  E028, 7-33, 8-44, 8-51
  E029, 8-41
  E030, 7-85
  E031, 8-44
  E032, 8-44
  E033, 8-14, 8-44, 8-48
  E034, 7-79, 8-44
  E035, 7-79, 8-44
  E036, 7-79, 8-44
  E037, 7-79, 8-44
  E038, 7-33, 8-44
  E039, 8-44
  E040, 7-85
  E041, 8-41
  E042, 8-41
  E043, 8-14, 8-41
  E044, 8-41
  E045, 8-41
  E046, 8-41
  E047, 8-41
  E048, 8-41
  E049, 8-44
  E050, 8-39
  E051, 7-33, 8-44
  E052, 7-33, 8-44
  E053, 8-44

jumpers (continued)
E054, 8-41
E055, 7-33, 8-41
E056, 8-10, 8-44
E057, 8-39
E058, 8-44
E059, 8-41, 8-42
E060, 8-39
E061, 7-24, 7-33
E062, 7-24, 7-33
E063, 8-36
E064, 8-36
E065, 7-26, 7-33
E066, 7-26, 7-33
E067, 7-26, 7-33
E068, 7-12
E069, 7-12
E070, 7-12
E071, 7-24, 7-33
E072, 7-24, 7-33
E073, 7-74
E074, 7-74
E075, 7-79, 8-14
E076, 8-14
E077, 7-79, 8-14
E078, 7-12
E079, 7-12
E080, 7-12
E081, 7-24, 7-33
E082, 7-24, 7-33
E083, 7-83
E084, 7-48, 7-83
E085, 7-79, 8-14
E086, 8-14
E087, 7-79, 8-14
E088, 7-12
E089, 7-12
E090, 7-12
E091, 4-12
E092, 4-9
E093, 4-6
E094, 4-7
E095, 5-7
E096, 5-8
E097, 6-12
E101, 4-12
E102, 4-9
E103, 4-6
E104, 4-7

jumpers (continued)
E105, 5-7
E106, 5-8
E107, 6-12
E111, 4-12
E112, 4-9
E113, 4-6
E114, 4-7
E115, 5-7
E116, 5-8
E117, 6-12
E121, 4-12
E122, 4-9
E123, 4-6
E124, 4-7
E125, 5-7
E126, 5-8
E127, 6-12
E131, 4-12
E132, 4-9
E133, 4-6
E134, 4-7
E135, 5-7
E136, 5-8
E137, 6-12
E141, 4-12
E142, 4-9
E143, 4-6
E144, 4-7
E145, 5-7
E146, 5-8
E147, 6-12
E151, 5-10
E152, 4-9
E153, 4-6
E154, 4-7
E155, 5-7
E156, 5-8
E157, 6-12
E161, 5-10
E162, 4-9
E163, 4-6
E164, 4-7
E165, 5-7
E166, 5-8
E167, 6-12
E171, 7-63, 7-65, 7-66
E172, 7-63, 7-65, 7-66
E173, 7-63, 7-65, 7-66

jumpers (continued)
E174, 7-63, 7-65, 7-66
E175, 7-63, 7-65, 7-66
E176, 7-63, 7-65, 7-66
E177, 7-63, 7-65, 7-66
E178, 7-63, 7-65, 7-66
E179, 7-63, 7-65, 7-66
E180, 7-63, 7-65, 7-68, C-2
E181, 7-63, 7-65, 7-66
E182, 7-63, 7-65, 7-66
E183, 7-63, 7-65, 7-66
E184, 7-63, 7-65, 7-66
E185, 7-63, 7-65, 7-66
E186, 7-63, 7-65, 7-66
E187, 7-63, 7-65, 7-66
E188, 7-63, 7-65, 7-66
E189, 7-63, 7-65, 7-66
E190, 7-63, 7-65, 7-68, C-2
E191, 7-63, 7-65, 7-66
E192, 7-63, 7-65, 7-66
E193, 7-63, 7-65, 7-66
E194, 7-63, 7-65, 7-66
E195, 7-63, 7-65, 7-66
E196, 7-63, 7-65, 7-66
E197, 7-63, 7-65, 7-66
E198, 7-63, 7-65, 7-66
E199, 7-63
E200, 7-67
E201, 7-63, 7-65, 7-66
E202, 7-63, 7-65, 7-66
E203, 7-63, 7-65, 7-66
E204, 7-63, 7-65, 7-66
E205, 7-63, 7-65, 7-66
E206, 7-63, 7-65, 7-66
E207, 7-63, 7-65, 7-66
E208, 7-63, 7-65, 7-66
E209, 7-63
E210, 7-67
E211, 7-48
E212, 7-48
E213, 7-48
E214, 7-48
E215, 7-63, 7-64, 7-65, 7-67
E216, 7-63, 7-64, 7-65, 7-67
E217, 7-63, 7-64, 7-65, 7-67
E218, 7-63
E219, 7-63
E220, 7-63
E221, 7-48

