ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Pentek 4270* at our website: *Click HERE*

OPERATING MANUAL

MODEL 4270

Quad TMS320C40 Digital Signal Processor
VMEbus Board / MIX Module

Pentek, Inc.
One Park Way
Upper Saddle River, NJ 07458
(201) 818-5900
http://www.pentek.com/

Copyright © 1993-1999

Manual Part No: 800.42700 Rev: G - June 28, 1999

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Pentek Model 4270 Operating Manual Revision History

Date	Rev	Applicable Serial #'s	Comments
7/9 - 12/3/93	Preliminary	9320XX - Forward	Contact Factory for details
02/11/94	A	9320XX - Forward	Finish update of tables. Flesh out MIX Mastering Section. Add text and additional tables to VME Section. Rearrange Appendices. Add more Programming Examples to Appendix A.
02/25/94	A.1	9320XX - Forward	Correct error in 'C40 Command Register tables, pgs. 46 & 47 (DMA Path Lock & Unlock register addresses reversed). Correct same error in text & tables, pgs. 43, 51, 55, and 63. Add section on installing MIX stack in VME cage to Appendix B
07/07/94	B	9320XX - Forward	Renamed VME Bank Size Jumper Block (JB11) to Window Size, and made some changes to table describing it. Enable DMA after access, not before. Added table of AM codes. Listed Pentek part# for ordering Comm Port cables. Added tables for valid VMEbus A24 base addressing as a function of SRAM window size. Added section on Global SRAM usage and transfer timing table. Numbered all Figures and Tables. Font size reduced.
10/03/94	C	9320000 - Forward	Correct error in MIX Interrupt Source ID Table (3-10). Change description of bits described in this table from Baseboard Interrupts to Module Interrupts. Clarify difference between these interrupts & MIX Master Interrupt. Clarify interrupt source identification in Interrupt Handling Programming Example (Section A.1). Change all references to DMA Enable to Path Lock, to more correctly describe function. Re-format Section 3.1.2 ('C40 Status Register Bit Description) with sub-sections.
03/10/95	D	9320000 - Forward	Changed name of Path Lock feature to Global bus Lock, to more accurately characterize its function. Add Chapter 4 (Working with the Model 4270), and moved appropriate material from Chapter 3 into the new chapter. Add new section on use of Global bus lock, indicating that it need not be invoked to perform DMA. Separate MIX Slave Memory Maps for 4200 series, 4283 and 4284 Baseboards. Corrected last 3 entries (Ext. Local RAM Banks & Ext. Local Bus) in Local 'C40 Memory Map (there was one 'F' too many in each entry). Table 2-8, TBC can interrupt VME on IRQ1 or IRQ7, not IRQ3.
10/11/96	E	9610060 - Forward	Identify EEPROM type in Specs. Pinout of S1 changed because of change to silk screen markings. Figure 2-1a added and Tables 2-1 & 2-2 changed for same reason. Identify mating connectors for Timer-I/O connector in Sec. 2.5. Add Sections 2.8 (MIX Stack Assembly), 2.9 (Card Cage Installation) & 2.10 (POST). Table 3-1 - Indicate that RAM Bank 2 is optional. Correct D7 setting in Table 3-52 (1, not 0 . . . Hex setting in table is correct). Correct MIX_base addresses for 4202 & 4284 in Sec. 4.6.1. Add a table giving initial access overheads in Sec. 4.8. Add info on Global Bus arbitration priorities to Sec. 4.8. Sec. A.1: add a global interrupt disable statement to IntInit2 and IntInit3. Update MIX assembly Appendix to Rev. F. Add Tables & Figures to Table of Contents.
02/13/97	F	9710060 - Forward	Add Option 060 to (60 MHz 'C40) to specs. NMI timeout interrupts disabled under Option 060. Add Memory Map for use on 4285 Baseboard Adaptor. Add timing tables for Option 060. Change MIX Appendix designations from C, D & E to X, Y & Z.
11/24/97	F.1	9710060 - Forward	Sec. 2.5.1 - LEDs are ON when TCLK0 pin is at logic '0', OFF when TCLK0 is a logic '1'. Provide references to the TCLK programming information in the 'C40 Manual (Sec. 9.13) in Secs. 2.5.1 & 2.5.2. Sec. 4.2 - 4270 has NO power-up reset circuit. Clarify exactly what gets reset by all reset sources - this affects the following Sections: 2.5.2, 3.4.1.3, 3.5.2.3, and 4.2.
01/08/99	F.2	9710060 - Forward	Removed MIX Appendices (Appendices X through Z). Sec. 2.3 - corrected the model number of the Comm Port Cable. Previously called out Model 2014, changed to Model 2104.
06/28/99	G	9710060 - Forward	Updated Boot Code Listing in Appendix B. In Table 4-3, added 'y' to MIX/VME address colums to designate processor #. Add material about modifying 4270 board for 4283 Global SRAM access from Addendum (800.42702). Merged Sec. 3.1.1 w/3.1.2, & (former) 3.1.3 w/3.1.4 as new 3.1.2. Corrections to last paragraph of Sec. 4.8 & Table 4-31, such that they agree w/ each other

References made in this manual to the **Texas Instruments TMS320C4X** User's Guide are made specifically to Revision C of that publication, dated August 1993.

WARRANTY

Pentek warrants that all products manufactured by Pentek conform to published Pentek specifications and are free from defects in materials and workmanship for a period of one year from the date of delivery when used under normal operating conditions and within the service conditions for which they were furnished.

The obligation of Pentek arising from a warranty claim shall be limited to repairing or at its option, replacing without charge, any product which in Pentek's sole opinion proves to be defective within the scope of the warranty.

Pentek must be notified in writing of the defect or nonconformity within the warranty period and the affected product returned to Pentek within thirty days after discovery of such defect or nonconformity.

Buyer shall prepay shipping charges, taxes, duties and insurance for products returned to Pentek for warranty service. Pentek shall pay for the return of products to buyer except for products returned from another country.

Pentek shall have no responsibility for any defect or damage caused by improper installation, unauthorized modification, misuse, neglect, inadequate maintenance, accident or for any product which has been repaired or altered by anyone other than Pentek or its authorized representatives.

The warranty described above is buyer's sole and exclusive remedy and no other warranty, whether written or oral, is expressed or implied. Pentek specifically disclaims fitness for a particular purpose. Under no circumstances shall Pentek be liable for any direct, indirect, special, incidental or consequential damages, expenses, losses or delays (including loss of profits) based on contract, tort, or any other legal theory.

Printed in the United States of America. All rights reserved. Contents of this publication may not be reproduced in any form without written permission.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

Page

Chapter 1: General Description

1.1 Introduction 11
1.2 Features 11
1.2.1 Standalone VMEbus Board or MIX Module Operation 11
1.2.2 MIX Bus Master Operation 12
1.2.3 MIX Bus Slave Operation 12
1.2.4 Global SRAM 12
1.2.5 Local SRAM 13
1.2.6 Local EPROM 13
1.2.7 J-TAG Support 13
1.2.8 Communication Ports 13
1.3 Specifications 14
1.4 Block Diagram 16
Figure 1-1: Block Diagram - Pentek Model 4270 16

Chapter 2: Installation and Connections

2.1 Inspection 17
2.2 Jumper Settings 17
Figure 2-1: Jumper Block Pin Numbering 17
Figure 2-2: Model 4270 VME board drawing showing jumper block locations 18
Figure 2-3: Model 4270 MIX board drawing showing jumper block locations 19
2.2.1 A16 Base Address Jumpers 17
Figure 2-4: Pin Numbering, Jumper Block S1 20
Table 2-1: Jumper Block S1 - A16 Base Address 20
Table 2-2: Jumper Block S1 - A16 Base Address 21
2.2.2 TMS ID Jumpers 22
Table 2-3: TMS ID Jumpers 22
2.2.3 'C40 Boot Load Address 22
Table 2-4: Boot Load Address Jumper Blocks 23
Table 2-5: 'C40 Boot Code Address Jumper Settings 23
2.2.4 Global Memory Window Size Jumpers 24
Table 2-6: A24 Global SRAM Window Size Jumpers 24
Table 2-7: MIX Bus Global SRAM Window Size Jumpers 25
2.2.5 VME Interrupt Sources 25
Table 2-8: VME Interrupt Source Selection 26
2.2.6 VME IRQ Level 26
Table 2-9: VME IRQ Level 26
2.2.7 MIX IRQ/DRQ Select Jumper 27
2.2.8 Test Bus Controller Jumpers 27
2.2.9 Other Jumper Blocks 27

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents (continued)

Page

Chapter 2: Installation and Connections (continued)

2.3 Front Panel Comm Port Connectors ... 27
Figure 2-5: Model 4270 - Front Panel ... 28
Figure 2-6: Comm Port Cable ... 28
2.4 Comm Port Configuration ... 28
Table 2-10: Comm Port Connectivity ... 29
Figure 2-7: Pentek Model 4270 Comm Port Allocation ... 30
2.5 Front Panel Timer I/O Pins and LED Indicators ... 31
Figure 2-8: Timer Pins and LED Indicators ... 31
2.5.1 Front Panel LED Indicators ... 31
2.5.2 Front Panel TCLK1 Timer Pins ... 32
2.5.3 Front Panel Reset Pins ... 33
2.6 XDS Emulator Connector ... 33
Figure 2-9: XDS Emulator Connector ... 33
2.7 Disassembly and Assembly Procedures ... 33
2.7.1 Naming Conventions ... 34
2.7.2 Tools Required ... 34
2.7.3 Removing the Front Panel ... 35
2.7.4 Separating the Printed Circuit Assemblies ... 35
2.7.5 Spacers and Insulators ... 36
Figure 2-10: Spacers and Insulators ... 36
2.7.6 The Model 4270 Accessory Kit ... 36
2.7.7 Installing (or Removing) the XDS Emulator Cable Assembly ... 37
Figure 2-11: XDS Emulator Cable Assembly ... 37
2.7.8 Preparing to Attach the Printed Circuit Assemblies ... 38
Figure 2-12: VME Board - front of solder side - top view ... 38
Figure 2-13: VME Board - front of component side - front view ... 39
2.7.9 Attaching the Printed Circuit Assemblies ... 39
Figure 2-14 Aligning the Docking Connectors ... 40
2.7.10 Replacing the Front Panel ... 41
2.8 Installing the MIX Module on a MIX Baseboard ... 41
2.9 Installing the Assembly in the VMEbus Card Cage ... 41
2.10 Power-Up Self-Test ... 42

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents (continued)

Page

Chapter 3: Memory Maps and Operating Procedures

3.1 Local Memory Map 43
Table 3-1: Model 4270 - Local Memory Map 43
Table 3-2: Model 4270 - Internal Peripherals 43
3.1.1 TMS320C40 Status Registers 44
Table 3-4: TMS320C40 Status Registers 45
3.1.1.1 User Jumper Status 44
3.1.1.2 TMS ID 44
Table 3-3: TMS ID Bits 44
3.1.1.3 Processor Interrupt to VMEbus 46
Table 3-5: VME Clear Interrupt Registers 46
3.1.1.4 Processor Interrupt to MIX Bus 46
Table 3-6: MIX Clear Interrupt Registers 46
3.1.1.5 Interrupts to Other Processors 46
Table 3-7: Outgoing Inter-Processor Interrupt Bits 46
Table 3-8: Processor Set Interrupt Registers 47
Table 3-9: Processor Clear Interrupt Registers 47
3.1.1.6 MIX Module Interrupt 47
Table 3-10: MIX Interrupt Source ID 47
3.1.1.7 Global Bus Lock 48
Table 3-11: Global Bus Lock Bits 48
3.1.1.8 Interrupt from VMEbus 48
3.1.1.9 Interrupt from MIX Bus Master 48
3.1.1.10 Interrupts from Other Processors 49
Table 3-12: Incoming Inter-Processor Interrupt Bits 49
Table 3-13: Processor Set Interrupt Registers 49
Table 3-14: Processor Clear Interrupt Registers 49
3.1.2 TMS320C40 Command Registers - Processors A, B, C and D 49
Table 3-15: TMS320C40 Command Registers 50
3.1.2.1 Clear Interrupt Registers 51
Table 3-16: Processor Clear Interrupt Registers 51
Table 3-17: Status Register - Processor Interrupt Bits 51
Table 3-18: Processor Set Interrupt Registers 51
Table 3-19: Bus Master Set Processor Interrupt Registers 52
3.1.2.2 Set Interrupt Registers 52
Table 3-20: Processor Set Interrupt Registers 52
Table 3-21: Status Register - Processor Interrupt Bits 52
Table 3-22: Processor Clear Interrupt Registers 52
Table 3-23: Bus Master Clear Processor Interrupt Registers 53
3.1.2.3 Global Bus Lock Registers 53
Table 3-24: Global Bus Lock Registers 53
3.1.2.4 Non-Maskable Interrupt (NMI) Registers 53
Table 3-25: Non-Maskable Interrupt Registers 54
3.1.3 Internal 'C40 MIX Bus Control Registers 54
Table 3-26: Model 4270 - MIX Bus Control Registers 54

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents (continued)

Page

Chapter 3: Memory Maps and Operating Procedures (continued)

3.2 Global Memory Map ... 55
Table 3-27: Global Bus and SRAM Addressing ... 55
3.3 MIX Bus Slave Interface Memory Maps ... 56
3.3.1 Model 4270 Address Map for the 4200/01 Series Baseboards ... 57
Table 3-28: Memory Map for Use with Models 4200/01(A) MIX Baseboards ... 58
3.3.2 Model 4270 Address Map for the Model 4202 Baseboard ... 59
Table 3-29: Memory Map for Use with Model 4202 MIX Baseboard ... 59
3.3.3 Model 4270 Address Map for the Model 4283 Baseboard ... 60
3.3.3.1 Module Selection ... 60
Table 3-30: Module Selection - Model 4283 Page Register ... 60
3.3.3.2 Resource Selection Within the Module ... 61
Table 3-31: Memory Map for Use with Model 4283 MIX Baseboard. 61
3.3.4 Model 4270 Address Map for the Model 4284 Baseboard ... 62
Table 3-32: Memory Map for Use with Model 4284 MIX Baseboard ... 62
3.3.5 Model 4270 Address Map for the Model 4285 MIX Baseboard Adapter ... 63
Table 3-33: Memory Map for Use with Model 4285 MIX Baseboard ... 64
3.4 MIX Bus Slave Register Descriptions ... 64
3.4.1 MIX 'C40 Reset Control Register ... 64
Table 3-34: MIX 'C40 Reset Control Register ... 65
3.4.1.1 MIX Global Bus Lock ... 65
Table 3-35: Global Bus Lock Registers ... 65
3.4.1.2 MIX Broadcast Mode Enable ... 65
3.4.1.3 Processor Reset ... 66
Table 3-36: Processor Reset Bits ... 66
Table 3-37: Reset Vector Select Bits ... 66
3.4.2 MIX Interrupt and Bus Lock Status Register ... 66
Table 3-38: MIX Interrupt and Bus Lock Status Register ... 67
3.4.2.1 Global Bus Locks ... 67
3.4.2.2 Interrupt Source Identification ... 67
3.4.3 MIX Command Registers ... 68
3.4.3.1 MIX Set Interrupt Registers ... 68
Table 3-39: MIX Set Interrupt Registers ... 68
3.4.3.2 MIX Clear Interrupt Registers ... 68
Table 3-40: MIX Clear Interrupt Registers ... 68

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents (continued)

Page

Chapter 3: Memory Maps and Operating Procedures (continued)

3.5 VMEbus Slave Interface Memory Map 68
Table 3-41: VMEbus Slave Memory Map 69
3.5.1 VMEbus Address Modifiers 69
Table 3-42: Address Modifier Codes 70
3.5.2 VME 'C40 Reset Control Register 70
Table 3-43: VME 'C40 Reset Control Register 70
3.5.2.1 VME Global Bus Lock 70
Table 3-44: Global Bus Lock Registers 71
3.5.2.2 VME Broadcast Mode Enable 71
3.5.2.3 Processor Reset 71
Table 3-45: Processor Reset Bits 72
Table 3-46: Reset Vector Select Bits 72
3.5.2.4 VME Test Bus Controller 72
3.5.3 VME Address Mapping and Control Register 73
Table 3-47: VME Address Mapping and Control Register 73
3.5.3.1 VME Base Address Selection 73
Table 3-48: A24 SRAM Base Address - Hex Weights of Address Bits 73
Table 3-49: Address Bit Setting for 1 MByte SRAM Window 74
Table 3-50: Address Bit Setting for 2 MByte SRAM Window 75
Table 3-51: Address Bit Setting for 4 MByte SRAM Window 75
Table 3-52: Address Bit Setting for 8 MByte SRAM Window 75
Table 3-53: Address Bit Setting for 16 MByte SRAM Window 75
Table 3-54: A32 Global SRAM Base Address 76
3.5.3.2 Paged Addressing Mode 76
3.5.3.3 Word Swap Mode 76
3.5.4 Address Page Register 77
Table 3-55: VME Address Page Register 77
3.5.5 VME Interrupt and Bus Lock Status Register 78
Table 3-56: VME Interrupt and Bus Lock Status Register 78
3.5.5.1 TBC Ready 78
3.5.5.2 Global Bus Locks 78
3.5.5.3 Interrupt Source Identification 79
Table 3-57: VME Interrupt ID Bits 79
3.5.6 VME Command Registers 79
3.5.6.1 VME Set Interrupt Registers 79
Table 3-58: VME Set Interrupt Registers 79
3.5.6.2 VME Clear Interrupt Registers 79
Table 3-59: VME Clear Interrupt Registers 79
3.5.7 VME IACK Vector Address Register 80
Table 3-60: VME IACK Vector Address Register 80
3.5.8 Test Bus Controller Registers 80

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents (continued)

Page

Chapter 4: Working with the Model 4270

4.1 Introduction 81
4.2 Resetting and Booting the Model 4270's Processors 81
Table 4-1: Boot Address Jumper Blocks 81
Table 4-2: 'C40 Boot Code Address Jumper Settings 81
Table 4-3 SRAM Locations of Pointers to EPROM Functions 82
Table 4-4: Register States After Boot from Factory EPROM 83
4.2.1 Boot Loading User Code 84
4.3 Downloading Programs to the Model 4270 84
4.3.1 Downloading Programs with SwiftNet 85
4.3.2 Downloading SPOX Programs 85
4.3.3 Downloading Programs with the XDS-510 Emulator 85
4.4 Working with Interrupts on the Model 4270 86
4.4.1 The 4270 as a VMEbus Interrupter 86
Table 4-5: VME Interrupt Source Selection 86
Table 4-6: VME IRQ Level 87
Table 4-7: VME Interrupt ID Bits 88
Table 4-8: VME Clear Interrupt Registers 88
4.4.2 Interrupting 'C40s on the 4270 from VME 88
Table 4-9: VME Set Interrupt Registers 88
4.4.3 The 4270 as a MIX Bus Interrupter 89
Table 4-10: MIX Interrupt ID Bits 89
Table 4-11: MIX Clear Interrupt Registers 89
4.4.4 Interrupting 'C40s on the 4270 from MIX 90
Table 4-12: MIX Set Interrupt Registers 90
4.4.5 Inter-Processor Interrupts on the 4270 91
Table 4-13: Processor Set Interrupt Registers 91
Table 4-14: Outgoing Inter-Processor Interrupt Status Bits 91
Table 4-15: Status Register - Inter-Processor Interrupt Bits 91
Table 4-16: Outgoing Inter-Processor Clear Interrupt Registers 91
4.5 Using the 4270 with a Model 4283 MIX Baseboard 92
Figure 4-1: Jumper Locations After Modification 92
4.6 The Model 4270 as a Bus Master 94
4.6.1 Mastering the MIX Bus 94
4.6.2 Mastering the VMEbus 96
Table 4-17: MIX Bus Byte Enables and Data Bus Usage 97
4.7 Using the Model 4270's Global Bus Lock Feature 98
4.7.1 Global Bus Locked by External Devices 99
4.7.2 Global Bus Locked by 'C40s on the 4270 100

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

Page

Chapter 4: Working with the Model 4270 (continued)

4.8 Using the Global Memory 101
Table 4-18: Transfer Times - Parallel Instruction Mode - 50 MHz 'C40 103
Table 4-19: Transfer Times - 'C40 DMA Mode - 50 MHz 'C40 103
Table 4-20: Transfer Times - Parallel Instructions w/ Bus Lock - 50 MHz 'C40 103
Table 4-21: Transfer Times - 'C40 DMA w/ Bus Lock - 50 MHz 'C40 103
Table 4-22: Transfer Times - Parallel Instruction Mode - 60 MHz 'C40 104
Table 4-23: Transfer Times - 'C40 DMA Mode - 60 MHz 'C40 104
Table 4-24: Transfer Times - Parallel Instructions w/ Bus Lock - 60 MHz 'C40 104
Table 4-25: Transfer Times - 'C40 DMA w/ Bus Lock - 60 MHz 'C40 104
Table 4-26: Transfer Times - Parallel Instruction Mode - 40 MHz 'C40 105
Table 4-27: Transfer Times - 'C40 DMA Mode - 40 MHz 'C40 105
Table 4-28: Transfer Times - Parallel Instructions w/ Bus Lock - 40 MHz 'C40 105
Table 4-29: Transfer Times - 'C40 DMA w/ Bus Lock - 40 MHz 'C40 105
Table 4-30: Global Bus Access for First Access 106
Table 4-31: VMEbus Timing 107

Appendix A: Programming Examples

A.1 Servicing MIX Module Interrupts A-1
A.2 Slo_Blnk.c A-2
A.3 p4243.c A-4
A.4 p4271.c A-6
A.5 mod.h A-11
A.6 Timer Routines A-15
A.7 Using Timer Interrupts to Avoid Bus Timeouts A-16
A.8 Emulator Support Files A-19

Appendix B: Boot EPROM Source Code Listing

B.1 Boot EPROM Source Code Listing B-1

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page is intentionally blank

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 1: General Description

1.1 Introduction

The Pentek Model 4270 incorporates four Texas Instruments' TMS320C40s in a standard 6U single-slot VMEbus board with a complete VMEbus slave interface.

Since it is compatible with Pentek's extensive family of MIX expansion modules, you can also use the Model 4270 as a quad 'C40 co-processor for MIX-based subsystems. Its MIX bus master and slave capabilities support a host of real-time DSP and data acquisition applications.

With 4 MBytes of Local SRAM and 4 MBytes of shared Global SRAM, the Model 4270 presents a powerful yet simple programming model, to speed development time and boost real-time operation.

1.2 Features

1.2.1 Standalone VMEbus Board or MIX Module Operation

Use the Model 4270 as a standalone VMEbus board, a MIX bus expansion module, or both!

As a standalone VME board, the Model 4270 occupies a single card cage slot. It presents a slave interface to the VMEbus, through which all four 1 MByte Global SRAMs can be accessed. This permits easy movement of programs and data by a VME host processor. The unit can also generate VMEbus interrupts, vital for many real-time applications.

As a MIX expansion module, the Model 4270 represents Pentek's first full-depth MIX module. Unlike standard MIX modules, these new full-depth modules utilize the full card cage depth, from the front panel to the backplane, accommodating more circuitry within the module.

The Model 4270 attaches to any of Pentek's MIX Baseboards, using standard MIX connectors, and occupies its own single slot in the VME card cage. In these MIX subsystems, all four 1 MByte Global SRAMs are directly mapped to MIX bus address space.

Some applications may best be carried out with dual bus access to the Shared Global SRAM - both from the VMEbus and the MIX bus. For example, analog input data may be gathered from a high-speed A/D converter MIX module for FFT processing by the 'C40s. Results are then placed in the Shared Global SRAM, where they may be read by a VMEbus CPU.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.2 Features (continued)

1.2.2 MIX Bus Master Operation

Like Pentek's single and dual 'C40 MIX modules (Models 4254 and 4257, respectively), the Model 4270 is capable of mastering data transfers on the MIX bus. Any of Pentek's newer MIX Baseboards (Models 4200, 4201, 4202, 4284, and 4285 Option 031) support these MIX bus externally mastered transfers.

With this bus master capability, each 'C40 has direct access to Pentek's extensive variety of analog and digital I/O MIX modules. With multiple 4270s in the MIX subsystem, access to any Shared Global SRAM in another module is also supported.

1.2.3 MIX Bus Slave Operation

The Model 4270 operates as a MIX slave on all Pentek Baseboards, including the popular Model 4283 'C30 processor board. Program and data transfers, to and from the Shared Global SRAMs, over the MIX bus, are initiated by the Baseboard processor. Full MIX bus support of mailbox registers and interrupts is included.

1.2.4 Global SRAM

A total of 4 MBytes of shared Global SRAM is accessible by all four 'C40s, by MIX bus Masters and by VMEbus Masters, simply by addressing the space allocated for the desired SRAM. This powerful resource simplifies nearly all applications by minimizing data moves required in less powerful architectures.

Zero-wait access is provided between each 'C40 and its own Global SRAM, and all four 'C40s can execute from their own Global SRAMs simultaneously. Both Local and Global 'C40 busses can operate at full speed with zero-wait performance.

When a 'C40 addresses one of the other three Global SRAMs, a single wait state is generated.

The Model 4270 can be programmed such that all four shared Global SRAMs can be written to simultaneously by a VMEbus or MIX bus Master. This provides fast downloading of data and programs to these memories using a single write sequence instead of four.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.2 **Features** (continued)

1.2.5 Local SRAM

A private 1 MByte Local SRAM for each 'C40 maximizes the use of its dual bus architecture, and its ability to conduct data and program cycles in parallel on the two busses. These zero-wait state SRAMs are ideal for storing program code or private tables, while the Global bus processes the data.

1.2.6 Local EPROM

In order to support non-volatile storage of program or data, each 'C40 is equipped with a user-programmable 32 kByte EPROM memory on its Local bus. Ideal for embedded standalone systems that self-boot from power up, these memories can also be used for system firmware such as popular subroutines or development system monitor support.

1.2.7 J-TAG Support

J-TAG Emulator Support is also included in the 4270. Two methods are provided for accessing the J-TAG scan path feature of all four 'C40s. First, a J-TAG interface circuit is mapped into the VMEbus slave address space, allowing access by any VMEbus master. Secondly, a standard J-TAG connector supports the Texas Instruments XDS-510 Emulator.

1.2.8 Communication Ports

Three of the six 20 MByte/sec Communication Ports on each 'C40 are connected to the other three 'C40s. The other three Comm Ports are brought out to convenient front panel connectors.

These connectors use standard 0.050 inch ribbon cable and readily-available connectors of the type manufactured by 3M Corporation. They are compatible with all of Pentek's Comm Port Devices, including the following Models: 4254, 4257, 4260, 4265, 4269, 4274, 4284, 4285, 4288, 6099A, 6102, 6106, 6109, 6110, 6223, 6504, 6505, 6508, 6509, 6510, 6514, 6516, 6532 and 7110.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3 Specifications

Processors (4)

Type:	Texas Instruments' TMS320C40
Clock	
Standard:	40 MHz
Option 015:	50 MHz
Option 060:	60 MHz

Local SRAMs (4)

Memory Size:	256k x 32, for each 'C40
Speed:	0 wait states
Access:	TMS320C40 Local bus

Global SRAMs (4)

Memory Size:	256k x 32 (1 MByte)
Speed:	0 'C40 wait state (after arbitration) for processor's own Global SRAM, 1 'C40 wait state (after arbitration) for another processor's Global SRAM
Access:	TMS320C40 Global bus, VMEbus and MIX bus

EPROMs (4)

Memory Size:	32k x 8, for each 'C40
Type:	Waferscale WS57C256F-70T or equivalent
Speed:	3 wait states
Access:	TMS320C40 Local bus

Front Panel Comm Port Connectors

Number:	12 total, 3 per 'C40
Connector Type:	16-pin, 3M 50116-B000

Miscellaneous Front Panel Connectors & Indicators

Emulation:	Texas Instruments J-TAG XDS Support (Option 005)
Misc.:	Two 6-pin headers, each with one (common) $\overline{\text{Reset}}$ Input for two 'C40s, and two 'C40 TCLK1 I/O connections
Indicators:	1 LED per 'C40 (four total) driven by TCLK0 output

VMEbus Slave Interface

Memory Map:	Global SRAM mapped as a region in VME A24 or A32 address space
Control:	Memory Mapped registers in A16 space for run/halt, status, and interrupt control

MIX Bus Master/Slave Interface

Memory Map:	Global SRAM mapped as a region in MIX address space
Control:	Memory Mapped registers for run/halt, status, and interrupt control

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.3 Specifications (continued)

Power Requirements

VMEbus:	4.5 A typ. @ +5V_{DC}

Dimensions

Height:	9.20"
Depth:	6.30"
Width:	0.80"

Ordering Info

Model 4270:	40 MHz 'C40's w/ 4x 1 MB Local SRAM, 4x 1 MB Global SRAM
Model 4270-015:	50 MHz 'C40's w/ 4x 1 MB Local SRAM, 4x 1 MB Global SRAM
Model 4270-005:	Front Panel XDS Emulator Connector Installed
Model 4270-060:	60 MHz 'C40's w/ 4x 1 MB Local SRAM, 4x 1 MB Global SRAM

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.4 Block Diagram

The block diagram of the Pentek Model 4270 Quad TMS320C40 Digital Signal Processor VME Board / MIX Module is presented in Figure 1-1, below.

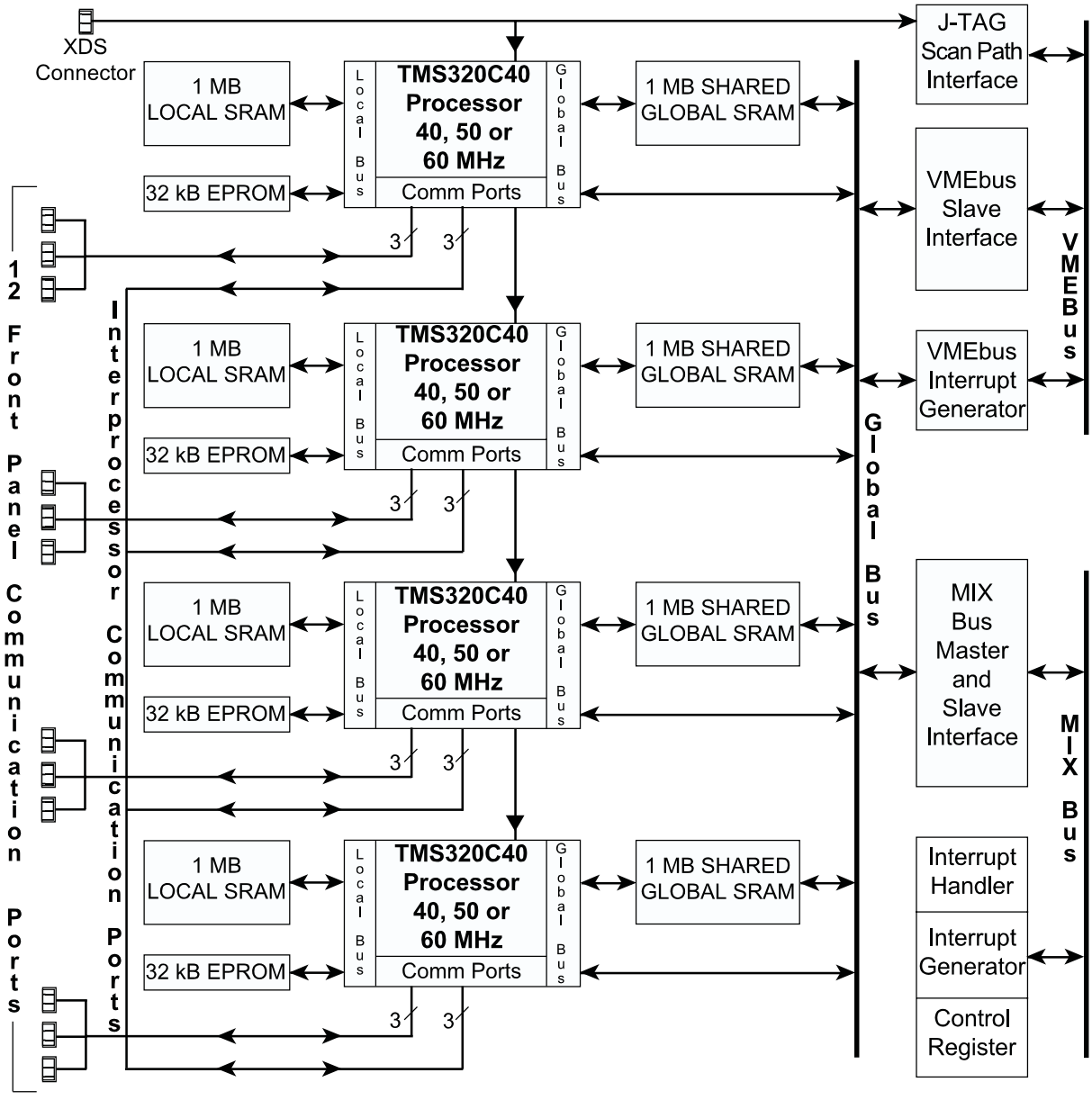


Figure 1-1: Block Diagram - Pentek Model 4270 Quad TMS320C40 DSP VME Board/MIX Module

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 2: Installation and Connections

2.1 Inspection

After unpacking the unit, inspect it carefully for possible damage to connectors or components. If you discover any damage, please contact Pentek immediately, at the phone number shown on this manual's title page. Please save the original shipping container and packing material, in case re-shipment is required.

2.2 Jumper Settings

On the two following pages are Figure 2-2 and Figure 2-3, which are drawings of the two circuit boards that comprise the Model 4270, showing the locations of the jumper blocks referred to in the sections below. We refer to these boards by the bus connectors they possess, i. e. as the VME board and the MIX board. The VME board, shown in Figure 2-2, on the next page, houses Processors A and B. The MIX board, shown in Figure 2-3, on the page after next, contains Processors C and D.

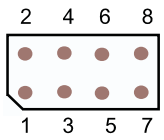


Figure 2-1: Jumper Block Pin Numbering

Figure 2-1, above right, shows the pin numbering scheme used for most of the double-row jumper blocks used in the Model 4270 (the only exception is jumper block S1, discussed in Section 2.2.1, below). Pins in single-row jumper blocks are simply numbered consecutively, with pin 1 identified by the notched corner of the box screened on the circuit board, underneath the jumper block. The shorting jumpers used on the Model 4270 are spaced on 2 mm centers. They are DuPont Electronics Model #86730-001 or the equivalent. Pentek's part number for these jumpers is 356-00010.

2.2.1 **A16 Base Address Jumpers** - VME Board Jumper Block S1

The upper byte of the Model 4270's A16 base address is set by the installation of shorting jumpers on jumper block S1, on the VME board. All of the registers described in Section 3.5 of this manual (VMEbus Interface Memory Map) are referenced as offsets from the A16 base address set by this jumper block.