jumpers (continued)
E222, 7-48
E223, 7-48
E224, 7-48
E225, 7-63, 7-64, 7-65, 7-67
E226, 7-63, 7-64, 7-65, 7-67
E227, 7-63, 7-64, 7-65, 7-67
E228, 7-63
E229, 7-63
E230, 7-63
E231, 4-12
E232, 4-12
E233, 4-12
E234, 4-12
E236, 4-12
E237, 4-12
E238, 4-12
E239, 4-12
E241, 4-12
E242, 4-12
E243, 4-12
E244, 4-12
E246, 7-84, 8-43
E247, 7-84, 8-43
E248, 7-8,4 8-43
E249, 7-84, 8-43
E251, 7-84, 8-43
E252, 7-33, 7-84, 8-43
E253, 7-33, 7-84, 8-43
E254, 7-84, 8-43
E256, 7-87
E257, 7-84, 8-43
E258, 7-84, 8-43
E259, 7-84, 8-43
E261, 7-87
E262, 7-87
E263, 7-87
E264, 7-82, 7-83
E266, 7-81
E267, 7-80
E268, 7-87
E269, 7-82, 7-83
E271, 7-81
E272, 7-80

L

LED, 8-55
  DS1, 8-55
  DS2, 8-51
  DS3, 7-26
  DS4, 7-25, 7-32
  DS5, 8-50
line printer interrupt, 8-49
LOCK* signal, 7-85, 9-9
LPT Acknowledge* signal, 7-25
LPT Busy signal, 7-26
LPT Error signal, 7-26
LPT Interrupt latch clear, 7-31
LPT Interrupt, 7-24, 8-49
LPT Strobe* signal, 7-27
LPTINT signal, 8-44, 8-49

M

MA<23:20> signal, termination, 7-83
master PIC, 8-41
math coprocessor, *see* floating point unit, 8-12
maximum byte count, DMA, 8-27
MB87033B SPC, 7-44
MBINT<7:0> signals, 8-44
MBTOUTINT signal, 8-44, 8-47
MCS0 signal, 7-71, 9-30
MCS1 signal, 7-71, 9-30
MDACK* signal, 9-31
MDRQT signal, 9-31
Memory Address Page Registers, 8-24
memory module installation, 3-5
MINTR0 signal, 9-31
MINTR1 signal, 9-31
MM3 memory module connector, 9-14
mode bit I/O address, 7-9
MPSC, 7-55
  default configuration, 7-62
  default jumpers, 7-63
  I/O addresses, 7-62
  interrupt, 8-48
  programming, 7-68
  vectored mode, 8-53
Multibus
  address line termination resistors, 7-83
  address space in 3rd Gbyte page, 5-11
  arbitration priority, 7-85
  arbitration, 7-85
  BCLK* signal, 7-80
  buffered I/O write operations, 7-85
Multibus (continued)
  bus release mode, 7-87
  CCLK* signal, 7-80
  connector, 9-7
  INIT* signal, 7-81
  interrupt destinations, 8-43
  interrupt drive signals, 7-84
  interrupt driver, PER*, 8-52
  interrupt drivers, 7-30, 8-52
  interrupts, 7-84, 8-45
  LOCK* signal, 7-85
  memory space configuration, real mode, 4-13
  P1 configuration, 7-80
  request error interrupt, 8-49
  termination resistor, 3-19
  timeout, *see also* BUSREQE, 8-9, 8-10
    clearing, 7-28
    interrupt, 8-44, 8-47
  window ending address, 4-7
  window starting address, 4-6
  window, 4-5
Multibus:ADR<23:20>* signal, 9-12, 9-13
Multibus:INT<7:0>* signal, 7-84
multidrop
  RS-422A operation, C-1
multidrop serial configuration, 7-68
multiprocessing configuration, 6-21