This jumper block is accessible without disassembling the unit. It is located at the top of the VME board, near the VME P1 connector. Pin 1 of the jumper block is on the side nearest the front panel. This jumper block was originally envisioned as a DIP-switch (which explains the designation S, as opposed to JB). As the layout process continued, it was determined that there was not sufficient room for the DIPswitch. A jumper block was substituted, but the pin numbering was retained. The pin numbering scheme for S1, therefore, resembles that of an DIP-packaged IC, as shown in Figure 2-4, at the top of page 20.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

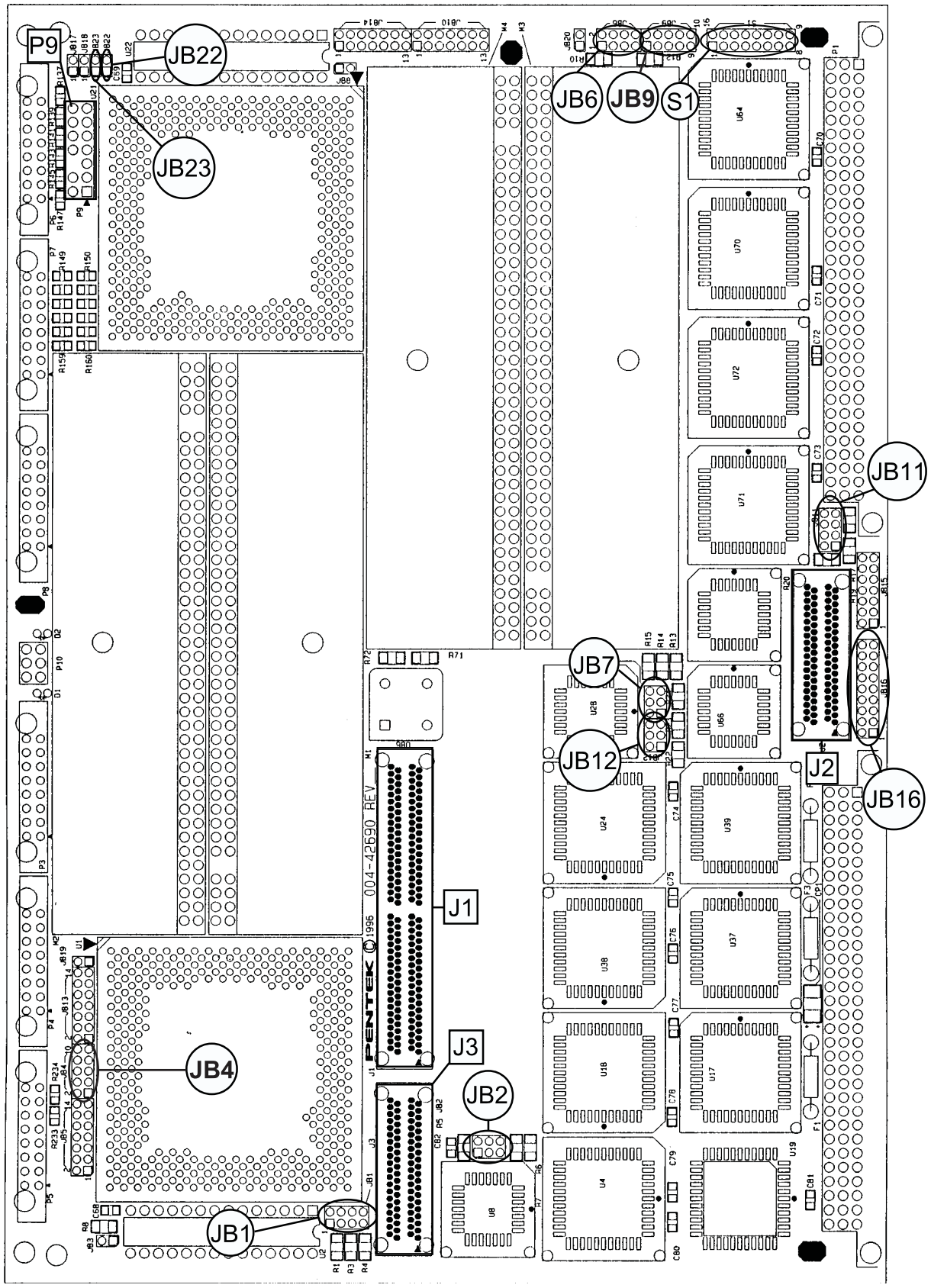


Figure 2-2: Model 4270 VME board drawing showing jumper block locations

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

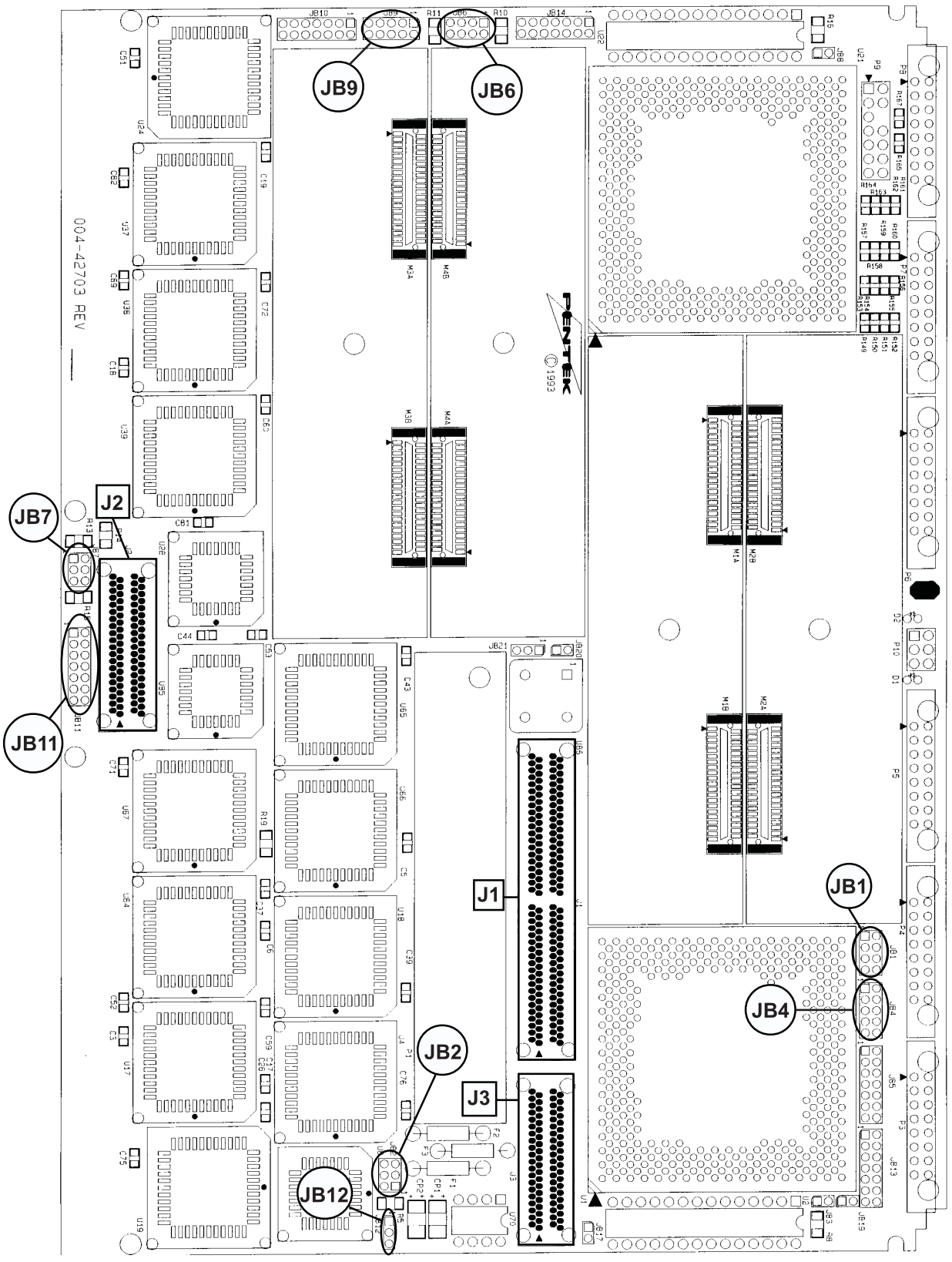


Figure 2-3: Model 4270 MIX board drawing showing jumper block locations

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.1 A16 Base Address (continued)

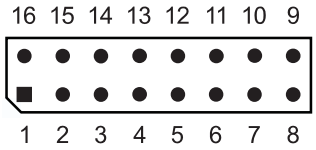


Figure 2-4: Model 4270 - Pin Numbering, Jumper Block S1

The jumper positions in this block are weighted for binary values. Installing a jumper in any position puts a '0' in that binary weighted position, and leaving the jumper off puts a '1' in that binary position. The jumper block sets the value for the two most significant hex digits of the base address. Jumpers placed on pins 9 through 16 will set the value of the most significant hex digit, and jumpers on pins 1 through 8 are used for the next less significant digit. The two least significant digits are 00.

Table 2-1, below, and Table 2-2, at the top of the next page, give the binary and hex weights of each pin pair in the block, and the jumper settings for each hex digit. The factory default setting for this block is all positions installed, which results in an A16_base address of 0x0000.

Table 2-1: Model 4270 - Jumper Block S1 - A16 Base Address

Binary *(Hex)* Weights of Jumper Positions							
Pins 1 - 16	**Pins 2- 15**	**Pins 3- 14**	**Pins 4- 13**	**Pins 5- 12**	**Pins 6- 11**	**Pins 7- 10**	**Pins 8 - 9**
256 *(0x0100)*	512 *(0x0200)*	1024 *(0x0400)*	2048 *(0x0800)*	4096 *(0x1000)*	8192 *(0x2000)*	16384 *(0x4000)*	32768 *(0x8000)*

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.1 A16 Base Address (continued)

Table 2-2: Model 4270 - Jumper Block S1 - A16 Base Address

Hex Digit Settings - *Positions for Most Significant Hex Digit in Italics*

Hex Digit	Pins 4-13 *Pins 8 - 9*	Pins 3-14 *Pins 7-10*	Pins 2-15 *Pins 6-11*	Pins 1-16 *Pins 5-12*	Hex Digit	Pins 4-13 *Pins 8 - 9*	Pins 3-14 *Pins 7-10*	Pins 2-15 *Pins 6-11*	Pins 1-16 *Pins 5-12*
0*	ON	ON	ON	ON	**8**	OFF	ON	ON	ON
1	ON	ON	ON	OFF	**9**	OFF	ON	ON	OFF
2	ON	ON	OFF	ON	**A**	OFF	ON	OFF	ON
3	ON	ON	OFF	OFF	**B**	OFF	ON	OFF	OFF
4	ON	OFF	ON	ON	**C**	OFF	OFF	ON	ON
5	ON	OFF	ON	OFF	**D**	OFF	OFF	ON	OFF
6	ON	OFF	OFF	ON	**E**	OFF	OFF	OFF	ON
7	ON	OFF	OFF	OFF	**F**	OFF	OFF	OFF	OFF

*-Factory Default Setting

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.2 TMS ID Jumpers - JB1 and JB6 (both boards)

These jumper blocks are used by software routines to identify the four individual processors on the 4270, and to allow the implementation of user-defined functions.

Jumpers placed between pins 5 & 6 or pins 7 & 8 of these blocks are used to implement user functions. The states of these jumpers can be read under program control by reading the Local status registers. Installing a jumper will write a zero to the appropriate bit of the status register. See Section 3.1.1.1 for further details.

The TMS ID is used by software programs to differentiate between the processors in a multiple-processor system. The TMS ID of the processor at the bottom of the Model 4270's VME board is set to A by jumpers between pins 1 & 2, and pins 3 & 4 of jumper block JB1 on the VME board. A jumper between pins 3 & 4 of jumper block JB6 on the VME board sets the TMS ID of the processor at the top of the VME board to B. The TMS ID of processor at the bottom of the 4270's MIX board is set to C by a jumper between pins 1 & 2 of JB1 on the MIX board, and the processor at the top of the MIX board gets its TMS ID of D from the absence of jumpers between pins 1 & 2 and pins 3 & 4 of JB6 on the MIX board. Table 2-3, below, shows the factory settings of these jumper blocks. We strongly discourage users from changing the ID jumper settings.

Table 2-3: Model 4270 - TMS ID Jumpers

Board	Jumper Block	pins 1 & 2	pins 3 & 4	TMS ID	pins 5 & 6	pins 7 & 8
VME	**JB1**	ON	ON	Processor A	ON*	ON*
VME	**JB6**	OFF	ON	Processor B	ON*	ON*
MIX	**JB1**	ON	OFF	Processor C	ON*	ON*
MIX	**JB6**	OFF	OFF	Processor D	ON*	ON*

* - User jumpers - Read from TMS Status Register

2.2.3 'C40 Boot Load Address - JB2 and JB7 (both boards)

The address of the code you wish to use to boot the 'C40 processors on your 4270 is partially determined by these jumpers. When a 'C40 is reset, it jumps to an address called the Reset Vector. The Reset Vector for each processor is determined by a pair of bits in the Reset Control Registers (see Sections 3.4.1 and 3.5.2 of this manual). The power-up default setting of these bits selects the 'C40's internal ROM. The ROM program tells the 'C40 to jump to another address and boot load the code found there. That is the address determined by these jumper blocks. For further details on the reset operation of the 'C40, see Sections 6.7 (Reset Operation) and 12.1 (Processor Initialization) of the Texas Instruments **TMS320C4x User's Guide**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.3 'C40 Boot Load Address (continued)

The boot load address for Processor A is set by jumper block JB2 on the VME board. Jumper block JB7 on the VME board sets the boot load address for Processor B. In a like manner, jumper blocks JB2 and JB7 on the MIX board are used to determine the boot load address for Processors C and D, respectively. Tables 2-4 and 2-5, below, describe these jumper settings and the available boot load addresses. In the 'C40 implementation used on the Model 4270, the only boot load sources that should ever be selected are the external EPROM and the Comm Port. The factory default settings of these jumper blocks point to the EPROM, where we put the program that initializes the unit and blinks the front panel LEDs

Table 2-4: Model 4270 - Boot Load Address Jumper Blocks

Board	JB2	JB7
VME	Processor A	Processor B
MIX	Processor C	Processor D

Table 2-5: Model 4270 - 'C40 Boot Code Address Jumper Settings

Address	Location	Pins 1 & 2	Pins 3 & 4	Pins 5 & 6
0x0030 0000*	Ext. PROM	ON	OFF	OFF
COMM PORT	Comm Port	OFF	OFF	OFF
0x4000 0000	Do Not Use	OFF	ON	OFF
0x6000 0000	Not Available	ON	ON	OFF
0x8000 0000	Do Not Use	OFF	OFF	ON
0xA000 0000	Not Available	ON	OFF	ON
0xC000 0000	Do Not Use	OFF	ON	ON
Undefined	Do Not Use	ON	ON	ON
* - Factory Default Setting				

NOTES: (1) During the 'C40 reset cycle, jumper blocks JB2 and JB7 drive the $\overline{\text{IIOF}}$(1-3) pins on their respective 'C40's, via a multiplexer. Following initialization, the multiplexer switches states, causing the $\overline{\text{IIOF}}$ pins to be driven by the interrupt sources for the 'C40. See Sections 4.4, 3.1.2, 3.4.3, and 3.5.6 of this manual for further information about handling interrupts to the 'C40s on the 4270. See the Texas Instruments **TMS320C4x User's Guide**, Section 3.1.10 ($\overline{\text{IIOF}}$ Flag Register), for further details.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.3 'C40 Boot Load Address (continued)

NOTES: (continued)

(2) Addresses 0x4000 0000 and 0x8000 0000 point to Local and Global SRAM, respectively. These are volatile memory banks that will not retain code after power-down and are thus not usable as boot load addresses. Address 0xC000 0000 points to a MIX module.

(3) During the standard EPROM boot, the first Global SRAM address is read. Based on the data value found there, the remainder of the boot code may be taken from the Global, Local or Internal RAM banks.

2.2.4 Global Memory Window Size Jumpers - JB11 (both boards)

Jumper block JB11 on the VME board sets the VMEbus window size in A24 space for the Global memory, which is accessible by all four processors, by VMEbus Masters and by MIX bus Baseboards and Masters. In A32 VME space, the Model 4270's Global memory will occupy a 16 MByte memory region, regardless of the settings of these jumpers. There is also a JB11 jumper block on the MIX board, which determines the Global memory window size from the MIX bus's point of view.

As shipped from the factory, the 4270's four 1 MByte Global SRAMs will occupy an 8 MByte region in A24 VME space (i. e., each 1 MByte bank occupies a 2 MByte region). This can be reduced, if need be, by changing the settings of jumper block JB11 on the VME board. If the window size is reduced to 1 MByte, the page register can be activated to allow access to the other processors' memories. See Sections 3.5.3.2 and 3.5.4 for further details. Table 2-6, below, describes the settings of the VMEbus window size jumper block.

Table 2-6: Model 4270 - VMEbus A24 Global SRAM Window Size Jumpers - Jumper Block JB11

SRAM Window	Pins 7-8 (A23)	Pins 5-6 (A22)	Pins 3-4 (A21)	Pins 1-2 (A20)
1 Megabyte	ON	ON	ON	ON
2 Megabytes	ON	ON	ON	OFF
4 Megabytes	ON	ON	OFF	OFF
*8 Megabytes	ON	OFF	OFF	OFF
16 Megabytes	OFF	OFF	OFF	OFF

* Factory Default Setting

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.4 Global Memory Window Size Jumpers (continued)

Be aware that reducing the VME window size setting will cause access problems in the memory above the allocated window size for users of Pentek's SwiftNet Communication Protocols, and the SwiftTools Development Environment. These programs do not utilize the page register. (Note that only SwiftNet Versions 2.0d Beta and higher provide 4270 support. Contact Pentek at the phone number shown on this manual's title page if you need an upgrade.)

The setting of this jumper block also affects the valid VMEbus addresses for the Global SRAM. See Section 3.5.3.1 for further details.

Jumper block JB11 on the MIX board serves a similar function for the Global memory from the MIX bus' viewpoint. This jumper block is also set for a window 8 MByte at the factory, and it should generally be left that way, with one exception. Those who wish to use their 4270 as a MIX module on a Model 4283 Baseboard will need to re-configure JB11 on the MIX board for the 4 MByte window size. This is due to the inability of the Page Register on the 4283 to generate address bit MXA22. Fortunately, this jumper block is accessible without disassembling the boards. See Section 4.5 of this manual for further details. Table 2-7, below, summarizes the useful settings of the MIX memory window size jumpers.

Table 2-7: MIX Bus Global SRAM Window Size Jumpers - JB11, MIX Board

Global SRAM Block Size	MIX Baseboard Compatibility	Jumper Pins Shorted
4 MByte	**4283**	1-2 and 3-4
***8 MByte**	**All others**	3-5 and 4-6
* Factory Default Setting		

2.2.5 VME Interrupt Sources - VME Board, JB16

Jumper Block JB16 on the VME board assigns the various on-card interrupt sources to VME IRQ levels, to allow the 4270 to send interrupts to the VMEbus. Table 2-8, on the next page, lists the interrupt sources, and describes which IRQs they may be assigned to, by the placement of jumpers at the indicated positions. The selected IRQ level should correspond to the one selected by JB12 (see Section 2.2.6). Since the 4270 may only be assigned to a single IRQ level, this jumper block essentially allows the user to select either any single processor, or all four processors, as valid VME interrupt sources.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.5 VME Interrupt Sources (continued)

Table 2-8: Model 4270 - VME Interrupt Source Selection - JB16							
	IRQ 1	**IRQ 2**	**IRQ 3**	**IRQ 4**	**IRQ 5**	**IRQ 6**	**IRQ 7**
TBC	1-3	-	-	-	-	-	1-2
Processor A	5-3	-	5-7	-	-	-	-
Processor B	-	8-6	-	8-10	-	-	-
Processor C	-	-	9-7	-	9-11	-	-
Processor D	-	-	-	12-10	-	12-14	-
Procs ABC&D	4-3	4-6	*15-17	15-13	16-18	16-14	4-2
* - Factory Default Setting NOTE: Procs ABC&D = any processor may interrupt VME - TBC = Test Bus Controller							

2.2.6 VME IRQ Level - VME Board, JB12

Jumper Block JB12 on the VME board sets the 4270's VME IRQ Level. This block should be configured for the same level as JB16 (see above) to allow the 4270 to send interrupts to the VMEbus. Table 2-9, below, shows the settings.

Table 2-9: Model 4270 - VME IRQ Level - JB12			
Level	**Pins 1 & 2**	**Pins 3 & 4**	**Pins 5 & 6**
IRQ 0	ON	ON	ON
IRQ 1	OFF	ON	ON
IRQ 2	ON	OFF	ON
IRQ 3	OFF	OFF	ON
IRQ 4	ON	ON	OFF
IRQ 5	OFF	ON	OFF
IRQ 6	ON	OFF	OFF
IRQ 7	OFF	OFF	OFF
* - Factory Default Setting IRQ0 = VME Interrupts disabled.			

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 Jumper Settings (continued)

2.2.7 MIX IRQ/DRQ Select Jumper - MIX Board, JB12

Jumper block JB12 on the MIX board can re-route a MIX Interrupt Request (IRQ) signal, generated by any of the 4270's processors, to the MIX DMA Request (DRQ) line. At the time of this writing, none of Pentek's MIX Baseboards or Expansion Modules make use of the DRQ signal. Therefore, this jumper should always be left in the factory default position, between pins 1 and 2.

2.2.8 Test Bus Controller Jumpers - VME Board, JB22 & JB23

In order to use the 74ACT8990 Test Bus Controller, shorting jumpers must be installed at JB22 (to connect the clock signal to the controller) and at JB23 (to connect the reset signal). These signals may also be applied externally, via the XDS JTAG connector (if installed, see Section 2.7), in which case these jumpers would be left off (the factory default). **These jumpers must NOT be installed if the XDS510 Emulator is connected.**

2.2.9 Other Jumper Blocks

Jumper blocks which are not discussed in the sections above are present on the boards for testing and configuration purposes, and should not be tampered with by anyone other than factory-trained, Pentek personnel. If you suspect that some of these jumper blocks have been re-configured, please contact Pentek at the phone number on the title page of this manual for assistance.

2.3 Front Panel Comm Port Connectors

The Model 4270 provides front panel access to three communications ports, referred to as 'Comm' Ports, for each of the four TMS320C40s. There are twelve Comm Port connectors on the front panel with markings on the panel showing the Comm Port number and the processor designation. For example the designation "COM5A" refers to Comm Port 5 for Processor 'A'.

Figure 2-5, on the right side of the next page, shows the front panel Comm Port layout and the pin assignments. Note that the left row and right row connectors are oriented at 180° relative to each other. The connectors are keyed to prevent improper insertion.

Comm Port cables (See Figure 2-6, on the next page) can be easily assembled by using a 3M hand press with locator plate #3443-110 and standard 16-conductor, 0.050 pitch flat ribbon cable such as 3M #3365/16. Every Model 4270 is shipped with twelve mating connectors (3M #50116-B000) and the optional strain relief (3M #3448-50116). The use of these standard components allows the user to easily configure the Comm Port cables to minimize excess cable length for any given interconnect pattern.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.3 Front Panel Comm Port Connectors (continued)

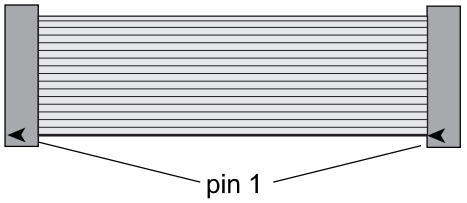


Figure 2-6: Comm Port Cable

As an aid to consistent assembly techniques, make sure that pin 1 of each socket at the end of the cable is connected to pin 1 of the ribbon cable. Pin 1 of the cable is identified by the colored stripe.

All twelve signal lines on the Comm Ports are equipped with series 24Ω resistors to properly match the characteristic impedance of the flat ribbon wiring used in the Comm Port cables.

Comm Port cables are available preassembled from Pentek as Model 2104. Cable lengths of up to 18" can be accommodated.

2.4 Comm Port Configuration

Table 2-10, on the next page, and Figure 2-7, on page 30, describe the connectivity and the default I/O configuration of the six Comm Ports on each of the four 'C40 processors in the Model 4270. In the table, I indicates a port configured as an input, O indicates a port configured as an output, and ports are referred to as processor/port#. Thus, O-B/3 indicates a port configured as an output, driving Processor B, port 3. I-C/0 would indicate a port configured as an input, driven by Processor C, port 0. FP indicates a user-accessible, front panel Comm Port.

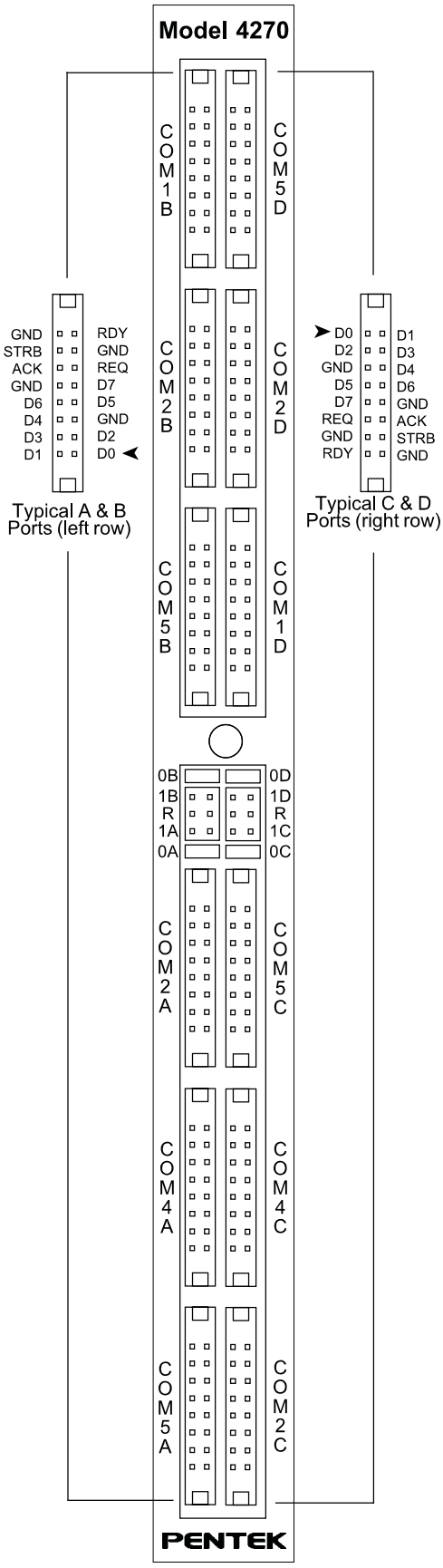


Figure 2-5: Model 4270 - Front Panel

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.4 Comm Port Configuration(continued)

Changing a Comm Port's I/O configuration is a simple matter of addressing it for the opposite function. In other words, if you wish to use Port 3 of Processor A as an output, all you need to do is address it to write. That port is connected to Port 0 of Processor C. When the data written by Processor A, Port 3 arrives at Port 0 of Processor C, that port will reconfigure itself as an input. See Chapter 8 of the Texas Instruments **TMS320C4x User's Guide** for further details.

CAUTION!!

(1) If two comm ports are cabled together, NEVER attempt to have both of them write simultaneously. This will cause SERIOUS DAMAGE to BOTH ports.

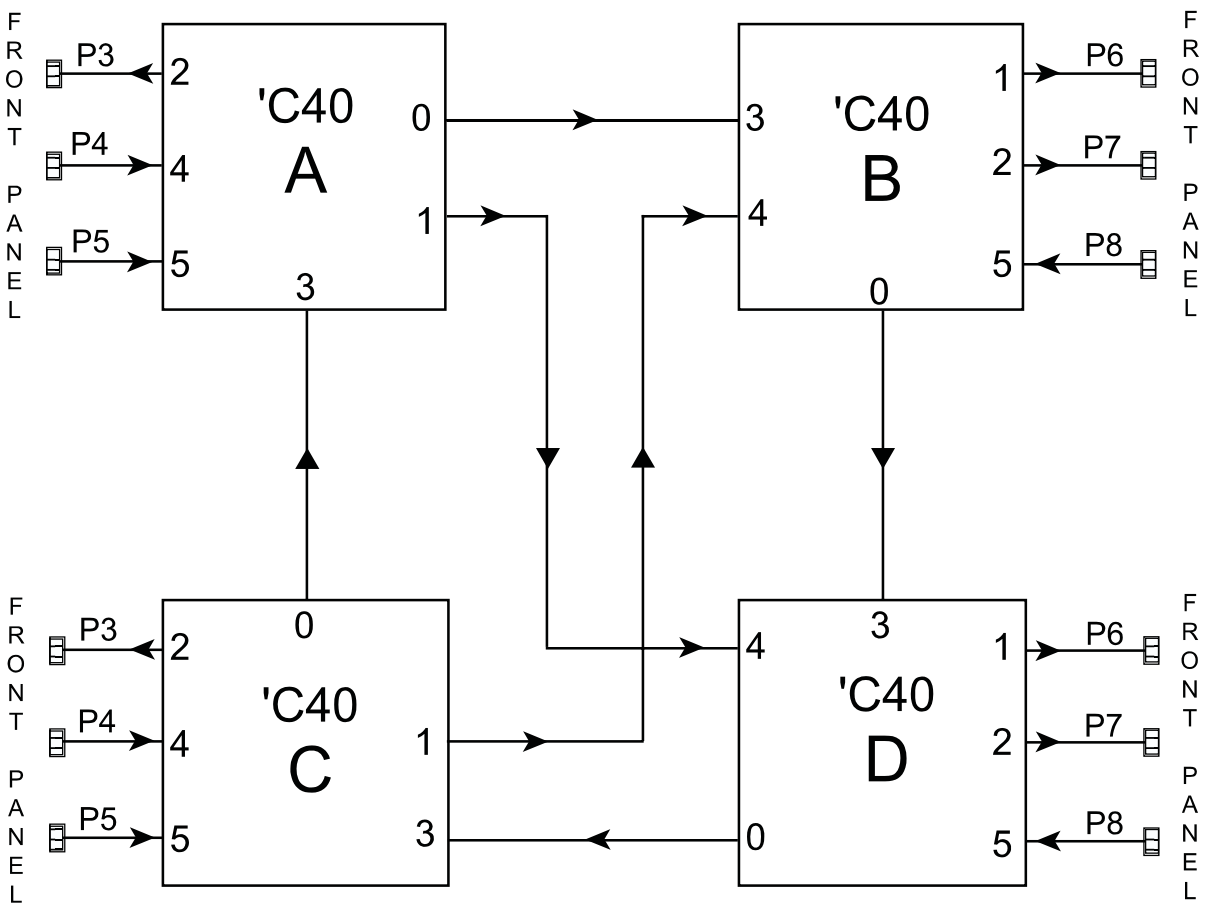
(2) It is recommended that all comm port cabling be completed BEFORE APPLYING POWER to the 4270.

Table 2-10: Model 4270 - Comm Port Connectivity

	Processor A	Processor B	Processor C	Processor D
Port 0	O-B/3	O-D/3	O-A/3	O-C/3
Port 1	O-D/4	O-FP	O-B/4	O-FP
Port 2	O-FP	O-FP	O-FP	O-FP
Port 3	I-C/0	I-A/0	I-D/0	I-B/0
Port 4	I-FP	I-C/1	I-FP	I-A/1
Port 5	I-FP	I-FP	I-FP	I-FP

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Figure 2-7: Pentek Model 4270 Comm Port Allocation



Default input/output configuration for comm ports indicated by arrows

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5 Front Panel Timer I/O Pins and LED Indicators

The Model 4270 provides access to the two timers contained in each of the four 'C40s. Two 6-pin headers and four green LEDs are provided in the center of the panel as shown in Figure 2-8, below. The pin definitions and LED assignments are also shown in the figure. Two styles of mating connector may be used here, either discrete 2-pin sockets or a 6-pin block. The two-pin connector housing may be Berg part number 65039-035 (Pentek #353.00201) or an equivalent, and the 6-pin block is Berg # 65043-034 (Pentek #353.00601) or equivalent. These connectors both use discrete .025" square socket pins, Berg #48254-000 (Pentek #354.00104). Berg makes two hand tools used in assembling these connectors, part #'s HT-208 (the pin crimper for 22 - 26 AWG wire) and HT-80PV (a pin removal tool).

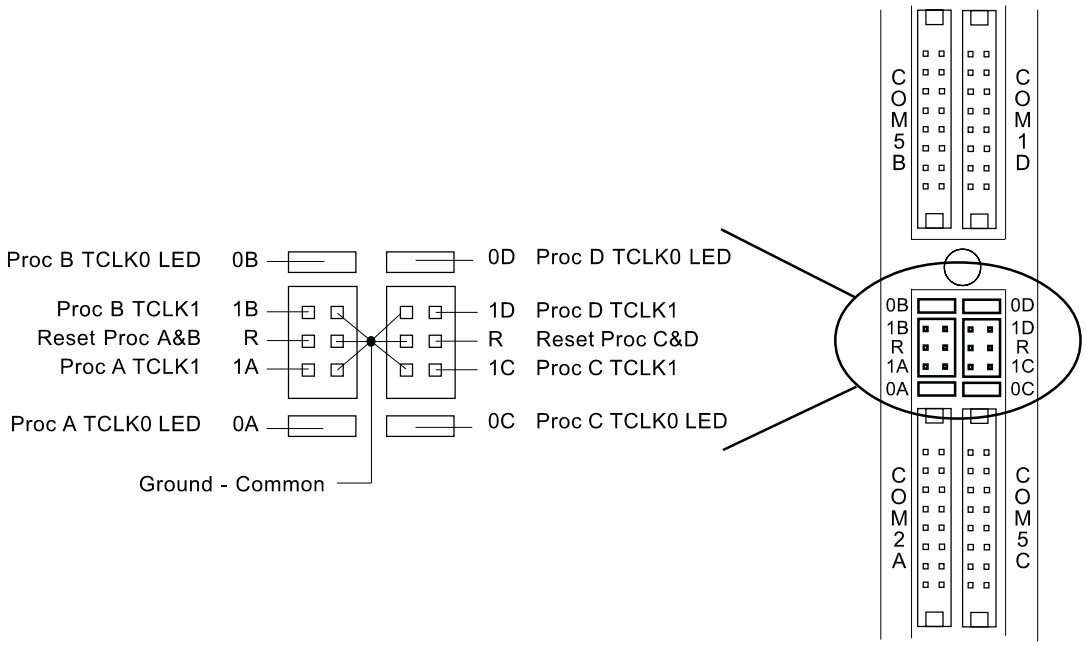


Figure 2-8: Model 4270 - Timer Pins and LED Indicators

2.5.1 Front Panel LED Indicators

Each of the four LEDs is driven through a TTL buffer by the TCLK0 pin on each of the four 'C40 processors. This pin is associated with Timer 0 of each 'C40. The function of the TCLK0 pin is programmable, and it can be used as a general purpose, software-driven flag to indicate program activity or status. This feature can be very useful in developing applications. When the TCLK0 pin is set to logic '1', the LED is OFF and when the TCLK0 pin is cleared to logic '0', the LED is ON. For information on programming the 'C40's Timer 0, please refer to Section 9.13 of the Texas Instruments **TMS320C4x User's Guide**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5 Front Panel Timer I/O Pins and LED Indicators (continued)

2.5.2 Front Panel TCLK1 Timer Pins

Each of the TCLK1 signals, associated with Timer 1 of each 'C40, is brought to a connector pin as shown in Figure 2-8, on the previous page. This pin is software configurable as an input or an output. Like the LED function described above, these signals can be very helpful in program development and are also useful for interfacing to external signals.

The pins are spaced at 0.1 inch centers, and are compatible with both standard flat cable sockets and crimp style discrete wire sockets. The adjacent signal ground pins provide an easily accessible ground reference for two conductor plug assemblies.

For example, to measure the execution time of a software code segment, temporarily embed an instruction to set the TCLK1 bit at the beginning of the segment, and another instruction to clear it at the end of the segment. Then connect an oscilloscope probe to TCLK1 and measure the pulse width. TCLK1 might also be used as a pollable flag, to be tested in a software loop. For information on programming the 'C40's Timer 1, please refer to Section 9.13 of the Texas Instruments **TMS320C4x User's Guide**.

When configured as an input, the TCLK1 pin may also be used for synchronizing a software task to an external TTL signal or from the TCLK1 output of another 'C40. TCLK1 can also be set as a clock input to Timer 1, and Timer 1 can be programmed to generate an interrupt when it receives one clock pulse. In this way, an external signal on TCLK1 can be used to generate a processor interrupt.

CAUTION!!

Because the TCLK1 pins can be used as either inputs or outputs, there is no buffering between the TCLK1 front panel pins and the 'C40 chips. Be careful to comply with the ESD precautions and the voltage and current limits specified in the 'C40 User's Guide. Whether used as in input or an output, voltage applied to TCLK1 must be between 0V and 5V, relative to the ground pin. When used as an output, limit the output current at logic '0' to less than 2 mA. See Section 14.4 of the Texas Instruments TMS320C4x User's Guide for more information.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.5 Front Panel Timer I/O Pins and LED Indicators (continued)

2.5.3 Front Panel Reset Pins

In addition to the TCLK1 pins, there are two $\overline{\text{Reset}}$ pins, one for 'C40 Processors A and B and another for Processors C and D. These pins are TTL compatible and are internally pulled up to +5V with a pullup resistor, which provides the inactive logic '1' level.

The active ('true') level for these $\overline{\text{Reset}}$ pins is logic '0'. An easy way to use the external reset function is to momentarily short the $\overline{\text{Reset}}$ line to the adjacent ground pin. This can be done by using a two-conductor plug and cable assembly wired to a shorting switch. The $\overline{\text{Reset}}$ line can also be driven from any TTL compatible logic device. Both $\overline{\text{Reset}}$ lines can be tied together to provide a common reset to all four 'C40 processors. The reset cycle initiated by these pins reset the associated 'C40 processors, but do not reset any of the logic associated with the MIX or VMEbus interfaces.

2.6 XDS Emulator Connector

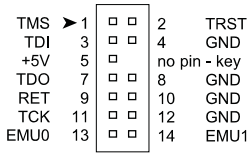


Figure 2-9: Model 4270 - XDS Emulator Connector

The Model 4270 has provisions for use with the Texas Instruments XDS510 Emulator. The emulator connector consists of a header on the 4270 VME board, to which a ribbon cable may be attached. This ribbon cable extends through the front panel, where the emulator connector is installed. The Emulator connector pinout is shown in Figure 2-9, at the right. These connectors are keyed for proper alignment with the socket connector at the end of the emulator cable. See Section 2.7 of this manual for installation and removal procedures for the emulator connector cable. Section 4.3.3 of this manual provides basic information about the use of the emulator. Refer to the Texas Instruments **TMS320C40 Emulator User's Guide** or the documentation supplied with the XDS system for complete operating details.

2.7 Disassembly and Assembly Procedures

Installation of the XDS Emulator connector described above on a standard Model 4270 requires that the front panel be removed and the two PC boards be separated. Detailed procedures for accomplishing these goals, and for reassembling the unit, are given below. These procedures may also be followed if you find it is necessary to change some of the factory default jumper settings.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

An installed XDS emulator cable is provided on Model 4270s purchased with option 005. In this case, the procedures below may be used if one wishes to remove the cable.

2.7.1 Naming Conventions

If the Model 4270 is positioned such that the front panel faces you, in the orientation shown in Figure 2-4 (see page 20), we refer to the circuit board to your left as the VME Board, and the circuit board to your right as the MIX Board. Processors A and B are found on the VME Board, and Processors C and D are on the MIX Board. The VME Board is slightly larger than the MIX Board, so that the edges of the VME Board can engage the mounting rails of the VME card cage when the unit is installed.

When we refer to the top and bottom of the unit, we also make the assumption that it is oriented as shown in Figure 2-4.

CAUTION!!

Perform all assembly steps in a static controlled work environment.

2.7.2 Tools Required

1) #0 Philips screwdriver

2) #1 Philips screwdriver

3) 3/16" flat-blade screwdriver

4) Long-nose pliers (optional)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.3 Removing the Front Panel

The Model 4270 front panel is secured to the main assembly by four screws. All four of these screws are found on the PC assemblies, i.e., none of the hardware on the front panel itself needs to be removed. Three of these four screws are located on the VME Board. The screws at the top and bottom of this board pass through holes at the ends of the arms of the card ejector assemblies, and can be removed with the flat-blade screwdriver. The third screw is located roughly at the front and center of the VME Board, and can be removed with the #1 Philips screwdriver. The fourth screw is located roughly at the front and center of the MIX Board, and can also be removed with the #1 Philips screwdriver. After these four screws are removed, slide the panel forward to separate it from the main assembly. If the panel mounting bracket does not slide out easily from between the PC assemblies, it may be necessary to first loosen the 0-80 Philips-head screws, described in the section below.

2.7.4 Separating the Printed Circuit Assemblies

The two printed circuit cards that comprise the Model 4270 are mechanically secured to one another using four 4-40 nylon screws (these may be either #1 Philips- or slotted-head screws, located near the corners of the unit), threaded into standoffs, and eight 0-80 Philips-head screws which pass through the Comm Port connectors and thread into hex nuts. The heads of all these screws are easily accessible if the unit is laid down on a tabletop, with the VME Board facing up. Remove all twelve of these screws (you may use either the long-nose pliers or a fingertip to hold the hex nuts still while turning the 0-80 screws), but DO NOT ATTEMPT TO SEPARATE THE BOARDS UNTIL YOU HAVE READ THE REST OF THIS SECTION.

The electrical connection between these two cards is accomplished by means of three pairs of high-density board-docking connectors. While these connectors do provide very reliable connectivity in a minimum of board space, the connectors themselves are relatively fragile, and are susceptible to damage if stressed in the wrong direction. These connector blocks can be located in Figures 2-2 and 2-3, on pages 18 & 19 of this manual, as J1, J2 and J3, on both assemblies.

To separate the boards without damaging the connectors, start at the rear of the unit, between the VME connectors, and gently push the boards apart using the thumb and forefinger of one hand until the J2 connectors disengage. Next, locate the J3 connectors at the bottom of the unit, and again gently push the boards apart in that area using the thumb and forefinger of one hand. This action completes the separation of the boards, and the disassembly procedure.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.5 Spacers and Insulators

To maintain proper spacing between the printed circuit assemblies, two spacers and an insulator are mounted between each of the six Comm Port connectors on the two boards. These are shown in Figure 2-10, below. We strongly urge you to retain these parts and install them properly when reassembling the unit. To accommodate the installation of the XDS Emulator Cable Assembly (which will pass through the front panel, between the Comm Port connectors), a shorter insulator and shorter spacers are supplied in the accessory kit. See the sections below for further details.

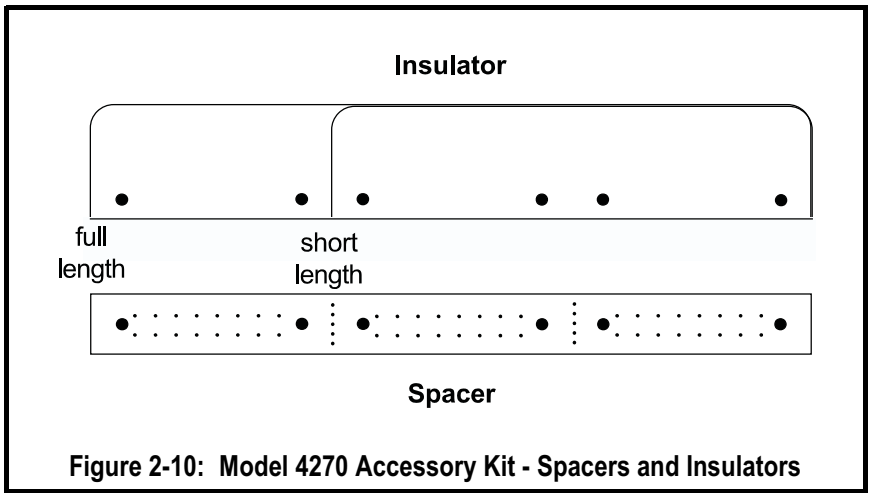


Figure 2-10: Model 4270 Accessory Kit - Spacers and Insulators

2.7.6 The Model 4270 Accessory Kit

The items listed below are included in the accessory kit supplied with your Model 4270. Pentek part numbers are provided in case you need to contact us for replacements of any of these items.

Item	Pentek Part Number
XDS Emulator Cable Assembly	009.42700
Short Insulator	315.00024
Short Spacers (2 pcs)	315.42694
MIX Stacking Connector w/ 4 Jackscrews	351.13000
Comm Port Cable Connectors (12 pcs)	353.01605
Strain Reliefs for Cable Connectors (12 pcs)	353.01606
0-80x11/16" Philips head screws (8 pcs)	385.21012
2.5x6mm Philips head screws (2 pcs)	385.30200
2.5x14mm Fillister head screws (2 pcs)	385.30600
4-40x1/4" nylon screws (4 pcs)	385.40351
0-80 hex nuts (8 pcs)	387.00100

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.6 The Model 4270 Accessory Kit (continued)

NOTE: Model 4270 option 005 is supplied with the XDS Emulator Cable Assembly installed. The accessory kit for this unit therefore excludes the cable assembly, replaces the short insulator and spacers with long ones (insulator part # 315.00023, spacer part # 320.42692).

2.7.7 Installing (or Removing) the XDS Emulator Cable Assembly

The XDS Emulator Cable Assembly is shown in Figure 2-11, below. The female end of the cable assembly connects to the header labeled P9 on the VME Board, located just behind the top Comm Port connector. To prevent contact misalignment, these connectors are keyed. Note that a pin is missing from the P9 header block, in a position corresponding to a socket that is blocked on the female connector. The same pin is missing from the male connector at the other end of the Cable Assembly, which plugs into a similarly keyed female connector from the XDS 510.

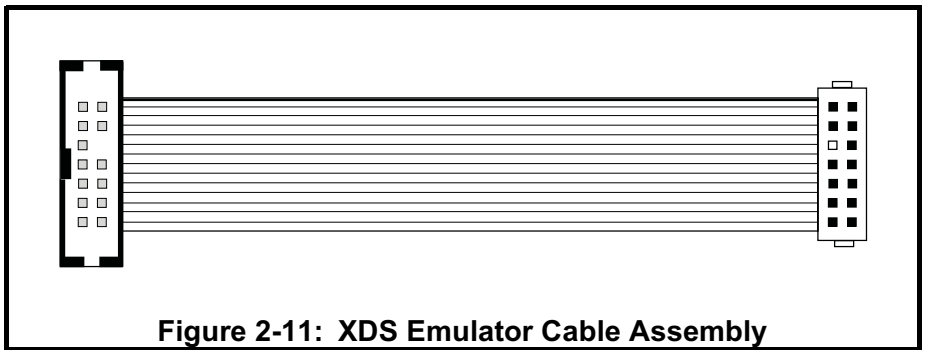


Figure 2-11: XDS Emulator Cable Assembly

Note that when the female end of the Cable assembly is connected to P9, the ribbon cable itself rests slightly above the Comm Port connector in front of it. To prevent stressing of the cable when the MIX board is replaced, use a straightedge or your finger to bend the ribbon cable such that it rests directly on top of the Comm Port connector.

Removing the cable assembly is a simple matter of pulling the female end of the assembly off the P9 header block on the VME board.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.8 Preparing to Attach the Printed Circuit Assemblies

After the XDS Emulator Cable Assembly has been installed (or removed), we are ready to rebuild the unit. The first step in this procedure is to place the VME Board on the worktable, with the component side (the side with the 'C40's) down. Next, install eight 0-80 x 11/16" Philips-head screws in the holes that pass through the Comm Port connectors (these were removed while disassembling the unit, and spares are included in the accessory kit). For both three-connector groups, screws are installed in the top hole of the top connector, in both holes of the middle connector and in the bottom hole of the bottom connector. Figure 2-12, below, shows the proper placement of these eight screws.

Next, place a piece of paper, a file folder, a straightedge, or some other implement over the screws to keep them from falling out. Then, turn the board over, such that the 'C40's are face up. Where you go from here depends on whether the Emulator Cable Assembly is installed or not.

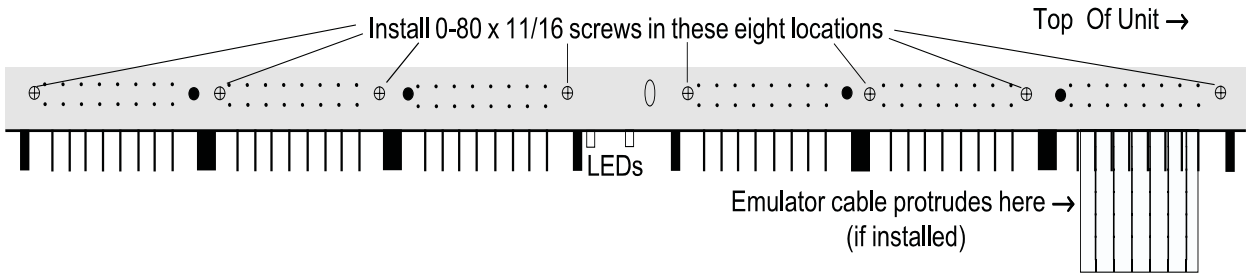


Figure 2-12: Model 4270 VME Board - front of solder side - top view

If the Emulator Cable Assembly is Installed - Locate the two short spacers and the short insulator in your accessory kit. Install one of the short spacers over the two Comm Port connectors next to the one that the cable assembly goes over (the screw shafts pass through the larger holes at the ends of the spacer). Install the short insulator in the same location, above the spacer. The insulator should be oriented such that it covers the shroud of the Comm Port connector (the spacers may be oriented in either direction). Install the other short spacer over the short insulator you just installed. Return the long spacers and insulator that were originally installed here to the accessory kit for possible future use, i. e., if the cable assembly is ever removed.

Next, take the two remaining full length spacers and the full length insulator, and install those parts over the other three Comm Port connectors in the manner described above (spacer, insulator, spacer). Figure 2-13, at the top of the next page, shows a front view of the VME Board when the steps above have been completed. Separation between the spacers and insulators has been exaggerated in the figure for clarity.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.8 Preparing to Attach the Printed Circuit Assemblies (continued)

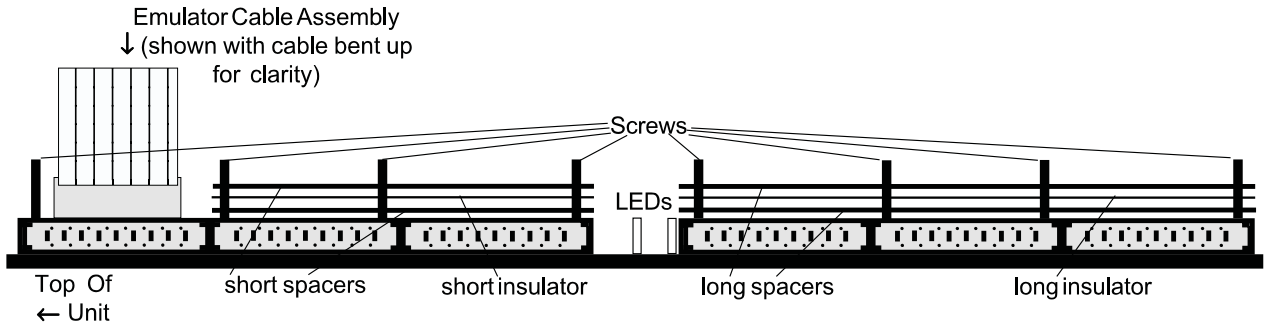


Figure 2-13: Model 4270 VME Board - front of component side - front view

If the Emulator Cable Assembly is Absent - In the accessory kit shipped with your unit, locate two full-length spacers and a full-length insulator. Another set of these components (two full length spacers and a full length insulator) was removed from the front of the board during disassembly. Install all of these parts over the two sets of Comm Port connectors, as described in the preceding paragraphs (spacer, insulator, spacer). Keep the short spacers and insulator with the accessory kit for possible future use (i.e., if you ever need to install the cable assembly). Upon completion of this portion of the procedure, the front of the unit should look similar to the figure above, but without the cable assembly and two sets of full length spacers and insulators.

2.7.9 Attaching the Printed Circuit Assemblies

The VME Board should now be lying flat on the worktable top, oriented as shown in Figure 2-13. Hold the MIX Board, with the 'C40 side down and the Comm Port connectors toward you. Align the corresponding holes in the MIX Board's Comm Port connectors with the screws that pass through the VME Board's Comm Port connectors, the spacers, and the insulators. Gently lower the MIX board such that it comes to rest on top of the VME board, but DO NOT APPLY ANY DOWNWARD PRESSURE AT THIS TIME.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.9 Attaching the Printed Circuit Assemblies (continued)

Rotate the board a half-turn (such that the Comm ports face away from you) and check the alignment of the rear docking connectors, located between the VMEbus connectors. Alignment should be adjusted such that the tab on the connector on the VME Board will fit within the slot in the connector on the MIX Board. Another set of docking connectors is located at the bottom of the assembly, which should now be on the side to your left (if you have followed the recommended orientations given here). Check the alignment of these connectors to verify that the male portion of the connector (on the VME Board) is properly positioned to engage the female end of the connector (on the MIX Board). Once you are certain that the connectors are properly aligned, GENTLY press down on the MIX board in the areas over the two sets of connectors until they are fully engaged.

Figure 2-14, below, shows the docking connectors appropriately aligned. The jumper block headers located in front of the rear docking connectors on both boards have been intentionally left out of the drawing.

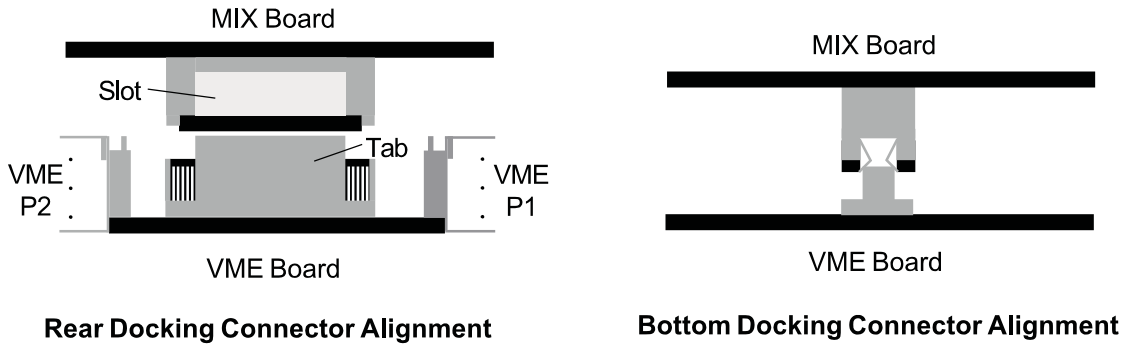


Figure 2-14: Model 4270 - Aligning the Docking Connectors

You should now begin to thread the 0-80 hex nuts on to the screws that pass through the Comm Port connectors, but do not tighten them fully at this time. After this, turn the assembly over such that the solder side of the VME Board is face up, and thread the four 4-40 nylon screws through the VME Board and into the four nylon standoffs near the corners of the MIX Board. These screws may be fully tightened at this time, but leave the 0-80 screws loose until the front panel has been replaced.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.7 Disassembly and Assembly Procedures (continued)

2.7.10 Replacing the Front Panel

To replace the Model 4270's Front Panel, start by simply sliding it over the Comm Port connector housings. If the XDS Emulator Cable Assembly is installed, it will be necessary to rotate the panel to pass the male end of the cable through before sliding it over these housings. Note that the longer of the two slots in the front panel goes toward the bottom of the assembled unit (i. e., the emulator cable, if installed, passes through the smaller slot).

The panel is secured to the rest of the assembly by means of four screws. Three of these are installed through the VME Board. Two 2.5x14mm Fillister head screws are installed through holes at the top and bottom front corners of the VME board, and pass through holes at the ends of the arms of the ejector handles on the front panel. These screws thread into threaded brackets at the corners of the MIX Board, using the flat-blade screwdriver. One 2.5x6mm Philips-head screw is installed through the elongated hole near the center of the front of the VME Board, and is threaded into a mounting bracket on the panel using the #1 Philips screwdriver. The other 2.5x6mm Philips screw is installed in a similar location from the MIX Board side of the unit.

After the front panel has been fully secured to the rest of the assembly, the 0-80 Philips head screws that pass through the Comm Port connectors may be fully tightened into their hex nuts. You may hold the nuts still with either a long nose plier or a finger.

This completes the reassembly procedure.

2.8 Installing the MIX Module on a MIX Baseboard

The Model 4270 is a full-depth MIX module. For MIX module stack assembly instructions, see Pentek's **MIX Bus Tutorial** (Part # 800.00001) of this manual.

2.9 Installing the Assembly in the VMEbus Card Cage

Pentek's **MIX Bus Tutorial** (Part # 800.00001) of this manual also covers the installation of the assembled MIX stack in the VME card cage.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.10 Power-Up Self-Test

When power is applied to the Model 4270, some basic diagnostic routines are run from the EPROM. If no problems are encountered during this procedure, the four LEDs on the 4270's front panel will blink about five times a second (Timer 0 in each 'C40 is used as a counter). If the factory boot code was executed and any LED only blinks around once per second, then that 'C40 has had trouble accessing its Global SRAM. If the factory boot code was executed and a given LED does not blink at all, then a more serious problem exists with the circuitry associated with that processor.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 3: Memory Maps and Operating Procedures

3.1 Local Memory Map

The map for the resources assigned to Local memory, as seen by each of the four TMS320C40 processors on the Model 4270, is presented in Tables 3-1 and 3-2, below. The following sections give more detailed information about the Local memory resources and their usage.

Table 3-1: Model 4270 - Local Memory Map
(for each processor)

0x0000 0000-0x0000 0FFF	Internal ROM (Boot Loader)
0x0000 1000-0x000F FFFF	Reserved
0x0010 0000-0x0010 00FF	Internal Peripherals (see Table 3-2, at right)
0x0010 0100-0x002F F7FF	Reserved
0x002F F800-0x002F FBFF	1k Internal RAM (Block 0)
0x002F FC00-0x002F FFFF	1k Internal RAM (Block 1)
0x0030 0000-0x0030 7FFF	External 32k x 8 Boot ROM (Read Only)
0x0030 8000-0x0033 FFFF	Reserved
0x0034 0000	Local Status Register (Read Only)
0x0034 0000-0x0034 000D	Local Command Registers (Write Only)
0x0034 000E-0x006F FFFF	Reserved
0x0070 0000	MIX Bus Control Register (Read/Write)
0x0070 0001-0x3FFF FFFF	Reserved
0x4000 0000-0x4003 FFFF	External Local RAM - Bank 0 (Standard - Read/Write)
0x4004 0000-0x4007 FFFF	External Local RAM - Bank 1 (Optional - Read/Write)
0x4008 0000-0x400B FFFF	External Local Bus (Read/Write)

Table 3-2: Model 4270 - Internal Peripherals
(for each processor)

0x0010 0000-0x0010 000F	Local and Global Control Port
0x0010 0010-0x0010 001F	Analysis Module Block Registers
0x0010 0020-0x0010 002F	Timer 0 Registers
0x0010 0030-0x0010 003F	Timer 1 Registers
0x0010 0040-0x0010 004F	Comm Port 0
0x0010 0050-0x0010 005F	Comm Port 1
0x0010 0060-0x0010 006F	Comm Port 2
0x0010 0070-0x0010 007F	Comm Port 3
0x0010 0080-0x0010 008F	Comm Port 4
0x0010 0090-0x0010 009F	Comm Port 5
0x0010 00A0-0x0010 00AF	DMA Channel 0
0x0010 00B0-0x0010 00BF	DMA Channel 1
0x0010 00C0-0x0010 00CF	DMA Channel 2
0x0010 00D0-0x0010 00DF	DMA Channel 3
0x0010 00E0-0x0010 00EF	DMA Channel 4
0x0010 00F0-0x0010 00FF	DMA Channel 5

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.1 TMS320C40 Status Registers

- 'C40 address 0x0034 0000 (Procs A, B, C & D)

Table 3-4, on the following page, shows the contents of the four 'C40 Status Registers. The registers appear at 'C40 address 0x0034 0000 for each of the four respective processors. Bits D17 - D13 and D10 - D0 are cleared to the logic '0' state at power-up reset. The states of bits D22 - D20 after reset are dependent on jumper placements. The states of D12 and D11 after reset are dependent on other MIX expansion modules (these are active low, and usually initialize to the '1' state), and the post-reset states of D19 and D18 (which are unused) are not defined. Detailed descriptions of each bit in these registers can be found in Section 3.1.2.

Note that the bit arrangement is identical for all four status registers. The only differences occur when reference is made to a specific processor (A, B, C or D). The individual bits in the 'C40 Status Registers are discussed in detail in the subsections below.

3.1.1.1 User Jumper Status - Bits D23 & D22

These two bits correspond to the states of the two User Jumpers (pins 5-6 and pins 7-8) in the TMS ID Jumper Blocks, identified as JB1 and JB6 on both boards (see Section 2.2.2). A '0' in any of these bits indicates that a jumper is installed. The states of these bits may be read under program control and used to initiate user-defined functions.

3.1.1.2 TMS ID - Bits D21 & D20

These two bits correspond to the states of the two TMS ID Jumpers (pins 1-2 and pins 3-4) in the TMS ID Jumper Blocks, identified as JB1 and JB6 on both boards (see Section 2.2.2). The states of these bits are interpreted to identify each processor as shown in Table 3-3, below.

Table 3-3: Model 4270 - TMS ID Bits

Processor	D21	D20
A	0	0
B	0	1
C	1	0
D	1	1

As previously noted in Section 2.2.2, we strongly discourage users from changing the configuration of the ID jumpers since standard 4270 software routines utilize these bits.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table 3-4: TMS320C40 Status Registers - Local ID, Interrupt and Path Lock Status
Read Only at 'C40 Address 0x0034 0000 (D31- D24 not used)

Processor A

D23	D22	D21	D20	D19	D18	D17	D16
0 = user ID jumper #2 installed	0 = user ID jumper #1 installed	0,0 in these 2 bits = TMS ID for Processor A		N/U	N/U	1 = Proc. A Interrupt to VME	1 = Proc. A Interrupt to MIX
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
1 = Proc. A Interrupt to Proc. D	1 = Proc. A Interrupt to Proc. C	1 = Proc. A Interrupt to Proc. B	0 = MIX Interrupt 2	0 = MIX Interrupt 1	1=Global Bus Locked by VME	1=Global Bus Locked by MIX	1=Global Bus Locked by Processor D
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
1=Global Bus Locked by Processor C	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A	1 = VME Interrupt to Proc. A	1 = MIX Interrupt to Proc. A	1 = Proc. D Interrupt to Proc. A	1 = Proc. C Interrupt to Proc. A	1 = Proc. B Interrupt to Proc. A

Processor B

D23	D22	D21	D20	D19	D18	D17	D16
0 = user ID jumper #2 installed	0 = user ID jumper #1 installed	0,1 in these 2 bits = TMS ID for Processor B		N/U	N/U	1 = Proc. B Interrupt to VME	1 = Proc. B Interrupt to MIX
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
1 = Proc. B Interrupt to Proc. D	1 = Proc. B Interrupt to Proc. C	1 = Proc. B Interrupt to Proc. A	0 = MIX Interrupt 2	0 = MIX Interrupt 1	1=Global Bus Locked by VME	1=Global Bus Locked by MIX	1=Global Bus Locked by Processor D
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
1=Global Bus Locked by Processor C	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A	1 = VME Interrupt to Proc. B	1 = MIX Interrupt to Proc. B	1 = Proc. D Interrupt to Proc. B	1 = Proc. C Interrupt to Proc. B	1 = Proc. A Interrupt to Proc. B

Processor C

D23	D22	D21	D20	D19	D18	D17	D16
0 = user ID jumper #2 installed	0 = user ID jumper #1 installed	1,0 in these 2 bits = TMS ID for Processor C		N/U	N/U	1 = Proc. C Interrupt to VME	1 = Proc. C Interrupt to MIX
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
1 = Proc. C Interrupt to Proc. D	1 = Proc. C Interrupt to Proc. B	1 = Proc. C Interrupt to Proc. A	0 = MIX Interrupt 2	0 = MIX Interrupt 1	1=Global Bus Locked by VME	1=Global Bus Locked by MIX	1=Global Bus Locked by Processor D
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
1=Global Bus Locked by Processor C	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A	1 = VME Interrupt to Proc. C	1 = MIX Interrupt to Proc. C	1 = Proc. D Interrupt to Proc. C	1 = Proc. B Interrupt to Proc. C	1 = Proc. A Interrupt to Proc. C

Processor D

D23	D22	D21	D20	D19	D18	D17	D16
0 = user ID jumper #2 installed	0 = user ID jumper #1 installed	1,1 in these 2 bits = TMS ID for Processor D		N/U	N/U	1 = Proc. D Interrupt to VME	1 = Proc. D Interrupt to MIX
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
1 = Proc. D Interrupt to Proc. C	1 = Proc. D Interrupt to Proc. B	1 = Proc. D Interrupt to Proc. A	0 = MIX Interrupt 2	0 = MIX Interrupt 1	1=Global Bus Locked by VME	1=Global Bus Locked by MIX	1=Global Bus Locked by Processor D
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
1=Global Bus Locked by Processor C	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A	1 = VME Interrupt to Proc. D	1 = MIX Interrupt to Proc. D	1 = Proc. C Interrupt to Proc. D	1 = Proc. B Interrupt to Proc. D	1 = Proc. A Interrupt to Proc. D

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.1 'C40 Status Register Bit Descriptions (continued)

3.1.1.3 Processor Interrupt to VMEbus - Bit D17

Table 3-5: Model 4270 - VME Clear Interrupt Registers

Processor	VMEbus Address
A	A16_base + 0x20
B	A16_base + 0x24
C	A16_base + 0x28
D	A16_base + 0x2C

A '1' in this bit indicates that the processor whose status register you are reading has issued an interrupt to the VMEbus. The processor sets this interrupt by writing to address 0x0034 0009 in its own Memory Map. An interrupt is cleared when a VME Master writes to the appropriate Clear Interrupt register in the 4270's A16 Memory Map. Table 3-5, left, gives the addresses of the VME Clear Interrupt Registers for interrupts issued by each processor.

3.1.1.4 Processor Interrupt to MIX Bus - Bit D16

Table 3-6: Model 4270 - MIX Clear Interrupt Registers

Processor	Symbolic MIX Address
A	Clr_Int_A_MX
B	Clr_Int_B_MX
C	Clr_Int_C_MX
D	Clr_Int_D_MX

A '1' in this bit indicates that the processor whose status register you are reading has issued an interrupt to the MIX bus. The processor sets this interrupt by writing to address 0x0034 0008 in its own Memory Map. To clear an interrupt, a MIX Master writes to the appropriate Clear Interrupt register in the 4270's MIX Memory Map. Table 3-4, at the left, gives the symbolic addresses of the MIX Clear Interrupt Registers for interrupts issued by each processor. See Table 3-28, 3-29, 3-31 or 3-32 for the actual (Baseboard-dependent) addresses of these registers

3.1.1.5 Interrupts to Other Processors - Bits D15 thru D13

Table 3-7: Model 4270 - Outgoing Inter-Processor Interrupt Bits

Bit	Interrupting Processor			
	A	B	C	D
D13	B	A	A	A
D14	C	C	B	B
D15	D	D	D	C

A '1' in any of these bits indicates that the processor whose Status Register you are reading has issued an interrupt to another processor. This interrupt drives the $\overline{\text{IIOF0}}$ input of the target 'C40. Table 3-7, left, tells which bit indicates which processor has received the interrupt, as a function of the processor whose Status Register is being read (i. e., the processor that issued the interrupt).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.1 'C40 Status Register Bit Descriptions (continued)

3.1.1.5 Interrupts to Other Processors (continued)

Table 3-8: Model 4270 - Processor Set Interrupt Registers

Address	Interrupting Processor			
	A	B	C	D
0x0034 0005	B	A	A	A
0x0034 0006	C	C	B	B
0x0034 0007	D	D	D	C

A processor sets this interrupt by writing to the appropriate Set Interrupt Register in its own address map. Table 3-8, left, tells which address to write, to interrupt a given (other) processor as a function of which processor is the interrupter.

Table 3-9: Model 4270 - Processor Clear Interrupt Registers

Address	Interrupting Processor			
	A	B	C	D
0x0034 0000	B	A	A	A
0x0034 0001	C	C	B	B
0x0034 0002	D	D	D	C

The interrupt is cleared when the interrupted processor writes to the appropriate Clear Interrupt Register in its own Memory Map. Table 3-9, left, tells which address to write, to clear an interrupt from a given processor as a function the processor that has been interrupted.

3.1.1.6 MIX Module Interrupt - Bits D12 & D11

Table 3-10: Model 4270 -MIX Interrupt Source ID

4270 slot #	D12 (MXINT2) = 0	D11 (MXINT1) = 0
0	Slot 1	Slot 2
1	Slot 2	Slot 0
2	Slot 0	Slot 1

These two bits indicate to the processor in question that another MIX module has issued a MIX interrupt. The exact interpretation of these bits (with respect to the source of the interrupt) depends upon the 4270's location in the MIX stack. See Table 3-10, at the left, for details. On each 'C40, the MXINT2 signal drives the $\overline{\text{IIOF3}}$ input, and the MIXINT1 signal drives the $\overline{\text{IIOF2}}$ input, during normal operation. Section A.1 gives sample ISRs for these interrupts.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.1 'C40 Status Register Bit Descriptions (continued)

3.1.1.7 Global Bus Lock - Bits D10 thru D5

A '1' in any of these 6 bits indicates that the Global bus on the 4270 is locked. Locking the bus allows the master device that issued the lock exclusive access to the Global SRAM and enables faster block transfers.

Processors lock the Global bus by writing to address 0x0034 000B in their own Memory Maps, and unlock it by writing to 0x0034 000A. MIX bus Masters lock the Global bus by writing a '1' (and unlock it by writing a '0') to D15 of the MIX Reset Control Register (see Section 3.4.1.1). VMEbus Masters lock the Global bus by writing a '1' (and unlock it by writing a '0') to D15 of the VME Reset Control Register, at the A16 base address (see Section 3.5.2.1). In all cases, one access over the bus must be made before locking it.

Table 3-11, at the left, tells which Status Register bit gives the Global bus lock state for which resource.

Table 3-11: Model 4270 - Global Bus Lock Bits

Bit	Bus locked by ...
D10	VME
D9	MIX
D8	Processor D
D7	Processor C
D6	Processor B
D5	Processor A

3.1.1.8 Interrupt from VMEbus - Bit D4

A '1' in this bit indicates that the processor whose Status Register you are reading has received an interrupt over the VMEbus. The processor clears this interrupt by writing to address 0x0034 0004 in its own Memory Map. This interrupt is delivered to the 'C40 at its $\overline{\text{IIOF1}}$ input.

3.1.1.9 Interrupt from MIX Bus Master - Bit D3

A '1' in this bit indicates that the processor whose Status Register you are reading has received an interrupt from a MIX bus Master. The Master sets this interrupt by writing to an address dependent upon the processor to be interrupted and the MIX Baseboard in use. See Section 3.4.3.1 and Table 3-28, 3-29, 3-31, or 3-32 for further details. The processor clears this interrupt by writing to address 0x0034 0003 in its own Memory Map. This interrupt is delivered to the 'C40 at its $\overline{\text{IIOF1}}$ input.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.1 'C40 Status Register Bit Descriptions (continued)

3.1.1.10 Interrupts from Other Processors - Bits D2 thru D0

Table 3-12: Model 4270 - Incoming Inter-Processor Interrupt Bits

Bit	Interrupted Processor			
	A	B	C	D
D0	B	A	A	A
D1	C	C	B	B
D2	D	D	D	C

Table 3-13: Model 4270 - Processor Set Interrupt Registers

Address	Interrupting Processor			
	A	B	C	D
0x0034 0005	B	A	A	A
0x0034 0006	C	C	B	B
0x0034 0007	D	D	D	C

Table 3-14: Model 4270 - Processor Clear Interrupt Registers

Address	Interrupted Processor			
	A	B	C	D
0x0034 0000	B	A	A	A
0x0034 0001	C	C	B	B
0x0034 0002	D	D	D	C

A '1' in any of these bits indicates that the processor whose Status Register you are reading has received an interrupt from another processor. This interrupt is delivered to the 'C40 at its $\overline{\text{IIOF0}}$ input. Table 3-12, left, tells which bit indicates which processor has issued the interrupt, as a function of the processor whose Status Register is being read (i. e., the processor that has been interrupted).

A processor sets this interrupt by writing to the appropriate Set Interrupt Register in its own address map. Table 3-13, left, tells which address to write, to interrupt a given (other) processor as a function of which processor is the interrupter.

The interrupt is cleared when the interrupted processor writes to the appropriate Clear Interrupt Register in its own Memory Map. Table 3-14, left, tells which address to write, to clear an interrupt from a given processor as a function which processor has been interrupted.

3.1.2 TMS320C40 Command Registers - Processors A, B, C and D

Table 3-15, on the next page, summarizes the addresses and functions of the Command Registers associated with the four 'C40 processors on the Model 4270. In all cases, simply writing to the given address will cause the indicated function to be executed. The data written to these addresses is unused and therefore trivial. Detailed descriptions of all the Command Registers can be found after the table on the next page.

Once again, you will note that the arrangement of these registers is very nearly identical for all four processors. The only differences arise with respect to references to specific processors (A, B, C or D).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.2 TMS320C40 Command Registers (continued)

Table 3-15: Model 4270 - TMS320C40 Command Registers (write only) - Processors A, B, C and D

Processor A			Processor B		
Address	**Name**	**Description**	**Address**	**Name**	**Description**
0x0034 0000	B Int Rst	Clear interrupt from Proc. B	**0x0034 0000**	A Int Rst	Clear interrupt from Proc. A
0x0034 0001	C Int Rst	Clear interrupt from Proc. C	**0x0034 0001**	C Int Rst	Clear interrupt from Proc. C
0x0034 0002	D Int Rst	Clear interrupt from Proc. D	**0x0034 0002**	D Int Rst	Clear interrupt from Proc. D
0x0034 0003	MIX Int Rst	Clear interrupt from MIX bus	**0x0034 0003**	MIX Int Rst	Clear interrupt from MIX bus
0x0034 0004	VME Int Rst	Clear interrupt from VMEbus	**0x0034 0004**	VME Int Rst	Clear interrupt from VMEbus
0x0034 0005	Set B Int	Send interrupt to Proc. B	**0x0034 0005**	Set A Int	Send interrupt to Proc. A
0x0034 0006	Set C Int	Send interrupt to Proc. C	**0x0034 0006**	Set C Int	Send interrupt to Proc. C
0x0034 0007	Set D Int	Send interrupt to Proc. D	**0x0034 0007**	Set D Int	Send interrupt to Proc. D
0x0034 0008	Set MIX Int	Send interrupt to MIX bus	**0x0034 0008**	Set MIX Int	Send interrupt to MIX bus
0x0034 0009	Set VME Int	Send interrupt to VMEbus	**0x0034 0009**	Set VME Int	Send interrupt to VMEbus
0x0034 000A	A Bus Unlock	Processor A Global Bus Unlock	**0x0034 000A**	B Bus Unlock	Processor B Global Bus Unlock
0x0034 000B	A Bus Lock	Processor A Global Bus Lock	**0x0034 000B**	B Bus Lock	Processor B Global Bus Lock
0x0034 000C	A NMI Enable	Processor A NMI Enable*	**0x0034 000C**	B NMI Enable	Processor B NMI Enable*
0x0034 000D	A NMI Disable	Processor A NMI Disable*	**0x0034 000D**	B NMI Disable	Processor B NMI Disable*
Processor C			**Processor D**		
Address	**Name**	**Description**	**Address**	**Name**	**Description**
0x0034 0000	A Int Rst	Clear interrupt from Proc. A	**0x0034 0000**	A Int Rst	Clear interrupt from Proc. A
0x0034 0001	B Int Rst	Clear interrupt from Proc. B	**0x0034 0001**	B Int Rst	Clear interrupt from Proc. B
0x0034 0002	D Int Rst	Clear interrupt from Proc. D	**0x0034 0002**	C Int Rst	Clear interrupt from Proc. C
0x0034 0003	MIX Int Rst	Clear interrupt from MIX bus	**0x0034 0003**	MIX Int Rst	Clear interrupt from MIX bus
0x0034 0004	VME Int Rst	Clear interrupt from VMEbus	**0x0034 0004**	VME Int Rst	Clear interrupt from VMEbus
0x0034 0005	Set A Int	Send interrupt to Proc. A	**0x0034 0005**	Set A Int	Send interrupt to Proc. A
0x0034 0006	Set B Int	Send interrupt to Proc. B	**0x0034 0006**	Set B Int	Send interrupt to Proc. B
0x0034 0007	Set D Int	Send interrupt to Proc. D	**0x0034 0007**	Set C Int	Send interrupt to Proc. C
0x0034 0008	Set MIX Int	Send interrupt to MIX bus	**0x0034 0008**	Set MIX Int	Send interrupt to MIX bus
0x0034 0009	Set VME Int	Send interrupt to VMEbus	**0x0034 0009**	Set VME Int	Send interrupt to VMEbus
0x0034 000A	C Bus Unlock	Processor C Global Bus Unlock	**0x0034 000A**	D Bus Unlock	Processor D Global Bus Unlock
0x0034 000B	C Bus Lock	Processor C Global Bus Lock	**0x0034 000B**	D Bus Lock	Processor D Global Bus Lock
0x0034 000C	C NMI Enable	Processor C NMI Enable*	**0x0034 000C**	D NMI Enable	Processor D NMI Enable*
0x0034 000D	C NMI Disable	Processor C NMI Disable*	**0x0034 000D**	D NMI Disable	Processor D NMI Disable*
* - Applies only to board revisions D and higher.					

A detailed description of the Local Command Registers for each of the 'C40 processors on the Model 4270 is given below. For all the registers listed in the following section, the indicated function is caused to execute by writing to the given address, regardless of what data is written.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.2 'C40 Command Register Descriptions (continued)

3.1.2.1 Clear Interrupt Registers - W/O @ 0x0034 0000 - 0x0034 0004

Table 3-16: Model 4270 - Processor Clear Interrupt Registers

Address	Interrupted Processor			
	A	B	C	D
0x0034 0000	B	A	A	A
0x0034 0001	C	C	B	B
0x0034 0002	D	D	D	C
0x0034 0003	MIX			
0x0034 0004	VME			

Table 3-17: Model 4270 - Status Register - Proc. Interrupt Bits

Bits	Interrupted Processor			
	A	B	C	D
D0	B	A	A	A
D1	C	C	B	B
D2	D	D	D	C
D3	MIX			
D4	VME			

Table 3-18: Model 4270 - Processor Set Interrupt Registers

Address	Interrupting Processor			
	A	B	C	D
0x0034 0005	B	A	A	A
0x0034 0006	C	C	B	B
0x0034 0007	D	D	D	C

An interrupted processor clears an interrupt issued to it by writing to the appropriate Clear Interrupt Register, depending upon the source of the interrupt. Table 3-16, left, tells which address should be written to clear an interrupt from which resource, as a function of which processor has been interrupted.