N

NMI
  default configuration, 8-40
  mask, 7-28, 8-42
NMI mask, use of, 7-42
NMIR signal, 8-41, 8-42
NMIR, PERINT input, 8-50

O

one-cycle ADMA modes, 8-16, 8-28
one-cycle page registers, DAG, 8-18, 8-24
one-cycle transfers, 8-30
OPT0 signal, 9-31
OPT1 signal, 9-31
OR gate, spare, 8-43, 8-51

P

P1 configuration, 7-80
P1 connector, 9-5
P2 connector, 9-5, 9-12

P3 connector, 9-5, 9-14
page registers, DAG, 8-18
paged/non-paged modes, 8-29
parallel cable, parts, 9-6
parallel interface
  cable, 9-15
  connector, 9-15
  PPI, 7-17
parallel Multibus arbitration, 7-85
parity error interrupt driver, 8-52
parity error interrupt, 8-50
parity error, clearing, I/O address, 7-9
parity error, PER* signal, 7-84
PCI server, E-2
PER* signal, 7-84, 8-43, 8-52
performance, DMA transfer rate, 8-26
PERINT signal, 8-44, 8-50
PFINT signal, 8-51
PFINT* signal, 8-44, 9-13
PFSN* signal, 7-25, 9-13
PIC, 8-36
  default configuration, 8-40
  direct-vectored mode, 8-53
  programming, 8-53
PIT, 7-10
  count values, 7-16
  counter interrupts, 8-45
  programming, 7-13
power fail interrupt, PFINT*, 9-13
power failure sense signal, PFSN*, 9-13
power requirements, 1-6
power up reset, 8-7
power-fail interrupt, 8-51
Power-Fail Sense Signal (PFSN*), 7-25
PPI, 7-17
  bit set/reset, 7-38
  I/O addresses, 7-34
  initialization, 7-36
  modes, 7-34
  PB3, 7-24
  PB4, 7-25
  PB5, 7-25
  PB6, 7-26
  PB7, 7-26
  PC0, 7-27
  PC1, 7-27
  PC2, 7-28
  PC3, 7-28
  PC4, 7-29
PPI (continued)
  PC7, 7-31, 7-32
  port A, 7-20
  port B, 7-23
  port C, 7-27
  ports, default assignments, 7-19
  programming, 7-34
  stake pins, 7-33
PPI, PC<6:5>, 7-30
PPI:PB<2:0>, 7-23
printer
  application of PPI, 7-39
  cable, 9-15
  connector, 9-15
Programmable Interrupt Controller, 8-36
programming
  ADMA General Mode Register, 8-15
  ADMA, 8-14
  DAG, 8-22
  MPSC, 7-68
  PIC, 8-53
  PIT, 7-13
  PPI, 7-34
  SPC, 7-51
PVAM
  switching to PVAM, 8-4
  switching to real mode, 7-42

R

real mode
  DRAM configuration, 4-8
  EPROM configuration, 4-10
  memory map, 4-3
  Multibus memory space configuration, 4-13
  Multibus window, 4-5
  switching to PVAM, 7-42
real time clock count values, 7-17
reset, 8-7
  flag, I/O address, 7-9
  INIT* signal, 7-81
  SCZRST register, 7-54
  SWRESET* configuration, 7-82
  AUXRST* signal, 9-13
resistor packs, 3-2
  default, B-24
  RP1, 7-48
  RP2, 7-48
  RP3, 7-48
  RP4, 3-20, 7-64, 7-66, 7-67, C-2

resistor packs (continued)
RP5, 3-20, 7-64, 7-66, 7-67, C-2
RP6, 3-20, 7-64, 7-66, 7-67, C-2
RP7, 7-84
RSINT signal, 8-44, 8-51
RS-232C
cable, 9-25
connector, 9-23
DCE, MPSC channel A, 7-64
DTE, MPSC channel A, 7-64
RS-422A
bias and termination resistors, 3-20
line driver installation, 3-21
multidrop configuration, 7-68
RS-422A/449
cable, 9-27
connector, 9-24
DCE, MPSC channel A, 7-66
operation, use of DM signal, C-2