The processor determines where the interrupt came from by reading its Status Register at address 0x0034 0000. Table 3-17, left, tells which bit in the Status Register indicates an interrupt from which source.

Interrupts to processors are set by any other resource (i. e., another processor, the MIX bus or the VMEbus) by writing to the appropriate Set Interrupt Register in its own address map. During normal operation, the outputs of the processors' set interrupt registers are OR'ed together and drive the target 'C40's $\overline{\text{IIOF0}}$ input.

Table 3-18, at the left, shows which address a given processor must write, in order to interrupt another processor. Table 3-19, below, tells which address a bus master should write, in order to interrupt a given processor. The outputs of the MIX and VME Set Interrupt registers are OR'ed together and drive the 'C40's $\overline{\text{IIOF1}}$ input.

Table 3-19: Model 4270 - Bus Master Set Processor Interrupt Registers

Bus	To Interrupt Processor . . .			
	A	B	C	D
VME	A16base+0x10	A16base+0x14	A16base+0x18	A16base+0x1C
MIX*	MX_Int_A	MX_Int_B	MX_Int_C	MX_Int_D
* - Baseboard-dependent symbolic addresses - see Table 3-28, 3-29, 3-31 or 3-32 for the actual address				

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.2 'C40 Command Register Descriptions (continued)

3.1.2.2 Set Interrupt Registers - W/O @ 0x0034 0005 - 0x0034 0009

Table 3-20: Model 4270 - Processor Set Interrupt Registers

Address	Interrupting Processor			
	A	B	C	D
0x0034 0005	B	A	A	A
0x0034 0006	C	C	B	B
0x0034 0007	D	D	D	C
0x0034 0008	MIX			
0x0034 0009	VME			

Any processor may interrupt any other onboard resource by writing to the appropriate Set Interrupt Register in its own Local memory space, depending upon which processor is the interrupter. During normal operation, all the processor interrupts that are targeted for a given processor are OR'ed together and drive the $\overline{\text{IIOF0}}$ input on the target 'C40. Table 3-20, at left, shows which address to write, in order to interrupt a given resource as a function of which processor issues the interrupt. MIX bus or VMEbus Masters can interrupt any processor by writing to the address shown in Table 3-19, at the bottom of the previous page. The bus master interrupts for a given processor are also OR'ed together, and drive that processor's $\overline{\text{IIOF1}}$ input.

Table 3-21: Model 4270 - Status Register - Proc. Interrupt Bits

Bit	Interrupted Processor			
	A	B	C	D
D0	B	A	A	A
D1	C	C	B	B
D2	D	D	D	C
D3	MIX			
D4	VME			

The interrupted processor determines the source of the interrupt by reading its Status Register, at address 0x0034 0000 in its own Local memory space. Table 3-21, at the left, tells which Status Register bit indicates an interrupt from which source, as a function of the identity of the processor receiving the interrupt.

Table 3-22: Model 4270 - Processor Clear Interrupt Registers

Address	Interrupted Processor			
	A	B	C	D
0x0034 0000	B	A	A	A
0x0034 0001	C	C	B	B
0x0034 0002	D	D	D	C
0x0034 0003	MIX			
0x0034 0004	VME			

A processor clears the interrupt issued to it from another resource by writing to the appropriate Clear Interrupt Register in its own memory space, depending upon the source of the interrupt. Table 3-22, at the left, tells which address to write, in order to clear an interrupt, as a function of the interrupt receiver. A bus master clears interrupts issued to it from the 4270 by writing to the appropriate Clear Interrupt Register in its own memory space, depending upon the interrupt source. Table 3-23, at the top of the next page, shows which address a VMEbus or MIX bus Master should write in order to clear an interrupt as a function of the interrupt source.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.2 'C40 Command Register Descriptions (continued)

3.1.2.2 Set Interrupt Registers (continued)

Table 3-23: Model 4270 - Bus Master Clear Processor Interrupt Registers				
Bus	**To Clear an Interrupt from Processor**			
	A	**B**	**C**	**D**
VME	A16base+0x20	A16base+0x24	A16base+0x28	A16base+0x2C
MIX*	Clr_Int_A_MX	Clr_Int_B_MX	Clr_Int_C_MX	Clr_Int_D_MX
* - Baseboard-dependent symbolic addresses - see Table 3-28, 3-29, 3-31 or 3-32 for the actual address				

3.1.2.3 Global Bus Lock Registers - W/O @ 0x0034 000A & 0x0034 000B

To "lock" the data path between a given processor and another processor's Global SRAM bank, write to address 0x0034 000B in the Local Memory Map of the processor that will be mastering the transaction. Locking the Global bus gives a processor exclusive access to the bus and all resources connected to it, and allows block transfers to occur at a faster rate. One access to the SRAM in question must be made before the Global bus can be locked. For further information about the usage and limitations of the Global bus lock feature, please refer to Section 4.7.

When finished with the block transfer, the processor should write to address 0x0034 000A to unlock the bus, and allow other devices access to Global SRAMs. Table 3-24, below, summarizes the registers and bits involved in Global bus locking.

Table 3-24: Model 4270 - Global Bus Lock Registers		
Device	**Global Bus Lock**	**Global Bus Unlock**
Processors	Write @ 0x0034 000B	Write @ 0x0034 000A
VMEbus	A16base+0, D15 = 1	A16base+0, D15 = 0
MIX bus	MX_Rst_Ctl, D15 = 1	MX_Rst_Ctl, D15 = 0

3.1.2.4 Non-Maskable Interrupt (NMI) Registers - W/O @ 0x0034 000C & 0x0034 000D

The 4270's shared memory arbitration system sometimes needs to make a processor wait for a resource that is in use by another device. When a processor must wait for a resource, a timer is started (this timer can be disabled, see below). If the processor waits too long for the resource to become available, (about 1 msec), the processor cycle terminates. When this happens, a 'C40 non-maskable interrupt (NMI) is generated to notify the processor of the incomplete data cycle.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.2 'C40 Command Register Descriptions (continued)

3.1.2.4 Non-Maskable Interrupt (NMI) Registers (continued)

For 4270 board revisions D and higher, the timer (and thus the interrupt) can be disabled by writing to 'C40 address 0x0034 000D. (The revision level of the unit can be found etched in copper on the solder side of either board.) The timer (and interrupt) can be re-enabled by writing to 'C40 address 0x0034 000C. These settings are summarized in Table 3-25, below.

NOTE: The NMI timer does not operate properly in units with Option 060 (60 MHz 'C40s) installed. Attempts to use this feature will hang such units.

Table 3-25: Model 4270 - Non-Maskable Interrupt Registers*

Write @ Address	Function
0x0034 000C	Enable NMI Timer
0x0034 000D	Disable NMI Timer

* - Applies only to board revisions D and higher.

3.1.3 Internal 'C40 MIX Bus Control Registers - R/W @ 0x0070 0000

Table 3-26, below, gives the bitwise definition of these registers, which are used when a processor on the 4270 becomes an Upper MIX Bus Master (UMBM). They are defined identically for all four processors. At reset all bits are cleared to the '0' state, except for the Release When Done bits (D6), which are set to '1'. Refer to the Intel MIX specification for complete details about the usage of the bits in this register.

Table 3-26: Model 4270 - MIX Bus Control Registers - Read/Write at 'C40 Address 0x0070 0000 - Identical for all four Processors

D7	D6	D5	D4	D3	D2	D1	D0
N/U	Release When Done	MXMIO	MXDC	$\overline{\text{MXBE3}}$	$\overline{\text{MXBE2}}$	$\overline{\text{MXBE1}}$	$\overline{\text{MXBE0}}$

Briefly, the 4 LSBs of these registers ($\overline{\text{MXBE0}}$ - $\overline{\text{MXBE3}}$) are the MIX byte enable bits. These allow a byte sent over the MIX bus to be justified in one of four positions on the 32 bit data bus (D0 - D7, D8 - D15, D16 - D23, or D24 - D31), or a word to be justified in either of two positions (D0 - D15 or D16 - D31). For MIX bus transactions with Pentek MIX modules, these four bits should be left in their default (low) states.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.1 Local Memory Map (continued)

3.1.3 Internal 'C40 MIX Bus Control Registers (continued)

The MXDC (MIX Decode) and MXMIO (MIX Memory/IO select) bits should be left in the low state for communication with any Pentek MIX Module. The Release When Done bit, when set to the '1' state, causes the 'C40 to relinquish the MIX bus after each access, allowing other devices to arbitrate for bus mastership.

NOTE: Proper use of the bits in these registers is critical when one wishes to use a processor on a Model 4270, mounted on a Model 4200-series MIX Baseboard, as a VMEbus Master. Refer to Section 4.6.2 of this manual for further details.

3.2 Global Memory Map

In the standard configuration, the four 1 MByte Global memory banks on the 4270 are mapped into four 2 MByte regions. Table 3-27, below, gives the address boundaries for the four memory blocks, as seen by the 'C40 processors, by the VMEbus, and by the MIX bus.

Table 3-27: Model 4270 - Global Bus and SRAM Addressing (Read/Write)

Resource	'C40 Address Range (R/W, 256kWords X 32bits)	VME Address Range* (R/W, 1 MByte X 8 bits)	MIX Address Range† (R/W, 1 MByte X 8 bits)
Processor A Global SRAM	0x8000 0000 - 0x8003 FFFF	VME_base+0x0000 0000 - VME_base+0x000F FFFF	MIX_base+0x0000 0000 - MIX_base+0x000F FFFF
Processor B Global SRAM	0x8008 0000 - 0x800B FFFF	VME_base+0x0020 0000 - VME_base+0x002F FFFF	MIX_base+0x0020 0000 - MIX_base+0x002F FFFF
Processor C Global SRAM	0x8010 0000 - 0x8013 FFFF	VME_base+0x0040 0000 - VME_base+0x004F FFFF	MIX_base+0x0040 0000 - MIX_base+0x004F FFFF
Processor D Global SRAM	0x8018 0000 - 0x801B FFFF	VME_base+0x0060 0000 - VME_base+0x006F FFFF	MIX_base+0x0060 0000 - MIX_base+0x006F FFFF
MIX Bus Master Interface	0xC000 0000 - 0xEFFF FFFF	N/A	N/A

* - VME_base is programmed in the VME Address Mapping and Control Register, at A16_base + 0x04

† - Addresses given in this column assume byte addressing, as on the Models 4200,4200A, 4201, 4201A and 4202 MIX Baseboards. For Models 4283 or 4284 Baseboards, divide the listed offsets by four. The MIX_base address is Baseboard dependent. See the tables in Section 3.3.

It is important to remember that Global SRAM is accessed from the 'C40 side on long-word addresses. VME and MIX access to Global SRAM is generally on byte addresses (1 longword = 32 bits = 4 bytes). The exception is when the 4270 is used as a MIX Slave, and the Baseboard or Upper MIX Bus Master (UMBM) is a DSP-based device (e. g. the Models 4283, 4284 or 4285 Baseboards, or a 4254, 4257 or another 4270 as UMBM). In this case, MIX access is on longword address boundaries.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2 Global Memory Map (continued)

The VME_base address is the starting address of Global SRAM in VME A24 or A32 space, determined by the programming of the VME Address Mapping and Control Register (see Section 3.5.3). This should not be confused with A16_base, which is determined by jumper settings (see Section 2.2.1). Note that VMEbus access to the 4270 in both A24 and A32 space is blocked until **after** the first A16 access. The first A16 access should be to write the desired VME_base address to the aforementioned Mapping Register.

The MIX_base address is dependent upon both the 4270's MIX stack position and the type of MIX Baseboard used. Section 3.3 provides MIX slave address maps for all of Pentek's VME/MIX Baseboards.

The Models 4283 and 4284 Baseboards both utilize members of the TMS320 series of DSPs as the Baseboard processor. These devices always use longword addressing. For either of these devices, or when using one of Pentek's other 'C40 co-processor MIX modules as an Upper MIX Bus Master (UMBM), the address offsets given in Table 3-27 must be divided by 4.

The Model 4270 can be accessed as a slave device in MIX bus memory space only, i. e., slave I/O access to the 4270 is not supported. As an Upper MIX Bus Master (UMBM), however, the 4270 can access other MIX modules in slave I/O space.

3.3 MIX Bus Slave Interface Memory Maps

The following sections cover slave access to the Model 4270 from the MIX Baseboard and the issuing of commands for module control and data transfers. There are two cases to be considered, dependent upon the type of Baseboard used. This chapter gives details for Baseboards that use direct MIX addressing, such as Pentek's Models 4200, 4200A, 4201, 4201A, 4202, 4284 and 4285. General information on the use of these Baseboards is given in Pentek's **MIX Tutorial** (part # 800.00001). The tutorial also gives general information on access to MIX modules from Baseboards that use an Address Page Register, such as Pentek's Model 4283. The Memory Map for the Model 4283 is also included in this chapter.

If the Baseboard you will be using is a Model 4200, 4200A, 4201 or 4201A, use the address map presented in Section 3.3.1. If you will be using your Model 4270 with a Model 4202 Baseboard, please refer to Section 3.3.2. If the Model 4283 will be the MIX Baseboard for your Model 4270, skip to Section 3.3.3, and use the address map found there. If your 4270 will be mounted on a Model 4284 Baseboard, use the address map in Section 3.3.4. Finally, if you will be using your Model 4270 as a MIX slave on the Model 4285 MIX Baseboard Adapter, refer to the address map in Section 3.3.5.

By convention, the MIX addresses given in this manual are generally given as 8-digit hexadecimal codes, with a space between the first and second four digits, simply for notational convenience. The form is MIX_base+0xYYYY ZZZZ. The leading '0x' is common notation for a hex value in the 'C' programming language.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

Since each hex digit represents 4 bits, the 8-digit MIX address represents 32 address bits. The MIX_base part of the address is dependent upon the Baseboard in use, and is used to select each module by its stack position. This geographical addressing scheme avoids the need for address jumpers and possible conflicts, but requires that software must accommodate the physical placement of modules in the MIX stack.

The exception to this convention will be for the Model 4283 Baseboard, whose TMS320C30 processor only has a 24-bit address bus on the port it uses for the MIX bus. In this case, a page register is used for module selection (as opposed to different base addresses for each module), and the addresses are given as 6-digit hexadecimal codes, of the form 0xYY ZZZZ.

A final point must now be made about MIX nomenclature. Module 0 (or MIX position 0) refers to the closest module to the Baseboard, whether or not it is nested. Module 1 (or MIX position 1) is the module that may be mounted above Module 0, and Module 2 is the device that may be stacked above Module 1.

3.3.1 Model 4270 Address Map for the 4200/01 Series Baseboards

Pentek's Models 4200 and 4201 Baseboards use a 16 MHz 68030 processor as the MIX bus controller. The 4200A and 4201A use a 40 MHz 68030. From the 68030's viewpoint, the MIX bus is mapped identically for all four of these Baseboards. The MIX_base address for Module 0 is 0xC000 0000. For the next device in the stack, Module 1, the base address is 0xD000 0000. Finally for the last MIX Module in the stack, Module 2, the MIX_base address is 0xE000 0000.

Table 3-28, on the next page, gives the physical addresses, in the memory space of the 68030 processor on any of these four Baseboards, that correspond to the symbolic names used to represent the Model 4270's slave resources in subsequent sections of this manual.

Note that the 4200 series of VME/MIX Baseboards also support mastering of the MIX bus by Upper MIX Bus Masters (UMBMs), and by other VMEbus Masters. For further information about accessing the 4270 as a MIX slave on these Baseboards in this manner, please refer to Pentek's **MIX Tutorial** (part # 800.00001), or to the Baseboard's or UMBM's Operating Manual. Use of the Model 4270 as a UMBM is covered in Section 4.6.1 of this manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.1 Model 4270 Address Map for the 4200/01 Series Baseboards (continued)

Table 3-28: Model 4270 Memory Map for Use with Models 4200, 4200A, 4201 and 4201A MIX Baseboards- 68030 View

4200/01 68030 Address	Symbolic Name	Resource
MIX_base+0x0000 0000 - MIX_base+0x000F FFFF	SRAM_A	Global SRAM, Processor A
MIX_base+0x0020 0000 - MIX_base+0x002F FFFF	SRAM_B	Global SRAM, Processor B
MIX_base+0x0040 0000 - MIX_base+0x004F FFFF	SRAM_C	Global SRAM, Processor C
MIX_base+0x0060 0000 - MIX_base+0x006F FFFF	SRAM_D	Global SRAM, Processor D
MIX_base+0x0800 0000	MX_Rst_Ctl	MIX 'C40 Reset Control Register
MIX_base+0x0800 0004	MX_Status	MIX Interrupt and Lock Status Register (R/O)
MIX_base+0x0800 0008	MX_Int_A	Send MIX Interrupt to Processor A (W/O)
MIX_base+0x0800 000C	MX_Int_B	Send MIX Interrupt to Processor B (W/O)
MIX_base+0x0800 0010	MX_Int_C	Send MIX Interrupt to Processor C (W/O)
MIX_base+0x0800 0014	MX_Int_D	Send MIX Interrupt to Processor D (W/O)
MIX_base+0x0800 0018	Clr_Int_A_MX	Clear Interrupt from Processor A to MIX (W/O)
MIX_base+0x0800 001C	Clr_Int_B_MX	Clear Interrupt from Processor B to MIX (W/O)
MIX_base+0x0800 0020	Clr_Int_C_MX	Clear Interrupt from Processor C to MIX (W/O)
MIX_base+0x0800 0024	Clr_Int_D_MX	Clear Interrupt from Processor D to MIX (W/O)
MIX_base+0x0800 0040	MX_Int_All	Send MIX Interrupt to All Processors (W/O)

MIX_base Address:

If 4270 is in MIX position 0, MIX_base = 0xC000 000
If 4270 is in MIX position 1, MIX_base = 0xD000 000
If 4270 is in MIX position 2, MIX_base = 0xE000 000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.2 Model 4270 Address Map for the Model 4202 Baseboard

The Model 4202 is a VMEbus slave-only Baseboard, and contains no processor. This device maps the resources connected to its MIX bus into VMEbus A32, A24 or A16 slave address space. Because the Model 4270 already has its own VMEbus slave interface, using the Model 4202 Baseboard to re-map the 4270's slave resources to the VMEbus makes precious little sense.

The Model 4202 does provide support for the Model 4270 as an Upper MIX bus Master (UMBM). Use of the Model 4270 as a UMBM is covered in Section 4.6.1 of this manual. A 4270 can also be a MIX bus slave on a 4202 Baseboard that houses another 4270 or another UMBM MIX Module (e. g., a 4254 or a 4257). The MIX Memory Map for a Model 4270 on a Model 4202 Baseboard is very nearly identical to the map for the 4200 series Baseboards. Only the MIX_base addresses are different. The map is presented in Table 3-29, below.

Table 3-29: Model 4270 Memory Map for Use with the Model 4202 MIX Baseboard

4202 MIX Address	Symbolic Name	Resource
MIX_base+0x0000 0000 - MIX_base+0x000F FFFF	SRAM_A	Global SRAM, Processor A
MIX_base+0x0020 0000 - MIX_base+0x002F FFFF	SRAM_B	Global SRAM, Processor B
MIX_base+0x0040 0000 - MIX_base+0x004F FFFF	SRAM_C	Global SRAM, Processor C
MIX_base+0x0060 0000 - MIX_base+0x006F FFFF	SRAM_D	Global SRAM, Processor D
MIX_base+0x0800 0000	MX_Rst_Ctl	MIX 'C40 Reset Control Register
MIX_base+0x0800 0004	MX_Status	MIX Interrupt and Lock Status Register (R/O)
MIX_base+0x0800 0008	MX_Int_A	Send MIX Interrupt to Processor A (W/O)
MIX_base+0x0800 000C	MX_Int_B	Send MIX Interrupt to Processor B (W/O)
MIX_base+0x0800 0010	MX_Int_C	Send MIX Interrupt to Processor C (W/O)
MIX_base+0x0800 0014	MX_Int_D	Send MIX Interrupt to Processor D (W/O)
MIX_base+0x0800 0018	Clr_Int_A_MX	Clear Interrupt from Processor A to MIX (W/O)
MIX_base+0x0800 001C	Clr_Int_B_MX	Clear Interrupt from Processor B to MIX (W/O)
MIX_base+0x0800 0020	Clr_Int_C_MX	Clear Interrupt from Processor C to MIX (W/O)
MIX_base+0x0800 0024	Clr_Int_D_MX	Clear Interrupt from Processor D to MIX (W/O)
MIX_base+0x0800 0040	MX_Int_All	Send MIX Interrupt to All Processors (W/O)

MIX_base Address:

If 4270 is in MIX position 0, MIX_base = 0x0000 0000
If 4270 is in MIX position 1, MIX_base = 0x1000 0000
If 4270 is in MIX position 2, MIX_base = 0x2000 0000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.3 Model 4270 Address Map for the Model 4283 Baseboard

On the Model 4283 Baseboard, there are two levels of addressing: 1) selection of the particular expansion module by its stack position and 2) selection of one of the resources within the module.

3.3.3.1 Module Selection

The TMS320C30 selects among MIX Modules by using a two bit field in the Model 4283 Page Register, located at 'C30 address 0x80 0002. These two bits select one of the three possible MIX stack positions according to Table 3-30, below.

Table 3-30: Model 4270 - Module Selection - Model 4283 Page Register - 'C30 Address 0x80 0002

Page Register Bits>	D31 - D18	D17	D16	D15	D14	D13 - D0	Example
Stack Position 0	x	MXA27	1	1	MXA21	x	0x01 8000
Stack Position 1	x		1	0		x	0x01 0000
Stack Position 2	x		0	1		x	0x00 8000
x = other functions							

Access to some resources on certain MIX modules may require that either or both of the MIX bus A27 (MX_A27) and A21 (MX_A21) bits be set to the logic '1' state. On the 4283 Baseboard, these bits are also contained in the Address Page Register. MX_A27 is the D17 bit of this register, and MX_A21 is the D14 bit. MIX register accesses on the Model 4270 require that the MX_A27 (D17) bit be set.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.3 Model 4270 Address Map for the 4283 Baseboard (continued)

3.3.3.2 Resource Selection Within the Module

Once the module has been selected, all the slave resources of the Model 4270 appear in the TMS320C30 address map according to Table 3-31, below. Accessing these registers is accomplished by simply reading and writing these locations. The 24-bit addressing scheme accesses 32-bit data locations for the TMS320C30, even though some of the elements correspond to 8-bit devices.

Table 3-31: Model 4270 Memory Map for Use with the Model 4283 MIX Baseboard

4283 'C30 Address	Symbolic Name	Resource
0x80 4000 - 0x80 5FFF[A]	SRAM_A	Global SRAM, Processor A
0x80 4000 - 0x80 5FFF[B]	SRAM_B	Global SRAM, Processor B
0x80 4000 - 0x80 5FFF[C]	SRAM_C	Global SRAM, Processor C
0x80 4000 - 0x80 5FFF[D]	SRAM_D	Global SRAM, Processor D
0x80 4000*	MX_Rst_Ctl	MIX 'C40 Reset Control Register
0x80 4001*	MX_Status	MIX Interrupt and Lock Status Register (R/O)
0x80 4002*	MX_Int_A	Send MIX Interrupt to Processor A (W/O)
0x80 4003*	MX_Int_B	Send MIX Interrupt to Processor B (W/O)
0x80 4004*	MX_Int_C	Send MIX Interrupt to Processor C (W/O)
0x80 4005*	MX_Int_D	Send MIX Interrupt to Processor D (W/O)
0x80 4006*	Clr_Int_A_MX	Clear Interrupt from Processor A to MIX (W/O)
0x80 4007*	Clr_Int_B_MX	Clear Interrupt from Processor B to MIX (W/O)
0x80 4008*	Clr_Int_C_MX	Clear Interrupt from Processor C to MIX (W/O)
0x80 4009*	Clr_Int_D_MX	Clear Interrupt from Processor D to MIX (W/O)
0x80 4010*	MX_Int_All	Send MIX Interrupt to All Processors (W/O)

[A] - 32 pages, 8k x 32 each, pages selected by D12 - D8 in Page Register, with (D14, D13) = (0,0) and D17 = 0

[B] - 32 pages, 8k x 32 each, pages selected by D12 - D8 in Page Register, with (D14, D13) = (0,1) and D17 = 0

[C] - 32 pages, 8k x 32 each, pages selected by D12 - D8 in Page Register, with (D14, D13) = (1,0) and D17 = 0

[D] - 32 pages, 8k x 32 each, pages selected by D12 - D8 in Page Register, with (D14, D13) = (1,1) and D17 = 0

* - D17 in Page Register = 1

NOTE: The Global SRAM block size jumper block (JB11) on the MIX board must be set properly in order to access all four of the Model 4270's Global SRAM banks from a Model 4283 Baseboard. Please refer to Table 2-7, in Section 2.2.4 of this manual, for details.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.4 Model 4270 Address Map for the Model 4284 Baseboard

The Model 4284 Baseboard uses a TMS320C40 Digital Signal Processor to master transactions with the MIX bus. The three MIX module locations are each mapped into separate, 64 M-Longword regions of the 'C40's memory space. Module 0's MIX_base address in the 'C40 Memory Map is 0x2000 0000. The MIX_base address for Module 1 is 0x2400 0000, and the 'C40's MIX_base address for Module 2 is 0x2800 0000. Table 3-32, below, presents the Memory Map for the MIX slave resources on the Model 4270, as seen by the 'C40 processor on the Model 4284 Baseboard.

Table 3-32: Model 4270 Memory Map for Use with the Model 4284 MIX Baseboard

4284 'C40 Address	Symbolic Name	Resource
MIX_base+0x0000 0000 - MIX_base+0x0003 FFFF	SRAM_A	Global SRAM, Processor A
MIX_base+0x0008 0000 - MIX_base+0x000B FFFF	SRAM_B	Global SRAM, Processor B
MIX_base+0x0010 0000 - MIX_base+0x0013 FFFF	SRAM_C	Global SRAM, Processor C
MIX_base+0x0018 0000 - MIX_base+0x001B FFFF	SRAM_D	Global SRAM, Processor D
MIX_base+0x0200 0000	MX_Rst_Ctl	MIX 'C40 Reset Control Register
MIX_base+0x0200 0001	MX_Status	MIX Interrupt and Lock Status Register (R/O)
MIX_base+0x0200 0002	MX_Int_A	Send MIX Interrupt to Processor A (W/O)
MIX_base+0x0200 0003	MX_Int_B	Send MIX Interrupt to Processor B (W/O)
MIX_base+0x0200 0004	MX_Int_C	Send MIX Interrupt to Processor C (W/O)
MIX_base+0x0200 0005	MX_Int_D	Send MIX Interrupt to Processor D (W/O)
MIX_base+0x0200 0006	Clr_Int_A_MX	Clear Interrupt from Processor A to MIX (W/O)
MIX_base+0x0200 0007	Clr_Int_B_MX	Clear Interrupt from Processor B to MIX (W/O)
MIX_base+0x0200 0008	Clr_Int_C_MX	Clear Interrupt from Processor C to MIX (W/O)
MIX_base+0x0200 0009	Clr_Int_D_MX	Clear Interrupt from Processor D to MIX (W/O)
MIX_base+0x0200 0010	MX_Int_All	Send MIX Interrupt to All Processors (W/O)

MIX_base Address:

If 4270 is in MIX position 0, MIX_base = 0x2000 0000
If 4270 is in MIX position 1, MIX_base = 0x2400 0000
If 4270 is in MIX position 2, MIX_base = 0x2800 0000

Note that the Model 4284 Baseboard also supports mastering of the MIX bus by Upper MIX Bus Masters (UMBMs). For further information about accessing the 4270 as a MIX slave from another UMBM, please refer to Pentek's **MIX Tutorial** (part # 800.00001), or to the Baseboard's or UMBM's Operating Manual. Use of the Model 4270 as a UMBM is covered in Section 4.6.1 of this manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.5 Model 4270 Address Map for the Model 4285 MIX Baseboard Adapter

When the Model 4285 is equipped with one of the MIX Expansion Bus Adapters (Options 031 or 036) all eight of its 'C40 processors can access modules on the MIX bus via the 4285's Main Bus. Additionally, four of the processors (A, B, E and F) have a separate access path to devices on the MIX bus via their Global I/O expansion ports. Separate control registers are provided for each of these five interfaces to the MIX bus. Two addressing modes, Normal and Compressed, are available for any of these access paths.

The control registers mentioned above each provide a 5-bit address page field. For this Baseboard, however, the page setting is NOT used to determine which MIX module is being addressed (i. e., the direct addressing method is used). The mapping of the page field onto the address bus is a function of the selected addressing mode. For complete details, refer to the Operating Manual for the Model 4285. For purposes of this operation summary, we will say that in Normal Addressing Mode, page bit 2 is used to assert the MXA27 address bit, while in Compressed Mode, MXA27 is asserted by 'C40 Global Address bit GA20.

Table 3-33, on the next page, presents the memory map for the slave resources on the Model 4270, as seen by the 'C40 processors on the Model 4285 Baseboard. The table also provides the MIX_Base addresses, in the memory spaces of the 'C40 processors, for the Expansion bus and Global I/O access paths. For further information on the use of the 4285 as a MIX baseboard, please refer to Section 3.4 of the 4285 Operating Manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 MIX Bus Slave Interface Memory Maps (continued)

3.3.5 Address Map for the Model 4285 MIX Baseboard Adapter (continued)

Table 3-33: Model 4270 Memory Map for Use with the Model 4285 MIX Baseboard

4285 'C40 Address	Symbolic Name	Resource
MIX_base+0x0000 0000 - MIX_base+0x0003 FFFF	SRAM_A	Global SRAM, Processor A
MIX_base+0x0008 0000 - MIX_base+0x000B FFFF	SRAM_B	Global SRAM, Processor B
MIX_base+0x0010 0000 - MIX_base+0x0013 FFFF	SRAM_C	Global SRAM, Processor C
MIX_base+0x0018 0000 - MIX_base+0x001B FFFF	SRAM_D	Global SRAM, Processor D
MIX_base+0x0200 0000	MX_Rst_Ctl	MIX 'C40 Reset Control Register
MIX_base+0x0200 0001	MX_Status	MIX Interrupt and Lock Status Register (R/O)
MIX_base+0x0200 0002	MX_Int_A	Send MIX Interrupt to Processor A (W/O)
MIX_base+0x0200 0003	MX_Int_B	Send MIX Interrupt to Processor B (W/O)
MIX_base+0x0200 0004	MX_Int_C	Send MIX Interrupt to Processor C (W/O)
MIX_base+0x0200 0005	MX_Int_D	Send MIX Interrupt to Processor D (W/O)
MIX_base+0x0200 0006	Clr_Int_A_MX	Clear Interrupt from Processor A to MIX (W/O)
MIX_base+0x0200 0007	Clr_Int_B_MX	Clear Interrupt from Processor B to MIX (W/O)
MIX_base+0x0200 0008	Clr_Int_C_MX	Clear Interrupt from Processor C to MIX (W/O)
MIX_base+0x0200 0009	Clr_Int_D_MX	Clear Interrupt from Processor D to MIX (W/O)
MIX_base+0x0200 0010	MX_Int_All	Send MIX Interrupt to All Processors (W/O)
Expansion Bus MIX_base Address:		**Global I/O Bus MIX_base Address:**
If 4270 is in MIX position 0, MIX_base = 0x9000 0000		If 4270 is in MIX position 0, MIX_base = 0xC800 0000
If 4270 is in MIX position 1, MIX_base = 0x9020 0000		If 4270 is in MIX position 1, MIX_base = 0xC820 0000
If 4270 is in MIX position 2, MIX_base = 0x9040 0000		If 4270 is in MIX position 2, MIX_base = 0xC840 0000
NOTE: MXA27 is asserted via page bit 2 in Normal addressing mode, or via 'C40 address bit GA20 in Compressed addressing mode		

3.4 MIX Bus Slave Register Descriptions

The sections beginning below describe the bitwise arrangement of the MIX bus accessible registers on the Model 4270. The function of each bit or bit group in each register will be discussed in depth, and tables summarizing the register content will also be found.

3.4.1 MIX 'C40 Reset Control Register - R/W @ Address MX_Rst_Ctl

The contents of the MIX 'C40 Reset Control Register are summarized in Table 3-34, on the next page. The subsections that follow give detailed descriptions of the functions of each bit referred to in the table.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4 MIX Bus Slave Register Descriptions (continued)

3.4.1 MIX 'C40 Reset Control Register (continued)

Table 3-34: Model 4270 - MIX 'C40 Reset Control Register - R/W @ MX_Rst_Ctl							
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
1 = MIX Global Bus Lock	1 = MIX Broadcast Mode	Processor D Rst Vec 1	Processor D Rst Vec 0	1 = Reset Processor D	Processor C Rst Vec 1	Processor C Rst Vec 0	1 = Reset Processor C
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
N/U	N/U	Processor B Rst Vec 1	Processor B Rst Vec 0	1 = Reset Processor B	Processor A Rst Vec 1	Processor A Rst Vec 0	1 = Reset Processor A

3.4.1.1 MIX Global Bus Lock - Bit D15

To "lock" the data path between a MIX bus Master and any processor's Global SRAM, set the D15 bit of the MIX 'C40 Reset Control Register to the logic '1' state. Locking the Global bus gives the MIX bus Master exclusive access to the bus and all resources connected to it, and allows block transfers to occur at a faster rate. One access to the SRAM in question must be made before the path can be locked. For further information about the usage and limitations of the Global bus lock feature, please refer to Section 4.7.

When finished with the block transfer, the MIX bus Master should clear this bit to the logic '0' state, to unlock the Global bus and allow other devices to access the SRAMs. Table 3-35, below, summarizes the registers and bits involved in Global bus locking.

Table 3-35: Model 4270 - Global Bus Lock Registers		
Device	**Global Bus Lock**	**Global Bus Unlock**
Processors	Write @ 0x0034 000B	Write @ 0x0034 000A
VMEbus	A16base + 0, D15 = 1	A16base + 0, D15 = 0
MIX bus	MX_Rst_Ctl, D15 = 1	MX_Rst_Ctl, D15 = 0

3.4.1.2 MIX Broadcast Mode Enable - Bit D14

Setting the D14 bit of the MIX 'C40 Reset Control Register to the logic '1' state will cause the data written over the MIX bus to Processor A to also be written to Processors B, C, and D. This is known as MIX Broadcast Mode. MIX Broadcast Mode is disabled by clearing this bit to the logic '0' state.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4 MIX Bus Slave Register Descriptions (continued)

3.4.1 MIX 'C40 Reset Control Register (continued)

3.4.1.3 Processor Reset - Bits D13 thru D8 and D5 thru D0

'C40 Processor A will be held in the reset state whenever bit D0 of the MIX 'C40 Reset Control Register is set to the logic '1' state. Processor B will remain reset as long as bit D3 of this register is set to the logic '1' state. A logic '1' in D8 will hold Processor C reset, and a logic '1' in bit D11 holds Processor D reset. To initiate a reset cycle in any processor, toggle the appropriate bit from logic '0' to logic '1', and then back to logic '0' again. The reset cycle initiated by toggling one of these bits will reset the associated 'C40 processor ONLY. This reset has no effect upon the MIX or VME interface logic for the processor in question. The action taken by a processor when it is reset depends upon the states of that processor's Reset Vector bits.

The Reset Vector bits in the Reset Control Register drive the RESETLOC(1,0) pins on the 'C40 Processors. These pins tell the processor where to look to find the address of the initialization code to be run when the RESET input is driven low. See Section 2.2.3 or 4.2 of this manual for more details. For a complete description of 'C40 reset operation, See the Texas Instruments **TMS320C4x User's Guide**, Sections 6.7 (Reset Operation) and 12.1 (Processor Initialization).

Tables 3-36 and 3-37, below, summarize the operation of the Reset bits in the MIX 'C40 Reset Control Register.

Table 3-36: Model 4270 - Processor Reset Bits - R/W @ MX_Rst_Ctl

To Reset Processor...	Write '1', then '0' to bit...	Reset Vector Select bit 0	Reset Vector Select bit 1
A	D0	D1	D2
B	D3	D4	D5
C	D8	D9	D10
D	D11	D12	D13

Table 3-37: Model 4270 - Reset Vector Select Bits

Reset Vector Select bit 1	Reset Vector Select bit 0	Memory Block	'C40 Reset Vector Location
0	**0**	Local	0x0000 0000
0	**1**	Local	0x7FFF FFFF
1	**0**	Global	0x8000 0000
1	**1**	Global	0xFFFF FFFF

3.4.2 MIX Interrupt and Bus Lock Status Register - R/O @Address MX_Status

Table 3-38, on the following page, summarizes the bits in the MIX Interrupt and Bus Lock Status register. The text that follows provides a detailed explanation of the uses of the bits in the register.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4 MIX Bus Slave Register Descriptions (continued)

3.4.2 MIX Interrupt and Bus Lock Status Register (continued)

Table 3-38: Model 4270 - MIX Interrupt and Bus Lock Status Register - R/O @ MX_Status

D15	D14	D13	D12	D11	D10	D9	D8
N/U	N/U	1 = Global Bus Locked by VME	1 = Global Bus Locked by MIX	1 = Global Bus Locked by Processor D	1 = Global Bus Locked by Processor C	1 = Global Bus Locked by Processor B	1 = Global Bus Locked by Processor A
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
N/U	N/U	N/U	N/U	1 = Proc. D Interrupt to MIX	1 = Proc. C Interrupt to MIX	1 = Proc. B Interrupt to MIX	1 = Proc. A Interrupt to MIX

3.4.2.1 Global Bus Locks - Bits D13 thru D8

When one of these bits reads back in the logic '0' state, it indicates that the associated resource has locked the 4270's Global bus, to allow faster block transfers. D13 of this Status Register will be in the high state when a VME Master has locked the Global bus (see Section 3.5.2.1 for details). While a MIX bus Master has the Global bus locked, D12 will be set (see Section 3.4.1.1 for more information).

A '1' in D11 says that Processor D has locked the Global bus, and D10 will be high when Processor C locks the bus. If D9 is set, then Processor B has locked the Global bus, and a '1' in D8 says the same about Processor A (see Section 3.1.2.3 for further information). Uses and limitations of the Global bus lock feature are covered in Section 4.7 of this manual.

3.4.2.2 Interrupt Source Identification - Bits D3 thru D0

When an interrupt is issued over the MIX bus by any of the processors on the 4270, these bits are used to indicate which processor set the interrupt. If D3 is a '1', the interrupt came from Processor D. If the interrupt was set by Processor C, then D2 will be high. An interrupt from Processor B will put a '1' in D1 of this register, and if D0 is high, then Processor A issued the interrupt.

These bits are reset when the interrupt is cleared. To clear a MIX interrupt, the MIX bus Master must write to the appropriate Clear Interrupt Register. Table 3-40, at the bottom left of the following page, tells which register to write, in order to clear an interrupt from a given processor. The data written to these registers is not used; simply write anything you choose to the indicated address.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.4 MIX Bus Slave Register Descriptions (continued)

3.4.3 MIX Command Registers

The text and tables below summarize the addresses and functions of the Command Registers associated with the MIX bus interface on the Model 4270. In all cases, just write to the given address to cause the indicated function to execute.

3.4.3.1 MIX Set Interrupt Registers
- W/O @ Addresses MX_Int_A thru MX_Int_D and MX_Int_All

Table 3-39: Model 4270 - MIX Set Interrupt Registers

Write to this address...	to Interrupt Processor...
MX_Int_A	A
MX_Int_B	B
MX_Int_C	C
MX_Int_D	D
MX_Int_All	A, B, C & D

A MIX bus Master may interrupt any of the 'C40 processors on the 4270 by writing to the appropriate Set Interrupt Register. During normal operation, this interrupt is delivered to the 'C40's $\overline{\text{IIOF1}}$ input. Table 3-39, left, tells which address must be written, in order to interrupt a given processor. The processor receiving this interrupt must clear it, by writing to address 0x0034 0003 (the Clear MIX Interrupt Register) in its own Local memory.

Writing to address MX_Int_All will interrupt all four processors. This interrupt is reset when Processor D writes to its Clear Interrupt Register at 0x0034 0003.

3.4.3.2 MIX Clear Interrupt Registers
- W/O @ MIX_base + 0x0800 0018 - MIX_base + 0x0800 0024

Table 3-40: Model 4270 - MIX Clear Interrupt Registers

Write to this address...	to Clear Int from Proc...
Clr_Int_A_MX	A
Clr_Int_B_MX	B
Clr_Int_C_MX	C
Clr_Int_D_MX	D

After receiving an interrupt from any of the processors on the 4270, a MIX bus Master must clear the interrupt by writing to the appropriate Clear Interrupt Register. The processors set interrupts to the MIX bus by writing to address 0x0034 0008, in their own Local memory. Table 3-40, left, tells which address the MIX Master must write, in order to clear an interrupt, depending upon which processor set it.

3.5 VMEbus Slave Interface Memory Map

The VME Slave Interface Memory Map for the Model 4270 is described by Table 3-41, on the following page. This memory map includes the Global SRAMs for each Processor, which are accessible in A24 or A32 space, as well as the VMEbus control, status and command registers, which are accessed in A16 space. In the text and tables that follow, A16_base is the VME A16 base address setting, set at jumper block S1. See Section 2.2.1 for more information on the setting of this jumper block. VME_base is the A24 or A32 base address, set by writing to the VME Address Mapping and Control Register, at address A16_base+0x0004. See Section 3.5.3 for information abvout this register.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

Table 3-41: Model 4270 - VMEbus Slave Memory Map

VMEbus Address	Symbolic Name	Resource
VME_base+0x0000 0000 - VME_base+0x000F FFFF	SRAM_A	Global SRAM, Processor A
VME_base+0x0020 0000 - VME_base+0x002F FFFF	SRAM_B	Global SRAM, Processor B
VME_base+0x0040 0000 - VME_base+0x004F FFFF	SRAM_C	Global SRAM, Processor C
VME_base+0x0060 0000 - VME_base+0x006F FFFF	SRAM_D	Global SRAM, Processor D
A16_base+0x0000	VME_Rst_Ctl	VME 'C40 Reset Control Register
A16_base+0x0004	VME_Map	VME Address Mapping and Control Register
A16_base+0x0008	VME_Page	VME Address Page Register
A16_base+0x000C	VME_Status	VME Interrupt and Lock Status Register (R/O)
A16_base+0x0010	VME_Int_A	Send VME Interrupt to Processor A (W/O)
A16_base+0x0014	VME_Int_B	Send VME Interrupt to Processor B (W/O)
A16_base+0x0018	VME_Int_C	Send VME Interrupt to Processor C (W/O)
A16_base+0x001C	VME_Int_D	Send VME Interrupt to Processor D (W/O)
A16_base+0x0020	Clr_Int_A_VME	Clear Interrupt from Processor A to VME (W/O)
A16_base+0x0024	Clr_Int_B_VME	Clear Interrupt from Processor B to VME (W/O)
A16_base+0x0028	Clr_Int_C_VME	Clear Interrupt from Processor C to VME (W/O)
A16_base+0x002C	Clr_Int_D_VME	Clear Interrupt from Processor D to VME (W/O)
A16_base+0x003C	VME_Int_All	Send VME Interrupt to All Processors (W/O)
A16_base+0x0040	VME_IACK	VMEbus IACK Vector Address Register
A16_base+0x0080 - A16_base+00FC	TBC	Test Bus Control Registers

Base Addresses:
A16_base is set at jumper block S1 (see Section 2.2.1)
VME_base is programmable, by writing to the VME_Map register at A16_base+0x0004 (see Section 3.5.3)

3.5.1 VMEbus Address Modifiers

There are three addressing modes defined for VMEbus transfers: 32-bit, 24-bit, and 16-bit modes. These modes are provided to support a variety of bus devices ranging from simple I/O functions with only a few memory locations to large RAM arrays with hundreds of megabytes of storage.

The addressing mode is determined by a special 6-bit code which is placed on six dedicated VMEbus lines by the current VMEbus Master and is known as the Address Modifier Code, or AM Code.

The VMEbus specification defines the usage of certain AM codes. Table 3-42, on the next page, lists the Address Modifier codes that the Model 4270 will respond to. Codes not listed in the table will not be recognized by the Model 4270.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.1 VMEbus Address Modifiers (continued)

Table 3-42: Model 4270 - Address Modifier Codes	
AM Code	**Function**
0x3E	A24 - Standard Supervisory Program Access
0x3D	A24 - Standard Supervisory Data Access
0x3A	A24 - Standard Nonprivileged Program Access
0x39	A24 - Standard Nonprivileged Data Access
0x2D	A16 - Short Supervisory Access
0x29	A16 - Short Nonprivileged Access
0x0E	A32 - Extended Supervisory Program Access
0x0D	A32 - Extended Supervisory Data Access
0x0A	A32 - Extended Nonprivileged Program Access
0x09	A32 - Extended Nonprivileged Data Access

3.5.2 VME 'C40 Reset Control Register - R/W @ A16_base+0x0000 (VME_Ctl)

The contents of the VME 'C40 Reset Control Register are summarized in Table 3-43, below. The subsections beginning on the next page give detailed descriptions of the functions of each bit referred to in the table.

Table 3-43: Model 4270 - VME 'C40 Reset Control Register - R/W @ A16_base+0x0000							
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
1 = VME Global Bus Lock	1 = VME Broadcast Mode	Processor D Rst Vec 1	Processor D Rst Vec 0	1 = Reset Processor D	Processor C Rst Vec 1	Processor C Rst Vec 0	1 = Reset Processor C
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
1 = Test Bus Controller Enable	0 = Test Bus Controller Reset	Processor B Rst Vec 1	Processor B Rst Vec 0	1 = Reset Processor B	Processor A Rst Vec 1	Processor A Rst Vec 0	1 = Reset Processor A

3.5.2.1 VME Global Bus Lock - Bit D15

To "lock" the data path between a VMEbus Master and any processor's Global SRAM, set bit D15 of the VME 'C40 Reset Control Register, at the A16 base address, to the logic '1' state. Locking the 4270's Global bus gives a VMEbus Master exclusive access to the bus and all resources connected to it, and allows block transfers to occur at a faster rate. One access to the SRAM in question must be made before the bus can be locked. For further information about the usage and limitations of the Global bus lock feature, please refer to Section 4.7.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.2 VME 'C40 Reset Control Register (continued)

3.5.2.1 VME Global Bus Lock - Bit D15 (continued)

When finished with the block transfer, the VMEbus Master should clear this bit to the logic '0' state, to unlock the Global bus and allow other devices to access the SRAMs. Table 3-44, below, summarizes the registers and bits involved in Global bus locking.

Table 3-44: Model 4270 - Global Bus Lock Registers

Device	Global Bus Lock	Global Bus Unlock
Processors	Write @ 0x0034 000B	Write @ 0x0034 000A
VMEbus	A16base+0, D15 = 1	A16base+0, D15 = 0
MIX bus	MX_Rst_Ctl, D15 = 1	MX_Rst_Ctl, D15 = 0

3.5.2.2 VME Broadcast Mode Enable - Bit D14

Setting the D14 bit of the VME 'C40 Reset Control Register to the logic '1' state will cause the data written to Processor A to also be written to Processors B, C, and D. This is known as Broadcast Mode. Broadcast Mode is disabled by clearing this bit to the logic '0' state.

3.5.2.3 Processor Reset - Bits D13 thru D8 and D5 thru D0

'C40 Processor A will be held in the reset state whenever bit D0 of the VME 'C40 Reset Control Register is set to the logic '1' state. Processor B will remain reset as long as bit D3 of this register is set to the logic '1' state. A logic '1' in D8 will hold Processor C reset, and a logic '1' in bit D11 holds Processor D reset. To initiate a reset cycle in any processor, toggle the appropriate bit from logic '0' to logic '1', and then back to logic '0' again. The reset cycle initiated by toggling one of these bits will reset the associated 'C40 processor ONLY. This reset has no effect upon the VME or MIX interface logic for the processor in question. The action taken by a processor when it is reset depends upon the states of that processor's Reset Vector bits.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.2 VME 'C40 Reset Control Register (continued)

3.5.2.3 Processor Reset (continued)

The Reset Vector bits in the Reset Control Register drive the RESETLOC(1,0) pins on the 'C40 Processors. These pins tell the processor where to look to find the address of the initialization code to be run when the RESET input is driven low. See Section 2.2.3 or 4.2 of this manual for more details. For a complete description of 'C40 reset operation, see the Texas Instruments' **TMS320C4x User's Guide**, Sections 6.7 and 12.1.

Tables 3-43 and 3-43, below, summarize the operation of the reset bits in the VME 'C40 Reset Control Register.

Table 3-45: Model 4270 - Processor Reset Bits - R/W @ A16_base+0x0000

To Reset Processor...	Write '1', then '0' to bit...	Reset Vector Select bit 0	Reset Vector Select bit 1
A	D0	D1	D2
B	D3	D4	D5
C	D8	D9	D10
D	D11	D12	D13

Table 3-46: Model 4270 - Reset Vector Select Bits

Reset Vector Select bit 1	Reset Vector Select bit 0	Memory Block	'C40 Reset Vector Location
0	**0**	Local	0x0000 0000
0	**1**	Local	0x7FFF FFFF
1	**0**	Global	0x8000 0000
1	**1**	Global	0xFFFF FFFF

3.5.2.4 VME Test Bus Controller - Bits D7 and D6

The Texas Instruments SN74ACT8990 is utilized as the Test Bus Controller in the 4270. If bit D7 of the VME 'C40 Reset Control register is set to the logic '1' state, the Controller is enabled. By default, this bit is initialized to the logic '0' state, disabling the part. Bit D6 of the Reset Control register is used to reset the controller. When D6 is cleared to the logic '0' state, the controller will be held in the reset state (this is the default condition). Thus, to make the controller operational, D7 and D6 must both be set to the logic '1' state.

Note that the Test Bus Controller should **NEVER** be enabled if the XDS-510 Emulator is connected to the 4270. See Section 3.5.8 for more information about the Test Bus Controller.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.3 VME Address Mapping and Control Register - R/W @ A16_base+0x0004 (VME_Map)

The VME Address Mapping and Control Register, at the A16 VME address offset 0x0004, contains the base address of the 4270's Global SRAM in A24 or A32 space (referred to as VME_base in the Global Memory Map, Table 3-27, and the VME Slave Memory Map, Table 3-41). It also contains a bit that allows the user to configure the board to communicate with either an Intel or a Motorola card cage controller (the Word Swap bit), and another bit to enable Paged Memory mode. Note that VMEbus access to the 4270 in both A24 and A32 space is blocked until **after** the first A16 access. The first A16 access should, therefore, be to write the desired VME_base address to this register. Table 3-47, below, summarizes the bit configuration of this register, and the subsections that follow describe the usage of those bits.

Table 3-47: Model 4270 - VME Address Mapping and Control Register - R/W @ A16_base + 0x04

D15	D14	D13	D12	D11	D10	D9	D8
← VME_base Address Selection - A32 Space →							
A31	A30	A29	A28	A27	A26	A25	A24
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
← VME_base Address Selection - A24 Space →				N/U	N/U	1 = Paged Addressing Mode	1 = Word Swap (Intel) Mode
A23	A22	A21	A20				

3.5.3.1 VME Base Address Selection - Bits D15 thru D4

The upper 12 bits of the Address Mapping and Control Register are used to determine the Base Address used by VMEbus Masters when accessing the Global SRAM on the Model 4270. Bits D7 - D4 set the upper four bits (A23 - A20) of the A24 base address. The upper eight bits of the A32 base address (A31 - A24) are set by D15 - D8 in this register.

Table 3-48, below, lists the address bits set by D7 - D4 and their hex weights. Table 3-49, at the top of the next page, summarizes all available A24 SRAM base addresses that may be set with these bits. Note that many of the addresses listed in this table are not valid for SRAM window sizes greater than 1 MByte.

Table 3-48: Model 4270 - A24 SRAM Base Address - Hex Weights of Address Bits Set in VME Address Mapping and Control Register

D7	D6	D5	D4
A23 (0x80 0000)	A22 (0x40 0000)	A21 (0x20 0000)	A20 (0x10 0000)

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.3 VME Address Mapping and Control Register (continued)

3.5.3.1 VME Base Address Selection Bits (continued)

Table 3-49: Model 4270 - A24 Global SRAM Address - Bit Settings for Available Base Addresses - 1 MByte SRAM Window

Hex Address	D7 (A23)	D6 (A22)	D5 (A21)	D4 (A20)	Hex Setting*
0x00 0000	0	0	0	0	0x0z
0x10 0000	0	0	0	1	0x1z
0x20 0000	0	0	1	0	0x2z
0x30 0000	0	0	1	1	0x3z
0x40 0000	0	1	0	0	0x4z
0x50 0000	0	1	0	1	0x5z
0x60 0000	0	1	1	0	0x6z
0x70 0000	0	1	1	1	0x7z
0x80 0000	1	0	0	0	0x8z
0x90 0000	1	0	0	1	0x9z
0xA0 0000	1	0	1	0	0xAz
0xB0 0000	1	0	1	1	0xBz
0xC0 0000	1	1	0	0	0xCz
0xD0 0000	1	1	0	1	0xDz
0xE0 0000	1	1	1	0	0xEz
0xF0 0000	1	1	1	1	0xFz

* for low byte of register...z = setting of D1 & D0

As mentioned earlier (in Section 2.2.4), the proper setting of these bits is dependent upon the setting of the VME SRAM Window size jumper block, JB11 on the VME board. The bits set by this block are compared to VME address bits to determine if this is the board being addressed.

JB11 connects the VME address bits corresponding to the bits set in this register to an address comparator. Bits not connected by JB11 (i. e., jumpers NOT installed on JB11) are pulled high at the comparator's input. Thus, to match these bits, for a 2 MByte VME SRAM window setting on JB11, D4 in this register must be set to the logic '1' state. For a 4 MByte SRAM window, D5 and D4 must both be set to the logic '1' state. For an 8 MByte SRAM window, D6, D5 and D4 must be set to the logic '1' state. For the 16 MByte window size, all four bits (D7 - D4) must be set to the logic '1' state, and the A24 base address is fixed at 0x00 0000.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.3 VME Address Mapping and Control Register (continued)

3.5.3.1 VME Base Address Selection Bits (continued)

Tables 3-50 through 3-53, below, list the register settings for valid VME SRAM base addresses when the SRAM window size is greater than 1 MByte.

Table 3-50: Model 4270 - A24 Global SRAM Base Address - Bit Settings for Valid Base Addresses - 2 MByte SRAM Window

Hex Address	D7 (A23)	D6 (A22)	D5 (A21)	D4 (A20)	Hex Setting*
0x00 0000	0	0	0	1	0x1z
0x20 0000	0	0	1	1	0x3z
0x40 0000	0	1	0	1	0x5z
0x60 0000	0	1	1	1	0x7z
0x80 0000	1	0	0	1	0x9z
0xA0 0000	1	0	1	1	0xBz
0xC0 0000	1	1	0	1	0xDz
0xE0 0000	1	1	1	1	0xFz

* for low byte of register...z = setting of D1 & D0

Table 3-51: Model 4270 - A24 Global SRAM Base Address - Bit Settings for Valid Base Addresses - 4 MByte SRAM Window

Hex Address	D7 (A23)	D6 (A22)	D5 (A21)	D4 (A20)	Hex Setting*
0x00 0000	0	0	1	1	0x3z
0x40 0000	0	1	1	1	0x7z
0x80 0000	1	0	1	1	0xBz
0xC0 0000	1	1	1	1	0xFz

* for low byte of register...z = setting of D1 & D0

Table 3-52: Model 4270 - A24 Global SRAM Base Address - Bit Settings for Valid Base Addresses - 8 MB SRAM Window

Hex Address	D7 (A23)	D6 (A22)	D5 (A21)	D4 (A20)	Hex Setting*
0x00 0000	0	1	1	1	0x7z
0x80 0000	1	1	1	1	0xFz

* for low byte of register...z = setting of D1 & D0

Table 3-53: Model 4270 - A24 Global SRAM Base Address - Bit Settings for Valid Base Address - 16 MB SRAM Window

Hex Address	D7 (A23)	D6 (A22)	D5 (A21)	D4 (A20)	Hex Setting*
0x00 0000	1	1	1	1	0xFz

* for low byte of register...z = setting of D1 & D0

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.3 VME Address Mapping and Control Register (continued)

3.5.3.1 VME Base Address Selection Bits (continued)

Table 3-54, below, lists the hex weights of the address bits set by the upper byte of this register, D15 - D8, which is used to set the A32 base address for the 4270. The 4270's Global SRAM always occupies a 16 MByte region in A32 VME space, regardless of the settings of the window size jumper block, JB11. Also, note that the A24 base address bits (D7 - D4 in this register) have no effect on the A32 base address.

Table 3-54: Model 4270 - A32 Global SRAM Base Address - VME Address Mapping and Control Register - Data Bits and their Hexadecimal Address Weights

D15	D14	D13	D12	D11	D10	D9	D8
A31 (0x8000 0000)	A30 (0x4000 0000)	A29 (0x2000 0000)	A28 (0x1000 0000)	A27 (0x0800 0000)	A26 (0x0400 0000)	A25 (0x0200 0000)	A24 (0x0100 0000)

3.5.3.2 Paged Addressing Mode - Bit D1

There are two types of VME base address selection possible. Direct addressing is accomplished with the Paged Addressing Mode bit (D1) set to '0'. In Direct Mode, the 4270 responds to addresses that exactly match the address bits set in the Address Mapping Register. Again, for VME A24 space, in the factory default configuration, the device takes up 8 Mbytes of VME memory. Since this is half of the available space in A24, this may present a problem. The A20, A21 and A22 bits MUST be set high and the rest of the bits specify the base address.

Paged Addressing allows the board to occupy only 1 MByte of VME memory, and is accomplished with the Page Mode bit (D14 at A16_base+0x0004) set to the logic '1' state. Then the memory on the board can be addressed in 1 MByte pages using bits set in the Page Register, at A16_base+0x0008 (see Section 3.5.4).

3.5.3.3 Word Swap Mode - Bit D0

Word Swap Mode is used to allow Intel type processors to access the 4270 memory, by setting bit D0 in this register to the logic '1' state. Leave this bit in the logic '0' state if you are working with a Motorola host.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.3 VME Address Mapping and Control Register (continued)

3.5.3.3 Word Swap Mode Bit (continued)

This is a very common function in VME cards. Remember that Intel and Motorola processors address words differently; with an Intel host, the upper 16 bits of the 32 bit longword are stored at locations with the VME address line A1 = '1'. A Motorola host stores those same bits at locations with A1 = '0'. This can confuse the 'C40s, since they can only read and write 32 bit longwords.

3.5.4 Address Page Register - R/W @ A16_base + 0x0008 (VME_Page)

This 8-bit register is used in conjunction with the Paged Addressing Mode Bit in the Address Mapping and Control Register (D1 at A16_base+0x0004). When that bit is set to the logic '1' state, paged addressing is enabled. This mode allows the 4270 to occupy less space in the VME A24 address map, thus allowing more room for other devices. When paged mode is enabled, this register is used to address the 4270's memory in 1 MByte pages.

There are bits in the Page Register corresponding to VME Address bits A20 - A27. In practice, the only bits that will be used in this register are the bits representing A21 and A22 (i. e., D1 and D2). When both of those bits are cleared to the logic '0' state, the page register points at Processor A's Global SRAM bank. When D2 is cleared to the logic '0' state and D1 is set to the logic '1' state, Processor B's bank is pointed to. Processor C's Global SRAM would be accessed with D2 set to the logic '1' state and D1 cleared to the logic '0' state. The SRAM for Processor D would be selected when both bits are set to the logic '1' state. Table 3-55, below shows the correspondence between the bits in this register and the VME Address lines.

Table 3-55: Model 4270 - VME Address Page Register - R/W @ A16_base+0x0008

D7	D6	D5	D4	D3	D2, D1 = Proc...	D0
A27	A26	A25	A24	A23	0, 0 = Proc. A 0, 1 = Proc. B 1, 0 = Proc. C 1, 1 = Proc. D	A20

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.5 VME Interrupt and Bus Lock Status Register - R/O @ A16_base+0x000C (VME_Status)

Table 3-56, below, summarizes the bits in the VME Interrupt and Bus Lock Status Register, at A16_base+0x000C. The subsections that follow the table provide a detailed explanation of how the bits in this register are used.

Table 3-56: Model 4270 - VME Interrupt and Bus Lock Status Register - R/O @ A16_base+0x000C

D15	D14	D13	D12	D11	D10	D9	D8
0 = TBC Ready	N/U	1=Global Bus Locked by VME	1=Global Bus Locked by MIX	1=Global Bus Locked by Processor D	1=Global Bus Locked by Processor C	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
N/U	N/U	N/U	1 = Interrupt, TBC to VME	1 = Interrupt, Processor D to VME	1 = Interrupt, Processor C to VME	1 = Interrupt, Processor B to VME	1 = Interrupt, Processor A to VME

3.5.5.1 TBC Ready - Bit D15

This bit will read back in the logic '0' state when the SN74ACT8990 Test Bus Controller is ready for use. This bit will be low when the D6 and D7 bits (TBC Enable and TBC Reset) in the VME 'C40 Reset Control Register, at the A16_base address, are both set to the logic '1' state, and a clock is applied to the TBC from either the J-TAG connector (see Section 2.7), or by installing a jumper on JB22 on the VME board.

3.5.5.2 Global Bus Locks - Bits D13 - D8

Any of these bits will read back in the logic '1' state when the indicated resource has locked the 4270's Global data bus, to allow faster block transfers. D13 of this Status Register will be in the logic '1' state when a VME Master has locked the Global bus (see Section 3.5.2.1 for details). While a MIX bus Master has the Global bus locked, D12 will be set (see Section 3.4.1.1 for more information).

A logic '1' in D11 says that Processor D has locked the Global bus, and D10 will be high when Processor C locks the bus. If D9 is in the logic '1' state, then Processor B has locked the Global bus, and a logic '1' in D8 says the same about Processor A (see Section 3.1.2.3 for further information). Uses and limitations of the Global bus lock feature are discussed in Section 4.7 of this manual.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.5 VME Interrupt and Bus Lock Status Register (continued)

3.5.5.3 Interrupt Source Identification - Bits D4 - D0

Table 3-57: Model 4270 - VME Interrupt ID Bits

Status Bit	Interrupter
D4	TBC
D3	D
D2	C
D1	B
D0	A

When an interrupt is issued over the VMEbus by any of the resources on the 4270, these bits are used to indicate which one set the interrupt. If the D4 bit is high, then the interrupt was issued by the Test Bus Controller. If D3 is a '1', the interrupt came from Processor D. If the interrupt was set by Processor C, then D2 will be high. An interrupt from Processor B will put a '1' in D1 of this register, and if D0 is high, then Processor A issued the interrupt. Table 3-57, left, summarizes the definitions of the Interrupt Identification Status Bits.

3.5.6 VME Command Registers

The text and tables below summarize the addresses and functions of the Command Registers associated with the VMEbus Slave interface on the Model 4270. In all cases, just write to the given address to cause the indicated function to execute.

3.5.6.1 VME Set Interrupt Registers - W/O @ A16_base+0x0010 thru A16_base+0x001C (VME_Int_A - VME_Int_D) & A16_base+0x003C (VME_Int_All)

Table 3-58: Model 4270 - VME Set Interrupt Registers

Write to this address...	to Interrupt Processor...
A16_base+0x0010	A
A16_base+0x0014	B
A16_base+0x0018	C
A16_base+0x001C	D
A16_base+0x003C	A, B, C & D

A VMEbus Master may interrupt any of the 'C40 processors on the 4270 by writing to the appropriate Set Interrupt Register. During normal operation, this interrupt is delivered to the 'C40's $\overline{\text{IIOF1}}$ input. Table 3-58, left, tells which address must be written, in order to interrupt a given processor. The processor receiving this interrupt must clear it by writing to address 0x0034 0004 (the Clear VME Interrupt Register) in its own Local memory.

Writing to A16_base + 0x3C will interrupt all four processors. This interrupt is reset when Processor D writes to its Clear Interrupt Register at 0x0034 0004.

3.5.6.2 VME Clear Interrupt Registers - W/O @ A16_base+0x0020 thru A16_base+0x002C (Clr_Int_A_VME - Clr_Int_D_VME)

Table 3-59: Model 4270 - VME Clear Interrupt Registers

Write to this address...	to clear Int. from Proc...
A16_base+0x0020	A
A16_base+0x0024	B
A16_base+0x0028	C
A16_base+0x002C	D

After receiving an interrupt from any of the processors on the 4270, a VMEbus Master must clear the interrupt by writing to the appropriate Clear Interrupt Register. The processors set interrupts to the VMEbus by writing to address 0x0034 0009, in their own Local memory. Table 3-59, left, tells which address the VME Master must write, in order to clear the interrupt, depending upon which processor set it.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.5 VMEbus Slave Interface Memory Map (continued)

3.5.7 VME IACK Vector Address Register

- R/W @ A16_base+0x0040 (VME_IACK)

This 8-bit register is used to store the IACK Vector Address (VA in the table below), which will be placed on the VME Data bus when a resource on the 4270 issues an interrupt to VME. Table 3-60, below, shows the correspondence between the bits in this register and the VME Data Bus lines that are their destinations.

Table 3-60: Model 4270 - VME IACK Vector Address Register - R/W @ A16_base+0x0040

D7	D6	D5	D4	D3	D2	D1	D0
VA7	VA6	VA5	VA4	VA3	VA2	VA1	VA0

3.5.8 Test Bus Controller Registers

- R/W @ A16_base+0x0080 thru A16_base+0x00FC (TBC)

The Texas Instruments SN74ACT8990 is provided on the Model 4270 for use as a JTAG Test Bus Controller. That device interfaces a host computer to a 4-wire serial test bus on the 4270, compliant with the definition in IEEE Standard 1149.1. The standard defines a flexible test structure, combining control and interface circuits, and allowing complex systems to be partitioned into subgroups for efficient testing. This device may also be used as a JTAG Interface. It may be used as an alternative to the XDS-510 Emulator, but use of these devices is mutually exclusive (i. e., you may use one or the other at any given time, **but not both**). Contact Texas Instruments and request Application Report EB203 for more information about the device.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Chapter 4: Working with the Model 4270

4.1 Introduction

This chapter draws on (and, in some cases, repeats) information presented in the preceding chapters to provide the user with operating hints and tips. It is our hope that this will help in obtaining the maximum utility from this powerful processing device in your particular application.

4.2 Resetting and Booting the Model 4270's Processors

When the Model 4270 is reset, by any means, the RESET inputs on the TMS320C40s are pulled to a logic low. This activation of the RESET input places the 'C40 processors in the state described in Table 6-4 (Pin States After System Reset), in Section 6.7 (Reset Operation) of the Texas Instruments **TMS320C4x User's Guide**. When the Reset is released, control of the 'C40 passes to the code found at the Reset Vector location. This location is determined by the contents of the Rst_Vec(1,0) bits for each processor in the 'C40 Reset Control Registers (see Table 3-33, in Section 3.4.1, or Table 3-42 in Section 3.5.2 of this manual). The reset operation also clears these two bits, thus pointing to the 'C40's Internal ROM at location 0x0000 0000. The program found there instructs the 'C40 to read its $\overline{\text{IIOF}}$(1-3) pins and, based on the result of that read and an internally stored table (see Table 4-2, below), jump to an address where it will find the code to boot load.

The 'C40s' $\overline{\text{IIOF}}$(1-3) pins are driven by the outputs of multiplexers. During resets, one of these multiplexers transfers the logic states found at jumper block JB2 on the VME board to the $\overline{\text{IIOF}}$ pins of Processor A. Another transfers the settings of jumper block JB7 on the VME board to the $\overline{\text{IIOF}}$ pins on Processor B. A third Mux sends the settings of JB2 on the MIX board to Processor C's $\overline{\text{IIOF}}$ pins, and a fourth drives Processor D's $\overline{\text{IIOF}}$ pins with the logic states found at JB7 on the MIX board. The association of Boot Address jumper blocks with processors is summarized in Table 4-1, below. For all of these jumper blocks, an installed jumper is a logic '0', and an absent jumper is a logic '1'. The factory settings of these jumper blocks set the jump locations for all four processors to 0x0030 0000, which is the beginning address of each processor's EPROM.

Table 4-1: Model 4270 - Boot Address Jumper Blocks

Board	JB2	JB7
VME	Proc. A	Proc. B
MIX	Proc. C	Proc. D

Table 4-2: Model 4270 - 'C40 Boot Code Address Jumper Settings

Address	Location	Pins 1 & 2	Pins 3 & 4	Pins 5 & 6
0x0030 0000*	Ext. PROM	ON	OFF	OFF
0x4000 0000	Local SRAM	OFF	ON	OFF
0x6000 0000	Do Not Use	ON	ON	OFF
0x8000 0000	Do Not Use	OFF	OFF	ON
0xA000 0000	Do Not Use	ON	OFF	ON
0xC000 0000	Do Not Use	OFF	ON	ON
COMM PORT	Comm Port	OFF	OFF	OFF
N/A	Do Not Use	ON	ON	ON
* - Factory Default Setting				

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 Resetting and Booting the Model 4270's Processors (continued)

The code provided in the EPROM, after it is boot loaded, initializes some of the 'C40s' internal registers, among them the Local and Global Bus Interface Control Registers. Pointers to the Look, Set and other useful functions contained in the EPROM are planted in the Global SRAM (the functions themselves are stored in the 'C40's Internal RAM). These are described in Table 4-3, below. The EPROM code also initializes the registers on the 4270 that are external to the 'C40s, as described in Table 4-4, on the next page.

Table 4-3: Model 4270 - SRAM Locations of Pointers to EPROM Functions

Host Slave Offset (MIX/VME_base+...)	'C40 Address	Contents
0x00y0 0800	**0x80zz 0200**	Function (command) address from Host
0x00y0 0804	**0x80zz 0201**	1st Command Argument - Address for Look/Set function or Source Address for block move or Global Boot Address
0x00y0 0808	**0x80zz 0202**	Data from Host for set function or Destination address for block move
0x00y0 080C	**0x80zz 0203**	2nd Command Argument - Data to Host for Look or Word count -1 for block move
0x00y0 0810	**0x80zz 0204**	Ready to Host (0=not ready / 1=ready)
0x00y0 0814	**0x80zz 0205**	Pointer to LOOK function
0x00y0 0818	**0x80zz 0206**	Pointer to SET function
0x00y0 081C	**0x80zz 0207**	Pointer to BLOCK MOVE function
0x00y0 0820	**0x80zz 0208**	Aux ready flag for factory use
0x00y0 0824	**0x80zz 0209**	Test pattern 0xA5A5 A5A5
0x00y0 0828	**0x80zz 020A**	Test pattern 0x5A5A 5A5A
0x00y0 082C	**0x80zz 020B**	Test pattern 0x1234 5678
0x00y0 0830	**0x80zz 020C**	Test pattern 0x8765 4321
0x00y0 0834	**0x80zz 020D**	Test pattern 0xFFFF FFFF
0x00y0 0838	**0x80zz 020E**	Test pattern 0x0000 0000
0x00y0 083C	**0x80zz 020F**	Test pattern 0x0000 0000
0x00y0 0840	**0x80zz 0210**	Test pattern 0x80zz 0200
0x00y0 0844	**0x80zz 0211**	Pointer to GLOBAL BOOT function
0x00y0 0848	**0x80zz 0212**	Pointer to GLOBAL LOOK function
0x00y0 084C	**0x80zz 0213**	Pointer to GLOBAL SET function
0x00y0 0850	**0x80zz 0214**	Reserved
0x00y0 0854	**0x80zz 0215**	Reserved
0x00y0 0858	**0x80zz 0216**	Pointer to GLOBAL BYTE LOAD function
0x00y0 085C	**0x80zz 0217**	Pointer to GLOBAL BYTE UNLOAD function
0x00y0 0860	**0x80zz 0218**	Pointer to GLOBAL WORD LOAD function
0x00y0 0864	**0x80zz 0219**	Pointer to GLOBAL WORD UNLOAD function

NOTES:

(1) MIX/VME Addresssing: y = 0 for Processor A; y = 2 for Processor B; y = 4 for Processor C; y = 6 for Processor D

(2) 'C40 Addressing: zz = 00 for Processor A; zz = 08 for Processor B; zz = 10 for Processor C; zz = 18 for Processor D

(3) MIX Addressing: On 4283, 4284 & 4285 Baseboards, divide the offsets given in the left-hand column by four. In this case, the MIX and 'C40 offsets will be identical.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 Resetting and Booting the Model 4270's Processors (continued)

Table 4-4: Model 4270 - Register States After Boot from Factory EPROM

Register	Address	Condition	Comments
'C40 Status Registers	**0x0034 0000 ('C40)**	0x00zz 0000	D23-D20 reflect jumper states, D19 & D18 are indeterminate, D17-D0 are cleared (0).
'C40 MIX Control Register	**0x0070 0000 ('C40)**	All used bits cleared (0)	Release When Done, MXMIO & MXDC low, byte enables set for 32-bit, single-cycle transfers
MIX 'C40 Reset Control Register	**MX_Rst_Ctl**	All used bits cleared (0)	MIX Bus Lock and Broadcast Mode off, 'C40 resets inactive, 'C40 boot addresses=0x0000 0000 (Local)
MIX Status Register	**MX_Status**	All used bits cleared (0)	Bus locks off, interrupts cleared.
VME 'C40 Reset Control Register	**A16_base+0x00**	All used bits cleared (0)	VME Bus Lock and Broadcast Mode off, TBC disabled, TBC reset active, 'C40 resets inactive, 'C40 boot addresses = 0x0000 0000 (Local)
VME Address Mapping Register	**A16_base+0x04**	All used bits cleared (0)	Default VME_base=0x0000 0000. A32 Address decoding disabled until first A16 access. First A16 access should be a write of a valid address to this register. Paged Mode & Word Swap off.
VME Address Page Register	**A16_base+0x08**	Indeterminate	Must be initialized by user code before enabling Paged Mode in Mapping Reg, if Paged addressing is to be used.
VME Status Register	**A16_base+0x0C**	D15 set, others cleared	TBC not ready, bus locks off, Interrupts cleared
IACK Vector Register	**A16_base+0x40**	Indeterminate	Must be initialized by user code before interrupting VME.
TBC Registers	**A16_base+0x80 thru 0xFC**	Indeterminate	Must be initialized by user code before using TBC

NOTE: Unused bits will be in indeterminate states

After register initialization, the EPROM code directs the 'C40 to read the first location in its Global SRAM ('C40 address 0x8000 0000 for Processor A, 0x8008 0000 for Processor B, 0x8010 0000 for Processor C or 0x8018 0000 for Processor D). If the data at that address is 0xA55A 5AA5, then the 'C40 uses the location pointed to by the next Global SRAM address (0x80zz 0001) as the boot load address. If any other data pattern is found at the first Global SRAM address (which will obviously be the case at power-up), the 'C40 will return to EPROM and continue to execute the factory default boot code found there. The processor is left running a program that toggles its front panel LED and awaiting an interrupt to direct it elsewhere. A listing of the standard EPROM boot code is provided in Appendix B of this manual.

If no problems are encountered in the boot procedure, and the processors executed the standard EPROM boot, the LEDs on the 4270's front panel will blink about five times a second (the 'C40's Timer 0 is used as a counter). If the factory boot code was executed and one or more LEDs only blink around once per second, then problems were encountered with access to Global SRAM for that (those) 'C40(s). If the factory boot code was executed and any of the LEDs do not blink at all, then a more serious problem exists.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 Resetting and Booting the Model 4270's Processors (continued)

Grounding the $\overline{\text{Reset}}$ pin on the Model 4270's front panel, or asserting one of the Processor Reset Bits in the VME or MIX 'C40 Reset Control Registers will initiate a reset cycle identical to the one described above for the processor(s) in question. Assertion of the VMEbus System Reset signal will perform all of the operations described above for all four processors, and will also reset the board's VMEbus interface logic. Assertion of the MIX Reset signal by the baseboard will perform all of the operations described above for all four processors, and will also reset the board's MIX interface logic. The baseboard usually asserts the MIX Reset signal in response to a VME System Reset or a front panel reset.