S

SBX
address swap, 7-74
board installation, 3-21
configuration, 7-71
DMA I/O address, 7-9
DMA operations, 7-77
I/O addresses, 7-71
I/O command recovery time, 7-77
interrupts, 7-79, 8-44, 8-48
option lines, 7-79
SBX option lines, 9-31
SBX1, connector J5, 9-29
SBX2, connector J6, 9-29
SBXxIRx signals, 8-48
SCSI
blast mode, 7-53
boot diskettes, E-1
cable, 9-19
cable, parts, 9-6
connector, 9-18
control registers, 7-51
control registers, I/O addresses, 7-10
FIFO data, 7-54
fuse, 3-2, 7-49
ID jumpers, 7-48
iRMX support, E-2
reset register, 7-54
SCZBSY signal, 9-13
SCSI (continued)
signals, 7-46
termination power, 7-49
termination resistor and fuse, 3-18, 7-48
turbo mode, 7-54
SCSI Peripheral Controller, SPC, 7-44
SCZBSY signal, 9-13
SCZID register, 7-52
SCZINT signal, 8-44, 8-48
SCZINTDIS register, 7-52, 7-53
SCZRST register, 7-52, 7-54
self-test, CPU, 8-11
serial
bias resistors, default, 7-64
break detect clear, I/O address, 7-9
cable, parts, 9-6
connectors, 7-62
CONSLOCK signal, 9-13
interrupt, 8-48
MPSC, 7-55
Multibus arbitration, 7-85
SERINT signal, 8-44
SETEDGE1 signal, 8-41
slave PIC, 8-42
software reset, 8-7
configuration, 7-82
I/O address, 7-9
spare
flipflop, 8-51
gates, 8-43
inverter, 8-50
OR gate, 8-51
SPC, 7-44
interrupt mask, 7-53
interrupt, 8-48
programming, 7-51
registers, 7-49
reset, 7-54
specifications
clock speeds, 1-4
DMA transfer rate, 8-26
DRAM, 1-3, 1-4
environmental, 1-5
EPROM, 1-4
performance, 1-3
physical, 1-5
power, 1-6
SCSI, 1-3
SWRESET* signal, 7-82

system jobs, PCI, E-2
system memory configuration priority, 6-5
system memory, 6-4

T

TDMA* signal, 9-31
terminate DMA signal, SBX, 9-31
termination and bias resistors, 3-2
termination power, SCSI, 7-49
termination resistors, SCSI, 7-48
Test Register, DAG, 8-25
time-out interrupt, 8-47
time-out, Multibus, 8-9
time-out, Multibus, clearing, 7-28
TOUTINT signal, 8-9
TURBO register, 7-52, 7-54
two-cycle ADMA modes, 8-16
two-cycle page registers, DAG, 8-18
two-cycle transfers, 8-33

U

U14 (3487 driver), 7-64, 7-66, 7-67
unpacking and inspection, 3-4
user-defined code, 7-23

V

vectored mode
  MPSC, 8-53
  PIC, 8-53

W

wire list, default, B-22

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, 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, refer to 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 at 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_____PHONE ( )_____

CITY_____STATE_____ZIP CODE_____

(COUNTRY)_____

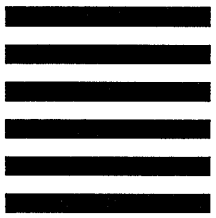
Please check here if you require a written reply. ❑

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.

If you are in the United States, use the preprinted address provided on this form to return your comments. No postage is required. If you are not in the United States, return your comments to the Intel sales office in your country. For your convenience, international sales office addresses are printed on the last page of this document.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



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

POSTAGE WILL BE PAID BY ADDRESSEE

ICD TECHNICAL PUBLICATIONS, HF3-72
INTEL CORPORATION
5200 NE ELAM YOUNG PKWY
HILLSBORO OR 97124-9978

International Sales Offices

AUSTRALIA
Intel Australia Pty. Ltd.
Spectrum Building
200 Pacific Hwy., Level 6
Crows Nest, NSE, 2065

BRAZIL
Intel Semicondutores do Brazil LTDA
Av. Paulista, 1159-CJS 404/405
01311-Sao Paulo - S.P.