4.2.1 Boot Loading User Code

During the power-up (or "cold") reset on the Model 4270, the boot load must always be from either the external EPROM or the Comm Port. However, processors on the 4270 can also be reset in the "warm" state, by using either the $\overline{\text{Reset}}$ pins on the front panel or the Reset Bits in the 'C40 Reset Control Registers (see Sections 3.4.1 and 3.5.2 for details about the Reset Control Registers). The boot load for a warm reset may come from any valid block in a processor's Memory Map.

First, the code to be boot loaded must be downloaded into the Global SRAM (see Section 4.3, below), at an offset from the given SRAM's base address of 0x0400 or greater. The code may then be boot loaded from the Global SRAM, or may first be moved into Local SRAM or the 'C40's Internal RAM. Once the code is in the desired location, the data pattern 0xA55A 5AA5 must be written to the first location in the Global SRAM bank for the given processor (the base address, i. e., 0x8000 0000 for Processor A, 0x8008 0000 for Processor B, 0x8010 0000 for Processor C or 0x8018 0000 for Processor D). Next, write the address at which your boot code starts into the next Global SRAM location (i. e., 0x8000 0001 for Processor A, 0x8008 0001 for Processor B, 0x8010 0001 for Processor C or 0x8018 0001 for Processor D). The processor may now be reset as described in the previous paragraph, and it will boot load the downloaded code.

4.3 Downloading Programs to the Model 4270

'C40 programs in Common Object File Format (COFF) can be transferred into the 4270's Global SRAM over the VMEbus. One recommended method for achieving this file transfer is via Pentek's SwiftNet Communications Protocols. The SPOX DSP Operating System also provides downloading utilities. Another downloading method is via the Texas Instruments XDS-510 Emulator.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 Downloading Programs to the Model 4270 (continued)

4.3.1 Downloading Programs with SwiftNet

Pentek's SwiftNet Communications Protocols are used to provide data transfer links over a distributed DSP network. Included with SwiftNet is a utility program, PNCFG, used to identify and provide essential information about the processors in the network. Another Program, PNLOAD, is used to transfer COFF files from the host system on which they were developed to target DSPs defined in PNCFG. COFF files may be developed in Pentek's SwiftTools Environment, which can call PNLOAD to accomplish the file transfer, or with any 'C40 compatible compiler/assembler/linker package. See the SwiftNet and SwiftTools Operating Manuals for further details.

4.3.2 Downloading SPOX Programs

COFF files created in the SPOX DSP Operating System with the MSPOX command can be downloaded to, and executed on, the 4270 with the PEXEC40 command. See the documentation provided with SPOX for the command formats and additional information.

4.3.3 Downloading Programs with the XDS-510 Emulator

Users of the Texas Instruments XDS-510 Emulator can transfer 'C40 COFF programs into the 4270 via the JTAG connector that protrudes from the front panel. (If the Emulator cable is not installed in your unit, refer to Section 2.7 of this manual for installation instructions.) Note that if the Emulator is to be connected to the Model 4270, jumpers must NOT be installed on jumper blocks JB22 and JB23 on the VME board, which Enable the use of the Test Bus Controller, but inhibit the use of the emulator (see Section 2.2.8). For that code to be properly executed, however, the 4270's register set must be put into its boot state. This must be done by the user, because operating environment of the emulator prevents the 4270 from executing its boot code at power-up.

There are also two registers internal to the 'C40 that must be initialized before you try to access the external registers. The Global Memory Interface Control Register, at address 0x0010 0000, must be set to 0x3DEC 0000, and the Local Memory Interface Control Register, at address 0x0010 0004, must be set to 0x3DEF 7F10. You may either load these values from the emulator's command interface, or place commands to initialize the registers at the beginning of any program you may load. For more information about the use of these registers, see Sections 7.3 (Memory Interface Control Registers) and 7.4 (Programmable Wait States) of the Texas Instruments **TMS320C4x User's Guide**.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 Downloading Programs to the Model 4270 (continued)

4.3.3 Downloading Programs with the XDS-510 Emulator (continued)

One method of initializing the external register set is to issue the RUN command from the emulator, and then ground the front panel Reset pin, before loading your code. Another method is to initialize the registers explicitly at the beginning of your code, after setting up the internal registers (see the paragraph above). An example is presented in Section A.7 (Emulator Support Files), in Appendix A of this manual. The example assigns the alias "myreset" to a routine that allows the card to initialize properly. See Table 4-4, in Section 4.2 of this manual, for the external register addresses and states to which they should be initialized.

4.4 Working with Interrupts on the Model 4270

The Model 4270 can issue interrupts to, and accept interrupts from, both the VMEbus and the MIX bus. Additionally, each 'C40 processor on the 4270 can interrupt any of the other 'C40s. The subsections below will provide details on how the 4270's processors interrupt other devices and each other, and how interrupts are handled by the processors on the 4270.

4.4.1 The 4270 as a VMEbus Interrupter

The Model 4270 can participate in the VMEbus IACK cycle as an interrupter only. Enabling the 4270 as a VME interrupter is a two-step process. First, it must be decided which devices on the 4270 can issue interrupts to VME. Any one processor, all four processors, or the Test Bus Controller (TBC) may be enabled as VME interrupt sources. This is achieved, and interrupt levels are assigned, by placing shorting jumpers on jumper block JB16 on the VME board. Table 4-5, below, tells which jumpers must be installed to enable a given device as a VME interrupt source, and which levels those devices may interrupt VME on. Remember that the 4270 can only be assigned to a single VMEbus interrupt level, which also means that only one of these jumpers should ever be installed at any given time.

Table 4-5: Model 4270 - VME Interrupt Source Selection - JB16, VME Board

	IRQ1	IRQ2	IRQ3	IRQ4	IRQ5	IRQ6	IRQ7
TBC	1 - 3	-	-	-	-	-	1 - 2
Processor A	5 - 3	-	5 - 7	-	-	-	-
Processor B	-	8 - 6	-	8 - 10	-	-	-
Processor C	-	-	9 - 7	-	9 - 11	-	-
Processor D	-	-	-	12 - 10	-	12 - 14	-
Procs ABC&D	4 - 3	4 - 6	*15 - 17	15 - 13	16 - 18	16 - 14	4 - 2

* - Factory Default Setting

NOTE: Procs ABC&D = any processor may interrupt VME - TBC = Test Bus Controller

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Working with Interrupts on the Model 4270 (continued)

4.4.1 The 4270 as a VMEbus Interrupter (continued)

Next, one of the seven VME interrupt levels must be enabled for the 4270, by the placement of jumpers on jumper block JB12 on the VME board. The same interrupt level selected on JB16 must be set on this jumper block. Table 4-6, below, describes the settings of this jumper block.

The 'C40 processors on the Model 4270 issue VMEbus interrupts by writing any data to address 0x0034 0009 in their Local Memory Maps. For information about using the SN74ACT8990 Test Bus Controller as an interrupter, please refer to that device's data sheet.

Table 4-6: Model 4270 - VME IRQ Level - JB12, VME Board

Level	Pins 1 & 2	Pins 3 & 4	Pins 5 & 6
IRQ 0	ON	ON	ON
IRQ 1	OFF	ON	ON
IRQ 2	ON	OFF	ON
IRQ 3*	OFF	OFF	ON
IRQ 4	ON	ON	OFF
IRQ 5	OFF	ON	OFF
IRQ 6	ON	OFF	OFF
IRQ 7	OFF	OFF	OFF
* - Factory Default Setting			

After the completion of the IACK cycle, the VME interrupt handler can determine which processor issued the interrupt by reading the VME Status Register, at address A16_base+0x0C. One of the five LSBs of this register is set to the logic '1' state to indicate a pending interrupt. D4 is the TBC's interrupt status bit, D3 is Processor D's, D2 is Processor C's, D1 is Processor B's, and D0 is Processor A's interrupt status bit. Table 4-7, at the top left of the next page, summarizes the VMEbus Interrupt Status Bits. Also, when a processor interrupts VME, the D17 bit in its own Status Register (Local bus address 0x0034 0000) is set to a logic '1'. Each of these status bits is cleared to the logic '0' state when the associated interrupt request is cleared.

To clear a pending interrupt request from Processor A on the 4270, the interrupt handler must write any data to 4270 address A16_base+0x20. A VME IRQ from Processor B is cleared by writing to A16_base+0x24. Writing any data to A16_base+0x28 will clear a VME interrupt from Processor C, and Processor D's IRQ is cleared by writing anything to A16_base+0x2C. The VME Clear Interrupt Registers are summarized in Table 4-8, at the top right of the next page. For information about clearing VME interrupts from the TBC, please see the data sheet for the 74ACT8990.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Working with Interrupts on the Model 4270 (continued)

4.4.1 The 4270 as a VMEbus Interrupter (continued)

Table 4-7: Model 4270 - VME Interrupt ID Bits

Status Bit	Interrupter
D4	TBC
D3	Processor D
D2	Processor C
D1	Processor B
D0	Processor A

Table 4-8: Model 4270 - VME Clear Interrupt Registers

Write to this Address...	to clear Int. from Proc...
A16_base+0x0020	A
A16_base+0x0024	B
A16_base+0x0028	C
A16_base+0x002C	D

4.4.2 Interrupting 'C40s on the 4270 from VME

By the strict VMEbus definition, the Model 4270 is NOT a VMEbus interrupt handler, in that it does not participate in the IACK cycle in the manner prescribed for a handler. A method is provided, however, by which VMEbus Masters may interrupt the activity of the 'C40 processors on the 4270.

A VMEbus Master can interrupt Processor A on the Model 4270 by writing any data to address A16_base+0x10. Any write to A16_base+0x14 on the 4270 will issue a VMEbus interrupt to Processor B. Processor C is interrupted from VME by writing anything to A16_base+0x18, and any write to A16_base+0x1C will issue a VME interrupt to Processor D. All four processors can be interrupted from VME by having a bus master write any data to address A16_base+0x3C. The VME set interrupt addresses are summarized in Table 4-9, below.

Table 4-9: Model 4270 - VME Set Interrupt Registers

Write to this Address...	to Interrupt Processor...
A16_base+0x0010	A
A16_base+0x0014	B
A16_base+0x0018	C
A16_base+0x001C	D
A16_base+0x003C	A, B, C, & D

VMEbus Master interrupts drive the $\overline{\text{IIOF1}}$ inputs on the 'C40 processors. This is also the interrupt input used by MIX bus Masters. To determine where this interrupt came from the 'C40 reads its Status Register, at Local bus address 0x0034 0000. A logic '1' in D4 of this register indicates that the interrupt came from VME. The interrupted 'C40 processor clears a pending VMEbus interrupt by writing any data to address 0x0034 0004 in its Local bus Memory Map. Clearing the interrupt also clears the associated status bit to logic '0'.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Working with Interrupts on the Model 4270 (continued)

4.4.3 The 4270 as a MIX Bus Interrupter

Any 'C40 processor on the Model 4270 may interrupt the MIX bus by writing any data to its own Local bus address 0x0034 0008. When a 'C40 interrupts MIX, the D16 bit in its own Status Register (Local bus address 0x0034 0000) is set to the logic '1' state.

The MIX bus Master (i. e., the Baseboard or an Upper MIX bus Master) knows which module interrupted by knowing which line the signal came in on. After determining that the interrupt came from the 4270, the Master reads the 4270's MIX status register to find out which processor issued the interrupt. The pending interrupt is indicated by a logic '1' in one of the Status Register's four LSBs. D3 is Processor D's interrupt status bit, D2 is Processor C's, D1 is Processor B's, and D0 is Processor A's interrupt status bit. Table 4-10, below left, summarizes the MIX Interrupt Status Bits. All associated status bits are cleared to the logic '0' state when the interrupt request is cleared.

To clear a pending interrupt request from Processor A on the 4270, the Baseboard must write any data to the Baseboard-dependent, 4270 symbolic MIX address Clr_Int_A_MX. A MIX IRQ from Processor B is cleared by writing to Clr_Int_B_MX. Writing any data to Clr_Int_C_MX will clear a MIX interrupt from Processor C, and Processor D's IRQ is cleared by writing anything to Clr_Int_D_MX. The MIX Clear Interrupt Registers are summarized in Table 4-11, below. For the physical addresses of these registers, please refer to Table 3-28 for a 4200-series Baseboard, Table 3-29 for a Model 4202 Baseboard, Table 3-31 for a Model 4283 Baseboard, Table 3-32 for a Model 4284 Baseboard or Table 3-33 for a Model 4285 Baseboard Adapter.

Table 4-10: Model 4270 - MIX Interrupt ID Bits

Status Bit	Interrupter
D3	Processor D
D2	Processor C
D1	Processor B
D0	Processor A

Table 4-11: Model 4270 - MIX Clear Interrupt Registers

Write to this Address...	to clear Int. from Proc...
Clr_Int_A_MX	A
Clr_Int_B_MX	B
Clr_Int_C_MX	C
Clr_Int_D_MX	D

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Working with Interrupts on the Model 4270 (continued)

4.4.4 Interrupting 'C40s on the 4270 from MIX

A MIX bus Master (i. e., the Baseboard or an Upper MIX bus Master) can interrupt Processor A on the Model 4270 by writing any data to the Baseboard-dependent, 4270 symbolic MIX address MX_Int_A. Any write to MX_Int_B on the 4270 will issue a MIX bus interrupt to Processor B. Processor C is interrupted from MIX by writing anything to MX_Int_C, and any write to MX_Int_D will issue a MIX interrupt to Processor D. All four processors can be interrupted from MIX by having a Master write any data to address MX_Int_All. The symbolic MIX set interrupt addresses are summarized in Table 4-12, below. For the physical addresses of these registers, please refer to Table 3-28 for a 4200-series Baseboard, Table 3-29 for a Model 4202 Baseboard, Table 3-31 for a Model 4283 Baseboard, Table 3-32 for a Model 4284 Baseboard or Table 3-33 for a Model 4285 Baseboard Adapter.

Table 4-12: Model 4270 - MIX Set Interrupt Registers

Write to...	to Interrupt...
MX_Int_A	Processor A
MX_Int_B	Processor B
MX_Int_C	Processor C
MX_Int_D	Processor D
MX_Int_All	Procs. A, B, C& D

MIX bus Master interrupts drive the $\overline{\text{IIOF1}}$ inputs on the 'C40 processors. This is also the interrupt input used by VMEbus Masters. To determine where this interrupt came from the 'C40 reads its Status Register, at Local bus address 0x0034 0000. A logic '1' in D3 of this register indicates that the interrupt came from MIX. The interrupted 'C40 processor clears a pending MIX bus interrupt by writing any data to address 0x0034 0003 in its Local bus Memory Map. Clearing the interrupt also clears the associated status bit to logic '0'.

The 4270 can also handle interrupts from MIX slave modules. The MX_INT1 signal drives the $\overline{\text{IIOF2}}$ inputs on the 'C40s, and its activation clears the D12 bit in the 'C40 Status Register to the logic '0' state. MX_INT2 drives $\overline{\text{IIOF3}}$, and clears D11 in the 'C40 Status Register when active. See Section A.1 (Servicing MIX Module Interrupts) in Appendix A of this manual for sample Interrupt Service Routines (ISRs) for these interrupts.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 Working with Interrupts on the Model 4270 (continued)

4.4.5 Inter-Processor Interrupts on the 4270

Any 'C40 processor on the Model 4270 can interrupt any of the other three processors. In each processor's Local bus Memory Map are three write-only addresses for interrupting other processors. The data written to these registers is unused, and therefore trivial. The addresses of the registers that set interrupts to other processors are summarized in Table 4-13, below left.

When a 'C40 on the 4270 interrupts one of the other three 'C40's, a bit is set in its own Status Register (as well as in the Status Register of the processor that it interrupts). Table 4-14, below right., tells which bit in a given processor's Status Register indicates a pending interrupt to which other processor.

Table 4-13: Model 4270 - Processor Set Interrupt Registers

Address	Interrupting Processor			
	A	B	C	D
0x0034 0005	B	A	A	A
0x0034 0006	C	C	B	B
0x0034 0007	D	D	D	C

Table 4-14: Model 4270 - Outgoing Inter-Processor Interrupt Status Bits

Bit	Interrupting Processor			
	A	B	C	D
D13	B	A	A	A
D14	C	C	B	B
D15	D	D	D	C

Interrupts from other on-board processors drive the $\overline{\text{IIOF0}}$ input of the target 'C40. The 'C40 that receives an interrupt on $\overline{\text{IIOF0}}$ reads his status register to determine which of the other processors issued the interrupt. The three LSBs of each 'C40's Status Register are the incoming inter-processor interrupt status bits. Table 4-15, below left, tells which of these bits indicates an interrupt from which other processor.

Three more Local bus addresses (again, the data written here is unused) are provided to clear interrupts from other processors. Table 4-16, below right, tells which of these addresses a given processor must write to clear an interrupt request from another on-board processor.

Table 4-15: Model 4270 Status Register - Inter-Processor Interrupt Bits

Bit	Interrupted Processor			
	A	B	C	D
D0	B	A	A	A
D1	C	C	B	B
D2	D	D	D	C

Table 4-16: Model 4270 - Outgoing Inter-Processor Clear Interrupt Registers

Address	Interrupted Processor			
	A	B	C	D
0x0034 0000	B	A	A	A
0x0034 0001	C	C	B	B
0x0034 0002	D	D	D	C

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.5 Using the 4270 with a Model 4283 MIX Baseboard

When the Model 4270 Quad 'C40 processor board is used as a MIX Module on Pentek's Model 4283 MIX Baseboard, the user may note an inability to access the Global SRAMs on the module from the Baseboard. This section describes a modification that can be made to the 4270, by the user, to facilitate access to the SRAM by the 4283.

First, the two circuit boards that comprise the 4270 must be separated. This procedure is described in Sections 2.7.1 through 2.7.4 of this manual.

CAUTION!!

Perform all disassembly and assembly steps in a static controlled work environment.

After the boards have been separated, locate the sets of PC pads labeled JB4 and JB9 on each of the circuit boards. Refer to Figures 2-2 and 2-3, on pages 18 and 19, for guidance. At each of these four locations there will be a PC trace connecting pin 1 to pin 2 (the pin numbering scheme is shown in Figure 4-1, below). These traces must be cut. Next, a small jumper wire must be installed connecting pins 9 and 10 at each of these four locations. Figure 4-1, below, shows how these jumper locations should appear after the modification.

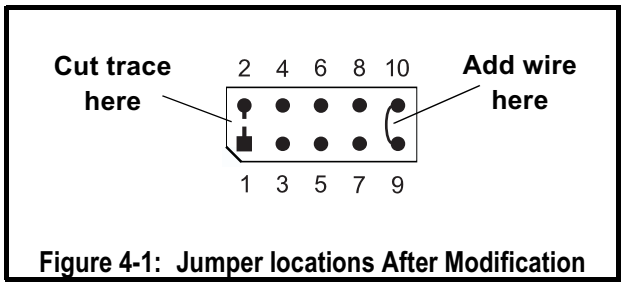


Figure 4-1: Jumper locations After Modification

NOTE: **Please read the next page of this manual BEFORE attempting to re-join the two PC assemblies!**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.5 Using the 4270 with a Model 4283 MIX Baseboard (continued)

The Model 4270 has four banks of Global SRAM that may be accessed via the MIX bus. In the standard configuration, these banks each occupy 2 Mbytes of MIX address space. Therefore, to select among the four spaces, MIX address lines MXA21 and MXA22 must be used.

The Model 4283 can directly address only 32 kBytes of MIX address space by providing MXA2 through MXA14. A page register on the Model 4283 allows the generation of MXA15 through MXA21. These circumstances may lead one to conclude that the 4283, lacking control of MXA22, can access only two of the 4270's four banks of Global SRAM. This is not the case.

The Model 4270 ships from the factory configured for 2 MBytes per SRAM bank. However, this configuration may be changed by modifying jumper settings on the 4270's MIX board. The following changes will enable a 4283 to address all of the shared SRAM via the MIX bus.

There is a jumper block on the MIX board visible between the P1 and P2 VME connectors. This jumper block is labeled JB11 (refer to Figure 2-3, on page 19 for the exact location). The factory default is to have the pins 4 & 6 jumpered together, and pins 3 & 5 jumpered together. To configure the board so that each SRAM block occupies only 1 MByte of MIX address space, pins 1 & 2 must be jumpered together, and pins 3 & 4 must be jumpered together.

This is most easily accomplished while the two boards that comprise the Model 4270 are still separated. It is also possible to accomplish this task without completely disassembling the unit, following the steps below:

1) Remove the two nylon screws holding the boards together near the P1 and P2 connectors.

2) Gently pry the boards apart just far enough to make the jumper changes.

The two circuit boards may now be re-attached to one another, following the instructions given in Sections 2.7.9 and 2.7.10 of this manual. See the **MIX Tutorial** (Pentek part # 800.00001) for more information about using MIX modules on the Model 4283 Baseboard.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 The Model 4270 as a Bus Master

The Model 4270 can function as an Upper MIX Bus Master (UMBM) when used with a MIX Baseboard that supports Upper Masters. When used on a 4200-series Baseboard, the 4270 can also master the VMEbus. The sections that begin on the next page describe the use of the 4270 as a bus master.

4.6.1 Mastering the MIX Bus

Although it is not a VME/MIX Baseboard, the Model 4270 has the capabilities to function as a UMBM when mounted on a Baseboard that supports external mastering. Pentek Baseboards that support this feature are the Models 4200, 4200A, 4201, 4201A, 4202, 4284 and 4285 Options 031 and 036 (the Model 4283 DOES NOT SUPPORT external mastering of the MIX bus).

To access a MIX module from any 'C40 on the 4270, there are a few considerations:

1) What is the MIX address of the target module that the 'C40 wants to access? This depends on the Baseboard that the 4270 and the target module(s) are mounted upon. For a Model 4200-series Baseboard, the MIX address for modules in stack positions 0, 1 and 2 are 0xC000 0000, 0xD000 0000, 0xE000 0000, respectively. For the Models 4202, 4284 and 4285 Baseboards, these addresses are 0xC000 0000, 0xC400 0000, 0xC800 0000.

2) What is the 'C40 address that the 'C40 must access in its own Memory Map to invoke a MIX bus Master access? For the Model 4270, this address is 0xC000 0000.

3) How is the final access address calculated? Remember that while the MIX address is a byte address, the 'C40 only uses longword addressing. Therefore, the equation for calculating the 'C40 address, based on a target MIX address is:

$$\text{'C40 Address} = \text{0xC000 0000} + \frac{\text{MIX Address}}{4}$$

For example, if a 4270 is mounted in stack position 1 on a Model 4200 Baseboard, with a Model 4271 Digital Drop Receiver Module in stack position 0, how would a 'C40 access the channel A FIFO on the 4271?

The 4271's channel A FIFO is at MIX address offset 0x0100. So the 'C40 would access:

$$\text{0xC000 0100} \div 4 + \text{0xC000 0000} = \text{0xF000 0040}$$

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 The Model 4270 as a Bus Master (continued)

4.6.1 Mastering the MIX Bus (continued)

It is an interesting (and occasionally confusing) coincidence that the 'C40 MIX Master access address is at address 0xC000 0000 in the C40's Global Memory Map, and that on the 4200, the stack position 0 MIX address is also 0xC000 0000, in the Global Memory Map. If the same modules were mounted on a Model 4202 Baseboard, the calculation would have been:

0x0000 0100 ÷ 4 + 0xC000 0000 = 0xC000 0040

As a bus master, the 4270 may also be a MIX bus interrupt handler. Appendix A of this manual presents an example of how one could code the interrupt service routines (ISRs) to handle interrupts from the other two modules that may be present in the MIX stack (see Section A.1).

Some of the older MIX module documents from Pentek provide 4283 addresses instead of MIX addresses. These are long word addresses, not byte addresses and have the offset value 0x0080 4000 added to them. This, plus the fact that there may be a 4283 Page Register manipulation, make the calculation:

'C40 Address = 4283 Address – (0x0080 4000 + the Page Register adjustment (if any))

The Page Register adjustment referred to above is important because the 4283 cannot directly produce MIX address bits above MXA14. Therefore, many MIX modules used with the 4283 specify in their documentation that some addresses must be produced in conjunction with the 4283 Page Register. Proceed carefully.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 The Model 4270 as a Bus Master (continued)

4.6.2 Mastering the VMEbus

When mounted on a Pentek Model 4200, 4200A, 4201 or 4201A VME/MIX Baseboard, the 'C40 processors on the 4270 can become VMEbus Masters. This is possible because the 68030 on the 4200-series Baseboard can access the VMEbus by accessing a portion of its Memory Map, and the 'C40, when it is a MIX bus master, can access any locations in the 68030's Memory Map. Throughout the discussion below, remember that MIX addresses are byte addresses and 'C40 addresses are longword addresses (1 longword = 32 bits = 4 bytes).

When the 'C40 accesses its address map in the range 0xC000 0000 - 0xFFFF FFFF, it is generating MIX addresses 0x0000 0000 - 0xFFFF FFFC. It turns out that 68030 addresses are exactly equivalent to MIX addresses (i. e., the MIX bus is the 68030's "Local" bus). Therefore, once a desired VMEbus address is determined, its 'C40 equivalent address can be calculated with the following equation:

$$\text{'C40 Address} = 0\text{xC000 0000} + \frac{\text{MIX Address}}{4}$$

Let's try an example. How would a 4270, mounted on a 4201, access a VME-bus device at A24 address 0x0080 0000? First, determine how the 68030 on the 4201 would access the device. The Memory Map for the 4201 shows that A24/D32 access is available at 68030 address 0xF400 0000. So the target 68030 address is 0xF480 0000. Now, since this is simply the desired MIX address, calculate as before:

$$\text{'C40 Address} = (0\text{xF480 0000} \div 4) + 0\text{xC000 0000} = 0\text{xFD20 0000}$$

Notice that this example uses D32 access on a longword boundary. Suppose one wanted D16 access on an odd word location, such as 0x0080 0002? This is where the byte enable lines, BE0 - BE3 come into play. These bits (D0 - D3 in the 'C40 MIX Bus Control Register, at 'C40 address 0x0070 0000) are used to allow the 'C40 to synthesize the effect of the address bit A1. To access the A24/D16 address 0x0080 0002, begin with the 4201 location corresponding to A24/D16. This is 0xF500 0000. Then calculate the 'C40 address:

$$\text{'C40 address} = (0\text{xF580 0002} \div 4) + 0\text{xC000 0000} = 0\text{xFD60 0000}$$

Notice that the division by four truncated the effective address to a long-word location. Before accessing the 'C40 address, set the D0 and D1 bits at 'C40 address 0x0070 0000 (BE0 and BE1 in the 'C40 MIX Bus Control Register) to the logic '1' state. Since these bits are low true, the remaining bits, BE2 and BE3, are appropriate for access to an odd word location. This causes the 4201 to assert VMEbus address line A1.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 The Model 4270 as a Bus Master (continued)

4.6.2 Mastering the VMEbus (continued)

To generate a D16 access to an even word location make sure that BE0 and and BE1 are cleared to the logic '0' state. Table 4-17, below, provides more detail on the use of the MIX Byte Enable bits, D0 - D3 in each processor's MIX Bus Control Register, at address 0x0070 0000 on the Local bus.

Table 4-17: Model 4270 - MIX Bus Byte Enables and Data Bus Usage

Transfer Type	Cycle # (Note 1)	Byte Enable Signals				Data Bus Utilization				Physical Address		Comment
		$\overline{MXBE3}$	$\overline{MXBE2}$	$\overline{MXBE1}$	$\overline{MXBE0}$	D31-D24	D23-D16	D15-D8	D7-D0	A1	A0	
32-bit	-	L	L	L	L	m	n	k	l	0	0	Single Cycle
32-bit	1	H	H	L	L	X	X	k	l	0	0	1 word +
2 cycle	2	L	L	H	H	X	X	m	n	1	0	1 word
24-bit	-	H	L	L	L	X	m	n	l	0	0	Single Cycle
24-bit	-	L	L	L	H	m	n	l	X	0	1	Single Cycle
24-bit	1	H	L	L	L	X	X	n	l	0	0	1 word +
2 cycle	2	H	L	H	H	X	X	X	m	1	0	1 byte
24-bit	1	L	L	L	H	X	X	l	X	0	1	1 byte +
2 cycle	2	L	L	H	H	X	X	m	n	1	0	1 word
16-bit	-	H	H	L	L	X	X	m	l	0	0	Single Cycle
16-bit	-	L	L	H	H	X	X	m	l	1	0	Single Cycle
16-bit,	1	H	L	L	H	X	X	l	X	0	1	2 bytes =
2 cycle	2	H	L	H	H	X	X	X	m	1	0	1 word
8-bit	-	H	H	H	L	X	X	X	l	0	0	Single Cycle
8-bit	-	H	H	L	H	X	X	l	X	0	1	Single Cycle
8-bit	-	H	L	H	H	X	X	X	l	1	0	Single Cycle
8-bit	-	L	H	H	H	X	X	l	X	1	1	Single Cycle

NOTES:
(1) Cycle # indicates the order in which multiple MIX cycles MUST be run.
(2) X = no byte, m = most significant byte, n = next more significant byte,
k = next less significant byte, l = least significant byte
(3) For Pentek MIX modules, use the 32-bit single cycle setting, which is the default.

It turns out that either values of 0x0 or 0xC in the MIX Bus Control Register will work for even word access, but only the value 0x3 works for odd word access. Also, some VME Slave boards are implemented such that the data written by the 'C40 must be shifted left 16 bits before being written. This is usually true if the slave is a register device, as opposed to a memory board.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 The Model 4270 as a Bus Master (continued)

4.6.2 Mastering the VMEbus (continued)

Another potential source of trouble with VME Mastering is that 'C40 processors have the characteristic of being "bus hogs". The 'C40's generally drops its bus request strobe, and leaves it in the low (i. e., active) state between accesses, which does not allow other devices to access the bus. This is a good characteristic for efficient block transfers, but sometimes not so good in a prioritized, multi-processor system. The Release When Done bit in the 'C40 MIX Bus Control Register (D6 at 0x0070 0000) can be set to make the 'C40 relinquish the MIX bus when it is in between transfers. This may be useful when working with 4200-series Baseboards. If the 68030 wants to get to its Local bus and the 'C40 has "hogged" it, one can run into problems. If you encounter this situation, try setting the Release When Done bit to the logic '1' state.

One final cautionary note about VMEbus Mastering on the 4270. If a 'C40 makes an attempt to become a VMEbus Master at the same time as another device is attempting to use its MIX or VME interface as a slave, a deadlock condition may develop. Consider the following situation . . . A 'C40 on a Model 4270, mounted on a 4200-series Baseboard successfully becomes a MIX bus Master. Simultaneous to this, an off-card VME Requester is granted VMEbus Mastership by the VME Arbiter. When the 'C40 tries to access the 68030's VME Master memory region (the mechanism the 'C40 uses to obtain VME Mastership), access is denied because the VME arbiter has already granted control to another device. When the VME Master tries to access the 4270's Global bus, he is also denied access, because the 'C40 has mastered it. This situation may be avoidable by the use of semaphores, or the deadlock interrupt on the Baseboard. If this situation should arise in your application, contact the Pentek at the phone number shown on this manual's cover page for assistance.

4.7 Using the Model 4270's Global Bus Lock Feature

The Model 4270's Global bus lock feature allows the device that is mastering a given data transaction to claim exclusive access to the board's Global data bus. When a bus master has invoked the Global bus lock, no other device can arbitrate for the use of any resource that it accesses via the Global bus. The Global bus lock allows block data transfers to occur at an accelerated rate. The master device that locked the Global bus retains uncontested access to the Global data bus, and to the Global SRAM bank it is using, until it unlocks the bus.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.7 Using the Model 4270's Global Bus Lock Feature (continued)

There are different limitations on, and characteristics of a locked Global bus, depending upon which master device locks the path. The two cases to be considered are bus locks issued from off-card (i. e., by a VMEbus Master, the MIX Baseboard or another Upper MIX Bus Master (UMBM)), and bus locks issued by one of the 'C40 processors on the 4270 itself. Each of these situations will be discussed in a subsection below.

NOTE: The Global bus lock feature will also help to increase the speed of DMA transfers involving the Global SRAMs on the 4270. Although it is recommended that the bus lock be invoked when using DMA, it is NOT required. Situations may arise in which is desirable to conduct DMA with a given block of memory, but still allow other devices to arbitrate for it. The combined use of DMA and the bus lock is left to the discretion of the user.

4.7.1 Global Bus Locked by External Devices

When a VMEbus Master, the MIX Baseboard, or a UMBM invokes the Global bus lock, it obtains uncontested use of the 4270's Global data bus and ONE (and ONLY ONE) Global SRAM bank. This means that no other off-card master can access ANY other bank of the 4270's shared memory, that none of the 'C40 processors on the 4270 can access any other processor's Global SRAM, and that the locked Global SRAM bank is inaccessible to the processor that "owns" it (the other three processors, however, can still access their own Global SRAMs). Additionally, the master device that imposed the lock (or any other device that may take over as master of that bus) will NOT be able to access any Global SRAM bank other than the one that it locked.

To invoke the Global bus lock, the master device must first make one access to the Global SRAM bank it wishes to lock. This access may or may not involve real data, as its purpose is simply to let the master's own requester circuitry know which path it wishes to lock. It must then set the Global bus Lock bit (D15 in its 'C40 Reset Control Register) to the logic '1' state. This register is at the A16_base address for VMEbus Masters, or at the Baseboard-dependent symbolic address MX_Rst_Ctl for MIX bus Masters (see the appropriate table in Section 3.3 for the physical address of this register). Once this bit has been set, all subsequent accesses by the master that set it to the 4270 are locked to the Global SRAM in question, and none of the 4270's other resources are available to this master.

When the Master device no longer needs exclusive use of the Global SRAM in question, it should promptly free the Global bus by clearing the Global bus lock bit referred to above to the logic '0' state. Remember that while the bus lock is in place, access to ALL Global memory on the 4270 is quite restricted. If access to another Global SRAM is necessary, the master must first unlock the bus, then access the other Global SRAM. It may then set the lock bit again.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.7 Using the Model 4270's Global Bus Lock Feature (continued)

4.7.1 Global Bus Locked by External Devices (continued)

It is worth noting at this point that while invoking the Global bus lock does guarantee that no other device can use the 4270's Global bus, it DOES NOT grant the master device exclusive use of its native bus. In other words, it may be necessary to take other steps to assure that, for example, a VMEbus Master that locks the 4270's Global bus ALSO keeps possession of the VME-bus for the duration of the block transfer. This issue may have even farther reaching implications in a multi-master MIX system. Suppose, for example, a UMBM locks 4270 Global SRAM A, and then loses MIX bus mastership to the Baseboard. If the Baseboard should then attempt to access the 4270, it will find that the lock applied by the prior master is still in effect, and that the only accessible Global SRAM bank is Processor A's.

4.7.2 Global Bus Locked by 'C40s on the 4270

The 'C40 processors on the 4270 can access their own Global SRAM at full processor speed (i. e., zero wait state), and the other three Global SRAMs with only one wait state (after Global bus arbitration). This access rate can only be sustained by utilizing either the 'C40's interlocked instructions, or by invoking the Global bus lock.

Any processor on the 4270 can lock any ONE Global SRAM bank. The Global bus lock CANNOT be used to move data between two different Global SRAM banks, or between a Global SRAM bank and the MIX bus. The Global bus lock can be used to accelerate transfers between the following resources:

Global SRAM and Local SRAM
Global SRAM and Internal 'C40 RAM
Different locations within the same Global SRAM
Local SRAM and the MIX bus
Internal 'C40 RAM and the MIX bus

Before locking the Global bus, one access must be made between the resources in question. The processor that wishes to lock the bus must then write any data to the Global Bus Lock Register, at address 0x0034 000B in its own Local Memory Map. When finished with the block transfer, the processor should promptly unlock the bus by writing to the Global Bus Unlock Register, at address 0x0034 000A.

When using the Global bus lock for MIX bus transactions, remember that the MIX bus itself is NOT locked. Basically, the only potential masters whose access to the MIX bus is inhibited by the locking of a 4270's Global bus are the other three 'C40s on the same 4270. This is because the 4270's arbitration logic must eventually yield to other bus masters' attempts to obtain the bus, to prevent a MIX deadlock condition.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.7 Using the Model 4270's Global Bus Lock Feature (continued)

4.7.2 Global Bus Locked by 'C40s on the 4270 (continued)

The Global bus lock can also be used to prevent problems that may occur while using the interlocked instructions of the 'C40. While attempting to maintain zero-wait access to one Global memory, there are conditions under which the 4270's arbitration may not react quickly enough to recognize an attempt to access a different Global memory. In this case, the next access will be to the same bank, rather than the new bank. One way to avoid this situation is to use the bus lock, which will require that the bus be unlocked before the access to the new bank.

NOTE: (1) When a 'C40 is performing interlocked accesses to a Global SRAM bank in a tight loop, the 4270 may actually behave as if the Global bus were locked by the 'C40, even though the lock was never set. This may cause timeouts or bus errors if an external device attempts to access the 4270's Global bus. A workaround for this situation is to program one of the 'C40's internal timers to interrupt the 'C40 periodically, to get it out of the loop and thus off the bus (the timer period must be less than the bus timeout interval). This is demonstrated in Section A.6 (Using Timer Interrupts to Avoid Bus Timeouts) in Appendix A of this manual.

(2) The above scenario can be further complicated if the access loop involves 'C40 DMA. If this is the case, the ISR for the timer interrupt should turn the DMA off briefly, then re-enable DMA before returning.

4.8 Using the Global Memory

The Global SRAM on the 4270 is accessible by any VMEbus Master, by MIX Baseboards and Upper MIX Bus Masters (UMBMs), and by all four 'C40 processors. This makes this memory area ideal for exchange of data and program information between processors in a multi-processor system. 'C40 programs in Common Object File Format (COFF) can be deposited here by the VME System Executive or the MIX Baseboard, and can either be executed directly from Global memory or moved by the 'C40s into their Local SRAM prior to execution.