CANADA
Intel Semiconductor of Canada, Ltd.
4585 Canada Way, Suite 202
Burnaby V5G 4L6
British Columbia

Intel Semiconductor of Canada, Ltd.
2650 Queensview Drive
Suite 250
Ottawa K2B 8H6
Ontario

Intel Semiconductor of Canada, Ltd.
190 Attwell Drive
Suite 500
Rexdale M9W 6H8
Ontario

Intel Semiconductor of Canada, Ltd.
620 St. Jean Boulevard
Pointe Claire H9R 3K2
Quebec

CHINA/HONG KONG
Intel PRC Corporation
15/F, Office 1, Citic Bldg.
Jian Guo Men Wai Street
Beijing, PRC

Intel Semiconductor Ltd.
10/F East Tower
Bond Center
Queensway, Central
Hong Kong

DENMARK
Intel Denmark A/S
Glentevej 61, 3rd Floor
2400 Copenhagen NV

FINLAND
Intel Finland OY
Ruosilantie 2
00390 Helsinki

FRANCE
Intel Corporation S.A.R.L.
1, Rue Edison-BP 303
78054 St. Quentin-en-Yvelines
Cedex

GERMANY
Intel Semiconductor GmbH
Dornacher Strasse 1
8016 Feldkirchen bei Muenchen

Intel Semiconductor GmbH
Hohenzollern Strasse 5
3000 Hannover 1

Intel Semiconductor GmbH
Abraham Lincoln Strasse 16-18
6200 Wiesbaden

Intel Semiconductor GmbH
Zettachring 10A
7000 Stuttgart 80

INDIA
Intel Asia Electronics, Inc.
4/2, Samrah Plaza
St. Mark's Road
Bangalore 560001

ISRAEL
Intel Semiconductor Ltd.
Atidim Industrial Park-Neve Sharet
P.O. Box 43202
Tel-Aviv 61430

(continued)

ITALY
Intel Corporation Italia S.p.A.
Milanofiori Palazzo E
20090 Assago
Milano

JAPAN
Intel Japan K.K.
5-6 Tokodai, Tsukuba-shi
Ibaraki, 300-26

Intel Japan K.K.
Daiichi Mitsugi Bldg.
1-8889 Fuchu-cho
Fuchu-shi, Tokyo 183

Intel Japan K.K.
Bldg. Kumagaya
2-69 Hon-cho
Kumagaya-shi, Saitama 360

Intel Japan K.K.
Mitsui-Seimei Musashi-kosugi Bldg.
915 Shinmaruko, Nakahara-ku
Kawasaki-shi, Kanagawa 211

Intel Japan K.K.
Nihon Seimei Atsugi Bldg.
1-2-1 Asahi-machi
Atsugi-shi, Kanagawa 243

Intel Japan K.K.
Ryokuchi-Eki Bldg.
2-4-1 Terauchi
Toyonaka-shi, Osaka 560

Intel Japan K.K.
Shinmaru Bldg.
1-5-1 Marunouchi
Chiyoda-ku, Tokyo 100

Intel Japan K.K.
Green Bldg.
1-16-20 Nishiki
Naka-ku, Nagoya-shi
Aichi 450

KOREA
Intel Korea, Ltd.
16th Floor, Life Bldg.
61 Yoido-Dong, Youngdeungpo-Ku
Seoul 150-010

NETHERLANDS
Intel Semiconductor B.V.
Postbus 84130
3099 CC Rotterdam

NORWAY
Intel Norway A/S
Hvamveien 4-PO Box 92
2013 Skjetten

SINGAPORE
Intel Singapore Technology, Ltd.
101 Thomson Road #21-05/06
United Square
Singapore 1130

SPAIN
Intel Iberia S.A.
Zurbaran, 28
28010 Madrid

SWEDEN
Intel Sweden A.B.
Dalvagen 24
171 36 Solna

SWITZERLAND
Intel Semiconductor A.G.
Zuerichstrasse
8185 Winkel-Rueti bei Zuerich

TAIWAN
Intel Technology Far East Ltd.
8th Floor, No. 205
Bank Tower Bldg.
Tung Hua N. Road
Taipei

UNITED KINGDOM
Intel Corporation (U.K.) Ltd.
Pipers Way
Swindon, Wiltshire SN3 1RJ