The 4270's Global SRAM is seen as four separate memory blocks of 256k longwords each from the 'C40 side, or 1 MByte each from the VMEbus and MIX bus sides. The starting address of these memory blocks, from the 'C40's point of view, is 0x8000 0000 for Processor A's Global SRAM, 0x8008 0000 for Processor B's Global SRAM, 0x8010 0000 for Processor C's Global SRAM and 0x8018 0000 for Processor D's Global SRAM. Any processor may access another processor's Global SRAM bank, but these transactions incur a 1 wait state penalty, after arbitration (processors encounter no wait states after arbitration when addressing their own Global SRAM bank).

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.8 Using the Global Memory (continued)

From the point of view of a VMEbus Master, Processor A's Global SRAM begins at the VME_base address. This address is programmed into the Address Mapping and Control Register, at address A16_base+0x04 (See Section 3.5.3.1 for further details). Processor B's bank of Global SRAM starts at address VME_base+0x0020 0000, Processor C's at VME_base+0x0040 0000, and Processor D's at VME_base+0x0060 0000.

From the MIX bus side, Processor A's Global SRAM starts at the MIX_base address (determined by the 4270's MIX stack position and the type of Baseboard used), and the offsets for the other banks are identical to those on the VME side, if the Baseboard uses byte addressing. The addressing information given above is summarized in Table 3-27, on page 55 of this manual. Physical MIX addressing, as a function of the Baseboard in use, is given in Tables 3-28, 3-29, 3-31, 3-32, and 3-33, in Section 3.3.

When multiple devices attempt to gain initial access the 4270's Global bus simultaneously, access will be granted in the following prioritized order: Processor A, Processor B, Processor C, Processor D, MIX (UMBM), and VME. On subsequent accesses, the processor that last used the Global bus has priority. If that device does not need the bus, priority would go to the next device in line. For example, if the Global bus was last accessed by Processor C, and C no longer needs the bus, priority would pass to Processor D, then MIX, then VME, then to Processor A. Use of the Global SRAM by the 'C40s is facilitated by the Global bus locking feature of the Model 4270. This feature is described in detail in Section 4.7 of this manual.

For applications involving real-time signal processing, it is important to know how quickly data can be moved from one memory area to another. Tables 4-18 through 4-29, on the next three pages, give the times needed to transfer a single longword from each resource available to a 'C40 processor on the Model 4270 to any another available resource, or to another region within that resource's address range.

The data in Tables 4-18 and 4-19 represent typical performance for a Model 4270 with 50 MHz 'C40s, using parallel instructions and 'C40 DMA transfer mode, respectively. Tables 4-20 and 4-21 represent the performance that can typically be expected under the same conditions, from a 50 MHz 'C40 on a 4270 with the Global bus lock used. Tables 4-22 and 4-23 represent the typical performance of a Model 4270 equipped with 60 MHz 'C40s, in parallel instruction and 'C40 DMA modes, respectively. Tables 4-24 and 4-25 show what can be expected when a 60 MHz 'C40 locks the Global bus. Tables 4-26 through 4-29 give the transfer timings for a 4270 with 40 MHz 'C40's under the same four conditions.

'C40 Processor A on the 4270 was the master device for all data transactions. In parallel instruction mode, the data transfers are mastered by the 'C40's CPU, in a loop using the RPTS instruction. See Example 12-16 (Block Move Under Program Control), in Section 12.3.2 (Block Moves) of the Texas Instruments **TMS320C4x User's Guide** for an example of the use of this instruction. Transfers performed in this manner (without the Global bus lock) are documented in Tables 4-18, 4-22 and 4-26, for 50 MHz, 60 MHz and 40 MHz devices, respectively. Tables 4-20, 4-24 and 4-28 describe such transfers with the Global bus lock applied.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.8 Using the Global Memory (continued)

Table 4-18: Model 4270 - Typical Transfer Times for One Longword (Parallel Instruction Mode) - 50 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	40 nsec	80 nsec	80 nsec	120 nsec	280 nsec	410 nsec
Local SRAM	40 nsec	160 nsec	80 nsec	120 nsec	280 nsec	425 nsec
Own Global SRAM	40 nsec	200 nsec	160 nsec	880 nsec	920 nsec	1120 nsec
Other Global SRAM	120 nsec	120 nsec	880 nsec	1040 nsec	1085 nsec	1282 nsec
MIX (no wait)	280 nsec	280 nsec	960 nsec	1090 nsec	560 nsec	720 nsec
MIX (wait)	425 nsec	415 nsec	1115 nsec	1285 nsec	720 nsec	870 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-19: Typical Transfer Times for One Longword ('C40 DMA Mode) - 50 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	80 nsec	80 nsec	80 nsec	120 nsec	280 nsec	425 nsec
Local SRAM	80 nsec	160 nsec	80 nsec	120 nsec	280 nsec	420 nsec
Own Global SRAM	360 nsec	360 nsec	160 nsec	880 nsec	920 nsec	1085 nsec
Other Global SRAM	520 nsec	520 nsec	880 nsec	1040 nsec	1095 nsec	1285 nsec
MIX (no wait)	565 nsec	565 nsec	960 nsec	1080 nsec	565 nsec	720 nsec
MIX (wait)	720 nsec	720 nsec	1120 nsec	1285 nsec	725 nsec	870 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-20: Model 4270 - Typical Transfer Times for One Longword (Parallel Instructions w/ Bus Lock) - 50 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	N/A	N/A	80 nsec	120 nsec	280 nsec	420 nsec
Local SRAM	N/A	N/A	80 nsec	120 nsec	280 nsec	425 nsec
Own Global SRAM	40 nsec	80 nsec	160 nsec	N/A	N/A	N/A
Other Global SRAM	120 nsec	120 nsec	N/A	240 nsec	N/A	N/A
MIX (no wait)	280 nsec	280 nsec	N/A	N/A	N/A	N/A
MIX (wait)	415 nsec	425 nsec	N/A	N/A	N/A	N/A
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-21: Model 4270 - Typical Transfer Times for One Longword ('C40 DMA w/ Bus Lock) - 50 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	N/A	N/A	80 nsec	120 nsec	285 nsec	425 nsec
Local SRAM	N/A	N/A	80 nsec	120 nsec	285 nsec	425 nsec
Own Global SRAM	80 nsec	80 nsec	160 nsec	N/A	N/A	N/A
Other Global SRAM	160 nsec	160 nsec	N/A	240 nsec	N/A	N/A
MIX (no wait)	525 nsec	525 nsec	N/A	N/A	N/A	N/A
MIX (wait)	685 nsec	685 nsec	N/A	N/A	N/A	N/A
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.8 Using the Global Memory (continued)

Table 4-22: Model 4270 - Typical Transfer Times for One Longword (Parallel Instruction Mode) - 60 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	33.3 nsec	66.7 nsec	66.7 nsec	100 nsec	233.3 nsec	342 nsec
Local SRAM	33.3 nsec	133.3 nsec	66.7 nsec	100 nsec	233.3 nsec	355 nsec
Own Global SRAM	33.3 nsec	166.7 nsec	133.3 nsec	733.3 nsec	767 nsec	934 nsec
Other Global SRAM	100 nsec	100 nsec	733.3 nsec	866.7 nsec	905 nsec	1070 nsec
MIX (no wait)	233.3 nsec	233.3 nsec	800 nsec	910 nsec	467 nsec	600 nsec
MIX (wait)	355 nsec	350 nsec	930 nsec	1075 nsec	600 nsec	725 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-23: Typical Transfer Times for One Longword ('C40 DMA Mode) - 60 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	66.7 nsec	66.7 nsec	66.7 nsec	100 nsec	355 nsec	505 nsec
Local SRAM	66.7 nsec	133.3 nsec	66.7 nsec	100 nsec	355 nsec	505 nsec
Own Global SRAM	300 nsec	300 nsec	133.3 nsec	733.3 nsec	1105 nsec	1265 nsec
Other Global SRAM	433.3 nsec	433.3 nsec	733.3 nsec	866.7 nsec	1305 nsec	1480 nsec
MIX (no wait)	471 nsec	471 nsec	800 nsec	900 nsec	471 nsec	600 nsec
MIX (wait)	600 nsec	600 nsec	934 nsec	1075 nsec	605 nsec	725 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-24: Model 4270 - Typical Transfer Times for One Longword (Parallel Instructions w/ Bus Lock) - 60 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	N/A	N/A	66.7 nsec	100 nsec	233.3 nsec	350 nsec
Local SRAM	N/A	N/A	66.7 nsec	100 nsec	233.3 nsec	355 nsec
Own Global SRAM	33.3 nsec	66.7 nsec	133.3 nsec	N/A	N/A	N/A
Other Global SRAM	100 nsec	100 nsec	N/A	200 nsec	N/A	N/A
MIX (no wait)	233.3 nsec	233.3 nsec	N/A	N/A	N/A	N/A
MIX (wait)	346 nsec	355 nsec	N/A	N/A	N/A	N/A
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-25: Model 4270 - Typical Transfer Times for One Longword ('C40 DMA w/ Bus Lock) - 60 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	N/A	N/A	66.7 nsec	100 nsec	238 nsec	355 nsec
Local SRAM	N/A	N/A	66.7 nsec	100 nsec	238 nsec	355 nsec
Own Global SRAM	66.7 nsec	66.7 nsec	133.3 nsec	N/A	N/A	N/A
Other Global SRAM	133.3 nsec	133.3 nsec	N/A	200 nsec	N/A	N/A
MIX (no wait)	438 nsec	438 nsec	N/A	N/A	N/A	N/A
MIX (wait)	571 nsec	571 nsec	N/A	N/A	N/A	N/A
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.8 Using the Global Memory (continued)

Table 4-26: Model 4270 - Typical Transfer Times for One Longword (Parallel Instruction Mode) - 40 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	50 nsec	100 nsec	100 nsec	150 nsec	350 nsec	515 nsec
Local SRAM	50 nsec	200 nsec	100 nsec	150 nsec	350 nsec	500 nsec
Own Global SRAM	50 nsec	250 nsec	200 nsec	1100 nsec	1100 nsec	1285 nsec
Other Global SRAM	150 nsec	150 nsec	1100 nsec	1300 nsec	1280 nsec	1465 nsec
MIX (no wait)	350 nsec	355 nsec	1200 nsec	1280 nsec	700 nsec	900 nsec
MIX (wait)	505 nsec	505 nsec	1250 nsec	1455 nsec	850 nsec	1100 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-27: Typical Transfer Times for One Longword ('C40 DMA Mode) - 40 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	100 nsec	100 nsec	100 nsec	150 nsec	355 nsec	505 nsec
Local SRAM	100 nsec	200 nsec	100 nsec	150 nsec	355 nsec	505 nsec
Own Global SRAM	450 nsec	450 nsec	200 nsec	1100 nsec	1105 nsec	1265 nsec
Other Global SRAM	650 nsec	650 nsec	1100 nsec	1300 nsec	1305 nsec	1480 nsec
MIX (no wait)	650 nsec	650 nsec	1200 nsec	1300 nsec	705 nsec	855 nsec
MIX (wait)	805 nsec	850 nsec	1250 nsec	1480 nsec	855 nsec	1105 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-28: Model 4270 - Typical Transfer Times for One Longword (Parallel Instructions w/ Bus Lock) - 40 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	N/A	N/A	100 nsec	150 nsec	355 nsec	530 nsec
Local SRAM	N/A	N/A	100 nsec	150 nsec	350 nsec	520 nsec
Own Global SRAM	50 nsec	100 nsec	200 nsec	N/A	N/A	N/A
Other Global SRAM	150 nsec	150 nsec	N/A	300 nsec	N/A	N/A
MIX (no wait)	350 nsec	350 nsec	N/A	N/A	N/A	N/A
MIX (wait)	505 nsec	505 nsec	N/A	N/A	N/A	N/A
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Table 4-29: Model 4270 - Typical Transfer Times for One Longword ('C40 DMA w/ Bus Lock) - 40 MHz 'C40						
From \ To	'C40 Int. RAM	Local SRAM	Own Global SRAM	Other Global SRAM	MIX (no wait)	MIX (wait)
'C40 Int. RAM	N/A	N/A	100 nsec	150 nsec	355 nsec	530 nsec
Local SRAM	N/A	N/A	100 nsec	150 nsec	355 nsec	505 nsec
Own Global SRAM	100 nsec	100 nsec	200 nsec	N/A	N/A	N/A
Other Global SRAM	200 nsec	200 nsec	N/A	300 nsec	N/A	N/A
MIX (no wait)	555 nsec	555 nsec	N/A	N/A	N/A	N/A
MIX (wait)	705 nsec	705 nsec	N/A	N/A	N/A	N/A
All timing results are given as average for 1 longword transfer based on 1000 longword transfers						

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.8 Using the Global Memory (continued)

In applications where more processing bandwidth is required, it may be advantageous to allow the 'C40's DMA controller to handle the data transfer tasks. In most cases, however, actual transfer times will be slower in 'C40 DMA mode than in parallel instruction mode. Section 12.6.3 (DMA Coprocessor) of the **TMS320C4x User's Guide** presents several DMA examples. Tables 4-19 and 4-21 document the performance of a 4270 with 50 MHz processors, in 'C40 DMA transfer mode, with the Global bus lock off and on, respectively. Tables 4-23 and 4-25 cover the same situations for 60 MHz processors, and Tables 4-27 and 4-29 give these timings for 40 MHz 'C40s.

To acquire the MIX bus transfer rate data presented in these tables, two different MIX co-processor modules were used, to illustrate the effect of MIX bus wait states. Pentek's Model 4247 'C30 processor card is an example of a device that does not impose wait states on MIX bus transactions. Transactions with that board are documented in the rows and columns of the tables marked MIX (no wait). The Model 4257 'C40 co-processor, on the other hand, does impose wait states when it transacts with the MIX bus. Results of the tests performed with that board are listed in the rows and columns marked MIX (wait).

As the tables above all state, the transfer times they list are averaged over 1000 cycles. One should not expect that the transfer of a single longword will occur as quickly as these tables state. The averaging process serves to mask some initial processing overhead encountered in Global bus accesses on the 4270. Accesses to the MIX bus may encounter an overhead of up to 24 machine cycles. Accesses to another processor's Global SRAM may encounter up to 11 machine cycles of overhead. Even when accessing it's own Global SRAM, a processor may encounter an overhead up to 7 machine cycles long. These overheads occur only on the first access, and subsequent accesses will occur at the times listed in Tables 4-18 through 4-27. Also note that the overheads listed above do not (and cannot) take possible arbitration delays into account. Table 4-30, below, summarizes the overhead information given above.

Table 4-30: Model 4270 - Global Bus Access Overheads - Maximum Initial Access Delays (Excluding Arbitration), for First Access

MIX Bus Access			Other Global SRAM Access			Own Global SRAM Access		
50 MHz 'C40	60 MHz 'C40	40 MHz 'C40	50 MHz 'C40	60 MHz 'C40	40 MHz 'C40	50 MHz 'C40	60 MHz 'C40	40 MHz 'C40
960 nsec	800 nsec	1200 nsec	440 nsec	366.7 nsec	500 nsec	280 nsec	233.3 nsec	350 nsec

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.8 Using the Global Memory (continued)

No VMEbus transfer data is included in Tables 4-18 through 4-29, because the 'C40s on the 4270 cannot master the VMEbus, unless assisted by the 68030 Processor on a Model 4200-series Baseboard. However, in a 4270 with 50 MHz 'C40s, the delay from the assertion of the $\overline{DS}$ signal from the master to the assertion of $\overline{DTACK}$ signal by the 4270 is 260 nsec if the Global bus is not locked, or 160 nsec if the Global bus is locked. The delay from the master's de-assertion of $\overline{DS}$ to a 50 MHz 4270's de-assertion of $\overline{DTACK}$ is 70 nsec, regardless of the state of the bus lock. A 60 MHz 4270 can assert $\overline{DTACK}$ 140 nsec after receiving a negative $\overline{DS}$ transition if its Global bus is locked, or 240 nsec if not. the de-assertion delay for a 60 MHz 4270 is around 66.7 nsec, regardless of the Global bus lock status. For a 40 MHz 4270, the assertion delay is 540 nsec if the Global bus is unlocked, or 300 nsec with the Global bus locked. The de-assertion delay on a 40 MHz board is 140 nsec if the bus is not locked, or 100 nsec if it is. This information is summarized in Table 4-31, below.

Table 4-31: Model 4270: VMEbus Timing

50 MHz 'C40 Processors			60 MHz 'C40 Processors			40 MHz 'C40 Processors		
VME Timing	Assertion (DS↓ to DTACK↓)	De-Assertion (DS↑ to DTACK↑)	VME Timing	Assertion (DS↓ to DTACK↓)	De-Assertion (DS↑ to DTACK↑)	VME Timing	Assertion (DS↓ to DTACK↓)	De-Assertion (DS↑ to DTACK↑)
Global Bus Unlocked	260 nsec	70 nsec	Global Bus Unlocked	240 nsec	66.7 nsec	Global Bus Unlocked	540 nsec	140 nsec
Global Bus Locked	160 nsec	70 nsec	Global Bus Locked	140 nsec	66.7 nsec	Global Bus Locked	300 nsec	100 nsec

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page is intentionally blank

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix A: Programming Examples

A.1 Servicing MIX Module Interrupts

To service an interrupt from a MIX module to the Model 4270, the proper 'C40 interrupt number is a function of the stack position of both the 4270 and the interrupting module. The MXINT1 signal will drive the $\overline{\text{IIOF2}}$ input on the 'C40s, and the MXINT2 signal drives $\overline{\text{IIOF3}}$. Some rules of thumb and example code are given below.

If the 4270 is in stack position 0:

Use **IntInit2** for the module 1 interrupt (MX_INT2) and **IntInit3** for the module 2 interrupt (MX_INT1)

If the 4270 is in stack position 1:

Use **IntInit3** for the module 0 interrupt (MX_INT1) and **IntInit2** for the module 2 interrupt (MX_INT2)

If the 4270 is in stack position 2:

Use **IntInit2** for the module 0 interrupt (MX_INT2) and **IntInit3** for the module 1 interrupt (MX_INT1)

```
/*********************************************************************/
unsigned int *int01_vector;

/*********************************************************************/
void IntInit2()      /* Use this routine to service MX_INT2, on IIOF3 */
{
   int01_vector = (unsigned int *) c_int01;
   asm("   ANDN 2000H,ST");
   asm("   LDEP IVTP,AR0");
   asm("   LDI @_int01_vector,R0");
   asm("   STI R0,*+AR0(6)");
   asm("   IACK @0");
   asm("   OR  09000H,IIF");
   asm("   OR 2000h,ST");
}
/*********************************************************************/
void IntInit3()      /* Use this routine to service MX_INT1, on IIOF2 */
{
   int01_vector = (unsigned int *) c_int01
   asm("   ANDN 2000H,ST");
   asm("   LDI @_int01_vector,R0");
   asm("   STI R0,*+AR0(5)");
   asm("   IACK @0");
   asm("   OR  00900H,IIF");
   asm("   OR  2000H,ST");
}
/*********************************************************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.1 Servicing MIX Module Interrupts (continued)

```
/*********************************************************************/
c_int01()
{
   /* put interrupt code here */
   asm("   IACK @0");
}
/*********************************************************************/
```

A.2 Slo_Blnk.C

The sample "C" source code below drives the four LEDs on the Model 4270's front panel with sinusoidally pulse-width modulated waveforms. Each processor drives one of the four LEDs. The signals from processors A and D are in phase with one another, and 180° out of phase with the signals generated by processors B and C.

```
/*****************************slo_blnk.c*****************************/
#include <math.h>
/*********************************************************************/
#define PROC_ID      ((unsigned long *)0x340000)
#define TIMER0_CTRL  ((unsigned long *)0x100020)
#define LED_ON       0x302
#define LED_OFF      0x306
#define COM0_CTRL    ((volatile unsigned long *)0x100040)
#define COM0_INPUT   ((unsigned long *)0x100041)
#define COM0_OUTPUT  ((unsigned long *)0x100042)
#define COM3_CTRL    ((volatile unsigned long *)0x100070)
#define COM3_INPUT   ((unsigned long *)0x100071)
#define COM3_OUTPUT  ((unsigned long *)0x100072)
#define PI3.1415927
/*********************************************************************/
int angleStep = 200;
int fullDuty = 10000;
/*********************************************************************/

main()
{
   if (((*PROC_ID & 0x300000) == 0)||
       ((*PROC_ID & 0x300000) == 0x300000))
   {
      procAD();                              /* processor A/D routine */
   }
   else
   {
      procBC();                              /* processor B/C routine */
   }
}
/*********************************************************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.2 **Slo_Blnk.C** (continued)

```
/*************************************************************************/
procAD()
{
   double angle;
   int onDelay, offDelay;
   while(1)
   {
      for(angle = 0.0; angle <= 2*PI;
               angle = angle + (2*PI/angleStep)
      {
         onDelay = (fullDuty/2.0) * (sin(angle)+1.0);
         offDelay = fullDuty - onDelay;
         *TIMER0_CTRL = LED_ON;                 /* turn proc A LED on */
         while((*COM0_CTRL & 0x1e0) != 0x0);
                                                  /* wait for COM0 o/p */
         *COM0_OUTPUT = LED_OFF;               /* turn proc D LED off */
         wait(onDelay);
         *TIMER0_CTRL = LED_OFF;               /* turn proc A LED off */
         while((*COM0_CTRL & 0x1e0) != 0x0);
                                               /* wait for COM0 output */
         *COM0_OUTPUT = LED_ON;                 /* turn proc D LED on */
         wait(offDelay);
      }
   }
}
/*************************************************************************/
wait(count)

int count;
{
   int i;
   for(i = 0; i < count; i++);
}
/*************************************************************************/
procBC()
{
   while(1)
   {
      while((*COM3_CTRL & 0x1e00) == 0x0);
                                               /* wait for COM3 input */
      *TIMER0_CTRL = *COM3_INPUT;              /* turn LED on or off */
   }
}
/*****************************end slo_blnk.c*****************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.3 p4243.c

The "C" code presented below uses a 4270, mounted in stack position 1 on a Model 4202 MIX Baseboard, as an Upper MIX Bus Master (UMBM). The 4270 takes data from, and returns data to, a Model 4243 A-D/D-A converter, in stack position 0. The 4243 issues an interrupt to the 4270 when its input data FIFO is half-full. The 4270 reads out the 4243's input data FIFO over the MIX bus, and writes the data to the 4243's output data FIFO.

```
/*****************************p4243.c*******************************/
#include "values.h"
#include "math.h"

/* The following are offsets from the beginning of SRAM */
#define SRAM_PROC_A   0x00000000u
#define SRAM_PROC_B   0x00080000u
#define SRAM_PROC_C   0x00100000u
#define SRAM_PROC_D   0x00180000u
#define PROC_A        0
#define PROC_B        1
#define PROC_C        2
#define PROC_D        3

/* Mix module addresses */
#define MOD0_ADDR     0xc0000000u
#define MOD1_ADDR     0xc4000000u
#define MOD2_ADDR     0xc8000000u

/* 4243 constants */
#defineMOD_ADDR      MOD0_ADDR
#defineStatusControl ((unsigned int *) (MOD_ADDR | 0u))
#defineCtcMode ((unsigned int *) (MOD_ADDR | 0x080))
#defineCh0 ((unsigned int *) (MOD_ADDR | 0x083u))
#defineCh1 ((unsigned int *) (MOD_ADDR | 0x082u))
#defineCh2 ((unsigned int *) (MOD_ADDR | 0x081u))
#defineDataBuffer ((unsigned int *) (MOD_ADDR | 0x01u))

/* 'C40 timer addresses */
#define timer0_address (unsigned int *)0x100020u
#define timer1_address (unsigned int *)0x100030u

unsigned int *int01_vector;
int int_occurred;
/*************************************************************/
void c_int02()
{
   unsigned int temp;
   unsigned int i;

   asm("   ldi iif,r0");
   asm("   sti r0,@_int_occurred");
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.3 p4243.c (continued)

```
   for (i=0;i<1024;i++)
      sig[i]=*DataBuffer;
   for (i=0;i<1024;i++)
      *DataBuffer=sig[i];
   toggle_led();

   asm("   ldi 0,iif");
   temp = *StatusControl;
   asm("   iack @0");
}
/*************************************************************************/
unsigned int sig[2048];

main()
{
   int i;
   *(unsigned int *) 0x34000d = 0xff;

   *StatusControl=0x0u;                              /* Reset FIFO's */
   *CtcMode=0x34000000u;                                /* Ch0 mode */
   *Ch0=0x64000000u;                 /* Input Sample rate divisor lsb */
   *Ch0=0x00000000u;                 /* Input Sample rate divisor msb */
   *CtcMode=0x74000000u;                                /* Ch1 mode */
   *Ch1=0x64000000u;               /* Output Sample rate divisor lsb */
   *Ch1=0x00000000u;               /* Output Sample rate divisor msb */
   *CtcMode=0x96000000u;                                /* Ch2 mode */
   *Ch2=0x06000000u;                   /* Filter Cutoff Freq = 20 kHz */

   int01_vector = (unsigned int *)c_int02;

   int_occurred = 0;

   i = *StatusControl;
      asm("   ldep ivtp,ar0");
      asm("   ldi @_int01_vector,r0");
      asm("   sti r0,*+ar0(5)");
      asm("   iack @0");

      asm("   or 00900h,iif");
      asm("   or 2000h,st");

*StatusControl=0x93000000u; /* Release and acquire in single-channel mode */

   while(1);
}
/*************************************************************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.3 p4243.c (continued)

```
/*************************************************************************/
led_on()
{
   *timer0_address = 0x302u;                                  /* led ON */
}
/*************************************************************************/
led_off()
{
   *timer0_address = 0x306u;                                 /* led OFF */
}
/*************************************************************************/
toggle_led()
{
   long int i;
   *timer0_address = ((*timer0_address & 0x306u) ^ 0x4);
      for (i=0; i<1000; i++);
}
/*****************************end p4243.c*****************************/
```

A.4 p4271.c

The "C" code presented below also uses a 4270, mounted on a Model 4202 MIX Baseboard, as a UMBM. This time, other module in the MIX stack is a Model 4271 Quad Digital Drop Reciever. Each processor on the 4270 takes responsibility for data management of one of the channels on the 4271, independently controlling the sample rate, local oscillator frequency, and decimation factor.

```
/*******************************p4271.c*******************************/
#define PROC_A            0
#define PROC_B            1
#define PROC_C            2
#define PROC_D            3

#define CTLA              0x41
#define CTLB              0x4100
#define CTLC              0x410000
#define CTLD              0x41000000

/* Mix module addresses */
#define MOD0_ADDR         0xc0000000u
#define MOD1_ADDR         0xc4000000u
#define MOD2_ADDR         0xc8000000u
/*************************************************************************/
#define MOD_ADDR          MOD0_ADDR
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.4 p4271.c (continued)

```
/* 4271 Constants */
#define GreyA        ((unsigned int *)(MOD_ADDR | 0x0000u))
#define GreyB        ((unsigned int *)(MOD_ADDR | 0x0010u))
#define GreyC        ((unsigned int *)(MOD_ADDR | 0x0020u))
#define GreyD        ((unsigned int *)(MOD_ADDR | 0x0030u))
#define CtlReg       ((unsigned int *)(MOD_ADDR | 0x0050u))
#define Stat         ((unsigned int *)(MOD_ADDR | 0x0051u))
#define InputCtlReg  ((unsigned int *)(MOD_ADDR | 0x0052u))
#define FifoA        ((unsigned int *)(MOD_ADDR | 0x0040u))
#define FifoB        ((unsigned int *)(MOD_ADDR | 0x0041u))
#define FifoC        ((unsigned int *)(MOD_ADDR | 0x0042u))
#define FifoD        ((unsigned int *)(MOD_ADDR | 0x0043u))

/* 'C40 Timer Addresses */
volatile unsigned int *timer0_address;
volatile unsigned int *timer1_address;

unsigned int *fifo;
unsigned int ctlmask;
unsigned int *int01_vector;

/*************************************************************************/
void c_int03()
{
   unsigned int temp;
   unsigned int i;

      toggle_led();

   for (i=0;i<256;i++)
   {
      temp=*fifo;
      temp=*fifo;
   }
   temp = *Stat;
   asm("   iack @0");
}
/*************************************************************************/
main()
{
   unsigned int *c40_vme_intr;
   unsigned int *c40_reg;
   unsigned int temp;
   int      proc;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.4 p4271.c (continued)

```
c40_vme_intr = (unsigned int *)0x340009u;
c40_reg = (unsigned int *)0x340000u;
timer0_address =(unsigned int *)0x100020u;
timer1_address =(unsigned int *)0x100030u;

*(unsigned int *) 0x34000d = 0xff;

/* Clear interrupt from VME */
*((unsigned int *)0x340004) = 0;

/* Enable Mix mod intr 0 */
int01_vector = (unsigned int *)c_int03;
asm("   ldep ivtp,ar0");
asm("   ldi @_int01_vector,r0");
asm("   sti r0,*+ar0(6)");
asm("   iack @0");

asm("   ldep ivtp,ar0");
asm("   ldi @_int01_vector,r0");
asm("   sti r0,*+ar0(5)");
asm("   iack @0");

asm("   ldi 09900h,iif");
asm("   or 2000h,st");

proc = (*c40_reg & 0x300000u) >> 20;

/* If processor A, initialize the 4271 */
if (proc == PROC_A )
{
   *CtlReg = 0;
   *InputCtlReg = 0;
   InitA();                          /* Initialize the Grey chips */
   InitB();
   InitC();
   InitD();
   fifo = FifoA;
   ctlmask = CTLA;
}

else if (proc == PROC_B)
{
fifo = FifoB;
ctlmask = CTLB;
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.4 p4271.c (continued)

```
   else if (proc == PROC_C)
   {
      fifo = FifoC;
      ctlmask = CTLC;
   }
   else
   {
      fifo = FifoD;
      ctlmask = CTLD;
   }
   temp = *Stat;
   *CtlReg &= ~ctlmask;
   *CtlReg |= ctlmask;
   while(1);
}
/*************************************************************************/
InitA()
{
   /* Sample rate = 10 MHz, LO = 10 MHz */
   /* Freq = ((2**28) / 25.6MHz) * LO */
   GreyA[0] = 0xcc;
   GreyA[1] = 0xcc;
   GreyA[2] = 0xcc;
   GreyA[3] = 0x04;
   GreyA[4] = 0x09;            /* Disable SS, load center freq always */
   GreyA[5] = 0x1;             /* quarter rate (real mode), normal BS */
   GreyA[6] = 0x06;                                    /* real mode */
   GreyA[7] = 0x0;
   GreyA[8] = 0x1f;               /* set overall decimation to 2D=64 */
   GreyA[9] = 0x0;
   GreyA[0xa] = 0x80;                                  /* real mode */
}
/*************************************************************************/
InitB()
{
   GreyB[0] = 0xcc;                                     /* 04e56041 */
   GreyB[1] = 0xcc;
   GreyB[2] = 0xcc;
   GreyB[3] = 0x04;                /* Sample rate=20 MHz, LO=9.9 MHz */
   GreyB[4] = 0x09;             /* Disable SS,load center freq always */
   GreyB[5] = 0x1;             /* quarter rate (real mode), normal BS */
   GreyB[6] = 0x06;                                    /* real mode */
   GreyB[7] = 0x0;
   GreyB[8] = 0x1f;                  /* set overall decimation to 64 */
   GreyB[9] = 0x0;
   GreyB[0xa] = 0x80;                                  /* real mode */
}
/*************************************************************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.4 p4271.c (continued)

```
InitC()
{
   GreyC[0] = 0xcc;                                        /* 04fdf3b6 */
   GreyC[1] = 0xcc;
   GreyC[2] = 0xcc;
   GreyC[3] = 0x04;              /* Sample rate=25.6 MHz, LO=9.8 MHz */
   GreyC[4] = 0x09;            /* Disable SS, load center freq always */
   GreyC[5] = 0x1;             /* quarter rate (real mode), normal BS */
   GreyC[6] = 0x06;                                    /* real mode */
   GreyC[7] = 0x0;
   GreyC[8] = 0x1f;                 /* set overall decimation to 64 */
   GreyC[9] = 0x0;
   GreyC[0xa] = 0x80;                                  /* real mode */
}
/*********************************************************************/
InitD()
{
   GreyD[0] = 0xcc;                                        /* 0516872b */
   GreyD[1] = 0xcc;
   GreyD[2] = 0xcc;
   GreyD[3] = 0x04;               /* Sample rate=20 MHz, LO=9.7 MHz */
   GreyD[4] = 0x09;            /* Disable SS, load center freq always */
   GreyD[5] = 0x1;             /* quarter rate (real mode), normal BS */
   GreyD[6] = 0x06;                                    /* real mode */
   GreyD[7] = 0x0;
   GreyD[8] = 0x1f;                 /* set overall decimation to 64 */
   GreyD[9] = 0x0;
   GreyD[0xa] = 0x80;                                  /* real mode */
}
/*********************************************************************/
led_on()
{
   *timer0_address = 0x302u;                                /* led ON */
}
/*********************************************************************/
led_off()
{
   *timer0_address = 0x306u;                               /* led OFF */
}
/*********************************************************************/
toggle_led()
{
   long int i;
   *timer0_address = ((*timer0_address & 0x306u) ^ 0x4);
      for (i=0; i<1000; i++);
}
/*****************************end p4271.c*****************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.5 mod.h

The header file **mod.h** is a general purpose file used to set up addressing for any slave MIX module on any MIX baseboard, with or without an Upper MIX Bus Master (UMBM) Processor module. It is presented here to illustrate the 4270's use as a UMBM on the Pentek baseboards that support such access (i. e., the 4200/01(A) series, 4202, 4284 and 4285).

```
/* mod.h -  This file determines a MIX Module's base address, based
            on the processor which is running the program, the MIX
            Baseboard, and the MIX stack position(s) of the Slave
            and/or Master module(s).

NOTE: This file only supports the following DSP processor boards:
               4283          4254          4257
               4284          4270          4285
Defining the Baseboard:
   Set the appropriate BB_42XX constant to a 1 and the others to 0

To run a program on a 4270, 4254, or 4257 Upper MIX Bus Master (UMBM) Processor:
   Set the MM_4270 constant if processor is on the 4270
   Set the MM_425X constant if processor is on the 4254 or 4257

To run a program on a 4285, 4284 or 4283 Baseboard Processor:
   Clear the MM_4270 and MM_425X constants

Specifying the MIX stack position of MIX Slave Module being addressed:
   For MIX stack position 0, set MOD to MOD0_ADDR
   For MIX stack position 1, set MOD to MOD1_ADDR
   For MIX stack position 2, set MOD to MOD2_ADDR

Running programs on the 4283 MIX Baseboard Processor (Setting the Page Register):
   To select MIX stack position 0, set MOD_SELECT to MOD0_SELECT
   To select MIX stack position 1, set MOD_SELECT to MOD1_SELECT
   To select MIX stack position 2, set MOD_SELECT to MOD2_SELECT      */

/* Select your Baseboard with a '1' - leave other constants at '0'    */
#define BB_4283   0
#define BB_4284   0
#define BB_4285   0
#define BB_4200   0
#define BB_4201   0
#define BB_4202   0

/* Define MIX Processor Module Type -
                     Only if program is to be run on a UMBM Processor */
#define MM_4270   1
#define MM_425X   0
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.5 **mod.h** (continued)

```
/* Specify MIX Stack Position of the Slave module being addressed     */
#define MOD            MOD1_ADDR

#if MM_4270 | MM_425X
   /* Specify MIX Stack Position for the 4270 or 425X UMBM Processor board */
   #define MIX_POS   0
#endif

#if BB_4283
   /* The 4283 baseboard selects MIX Slave Modules with its Page Register */
   #define MOD_SELECT   MOD1_SELECT
#else
   #define MOD_SELECT   MOD    /* Only being set to keep code similar */
#endif

#if BB_4285
   /* The 4285 has 2 independent paths for accessing the MIX bus      */
   /* resources;  the Expansion Bus and the Global I/O bus.           */
   /* The Expansion Bus may be mastered by any of the on-card 'C40    */
   /* processors or by off-card VMEbus masters.  The Global I/O Bus   */
   /* can be mastered only by 'C40 processors A, B, E or F.  Upper    */
   /* MIX bus Mastering is also supported by the 4285.                */
   /* NOTE: MIX interrupts are not supported over the Global I/O Bus. */

   #define EXP_BUS   1
   #define GLOB_BUS  0
#endif

/* Baseboard = 4283 - This Baseboard does not support Upper MIX Bus Masters */
#if BB_4283
   /* The 4283 Baseboard uses its page register to select
                                  the MIX Slave Module being accessed */
   #define MOD0_SELECT     0x18000
   #define MOD1_SELECT     0x10000
   #define MOD2_SELECT     0x8000

   #define  MOD0_ADDR      0x00804000
   #define  MOD1_ADDR      0x00804000
   #define  MOD2_ADDR      0x00804000

   /* Some MIX Modules use MIX Address Bit A27 -
      This bit is also set in the Page Register on this baseboard     */
   #define D17_DATA_BIT    0x00020000

   /* The "real" A27 address bit is unused on the 4283                */
   #define A27_ADDR_BIT    0x00000000

#endif
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.5 mod.h (continued)

```
/* Baseboard = 4284 */
#if BB_4284
  #if MM_4270 | MM_425X  /* UMBM Processor Module = 4270, 4254, or 4257 */
      #define  MOD0_ADDR   0xC0000000
      #define  MOD1_ADDR   0xC4000000
      #define  MOD2_ADDR   0xC8000000

      /* 'C40 Address Bit that corresponds to MXA27 */
      #define A27_ADDR_BIT    0x02000000

   #else                    /* No UMBM */
      #define  MOD0_ADDR   0x20000000
      #define  MOD1_ADDR   0x24000000
      #define  MOD2_ADDR   0x28000000

      /* 'C40 Address Bit that corresponds to MXA27 */
      #define A27_ADDR_BIT 0x02000000
   #endif
#endif

/* Baseboard = 4200 or 4201 */
#if BB_4200 | BB_4201
   #if MM_4270 | MM_425X  /* UMBM Processor Module = 4270, 4254, or 4257 */
      #define  MOD0_ADDR   0xF0000000
      #define  MOD1_ADDR   0xF4000000
      #define  MOD2_ADDR   0xF8000000

      /* 'C40 Address Bit that corresponds to MXA27 */
      #define A27_ADDR_BIT 0x02000000

   #else                    /* No UMBM */
      #define  MOD0_ADDR   0xC0000000
      #define  MOD1_ADDR   0xD0000000
      #define  MOD2_ADDR   0xE0000000

      /* 'C40 Address Bit that corresponds to MXA27 */
      #define A27_ADDR_BIT 0x02000000
   #endif
#endif
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.5 mod.h (continued)

```
/* Baseboard = 4202  */
#if BB_4202
   #if MM_4270 | MM_425X  /* UMBM Processor Module = 4270, 4254, or 4257 */
      #define  MOD0_ADDR      0xC0000000
      #define  MOD1_ADDR      0xC4000000
      #define  MOD2_ADDR      0xC8000000

      /* 'C40 Address Bit that corresponds to MXA27 */
      #define A27_ADDR_BIT    0x02000000
   #endif
#endif

/* Baseboard = 4285 */
#if BB_4285
   #if MM_4270 | MM_425X  /* UMBM Processor Module = 4270, 4254, or 4257 */
      #define  MOD0_ADDR   0xC0000000
      #define  MOD1_ADDR   0xC4000000
      #define  MOD2_ADDR   0xC8000000

      /* 'C40 Address Bit that corresponds to MXA27 */
      #define A27_ADDR_BIT 0x02000000

   #else                      /* No UMBM */

      #if EXP_BUS
         #define  MOD0_ADDR   0x90000000
         #define  MOD1_ADDR   0x90200000
         #define  MOD2_ADDR   0x90400000

         /* 'C40 Address Bit that corresponds to MXA27 */
         #define A27_ADDR_BIT 0x00100000
      #endif

      #if GLOB_BUS
         #define MOD0_ADDR    0xC8000000
         #define MOD1_ADDR    0xC8200000
         #define MOD2_ADDR    0xC8400000

         /* 'C40 Address Bit that corresponds to MXA27 */
         #define A27_ADDR_BIT 0x00100000
      #endif
   #endif
#endif
/************************* End of File ****************************/
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.6 Timer Routines

The two 'C' routines presented below can be used to start and stop Timer 1 on any of the Model 4270's 'C40 processors.

```
/* timer1.c */
/* timer1 routines */

#ifdef MACH_C40
#define TGCR1 0x100030
#define TC1   0x100034
#define TP1   0x100038
#else
#define TGCR1 0x808030
#define TC1   0x808034
#define TP1   0x808038
#endif


long far *addr;

/* ------------------------------------ */

void start_timer1()
{
  /* start timer1 counter */
  addr = (long far*)TC1;
  *addr = 0;
  addr = (long far*)TP1;
  *addr = 0xffffffff;
  addr = (long far*)TGCR1;
  *addr = 0x03c0;
}

/* ------------------------------------ */

long stop_timer1()
{
  /* stop and read timer1 counter */
  addr = (long far*)TGCR1;
  *addr = 0x0300;
  addr = (long far*)TC1;
  return(*addr);
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.7 Using Timer Interrupts to Avoid Bus Timeouts

This Example demonstrates the use of timer interrupts as a method of forcing a 'C40 off of the 4270's Global bus. This is beneficial when VME, MIX, and the 'C40s are all competing for access to the Global bus, a situation which can result in bus timeouts.

```
/* C40 Status Register - Used to determine the C40 Processor ID       */
#define C40_STATUS    ((unsigned long *)0x340000)

/* Processor Identification Constants */
#define PROC_A        0
#define PROC_B        1
#define PROC_C        2
#define PROC_D        3

/* Global SRAM memory map - Each processor has been offset by 0x1000  */
/*                          to avoid areas used by SwiftNet           */
#define PROC_A_SRAM   0x80001000
#define PROC_B_SRAM   0x80081000
#define PROC_C_SRAM   0x80101000
#define PROC_D_SRAM   0x80181000

/* C40 Timer Addresses */
#define TIMER0_CTRL((unsigned long *)0x100020)
#define TIMER1_CTRL((unsigned long *)0x100030)

/* C40 Timer Specific Constants */
#define TCLK_ON 0x302                        /* Turns TCK Signal ON  */
#define TCLK_OFF 0x306                       /* Turns TCK Signal OFF */
#define CPU_CLOCK_RATE 50.0e6          /* 50 MHz or 40 MHz clock rate */
#define CLOCKS_PER_SEC (CPU_CLOCK_RATE / 2.0)
#define TIMER_CLOCK (CLOCKS_PER_SEC / 2.0)
#define TIMER_FREQ 7.0e-6                  /* Time Between Interrupts */

/* TCK1 Interrupt Service Routine */
void c_int01()
{
   restartTimer();
}

unsigned int *int01_vector;                          /* Pointer to ISR */
unsigned long *SRAMP;                       /* Pointer to Global SRAM */
main()
{
   unsigned int proc;
   register unsigned int i;
   register unsigned long temp = 0x12345678;

   asm("  OR 800h,ST");
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.7 Using Timer Interrupts to Avoid Bus Timeouts (continued)

```
   /* Initialize TCK1 as a General Purpose IO Pin */
   tck1_on();

/* Timer 0 is used to interrupt the processor at a defined rate.      */
/* If VME is timing out on accesses to the Global SRAM, the timeout   */
/* should be less than the VME bus timeout.  This timeout should      */
/* be experimented with to find the best value for your application   */

   setupTimer();

   /* Determine processor */
   proc = (*C40_STATUS & 0x300000u) >> 20;

/* Based on the processor, choose another processor's global memory   */
   if (proc == PROC_A )
      SRAMP = (unsigned long *) PROC_C_SRAM;
   else if (proc == PROC_B)
      SRAMP = (unsigned long *) PROC_D_SRAM;
   else if (proc == PROC_C)
      SRAMP = (unsigned long *) PROC_A_SRAM;
   else
      SRAMP = (unsigned long *) PROC_B_SRAM;

/* To demonstrate the C40 locking up the global bus, a RPTB instruction*/
/* to loop on another processors global memory section.  The timer    */
/* interrupt will not work with the RPTS instruction because it is    */
/* non-interruptable.                                                 */

   asm(" LDI @_SRAMP,AR4");
   asm(" LDI 1234h,R2");
   while (1)
   {
      asm(" LDI 1000h,RC");
      asm(" RPTB ENDLOP");
      asm(" ENDLOP  STI R2,*AR4");

   /* Toggle TCK1 Pin - The TCK1 signal is output to Front Panel.     */
   /* This is useful for monitoring loop with an oscilloscope.        */

      toggle_tck1();
   }
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.7 Using Timer Interrupts to Avoid Bus Timeouts (continued)

```
void setupTimer()
{
   /* Initialize Timer0 Interrupt */
   int01_vector = (unsigned int *) c_int01;
   asm(" LDEP IVTP,AR0");                 /* Get current IVTP Address */
   asm(" LDI @_int01_vector,R0");    /* Store Address of ISR in Table */
   asm(" STI R0,*+AR0(2)");
   asm(" OR 0001H,IIE");                   /* Enable Timer0 Interrupt */
   asm(" OR 2000H,ST");                   /* Enable Global Interrupts */

   /* Initialize Timer to interrupt every 10 usec */

   *(unsigned long *)(TIMER0_CTRL+8) = TIMER_FREQ * TIMER_CLOCK;
   *(unsigned long *)(TIMER0_CTRL) = 0x2c1;
}
void restartTimer()                              /* Restarts the Timer */
{
   *(unsigned long *)(TIMER0_CTRL+8) = TIMER_FREQ * TIMER_CLOCK;
}

/* The following routines are used for measuring loop timing */
void tck1_on()
{
   *TIMER1_CTRL = 0x302u;                                    /* TCK ON */
}

void tck1_off()
{
   *TIMER1_CTRL = 0x306u;                                  /* TCK OFF */
}

void toggle_tck1()
{
   *TIMER1_CTRL = ((*TIMER1_CTRL & 0x306u) ^ 0x4);
}
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A.8 Emulator Support Files

The two files listed below are for use with the Texas Instruments XDS-510 'C40 Emulator. The first is a script file that provides the Emulator with the necessary memory mapping information for the Model 4270. The alias "myreset" may be called at the beginning of routines that the emulator will be used with, to insure that the board initializes properly. The second is a configuration file that tells the emulator how many processors the 4270 contains, and where they reside along the JTAG scan path.

```
; File name:  4269_70.cmd
; 4270 memory map
mr
alias setbus, "e *0x100000 = 0x3dec0000;e *0x100004 = 0x3def7f10"
alias myreset, "reset;setbus"
ma 0x00000000,0x800,ROM
ma 0x002ff800,0x400,RAM
ma 0x002ffc00,0x400,RAM
ma 0x00100000,1,RAM
ma 0x00100004,1,RAM
ma 0x00100020,9,RAM
ma 0x00100030,9,RAM
ma 0x00100040,3,RAM
ma 0x00100050,3,RAM
ma 0x00100060,3,RAM
ma 0x00100070,3,RAM
ma 0x00100080,3,RAM
ma 0x00100090,3,RAM
ma 0x001000a0,9,RAM
ma 0x001000b0,9,RAM
ma 0x001000c0,9,RAM
ma 0x001000d0,9,RAM
ma 0x001000e0,9,RAM
ma 0x001000f0,9,RAM
ma 0x300000,0x4000,ROM          ;Boot Eprom
ma 0x340000,16,RAM              ;C40 control registers
ma 0x700000,16,RAM              ;C40 mix control registers
ma 0xC0000000,0xc000000,RAM     ;MIXbus expansion
ma 0x40000000,0x80000,RAM       ;Local zero wait state memory
ma 0x80000000,0x200000,RAM      ;Dual Port memory
setbus

;File Name:  cpux4.cfg
"CPU_D"         TI320C4x        ;the first 'C40
"CPU_C"         TI320C4x
"CPU_B"         TI320C4x
"CPU_A"         TI320C4x        ;the last 'C40
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page is intentionally blank

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Appendix B: Boot EPROM Source Code Listing

B.1 Boot EPROM Source Code Listing

Below is a 'C40 assembly language listing of the boot code and look/set functions contained in the four factory-installed EPROMs provided by Pentek with your Model 4270. Note that references made in this code to Flash memory are really intended to mean the Global SRAM for a given processor. The table of pointers presented at the beginning of the listing applies specifically to Processor A on the 4270, as evidenced by the fact that all the addresses (from the 'C40 side) begin with 0x8000. For Processor B, that prefix would be 0x8008, for Processor C it would be 0x8010, and for Processor D it would be 0x8018.

```
.list
*
*       TITLE 'PROCESSOR INITIALIZATION'
*
        .global RESET,INIT,BEGIN
        .global NMI,INT0,INT1,INT2,INT3
        .global NON_MASK,ISR0,ISR1,ISR2,ISR3
        .global LOOK,SET
*
*************************************************************************
*                                                                       *
*                    LOOK/SET memory usage                              *
*       HOST      C40                                                   *
*       --------  ----------                                            *
*       0x000200  0x80000200  = Function address from Host              *
*       0x804201  0x80000201  = Address for look/set function           *
*                             = Source address for block move           *
*                             = Flash boot address                      *
*       0x804202  0x80000202  = Data from Host for set function         *
*                             = Destination address for block move      *
*       0x804203  0x80000203  = Data to Host from look function         *
*                             = Word count-1  for block move            *
*       0x804204  0x80000204  = Ready to Host  (0=not ready / 1=ready)  *
*       0x804205  0x80000205  = Pointer to LOOK function                *
*       0x804206  0x80000206  = Pointer to SET function                 *
*       0x804207  0x80000207  = Pointer to  BLOCK MOVE function         *
*       0x804208  0x80000208  = Aux ready flag for factory use          *
*       0x804209  0x80000209  = Test pattern 0xa5a5a5a5                 *
*       0x80420a  0x8000020a  = Test pattern 0x5a5a5a5a                 *
*       0x80420b  0x8000020b  = Test pattern 0x12345678                 *
*       0x80420c  0x8000020c  = Test pattern 0x87654321                 *
*       0x80420d  0x8000020d  = Test pattern 0xffffffff                 *
*       0x80420e  0x8000020e  = Test pattern 0x00000000                 *
*       0x80420f  0x8000020f  = Boot Code Version     -- 1.0a --        *
*       0x804210  0x80000210  = Test pattern 0x80??0200                 *
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
*       FLASH support                                                  *
*                                                                      *
*       0x000211  0x80000211= Pointer to FLASH BOOT function           *
*       0x000212  0x80000212= Pointer to FLASH LOOK function           *
*       0x000213  0x80000213= Pointer to FLASH SET function            *
*       0x000214  0x80000214= Pointer to FLASH ERASE function          *
*       0x000215  0x80000215= Pointer to FLASH ERASE SECTOR function   *
*       0x000216  0x80000216= Pointer to FLASH BYTE LOAD function      *
*       0x000217  0x80000217= Pointer to FLASH BYTE UNLOAD function    *
*       0x000218  0x80000218= Pointer to FLASH WORD LOAD function      *
*       0x000219  0x80000219= Pointer to FLASH WORD UNLOAD function    *
*                                                                      *
************************************************************************
*
*       PROCESSOR INITIALIZATION FOR THE TMS320C40.
*
*
*       IN THIS SECTION, CONSTANTS THAT CANNOT BE REPRESENTED
*       IN THE SHORT FORMAT ARE INITIALIZED.
   .text
*
RESET    .word    INIT        ; RS-load address INIT to HOST
NMI      .word    NON_MASK    ; Non maskable interrupt NMI
TINT0    .word    TIME0       ; Timer 0 interrupt processing
INT0     .word    ISR0        ; INT0-
INT1     .word    ISR1        ; INT1-
INT2     .word    ISR2        ; INT2-
INT3     .word    ISR3        ; INT3-
LOOKADR  .word    LOOK        ; LOOK function
SETADR   .word    SET         ; SET function
MOVEADR  .word    BLKMOV      ; Block move function
NOPFUNC  .word    NOTHIN      ; Do nothing function
PATRNS   .word    PATRN       ; Test patterns
FBOOTA   .word    FBOOT       ; FLASH BOOT function
FLOOKA   .word    FLOOK       ; FLASH LOOK function
FSETA    .word    FSET        ; FLASH SET function
FLDBA    .word    FLDB        ; FLASH LOAD BYTE function
FUNLDBA  .word    FUNLDB      ; FLASH UNLOAD BYTE function
FLDWA    .word    FLDW        ; FLASH LOAD WORD function
FUNLDWA  .word    FUNLDW      ; FLASH UNLOAD WORD function
ERASEA   .word    ERASE       ; FLASH ERASE function
SERASEA  .word    SERASE      ; FLASH SECTOR ERASE function
BWIDE    .word    B_WIDE      ;
WWIDE    .word    W_WIDE      ;
*
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
MASK        .word    0ffffffffH  ;
BOOTF       .word    0007fffffH  ; Boot failed blink value
BOOTP       .word    00007ffffH  ; Boot passed blink value
IIEMASK     .word    000000000H  ; Interrupt enable mask
IIFMASK     .word    000000090H  ; Interrupt enable mask external
IVTADR      .word    0002FFC00H  ;*Beginning address of interrupt vectors
*IVTADR     .word    04003FC00H  ;*Beginning address of interrupt vectors
CTRL        .word    000100000H  ; Pointer for peripheral-bus memory map
GLOINT      .word    03DEC0000H  ; Init of global memory interface control (0)
LOCALINT    .word    03DEF7F10H  ; Init of local memory interface control (4)
FLASHINT    .word    03DEF4110H  ; Init of local during flash access (4)
C40REG      .word    000340000H  ; Processor IO registers
STCK        .word    0002FFF00H  ; Beginning of stack
PATRN       .word    0A5A5A5A5H  ;
            .word    05A5A5A5AH  ;
            .word    012345678H  ;
            .word    087654321H  ;
            .word    0ffffffffH  ;
            .word    000000000H  ;
            .word    00000010aH  ; Version of Boot code
PROC_ADDR   .word    000000000H  ;
PROC_TEST   .word    000000000H  ; test address
*
PROCM       .word    000300000H  ; Mask for processor ID
PROCA       .word    000000000H  ; Processor ID  "A"
PROCB       .word    000100000H  ; Processor ID  "B"
PROCC       .word    000200000H  ; Processor ID  "C"
PROCD       .word    000300000H  ; Processor ID  "D"
*
PRO1_ADDR   .word    80000200H   ; Memory "A"
PRO2_ADDR   .word    80080200H   ; Memory "B"
PRO3_ADDR   .word    80100200H   ; Memory "C"
PRO4_ADDR   .word    80180200H   ; Memory "D"

FBASE       .word    80000000H   ;
FBOOTL      .word    80000000H   ; Points to boot flash flag
FBOOTJ      .word    80000001H   ; Contains flash address to boot from
FSEQ55      .word    00b05555H   ;
FSEQAA      .word    00b02AAAH   ;
FSEQRST     .word    0f0H        ; Read/Reset
FSEQWRT     .word    0a0H        ; Write
FSEQER1     .word    080H        ; Erase seq start
FSEQER2     .word    010H        ; Chip erase
FSEQER3     .word    030H        ; Sector erase
FLASHFG     .word    0A55A5AA5H  ; Boot flash flag
*FLASHFG    .word    012345678H  ; Boot flash flag TEMPORARY
FBOOTT      .word    0           ; Holds boot address
*
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
INIT                                ;
      LDPK  02Fh                    ; Point the DP register to page 2Fh
      LDA   @CTRL,AR0               ; Point to control register
      LDI   @GLOINT,R0              ; Init global memory interface control
      STI   R0,*AR0                 ;
      LDI   @LOCALINT,R0            ; Init local memory interface control
      STI   R0,*+AR0(4)             ;
*
      LDI   @STCK,SP                ; Init stack pointer to 2fff00h
      LDI   1800H,ST                ; Clear and enable cache and
                                    ; disable OVM
      LDI   @IVTADR,R0              ; Setup interrupt vectors
      LDPE  R0,IVTP                 ; Set expansion register
*
* Turn on front panel LED
      LDI   0302H,R0                ; LED on
      STI   R0,*+AR0(020H)          ; Set register
*
      LDA   @C40REG,AR3             ;
      LDI   *AR3,R0                 ; SEE WHICH PROCESSOR THIS IS
      AND   @PROCM,R0               ;
      BZ    PROC1S                  ; Is processor "A"
      CMPI  @PROCB,R0               ; See if processor "B"
      BZ    PROC2S                  ;
      CMPI  @PROCC,R0               ; See if processor "C"
      BZ    PROC3S                  ;
PROC4S
      LDI   @PRO4_ADDR,R0           ; Is processor "D"
      B     PROCG                   ;
PROC3S
      LDI   @PRO3_ADDR,R0           ;
      B     PROCG                   ;
PROC2S
      LDI   @PRO2_ADDR,R0           ;
      B     PROCG                   ;
PROC1S
      LDI   @PRO1_ADDR,R0           ;
PROCG
      STI   R0,@PROC_ADDR           ;
      ADDI  9,R0                    ;
      STI   R0,@PROC_TEST           ;
      LDI   @PROC_ADDR,R0           ;
      SUBI  0200H,R0                ;
      STI   R0,@FBASE               ; Make flash start point to Global
      STI   R0,@FBOOTL              ;
      ADDI  1,R0                    ;
      STI   R0,@FBOOTJ              ;
*
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
* Setup pointers for HOST to use
      LDA      @PROC_ADDR,AR2    ; Processor base address
      LDI      @NOPFUNC,R0       ; Init GO address
      STI      R0,*+AR2(0)       ;
      LDI      @LOOKADR,R0       ; Init LOOK address
      STI      R0,*+AR2(5)       ;
      LDI      @SETADR,R0        ; Init SET address
      STI      R0,*+AR2(6)       ;
      LDI      @MOVEADR,R0       ; Init BLOCK MOVE address
      STI      R0,*+AR2(7)       ;
      LDI      @FBOOTA,R0        ; Init BOOT address
      STI      R0,*+AR2(17)      ;
      LDI      @FLOOKA,R0        ; Init LOOK address
      STI      R0,*+AR2(18)      ;
      LDI      @FSETA,R0         ; Init SET address
      STI      R0,*+AR2(19)      ;
      LDI      @ERASEA,R0        ; Init ERASE address
      STI      R0,*+AR2(20)      ;
      LDI      @SERASEA,R0       ; Init SECTOR ERASE address
      STI      R0,*+AR2(21)      ;
      LDI      @FLDBA,R0         ; Init LOAD BLOCK BYTE address
      STI      R0,*+AR2(22)      ;
      LDI      @FUNLDBA,R0       ; Init UNLOAD BLOCK BYTE address
      STI      R0,*+AR2(23)      ;
      LDI      @FLDWA,R0         ; Init LOAD BLOCK WORD address
      STI      R0,*+AR2(24)      ;
      LDI      @FUNLDWA,R0       ; Init UNLOAD BLOCK WORD address
      STI      R0,*+AR2(25)      ;
*
* Store data patterns in memory
      LDA      @PATRNS,AR1       ; Processor test patterns
      LDA      @PROC_TEST,AR4    ; Processor test address
      LDI      *AR1++,R0         ;
      RPTS     6                 ; Move seven patterns
      STI      R0,*AR4++         ;
||    LDI      *AR1++,R0         ;
      STI      R0,*AR4++         ;
      NOP                        ;
      NOP                        ;
      NOP                        ;
*
* Check test patterns
      LDA      @PATRNS,AR1       ; Processor test patterns
      LDA      @PROC_TEST,AR4    ; Processor test address
      LDA      6,AR5             ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
CKDATA
      LDI      *AR1++,R0          ;
||    LDI      *AR4++,R1          ;
      CMPI     R0,R1              ;
      BNZ      READBD             ;
      DB       AR5,CKDATA         ;
CKEND
      NOP                         ;
      NOP                         ;
      NOP                         ;
*
READGD
      LDI      @BOOTP,R0          ; Timer period Pass
      B        SETTMR             ;
READBD
      LDI      @BOOTF,R0          ; Timer period Fail
SETTMR
      STI      R0,*+AR0(028H)     ;
      LDI      03C3H,R0           ; Start timer
      STI      R0,*+AR0(020H)     ;
*
* Set proccessor ready
      LDI      1,R0               ;
      STI      R0,*+AR2(4)        ; Set ready flag
* Check ready flag   -  Not used
*     LDI      *+AR2(4),R1        ; Check ready flag
*     CMPI     1,R1               ; Did it write in ????
*     BNZ      READBD             ; No so blink LED slowly
*
*
      LDA      @C40REG,AR1        ; Point to IO registers
      LDI      1,R0               ;
      STI      R0,*+AR1(3)        ; Clear MIX interrupt if set
      STI      R0,*+AR1(4)        ; Clear VME interrupt if set
      STI      R0,*+AR1(13)       ; Disable NMI timeout interrupt
*
      IACK     *+AR2(0)           ; Reset pld
      CALL     CKBOOT             ; See if boot from flash set
      LDI      @IIEMASK,IIE       ; Disable interrupts
      LDI      @IIFMASK,IIF       ; Enable interrupts - IIF1 only
      IACK     *+AR2(0)           ; Reset pld
      OR       2000H,ST           ; Global interrupt enable
*
*
WAIT
      IDLE                        ;
      B        WAIT               ;
*
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
*
NON_MASK                        ; Non maskable interrupt NMI-loads
                                ; address NMI to PC
      RETI                      ;
TIME0                           ; Timer 0 interrupt processing
      RETI                      ;
*
ISR0                            ; INT0-
      RETI                      ;
ISR1                            ; INT1- Mix interrupt processing
      PUSH    ST                ;
      PUSH    R0                ;
      PUSHF   R0                ;
      PUSH    R1                ;
      PUSHF   R1                ;
      PUSH    AR0               ;
      PUSH    AR1               ;
      PUSH    AR2               ;
      PUSH    AR3               ;
      PUSH    DP                ;
      LDPK    02Fh              ; Point the DP register to page 2Fh
      LDA     @PROC_ADDR,AR2    ; Processor base address
      LDI     2,R0              ;
      STI     R0,*+AR2(8)       ; Set Aux ready flag
      LDI     *+AR2(0),R0       ; Get address from HOST
      CALLU   R0                ;
RETURN
      LDI     1,R0              ;
      STI     R0,*+AR2(4)       ; Set ready flag to Host
      LDA     @C40REG,AR0       ; Point to IO registers
      LDI     1,R0              ;
      STI     R0,*+AR0(3)       ; Clear MIX interrupt
      STI     R0,*+AR0(4)       ; Clear VME interrupt
      POP     DP                ;
      POP     AR3               ;
      POP     AR2               ;
      POP     AR1               ;
      POP     AR0               ;
      POPF    R1                ;
      POP     R1                ;
      POPF    R0                ;
      POP     R0                ;
      POP     ST                ;
      RETI                      ;
LOOK
      LDI     *+AR2(1),AR0      ; Get address from 0x201
      LDI     *AR0,R0           ; Read word
      STI     R0,*+AR2(3)       ; Store into 0x203
      RETSU ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
SET
      LDI      *+AR2(1),AR0   ; Get address from 0x201
      LDI      *+AR2(2),R0    ; Get word to write
      STI      R0,*AR0        ; Store at pointer
      RETSU                   ;
*
BLKMOV
      LDI      *+AR2(1),AR0   ; Get src address from 0x201
      LDI      *+AR2(2),AR1   ; Get destination address
      LDI      *+AR2(3),RC    ; Get count
      RPTB     BLKMVL         ;
      LDI      *AR0++,R0      ;
BLKMVL
      STI      R0,*AR1++      ;
      RETSU                   ;
*
NOTHIN
      RETSU                   ;
FBOOT                         ; FLASH BOOT function
      CALL     FLASHI         ;
      LDI      *+AR2(1),AR1   ; Get BOOT address from 0x201
FBOOTE
      LDHI     010h,AR0       ;
      LDI      0,R0           ; Set start address flag off
      LDI      *AR1++(1),R1   ; Get word width
      LDI      @WWIDE,R10     ;
      LSH      26,R1          ;
      BN       LOAD0          ;
      NOP      *AR1++(1)      ;
      LDI      @BWIDE,R10     ;
      ADDI     2,AR1          ;
LOAD0
      CALLU    R10            ;
      STI      AR2,*AR0       ; Store gbus control reg
      CALLU    R10            ;
      STI      AR2,*+AR0(4)   ; Store lbus control reg
LOAD2
      CALLU    R10            ;
      SUBI3    1,AR2,RC       ; Set block size
      CMPI     -1,RC          ; If zero byte size start program
      BEQ      IVTP_LOAD      ;
      CALLU    R10            ;
      LDI      AR2,AR0        ; Set destination address
      LDI      R0,R0          ; Test start address flag
      LDIZ     AR2,R9         ; Load start address if flag is off
      LDI      -1,R0          ; Set start & dest flag on
      SUBI     1,R10          ; Sub address with loop
      CALLU    R10            ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
      LDI     1,R0          ; Set dest flag off
      ADDI    1,R10         ;
      B       LOAD2         ; Jump to load new block when looped
IVTP_LOAD
      CALLU   R10           ;
      LDPE    AR2,IVTP      ; Load IVTP pointer
TVTP_LOAD
      CALLU   R10           ;
      LDPE    AR2,TVTP      ; Load TVTP pointer
      CALLU   R10           ;
      IACK    *AR2          ; Send IACK signal out
      BU      R9            ; Branch to start of program
LOOP_B
      RPTB    LOAD_B        ; PGM load loop
B_WIDE
      LWL0    *AR1++(1),AR2 ;
      NOP                   ;
      LWL1    *AR1++(1),AR2 ;
      NOP                   ;
      LWL2    *AR1++(1),AR2 ;
      NOP                   ;
      LWL3    *AR1++(1),AR2 ;
      NOP                   ;
      LDI     R0,R0         ; Test load address flag
      BNN     B_END         ;
LOAD_B
      STI     AR2,*AR0++(1) ; Store word to destination
B_END
      RETSU                 ;
LOOP_W
      RPTB    LOAD_W        ; PGM load loop
W_WIDE
      LDI     *AR1++(1),AR2 ;
      LDI     R0,R0         ; Test load address flag
      BNN     W_END         ;
LOAD_W
      STI     AR2,*AR0++(1) ; Store word to destination
W_END
      RETSU                 ;
      CALL    FLASHU        ;
      RETSU                 ;
FLOOK                       ; FLASH LOOK function
      CALL    FLASHI        ;
      LDI     *+AR2(1),AR0  ; Get address from 0x201
      LDI     *AR0,R0       ; Read word
      STI     R0,*+AR2(3)   ; Store into 0x203
      CALL    FLASHU        ;
      RETSU                 ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
FSET                            ; FLASH SET function
      CALL     FLASHI           ;
      LDA      @FSEQ55,AR3      ; Write command
      LDI      0aaH,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQAA,AR3      ;
      LDI      055H,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQ55,AR3      ;
      LDI      @FSEQWRT,R0      ;
      STI      R0,*AR3          ;
      LDI      *+AR2(1),AR3     ; Get address from 0x201
      LDI      *+AR2(2),R0      ; Get word to write
      STI      R0,*AR3          ; Store at pointer
      AND      080H,R0          ;
FSETL
      LDI      *AR3,R1          ; Wait till write complete
      AND      080H,R1          ;
      CMPI     R0,R1            ;
      BNE      FSETL            ;
      CALL     FLASHU           ;
      RETSU                     ;
FLDB                            ; FLASH LOAD BYTE function
      CALL     FLASHI;
      LDI      *+AR2(1),AR0     ; Get src address from 0x201
      LDI      *+AR2(2),AR1     ; Get destination address
      LDI      *+AR2(3),RC      ; Get count
      RPTB     FLDBL            ;
      LDI      *AR0++,R0        ;
      CALL     WRITEFLASH       ;
FLDBL
      NOP      *AR1++           ;
      CALL     FLASHU           ;
      RETSU                     ;
FUNLDB                          ; FLASH UNLOAD BYTE function
      CALL     FLASHI           ;
      LDI      *+AR2(1),AR0     ; Get src address from 0x201
      LDI      *+AR2(2),AR1     ; Get destination address
      LDI      *+AR2(3),RC      ; Get count
      RPTB     FUNLDBL          ;
      LDI      *AR0++,R0        ;
FUNLDBL
      STI      R0,*AR1++        ;
      CALL     FLASHU           ;
      RETSU                     ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
FLDW                            ; FLASH LOAD WORD function
      CALL     FLASHI           ;
      LDI      *+AR2(1),AR0     ; Get src address from 0x201
      LDI      *+AR2(2),AR1     ; Get destination address
      LDI      *+AR2(3),RC      ; Get count
      RPTB     FLDWL            ;
      LDI      *AR0++,R0        ;
      CALL     WRITEFLASH       ;
      NOP      *AR1++           ;
      LSH      -8,R0            ;
      CALL     WRITEFLASH       ;
      NOP      *AR1++           ;
      LSH      -8,R0            ;
      CALL     WRITEFLASH       ;
      NOP      *AR1++           ;
      LSH      -8,R0            ;
      CALL     WRITEFLASH       ;
FLDWL
      NOP      *AR1++           ;
      CALL     FLASHU           ;
      RETSU                     ;
FUNLDW                          ; FLASH UNLOAD WORD function
      CALL     FLASHI           ;
      LDI      *+AR2(1),AR0     ; Get src address from 0x201
      LDI      *+AR2(2),AR1     ; Get destination address
      LDI      *+AR2(3),RC      ; Get count
      RPTB     FUNLDWL          ;
      LDI      0,R0             ;
      LDI      *AR0++,R1        ;
      AND      0FFh,R1          ;
      OR       R1,R0            ;
      LDI      *AR0++,R1        ;
      AND      0FFh,R1          ;
      LSH      8,R1             ;
      OR       R1,R0            ;
      LDI      *AR0++,R1        ;
      AND      0FFh,R1          ;
      LSH      16,R1            ;
      OR       R1,R0            ;
      LDI      *AR0++,R1        ;
      AND      0FFh,R1          ;
      LSH      24,R1            ;
      OR       R1,R0            ;
FUNLDWL
      STI      R0,*AR1++        ;
      CALL     FLASHU           ;
      RETSU                     ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
ERASE                           ; FLASH ERASE function
      CALL     FLASHI           ;
      LDA      @FSEQ55,AR3      ; Erase Chip command
      LDI      0aaH,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQAA,AR3      ;
      LDI      055H,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQ55,AR3      ;
      LDI      @FSEQER1,R0      ;
      STI      R0,*AR3          ;
      LDA      @FSEQ55,AR3      ;
      LDI      0aaH,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQAA,AR3      ;
      LDI      055H,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQ55,AR3      ;
      LDI      @FSEQER2,R0      ;
      STI      R0,*AR3          ;
ERASEL
      LDI      *AR3,R1          ; Wait till ERASE complete
      TSTB     080H,R1          ;
      BZ       ERASEL           ;
      CALL     FLASHU           ;
      RETSU                     ;
SERASE                          ; FLASH SECTOR ERASE function
      CALL     FLASHI           ;
      LDA      @FSEQ55,AR3      ; Erase Chip command
      LDI      0aaH,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQAA,AR3      ;
      LDI      055H,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQ55,AR3      ;
      LDI      @FSEQER1,R0      ;
      STI      R0,*AR3          ;
      LDA      @FSEQ55,AR3      ;
      LDI      0aaH,R0          ;
      STI      R0,*AR3          ;
      LDA      @FSEQAA,AR3      ;
      LDI      055H,R0          ;
      STI      R0,*AR3          ;
      LDI      *+AR2(1),AR3     ; Get address from 0x201
      LDI      @FSEQER3,R0      ;
      STI      R0,*AR3          ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
SERASEL
      LDI      *AR3,R1         ; Wait till ERASE complete
      TSTB     080H,R1         ;
      BZ       SERASEL         ;
      CALL     FLASHU          ;
      RETSU                    ;
FLASHI
      LDA      @CTRL,AR0       ; Point to control register
      LDI      @FLASHINT,R0    ; Init local memory interface control
      STI      R0,*+AR0(4)     ;
      CALL     RFLASH          ; Reset FLASH
      RETSU                    ;
FLASHU
      LDA      @CTRL,AR0       ; Point to control register
      LDI      @LOCALINT,R0    ; Init local memory interface control
      STI      R0,*+AR0(4)     ;
      CALL     RFLASH          ; Reset FLASH
      RETSU                    ;
RFLASH
      LDA      @FSEQ55,AR3     ; Read/Reset command
      LDI      0aaH,R0         ;
      STI      R0,*AR3         ;
      LDA      @FSEQAA,AR3     ;
      LDI      055H,R0         ;
      STI      R0,*AR3         ;
      LDA      @FSEQ55,AR3     ;
      LDI      @FSEQRST,R0     ;
      STI      R0,*AR3         ;
      LDA      @FBASE,AR3      ;
      LDI      *AR3,R0         ;
      RETSU                    ;
READFLASH                      ; Enter with address in AR1
                               ; and data returned in R0
      CALL     FLASHI          ;
      LDI      0,R0            ;
      LDI      *AR1++,R1       ;
      AND      0FFh,R1         ;
      OR       R1,R0           ;
      LDI      *AR1++,R1       ;
      AND      0FFh,R1         ;
      LSH      8,R1            ;
      OR       R1,R0           ;
      LDI      *AR1++,R1       ;
      AND      0FFh,R1         ;
      LSH      16,R1           ;
      OR       R1,R0           ;
      LDI      *AR1++,R1       ;
      AND      0FFh,R1         ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
      LSH     24,R1          ;
      OR      R1,R0          ;
      PUSH    R0             ;
      CALL    FLASHU         ;
      POP     R0             ;
      RETSU                  ;
WRITEFLASH                   ; Enter with address in AR1
                             ; and data in R0
      PUSH    R0             ;
      LDA     @FSEQ55,AR3    ; Write command
      LDI     0aaH,R1        ;
      STI     R1,*AR3        ;
      LDA     @FSEQAA,AR3    ;
      LDI     055H,R1        ;
      STI     R1,*AR3        ;
      LDA     @FSEQ55,AR3    ;
      LDI     @FSEQWRT,R1    ;
      STI     R1,*AR3        ;
      STI     R0,*AR1        ; Store data at destination
      AND     080H,R0        ;
FWRTL
      LDI     *AR1,R1        ; Wait till write complete
      AND     080H,R1        ;
      CMPI    R0,R1          ;
      BNE     FWRTL          ;
      POP     R0             ;
      RETSU                  ;
CKBOOT
      LDA     @FBOOTL,AR1    ;
      LDI     *AR1,R      0  ;
      CMPI    @FLASHFG,R0    ;
      BNE     CKBOOTX        ;
      LDA     @FBOOTJ,AR1    ;
      LDI     *AR1,R0        ;
      STI     R0,@FBOOTT     ;
*     CALL    FLASHI         ;
      LDA     @FBOOTT,AR1    ;
      B       FBOOTE         ; Do boot
CKBOOTX
      RETSU                  ;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

B.1 Boot EPROM Source Code Listing (continued)

```
CKERASE
      LDI      0,R0           ;
      STI      R0,*AR3        ;
      LDI      *AR3,R1        ;
      BNE      CKERASX        ;
      LDI      5,R0           ;
      STI      R0,*AR3        ;
      LDI      *AR3,R1        ;
      CMPI     R0,R1          ;
      BNE      CKERASX        ;
      LDI      0,R0           ;
      STI      R0,*AR3        ;
      LDI      *AR3,R1        ;
      BNE      CKERASX        ;
      CALL     ERASE          ;
CKERASX
      RETSU                   ;
ISR2                          ; INT2-
      RETI                    ;
ISR3                          ; INT3-
      RETI                    ;
      .end
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

This page is intentionally blank

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP