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 4269-015* at our website: *Click HERE*

OPERATING MANUAL

MODEL 4269

Dual TMS320C40 Digital Signal Processor
VMEbus Board

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

Copyright © 1993–1999

Manual Part No: 800.42690 Rev: G – September 17, 1999

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

Pentek Model 4269 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	Adapted from 4270 Manual, Rev. A
02/25/94	A.1	9320XX – Forward	Correct error in 'C40 Command Register tables, pg. 29 (DMA Path Lock & Unlock register addresses reversed). Correct same error in text & tables, pgs. 27, 32 and 36.
6/24/94	B	9320XX – Forward	Removed leftover references to Model 4270, changed total VME address size in Section 3.3.2.1 from 8MB to 4MB. Renamed Bank Size Jumper Block (JB11) to Window Size, and made some changes to table describing it. Numbered all Figures and Tables. Font size reduced.
07/07/94	C	9320XX – Forward	Listed Pentek part# for ordering Comm Port cables. Enable DMA after access, not before. Added tables for valid VMEbus A24 base addressing as a function of SRAM window size. Added table of AM codes. Added section on Global SRAM usage and transfer timing table.
10/03/94	D	9320XX – Forward	Change all references to DMA Enable to Path Lock, to more cor–rectly describe function. Re–format Section 3.1.2 ('C40 Status Reg–ister Bit Description) with sub–sections.
03/10/95	E	9320XX – Forward	Changed name of Path Lock feature to Global bus Lock, to more accurately characterize its function. Add Chapter 4 (Working with the Model 4269), 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 Base–boards. 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). Table2–8, TBC can interrupt VME on IRQ1 or IRQ7, not IRQ3. More Sample Code in Appendix A (Timer & Emulator Support)
10/11/96	F	9610060 – Forward	Identify EEPROM type in Specs. Add Option 060 to (60 MHz 'C40) to 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 2.7 (Card Cage Installation) & 2.8 (POST). Table 3–1 – Indicate that RAM Bank 2 is optional. Correct D7 set–ting in Table 3–28 (1, not 0 . . . Hex setting in table is correct). NMI timeout interrupts disabled under Option 060. Add timing tables for Option 060. Add a table giving initial access overheads in Sec. 4.8. Add info on Global Bus arbitration priorities to Sec. 4.8. Add Tables & Figures to Table of Contents.
09/17/99	G	9610060 – Forward	Combined Sec. 3.1.2 with Sec. 3.1.1, (now all Sec. 3.1.1), and Sec. 3.1.4 with Sec. 3.1.3 (now Sec. 3.1.2). Add shading to tables. Update boot code in Appendix B. Corrected part # for Comm Port Cable (2104, not 2014). A few other minor changes to carryover errors frrom 4270 manual (removed MIX references & references that might lead one to beleive that there are more than two proces–sors here).

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 mate–rials 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, indi–rect, 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 9
1.2 Features 9
1.2.1 Standalone VMEbus Board Operation 9
1.2.2 Global SRAM 9
1.2.3 Local SRAM 10
1.2.4 Local EPROM 10
1.2.5 J–TAG Support 10
1.2.6 Communication Ports 10
1.3 Specifications 11
1.4 Block Diagram 12
Figure 1–1: Block Diagram – Pentek Model 4269 12

Chapter 2: Installation and Connections

2.1 Inspection 13
2.2 Jumper Settings 13
Figure 2–1: Jumper Block Pin Numbering 13
Figure 2–2: Pin Numbering, Jumper Block S1 13
2.2.1 A16 Base Address Jumpers 13
Figure 2–3: Model 4269 PC board drawing showing jumper block locations 14
Table 2–1: A16 Base Address – Binary and Hex Weights of Jumper Positions 15
Table 2–2: A16 Base Address – Hex Digit Settings 15
2.2.2 TMS ID Jumpers 16
Table 2–3: TMS ID Jumpers 16
2.2.3 'C40 Boot Load Address 16
Table 2–4: Boot Address Jumper Blocks 17
Table 2–5: 'C40 Boot Code Address Jumper Settings 17
2.2.4 Global Memory Window Size Jumper 18
Table 2–6: VMEbus A24 Global SRAM Window Size Jumpers 18
2.2.5 VME Interrupt Sources 19
Table 2–7: VMEbus Interrupt Source Selection 19
2.2.6 VME IRQ Level 19
Table 2–8: VME IRQ Level 19
2.2.7 Test Bus Controller Jumpers 20
2.2.8 Other Jumper Blocks 20
2.3 Front Panel Comm Port Connectors 21
Figure 2–4: Model 4269 Front Panel 21
Figure 2–5: Comm Port Cable 21

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

Table of Contents (continued)

Page

Chapter 2: Installation and Connections (continued)

2.4 Comm Port Configuration 22
Figure 2–6: Comm Port Allocation 22
Table 2–9: Comm Port Connectivity 22
2.5 Front Panel Timer I/O Pins and LED Indicators 23
Figure 2–7: Timer Pins and LED Indicators 23
2.5.1 Front Panel LED Indicators 24
2.5.2 Front Panel TCLK1 Timer Pins 24
2.5.3 Front Panel Reset Pins 25
2.6 XDS Emulator Connector 25
Figure 2–8: XDS Emulator Connector 26
2.7 Inserting and Removing the Model 4269 in the VMEbus Card Cage 26
2.7.1 Inserting the Unit 26
2.7.2 Removing the Unit 26
2.8 Power–Up Self–Test 26

Chapter 3: Memory Maps and Operating Procedures

3.1 Local Memory Map 27
Table 3–1: Local Memory Map (for each processor) 27
Table 3–2: Internal Peripherals (for each processor) 27
3.1.1 TMS320C40 Status Registers 28
Table 3–3: TMS320C40 Status Registers 28
3.1.1.1 User Jumper Status 29
3.1.1.2 TMS ID 29
Table 3–4: TMS ID Bits 29
3.1.1.3 Processor Interrupt to VMEbus 29
Table 3–5: VME Clear Interrupt Registers 29
3.1.1.4 Interrupt to Other Processor 30
3.1.1.5 Global Bus Lock 30
Table 3–6: Global Bus Lock Bits 30
3.1.1.6 Interrupt from VMEbus 30
3.1.1.7 Interrupt from Other Processor 30
3.1.2 TMS320C40 Command Registers 31
Table 3–7: TMS320C40 Command Registers 31
3.1.2.1 Clear Interrupt Registers 32
Table 3–8: Processor Clear Interrupt Registers 32
Table 3–9: Status Register – Processor Interrupt Bits 32
Table 3–10: VME Set Interrupt to Processor Registers 32

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)

3.1 Local Memory Map (continued)
3.1.2 TMS320C40 Command Registers (continued)
3.1.2.2 Set Interrupt Registers 33
Table 3–11: Processor Set Interrupt Registers 33
Table 3–12: VME Clear Interrupt Registers 33
3.1.2.3 Global Bus Lock Registers 34
Table 3–13: Global Bus Lock Registers 34
3.1.2.4 Non–Maskable Interrupt (NMI) Registers 34
Table 3–14: Non–Maskable Interrupt Registers 35
3.2 Global Memory Map 35
Table 3–15: Global SRAM Addressing 35
3.3 VMEbus Slave Interface Memory Map 36
Table 3–16: VMEbus Slave Memory Map 36
3.3.1 VMEbus Address Modifiers 36
Table 3–17: Address Modifier Codes 37
3.3.2 VME 'C40 Reset Control Register 37
Table 3–18: VME 'C40 Reset Control Register 37
3.3.2.1 VME Global Bus Lock 38
Table 3–19: Global Bus Lock Registers 38
3.3.2.2 VME Broadcast Mode Enable 38
3.3.2.3 VME Test Bus Controller 38
3.3.2.4 Processor Reset 39
Table 3–20: Processor Reset Bits 39
Table 3–21: Reset Vector Select Bits 39
3.3.3 VME Address Mapping and Control Register 40
Table 3–22: VME Address Mapping and Control Register 40
3.3.3.1 VME Base Address Selection 40
Table 3–23: Hex Weights of A24 Bits Set in Address Register 41
Table 3–24: AddressBit Settings for 1 MByte SRAM Window 41
Table 3–25: Address Bit Settings for 2 MByte SRAM Window 42
Table 3–26: Address Bit Settings for 4 MByte SRAM Window 42
Table 3–27: Address Bit Settings for 8 MB SRAM Window 43
Table 3–28: Address Bit Settings for 16 MB SRAM Window 43
Table 3–29: Hex Weights of A32 Bits in Address Register 43
3.3.3.2 Paged Addressing Mode 44
3.3.3.3 Word Swap Mode 44
3.3.4 Address Page Register 44
Table 3–30: VME Address Page Register 45

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)

3.3 VMEbus Slave Interface Memory Map (continued)
3.3.5 VME Interrupt and Bus Lock Status Register 45
Table 3–31: VME Interrupt and Bus Lock Status Register 45
3.3.5.1 TBC Ready 45
3.3.5.2 Global Bus Locks 46
3.3.5.3 Interrupt Source Identification 46
Table 3–32: VME Interrupt ID Bits 46
3.3.6 VME Command Registers 46
3.3.6.1 VME Set Interrupt Registers 47
Table 3–33: VME Set Interrupt Registers 47
3.3.6.2 VME Clear Interrupt Registers 47
Table 3–34: VME Clear Interrupt Registers 47
3.3.7 VME IACK Vector Address Register 47
Table 3–35: VME IACK Vector Address Register 47
3.3.8 Test Bus Controller Register 48

Chapter 4: Working with the Model 4269

4.1 Introduction 49
4.2 Resetting and Booting the Model 4269's Processors 49
Table 4–1: Boot Address Jumper Blocks 49
Table 4–2: 'C40 Boot Code Address Jumper Settings 49
Table 4–3: SRAM Locations of Pointers to EPROM Functions 50
Table 4–4: Register States After Boot from Factory EPROM 51
4.2.1 Boot Loading User Code 52
4.3 Downloading Programs to the Model 4269 52
4.3.1 Downloading Programs with SwiftNet 52
4.3.2 Downloading SPOX Programs 53
4.3.3 Downloading Programs with the XDS–510 Emulator 53
4.4 Working with Interrupts on the Model 4269 54
4.4.1 The 4269 as a VMEbus Interrupter 54
Table 4–5: VME Interrupt Source Selection 54
Table 4–6: VME IRQ Level 55
Table 4–7: VME Interrupt ID Bits 55
Table 4–8: VME Clear Interrupt Registers 55
4.4.2 Interrupting 'C40s on the 4269 from VME 56
Table 4–9: VME Set Interrupt Registers 56
4.4.3 Inter–Processor Interrupts on the 4269 56
Table 4–10: Status Register – Processor Interrupt Bits 57

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

Table of Contents (continued)

Page

Chapter 4: Working with the Model 4269 (continued)

4.5 Using the Model 4269's Global Bus Lock Feature 57
4.5.1 Global Bus Locked by VME 57
4.5.2 Global Bus Locked by 'C40s on the 4269 58
4.6 Using the Global Memory 59
Table 4–11: Transfer Times (Parallel Instruction Mode) – 50 MHz 'C40 61
Table 4–12: Transfer Times ('C40 DMA Mode) – 50 MHz 'C40 61
Table 4–13: Transfer Times (Parallel Instructions w/ Bus Lock) – 50 MHz 'C40 61
Table 4–14: Transfer Times ('C40 DMA w/ Bus Lock) – 50 MHz 'C40 61
Table 4–15: Transfer Times (Parallel Instruction Mode) – 60 MHz 'C40 62
Table 4–16: Transfer Times ('C40 DMA Mode) – 60 MHz 'C40 62
Table 4–17: Transfer Times (Parallel Instructions w/ Bus Lock) – 60 MHz 'C40 62
Table 4–18: Transfer Times ('C40 DMA w/ Bus Lock) – 60 MHz 'C40 62
Table 4–19: Transfer Times (Parallel Instruction Mode) – 40 MHz 'C40 63
Table 4–20: Transfer Times ('C40 DMA Mode) – 40 MHz 'C40 63
Table 4–21: Transfer Times (Parallel Instructions w/ Bus Lock) – 40 MHz 'C40 63
Table 4–22: Transfer Times ('C40 DMA w/ Bus Lock) – 40 MHz 'C40 63
Table 4–23: Global Bus Access Overheads – for First Access 64
Table 4–24: VMEbus Timing 64

Appendix A: Programming Examples

A.1 Slo_Blnk.C A–1
A.2 Timer Routines A–3
A.3 Using Timer Interrupts to Avoid Bus Timeouts A–4
A.4 Emulator Support Files A–7

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 4269 incorporates two Texas Instruments' TMS320C40s in a standard 6U single−slot VMEbus board with a complete VMEbus slave interface.

With 2 MBytes of Local SRAM and 2 MBytes of shared Global SRAM, the Model 4269 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 Operation

As a standalone VME board, the Model 4269 occupies a single card cage slot. It presents a slave interface to the VMEbus, through which both 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.

1.2.2 Global SRAM

A total of 2 MBytes of shared Global SRAM is accessible by both 'C40s 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 both '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 one 'C40 addresses the other 'C40's Global SRAM, a single wait state is generated.

The Model 4269 can be programmed such that both shared Global SRAMs can be written to simultaneously by a VMEbus Master. This provides fast downloading of data and programs to these memories using a single write sequence instead of two.

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

1.2 Features (continued)

1.2.3 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.

1.2.4 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 sub−routines or development system monitor support.

1.2.5 J−TAG Support

J−TAG Emulator Support is also included in the 4269. Two methods are provided for accessing the J−TAG scan path feature of both '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.

1.2.6 Communication Ports

One of the six 20 MByte/sec Communication Ports on each 'C40 are connected to the other 'C40. Three of the remaining five 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, 4270, 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 (2)
- **Type:** Texas Instruments' TMS320C40
- **Clock**
  - **Standard:** 40 MHz
  - **Option 015:** 50 MHz
  - **Option 060:** 60 MHz

Local SRAMs (2)
- **Memory Size**
  - **Standard:** 256k x 32 (1 MByte), for each 'C40
  - **Option 002:** 512k x 32 (1 MByte), for each 'C40
- **Speed:** 0 wait states
- **Access:** TMS320C40 Local bus

Global SRAMs (2)
- **Memory Size**
  - **Standard:** 256k x 32 (1 MByte), for each 'C40
  - **Option 001:** 512k x 32 (2 MByte), for each 'C40
- **Speed:** 0 'C40 wait state (after arbitration) for processor's own Global SRAM, 1 'C40 wait state (after arbitration) for other processor's Global SRAM
- **Access:** TMS320C40 Global bus and VMEbus

EPROMs (2)
- **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:** 6 total, 3 per 'C40
- **Connector Type:** 16–pin, 3M 50116–B000

Miscellaneous Front Panel Connectors & Indicators
- **Emulation:** Texas Instruments J–TAG XDS Support
- **Misc.:** 6–pin header, with one (common) $\overline{\text{Reset}}$ Input for the two 'C40s, and two 'C40 TCLK1 I/O connections
- **Indicators:** 1 LED per 'C40 (two 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

Power Requirements
- **VMEbus:** 2.5 A typ. @ $+5V_{DC}$

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

1.3 Specifications (continued)

Dimensions

Height:	9.20″
Depth:	6.30″
Width:	0.80″

Ordering Info

Model 4269:	Dual 40 MHz TMS320C40, 2x 1 MB Local SRAM, 2x 1 MB Global SRAM
Option 001:	2x 2 MB Global SRAM
Option 002:	2x 2 MB Local SRAM
Option 015:	Dual 50 MHz TMS320C40
Option 060:	Dual 60 MHz TMS320C40

1.4 Block Diagram

The block diagram of the Pentek Model 4269 Dual TMS320C40 Digital Signal Processor VME Board is presented in Figure 1–1, below.

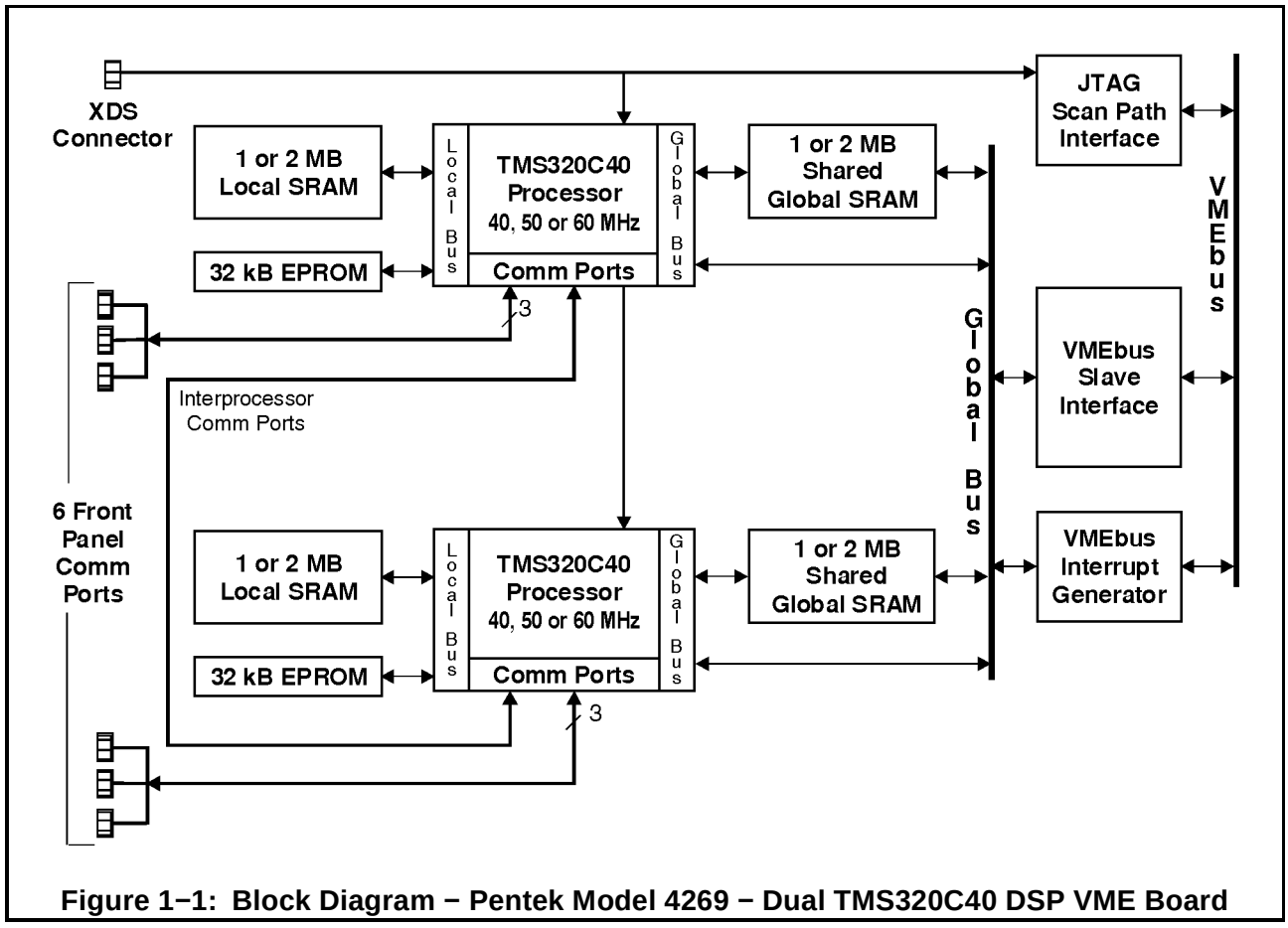


Figure 1–1: Block Diagram – Pentek Model 4269 – Dual TMS320C40 DSP VME Board

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

Figure 2–3, on the following page, is a drawing of the Model 4269's circuit board, showing the locations of the jumper blocks referred to in the sections below. Figure 2–1, at the right, shows the pin numbering for most of the double–row jumper blocks used in the Model 4269 (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 4269 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.

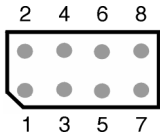


Figure 2–1: Jumper block Pin Numbering

2.2.1 A16 Base Address Jumpers – Jumper Block S1

The upper byte of the Model 4269's A16 base address is set by the installation of shorting jumpers on jumper block S1. All of the registers described in Section 3.3 of this manual (VMEbus Interface Memory Map) are referenced as offsets from the A16 base address set by this jumper block.

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 a DIP–packaged IC, as shown in Figure 2–2, below.

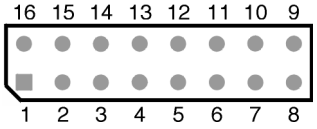


Figure 2–2: Model 4269 – Pin Numbering, Jumper Block S1

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

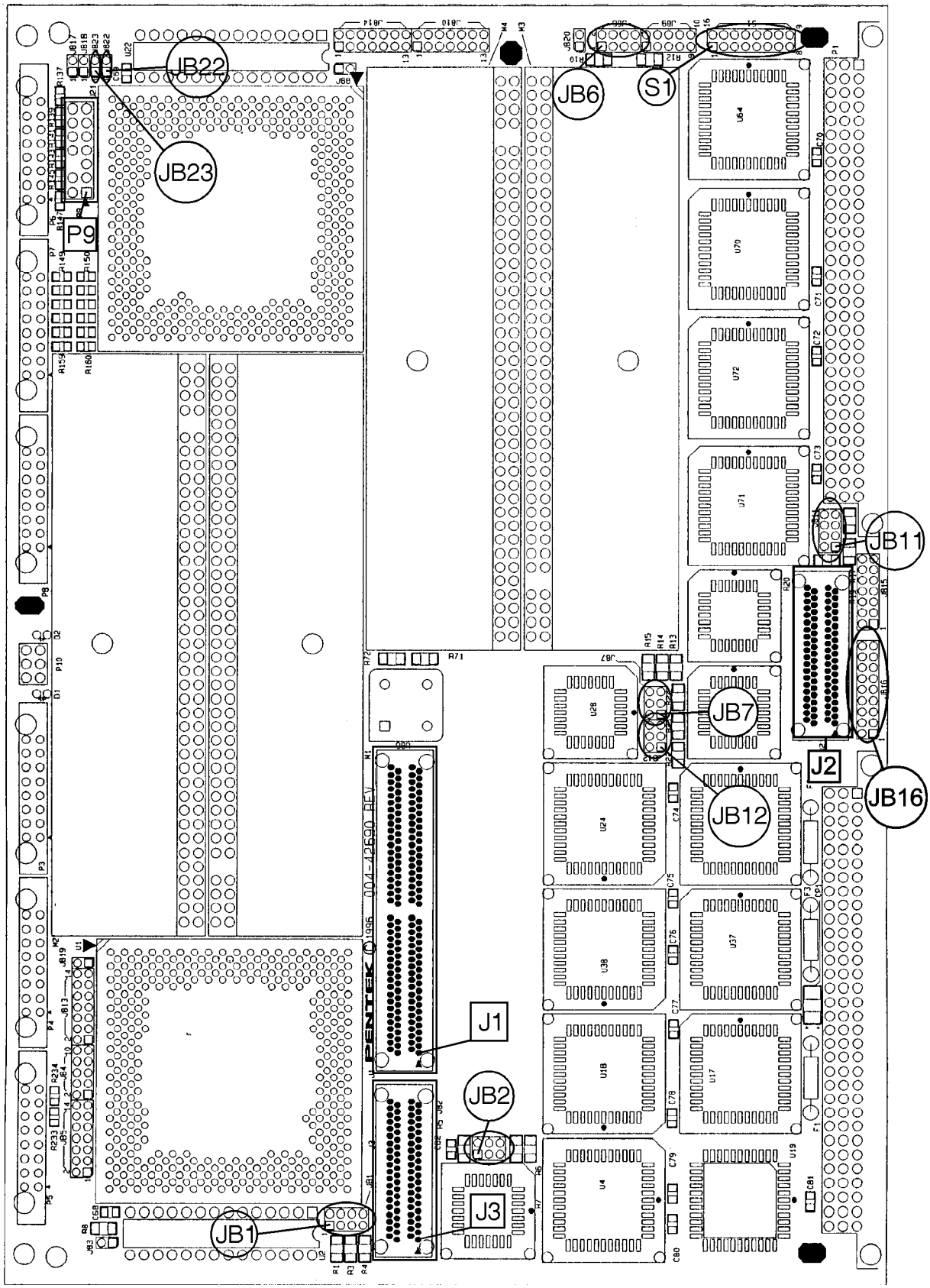


Figure 2−3: Model 4269 PC board drawing showing jumper block & connector locations − (lines point to pin 1)

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

2.2 Jumper Settings (continued)

2.2.1 A16 Base Address Jumpers (continued)

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.

Tables 2–1 and 2–2, below, 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 4269 – 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)*

Table 2–2: Model 4269 – 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

These jumper blocks are used by software routines to identify the two individual processors on the 4269, 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 4269 board is set to A by jumpers between pins 1 & 2, and pins 3 & 4 of jumper block JB1. A jumper between pins 3 & 4 of jumper block JB6 sets the TMS ID of the processor at the top of the board to B. 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 4269 – TMS ID Jumpers

Jumper Block	pins 1 & 2	pins 3 & 4	TMS ID	pins 5 & 6	pins 7 & 8
JB1	ON	ON	Processor A	ON*	ON*
JB6	OFF	ON	Processor B	ON*	ON*
* – User jumpers – Read from TMS Status Register					

2.2.3 'C40 Boot Load Address – JB2 and JB7

The address of the code you wish to use to boot the 'C40 processors on your 4269 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 Register (see Section 3.3.2 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. Jumper block JB7 sets the boot load address for Processor B. 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 4269, 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 4269 – Boot Address Jumper Blocks

JB2	JB7
Processor A	Processor B

Table 2–5: Model 4269 – '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	**Not Available**	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 and 3.3.6 of this manual for further information about handling interrupts to the 'C40s on the 4269. See the Texas Instruments **TMS320C4x User's Guide**, Section 3.1.10 (IIOF Flag Register), for further details.

(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.

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)

(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

Jumper block JB11 sets the VMEbus window size in A24 space for the Global Memory, which is accessible by both processors and by VMEbus Masters. In A32 VME space, the Model 4269's Global Memory will occupy a 16 MByte memory region, regardless of the settings of these jumpers.

As shipped from the factory, the 4269's two 1 MByte Global SRAMs will occupy a 4 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. If the window size is reduced to 1 MByte, the page register can be activated to allow access to the other processors' memory. See Sections 3.3.3.2 and 3.3.4 for further details. Table 2–6, below, describes the settings of the VMEbus window size jumper block.

Table 2–6: Model 4269 – 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				

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 4269 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 3.3.3.1 for further details.

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

2.2 Jumper Settings (continued)

2.2.5 VME Interrupt Sources – JB16

Jumper Block JB16 on the VME board assigns the various on–card interrupt sources to VME IRQ levels, to allow the 4269 to send interrupts to the VME–bus. Table 2–7, below, 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 4269 may only be assigned to a single IRQ level, this jumper block essentially allows the user to select either any single processor, or both processors, as valid VME interrupt sources.

Table 2–7: Model 4269 – VMEbus 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	–	–	–
Procs. A&B	4 – 3	4 – 6	*15 – 17	15 – 13	16 – 18	16 – 14	4 – 2
* – Factory Default Setting							
NOTES: (1) Procs A&B = either processor may interrupt VME (2) TBC = Test Bus Controller							

2.2.6 VME IRQ Level – JB12

Jumper Block JB12 on the VME board sets the 4269's VME IRQ Level. This block should be configured for the same level as JB16 (see above) to allow the 4269 to send interrupts to the VMEbus. Table 2–8, below, shows the settings.

Table 2–8: Model 4269 – 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 IRQ 0 = VME Interrupts disabled			

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

2.2 Jumper Settings (continued)

2.2.7 Test Bus Controller Jumpers – 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, in which case these jumpers would be left off (the fac–tory default). **These jumpers must NOT be installed if the XDS510 Emula–tor is connected.**

2.2.8 Other Jumper Blocks

Jumper blocks which are not discussed in the sections above are present on the board 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.

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

2.3 Front Panel Comm Port Connectors

The Model 4269 provides front panel access to three communications ports, referred to as "Comm Ports", for each of the two TMS320C40s. There are six 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 2–4, at the right, shows the front panel Comm Port layout and the pin assignments. The connectors are keyed to prevent improper insertion.

Comm Port cables (See Figure 2–5, below) 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 4269 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.

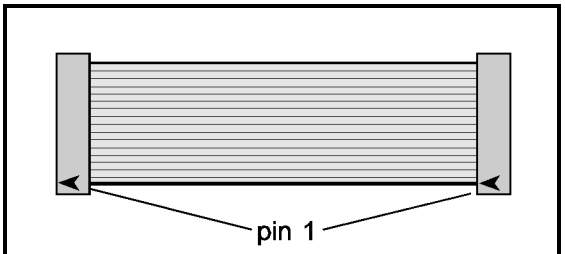


Figure 2–5: Comm Port Cable

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.

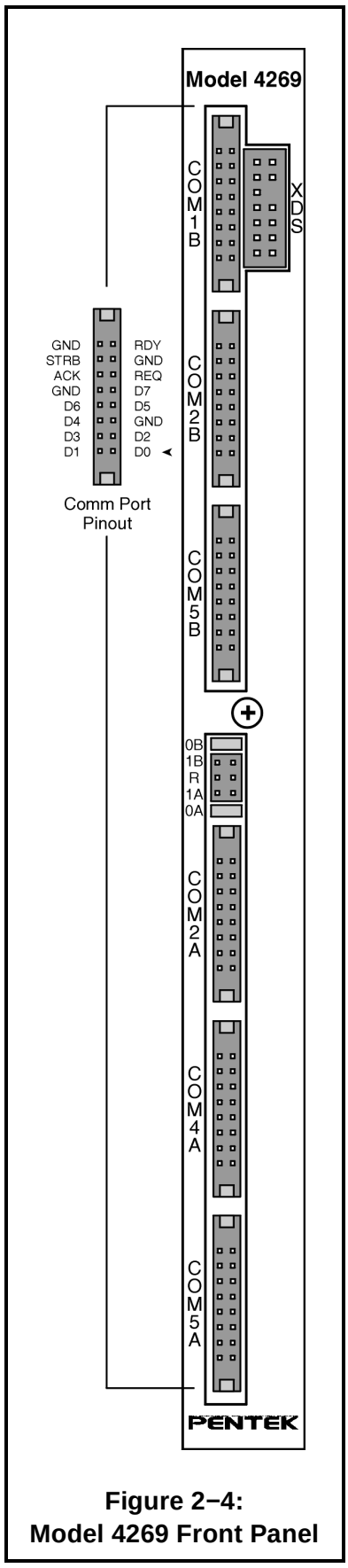


Figure 2–4: Model 4269 Front Panel

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

2.4 Comm Port Configuration

Figure 2–6 and Table 2–9, below, describe the connectivity and the default I/O configuration of the six Comm Ports on each of the two 'C40 processors in the Model 4269. 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-A/0** would indicate a port configured as an input, driven by Processor A, port 0. **FP** indicates a user–accessible, front panel Comm Port. **N/C** denotes an unconnected port.

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 B as an output, all you need to do is address it to write. That port is connected to Port 0 of Processor A. When the data written by Processor B, Port 3 arrives at Port 0 of Processor A, that port will reconfigure itself as an input. See Chapter 8 of the Texas Instruments **TMS320C4x User's Guide** for further details.

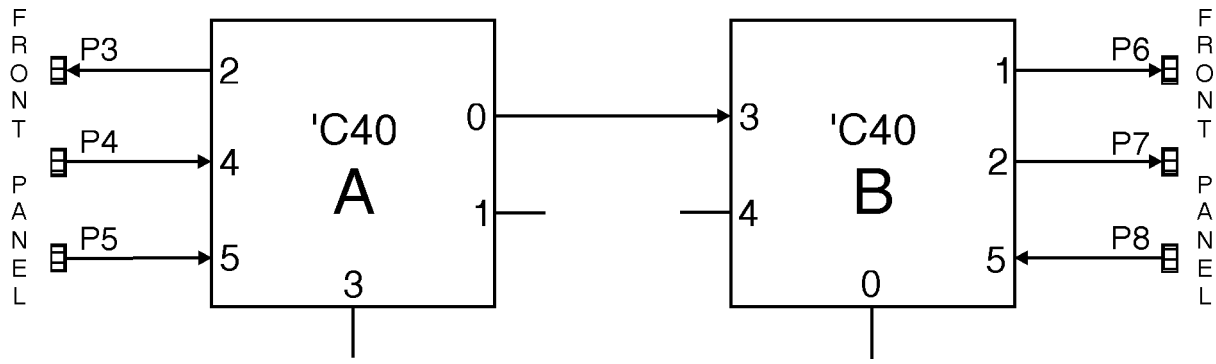


Figure 2–6: Pentek Model 4269 Comm Port Allocation – Default Input/Output Configuration for Comm Ports indicated by arrows

Table 2–9: Model 4269 – Comm Port Connectivity

	Processor A	Processor B
Port 0	O–B/3	N/C
Port 1	N/C	O–FP
Port 2	O–FP	O–FP
Port 3	N/C	I–A/O
Port 4	I–FP	N/C
Port 5	I–FP	I–FP

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

2.4 Comm Port Configuration (continued)

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 4269

2.5 Front Panel Timer I/O Pins and LED Indicators

The Model 4269 provides access to the two timers contained in each of the 'C40s. A 6–pin header and two green LEDs are provided in the center of the panel as shown in Figure 2–7, below. The pin definition and LED assignments are shown in the detail portion of 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).

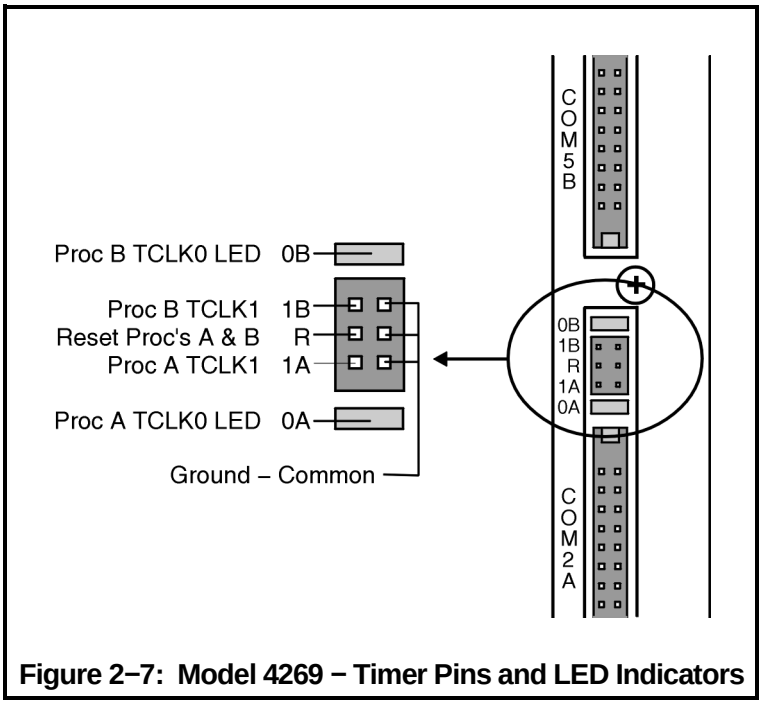


Figure 2–7: Model 4269 – Timer Pins and LED Indicators

Rev. G

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

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

2.5.1 Front Panel LED Indicators

Each of the LEDs is driven through a TTL buffer by the TCLK0 pin on each of the '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.

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 2−7, on the preceding 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, tem−porarily 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.

When configured as an input, the TCLK1 pin may also be used for synchro−nizing 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.

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 (continued)

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.

2.5.3 Front Panel Reset Pins

In addition to the TCLK1 pins, there is a common $\overline{\text{Reset}}$ pin, for both 'C40 Processors. This pin is TTL compatible and is internally pulled up to +5V with a pullup resistor, which provides the inactive logic '1' level.

The active ('true') level for this $\overline{\text{Reset}}$ pin 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.

2.6 XDS Emulator Connector

The Model 4269 has provisions for use with the Texas Instruments XDS510 Emulator. The emulator connector is on the front panel, to the right of the topmost Comm Port connector. The Emulator connector pinout is shown in Figure 2–8, on the top of the next page. These connectors are keyed for proper alignment with the socket connector at the end of the emulator cable. Section 4.3.3 of this manual provides basic informa–tion about the use of the emulator. Refer to the Texas Instruments **TMS320C40 Emula–tor User's Guide** or the documentation supplied with the XDS system for complete operating details.

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

2.6 XDS Emulator Connector (continued)

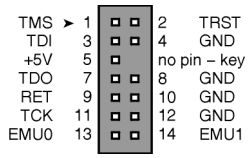


Figure 2-8: Model 4269 - XDS Emulator Connector

2.7 Inserting and Removing the Model 4269 in the VMEbus Card Cage

CAUTION!!

Never insert or remove the Model 4269 while power is turned on in the card cage.

2.7.1 Inserting the Unit

Align the Model 4269 card edge with the cage card guides and push evenly on both ejector handles until the front panel seats against the upper and lower rails. Secure the unit in the cage by tightening the upper and lower retaining screws in the front panel.

2.7.2 Removing the Unit

Loosen the upper and lower retaining screws; they are captive screws and will remain part of the panel assembly. Gently push both ejector handles apart (away from the center of the panel) until the unit unseats itself from the cage connectors. Then pull outward to remove the unit.

2.8 Power–Up Self–Test

When power is applied to the Model 4269, some basic diagnostic routines are run from the EPROM. If no problems are encountered during this procedure, the two LEDs on the 4269'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 fac–tory 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 two TMS320C40 processors on the Model 4269, 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 4269 – 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–0x3FFF FFFF	Reserved
0x4000 0000–0x4003 FFFF	External Local RAM – Bank 0 (Standard – Read/Write)
0x4004 0000–0x4007 FFFF	External Local RAM – Bank 1 (Option 002 – Read/Write)
0x4008 0000–0x400B FFFF	External Local Bus (Read/Write)

Table 3–2: Model 4269 – 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 – Processors A and B

Table 3–3, below, shows the contents of the two 'C40 Status Registers. The registers appear at 'C40 address 0x0034 0000 for each processor. With the exception of bits D23 – D20 (which reflect the states of jumper settings), all used bits in these registers are cleared to the logic '0' state at power–up reset. The post–reset states of the unused bits is undefined. Detailed descriptions of each bit in these registers can be found on the next few pages.

Note that the bit arrangement is identical for both status registers. The only differences occur when reference is made to a specific processor (A or B).

Table 3–3: 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	N/U
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
N/U	N/U	1 = Proc. A Interrupt to Proc. B	N/U	N/U	1=Global Bus Locked by VME	N/U	N/U
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
N/U	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A	1 = VME Interrupt to Proc. A	N/U	N/U	N/U	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	N/U
D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**
N/U	N/U	1 = Proc. B Interrupt to Proc. A	N/U	N/U	1=Global Bus Locked by VME	N/U	N/U
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
N/U	1=Global Bus Locked by Processor B	1=Global Bus Locked by Processor A	1 = VME Interrupt to Proc. B	N/U	N/U	N/U	1 = Proc. A Interrupt to Proc. B

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

3.1 Local Memory Map (continued)

3.1.1 TMS320C40 Status Registers (continued)

The individual bits in the 'C40 Status Registers ('C40 Address 0x0034 0000) for the two processors on the Model 4269 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 (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

Table 3–4: Model 4269 – TMS ID Bits

Processor	D21	D20
A	0	0
B	0	1

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 (see Section 2.2.2). The states of these bits are interpreted to identify each processor as shown in Table 3–4, left.

As previously noted in Section 2.2.2, we strongly discourage users from changing the configuration of the ID jumpers since standard 4269 software routines utilize these bits.

3.1.1.3 Processor Interrupt to VMEbus – Bit D17

Table 3–5: Model 4269 – VME Clear Interrupt Registers

Processor	VMEbus Address
A	A16_base + 0x20
B	A16_base + 0x24

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 4269's A16 Memory Map. Table 3–5, left, gives the addresses of the VME Clear Interrupt Registers for interrupts issued by each processor.

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)

3.1.1.4 Interrupt to Other Processor – Bit D13

A '1' in this bit indicates that the processor whose Status Register you are reading has issued an interrupt to the other processor. This interrupt drives the $\overline{\text{IIOF0}}$ input of the target 'C40.

A processor sets this interrupt by writing to the Set Interrupt Register, at address 0x0034 0005 in its own Memory Map. The interrupt is cleared when the interrupted processor writes to the Clear Interrupt Register, at address 0x0034 0000 in its own Memory Map.

3.1.1.5 Global Bus Lock – Bits D10, D6 and D5

A '1' in any of these three bits indicates that the Global bus on the 4269 is locked. Locking the bus allows the master device that issued the lock exclusive access to the Global SRAM and enables faster block transfers.

Table 3–6: Model 4269–Global Bus Lock Bits

Bit	Bus locked by...
D10	VME
D6	Processor A
D5	Processor B

Processors lock the Global bus by writing to address 0x0034 000B in their own Memory Maps, and unlock it by writing to 0x0034 000A. VMEbus Masters lock the Global bus by setting bit D15 of the VME Reset Control Register, at the A16_base address (see Section 3.3.2.1), to the logic '1' state, and unlock it by clearing that bit to the logic '0' state. In all cases, one access over the bus must be made **before** locking it.

Table 3–6, at the left, tells which Status Register bit gives the Global bus lock state for which resource.

3.1.1.6 Interrupt from VMEbus – Bit D4

A logic '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.

3.1.1.7 Interrupt from Other Processor – Bit D0

A logic '1' in this bit indicates that the processor whose Status Register you are reading has received an interrupt from the other processor. This interrupt is delivered to the 'C40 at its $\overline{\text{IIOF0}}$ 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)

3.1.1.7 Interrupt from Other Processor (continued)

A processor sets this interrupt by writing to the Set Interrupt Register, at address 0x0034 0005, in its own Memory Map. The interrupt is cleared when the interrupted processor writes to the Clear Interrupt Register, at address 0x0034 0000, in its own Memory Map.

3.1.2 TMS320C40 Command Registers – Processors A and B

Table 3–7, below, summarizes the addresses and functions of the Command Registers associated with the two 'C40 processors on the Model 4269. 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 on the next few pages.

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

Table 3–7: Model 4269 – TMS320C40 Command Registers (write only) – Processors A and B

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	N/U	Not Used (Reserved)	**0x0034 0001**	N/U	Not Used (Reserved)
0x0034 0002	N/U	Not Used (Reserved)	**0x0034 0002**	N/U	Not Used (Reserved)
0x0034 0003	N/U	Not Used (Reserved)	**0x0034 0003**	N/U	Not Used (Reserved)
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	N/U	Not Used (Reserved)	**0x0034 0006**	N/U	Not Used (Reserved)
0x0034 0007	N/U	Not Used (Reserved)	**0x0034 0007**	N/U	Not Used (Reserved)
0x0034 0008	N/U	Not Used (Reserved)	**0x0034 0008**	N/U	Not Used (Reserved)
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*
* – Applies only to board revisions D and higher.					

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)

A detailed description of the Local Command Registers for each of the 'C40 processors on the Model 4269 is given below. For all the registers listed in the following sections, the indicated function is caused to execute by writing to the given address, regardless of what data is written.

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

Table 3–8: Model 4269 – Processor Clear Interrupt Registers

Address	Interrupted Processor	
	A	B
0x0034 0000	B	A
0x0034 0004	V M E	

Table 3–9: Model 4269 – Status Register – Proc. Interrupt Bits

Bit	Interrupted Processor	
	A	B
D0	B	A
D4	V M E	

Table 3–10: Model 4269 – VME Set Interrupt to Processor Registers

A	B
A16base+0x10	A16base+0x14

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 3–8, 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 3–9, 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., the other processor or the VME–bus) by writing to the appropriate Set Interrupt iRegister in its own address map. In order to interrupt the other processor, either processor writes to address 0x0034 0005 in its own Local Memory Map. This interrupt drives the target 'C40's $\overline{\text{IIOF0}}$ input. Table 3–10, at the left, tells which address a VMEbus Master should write, in order to interrupt a given processor. The output of the VME Set Interrupt register drives the 'C40's IIOF1 input.

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

Any processor may interrupt any other onboard resource by writing to the appropriate Set Interrupt Register in its own Local memory space. This interrupt drives the IIOF0 input on the target 'C40. Table 3–11, at left, shows which address to write, in order to interrupt a given resource as a function of which processor issues the interrupt. VMEbus Masters can interrupt any processor by writing to the address shown in Table 3–10, at the bottom of the previous page. VMEbus Master interrupts drive the target processor's IIOF1 input.

Table 3–11: Model 4269 – Processor Set Interrupt Registers

Address	Interrupting Processor	
	A	B
0x0034 0005	B	A
0x0034 0009	V M E	

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 3–9, on the previous page, tells which Status Register bit indicates an interrupt from which source, as a function of the identity of the processor receiving the interrupt.

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 3–8, on the previous page, tells which address to write, in order to clear an interrupt, as a function of the interrupt receiver.

A VMEbus Master clears interrupts issued to it from the 4269 by writing to the appropriate Clear Interrupt Register in its own memory space, depending upon the interrupt source. Table 3–12, left, shows which address a VMEbus Master should write in order to clear an interrupt as a function of the interrupt source.

Table 3–12: Model 4269 – VME Clear Interrupt Reg's

A	B
A16base+0x20	A16base+0x24

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)

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

To "lock" the data path between one processor and the other 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 4.5.

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 3–13, below, summarizes the registers and bits involved in Global bus locking.

Table 3–13: Model 4269 – 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

3.1.2.4 Non–Maskable Interrupt (NMI) Registers – W/O @ 0x0034 000C & 0x034 000D

The 4269'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.

For 4269 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 the board.) The timer (and interrupt) can be re–enabled by writing to 'C40 address 0x0034 000C. These settings are summarized in Table 3–14, at the top of the next page.

NOTE: This feature is disabled in units equipped with Option 060 (60 MHz 'C40s).

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)

Table 3−14: Model 4269 − Non−Maskable Interrupt Registers

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

3.2 Global Memory Map

In the standard configuration, the two 1 MByte Global Memory banks on the 4269 are mapped into two 2 MByte regions. Table 3−15, below, gives the address boundaries for the two memory blocks, as seen by the 'C40 processors and by the VMEbus.

Table 3−15: Model 4269 − Global SRAM Addressing (Read/Write)

Resource	'C40 Address Range (R/W, 256kWords X 32bits)	VME 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
Processor B Global SRAM	0x8008 0000 − 0x800B FFFF	VME_base+0x0020 0000 − VME_base+0x002F FFFF

It is important to remember that Global SRAM is accessed from the 'C40 side on long−word addresses. VME access to Global SRAM is generally on byte addresses (1 long−word = 32 bits = 4 bytes).

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 3.3.3). This should not be confused with A16_base, which is determined by jumper settings (see Section 2.2.2). Note that VMEbus access to the 4269 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 Map−ping Register.

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

3.3 VMEbus Slave Interface Memory Map

The A16 VME Slave interface Memory Map for the Model 4269 is described by Table 3–16, below. 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.

Table 3–16: Model 4269 – VMEbus Slave Memory Map

4200/01 68030 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
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	N/U	Not Used (Reserved)
A16_base+0x001C	N/U	Not Used (Reserved)
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	N/U	Not Used (Reserved)
A16_base+0x002C	N/U	Not Used (Reserved)
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 3.3.3**)

3.3.1 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.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.1 VMEbus Address Modifiers (continued)

The VMEbus specification defines the usage of certain AM codes. Table 3–17, below, lists the Address Modifier codes that the Model 4269 will respond to. Codes not listed in the table will not be recognized by the Model 4269.

Table 3–17: Model 4269 – 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

3.3.2 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 3–18, below. The subsections beginning on the next page give detailed descriptions of the functions of each bit referred to in the table.

Table 3–18: Model 4269 – 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	N/U	N/U	N/U	N/U	N/U	N/U
D7	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
1 = Test Bus Controller Enable	0 = Test Bus Controller Enable	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

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.2 VME 'C40 Reset Control Register (continued)

3.3.2.1 VME Global Bus Lock – Bit D15

To "lock" the data path between a VMEbus Master and either 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 4269'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 4.5.

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 3–19, below, summarizes the registers and bits involved in Global bus locking.

Table 3–19: Model 4269 – Global Bus Lock Registers

DMA Master	Global Bus Lock	Global Bus Unlock
Processors	Write @ 0x0034 000B	Write @ 0x0034 000A
VMEbus	A16base + 0, D15 = 1	A16base + 0, D15 = 0

3.3.2.2 VME Broadcast Mode Enable – Bit D14

Setting bit D14 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 Processor B. This is known as Broadcast Mode. Broadcast Mode is disabled by clearing this bit to the logic '0' state.

3.3.2.3 VME Test Bus Controller – Bits D7 and D6

The Texas Instruments SN74ACT8990 is utilized as the Test Bus Controller in the 4269. 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 initializes in the low state, disabling the part. Bit D6 of the Reset Control Register is used to reset the Controller. Clearing D6 to the logic '0' state will hold the controller in the reset state (the default condition), and setting both D6 and D7 to the logic '1' state will make the Controller operational.

Note that the Test Bus Controller should **NEVER** be enabled if the XDS–510 Emulator is connected to the 4269. See Section 3.3.7 for more information about the Test Bus Controller.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.2 VME 'C40 Reset Control Register (continued)

3.3.2.4 Processor Reset – Bits D5 thru D0

'C40 Processor A will be held in the reset state while bit D0 of the VME 'C40 Reset Control Register contains a '1'. Processor B will remain reset as long as there is a '1' in bit D3 of this register. To initiate a reset cycle in any processor, toggle the appropriate bit from '0' to '1', and back to '0' again. 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 and 12.1.

Tables 3–20 and 3–21, below, summarize the operation of the reset bits in the VME 'C40 Reset Control Register.

Table 3–20: Model 4269 – 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

Table 3–21: Model 4269 – 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

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.3 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 4269's Global SRAM in A24 or A32 space (referred to as VME_base in the Global Memory Map, Table 3–15, and the VME Slave Memory Map, Table 3–16). 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 4269 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 3–22, below, summarizes the bit configuration of this register, and the subsections that follow describe the usage of those bits.

Table 3–22: Model 4269 – 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				

3.3.3.1 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 4269. 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 3–23, at the top of the next page, lists the address bits set by D7 – D4 and their hex weights. Table 3–24, also on 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.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.3 VME Address Mapping and Control Register (continued)

3.3.3.1 VME Base Address Selection (continued)

Table 3−23: Model 4269 − 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)

Table 3−24: Model 4269 − 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. The bits set by this block are compared to VME address bits to determine if this is the board being addressed.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.3 VME Address Mapping and Control Register (continued)

3.3.3.1 VME Base Address Selection (continued)

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 com–parator's input. Thus, to match these bits, for a 2 MByte VME SRAM window setting on JB11, D4 in this register must set to the '1' state. For a 4 MByte SRAM window, D5 and D4 must both be set to '1'. For an 8 MByte SRAM window, D6, D5 and D4 must be in the '1' state. For the 16 MByte window size, all four bits (D7 – D4) must be set, and the A24 base address is fixed at 0x00 0000.

Tables 3–25 through 3–28, below and on the following page, list the register settings for valid VME SRAM base addresses when the SRAM window size is greater than 1 MByte.

Table 3–25: Model 4269 – 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 3–26: Model 4269 – 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

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.3 VME Address Mapping and Control Register (continued)

3.3.3.1 VME Base Address Selection (continued)

Table 3−27: Model 4269 − 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 3−28: Model 4269 − A24 Global SRAM Base Address − Bit Settings for Valid Base Addresses − 16 MB SRAM Window

Hex Address	D7 (A23)	D6 (A22)	D5 (A21)	D4 (A20)	Hex Setting*
0x00 0000	0	1	1	1	0xFz
* for low byte of register...z = setting of D1 & D0					

Table 3−29, 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 4269. The 4269'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 3−29: Model 4269 − 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)

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.3 VME Address Mapping and Control Register (continued)

3.3.3.2 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) cleared to '0'. In Direct Mode, the 4269 responds to addresses that exactly match the address bits set in the Address Mapping Register. Again, in the factory default configuration, the device takes up 4 Mbytes of VME memory. Since this is one quarter 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 '1'. 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 3.3.4).

3.3.3.3 Word Swap Mode – Bit D0

Word Swap Mode is used to allow Intel type processors to access the 4269 memory, by setting D0 in this register to a '1'. Leave this bit in the '0' state if you are working with a Motorola host.

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.

3.3.4 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 '1' state, paged addressing is enabled. This mode allows the 4269 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 4269's memory in 1 MByte pages.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.4 Address Page Register (continued)

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 low, the page register points at Processor A's Global SRAM bank. A '1' in D1 and a '0' in D2 would point to Processor B's bank. Table 3–30, below shows the correspondence between the bits in this register and the VME Address lines.

Table 3–30: Model 4269 – 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	A20

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

Table 3–31, below, summarizes the bits in the VME Interrupt and Bus Lock Status Register, at A16_base+0x000C. The subsections beginning on the next page provide a detailed explanation of how bits in the register are used.

Table 3-31: Model 4269 - 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	N/U	N/U	N/U	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	N/U	N/U	1 = Interrupt, Processor B to VME	1 = Interrupt, Processor A to VME

3.3.5.1 TBC Ready – Bit D15

When this bit reads back in the logic '0' state, it indicates that the SN74ACT8990 Test Bus Controller is ready for use. This bit will be a logic '0' 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 2.6), or by installing a jumper on JB22 on the VME board.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.5 VME Interrupt and Bus Lock Status Register (continued)

3.3.5.2 Global Bus Locks – Bits D13, D9 and D8

When one of these bits reads back in the logic '0' state, it indicates that the associated resource has locked the 4269's Global data bus, to allow faster block transfers. D13 of this Status Register will read back in the logic '1' state when a VME Master has locked the Global bus (see Section 3.3.2.1 for details).

When D9 reads back in the logic '1' state, it says that Processor B has locked the Global bus, and D8 will be a logic '1' when Processor A locks the bus (see Section 3.1.2.3 for further information). Uses and limitations of the Global bus lock feature are discussed in Section 4.5 of this manual.

3.3.5.3 Interrupt Source Identification – Bits D4, D1 and D0

Table 3–32: Model 4269 – VME Interrupt ID Bits

Status Bit	Interrupter
D4	TBC
D1	B
D0	A

When an interrupt is issued over the VMEbus by any of the resources on the 4269, these bits are used to indicate which one set the interrupt. If the D4 bit reads back in the logic '1" state, then the interrupt was issued by the Test Bus Controller. If D1 is reads back as a logic '1', the interrupt came from Processor B. If the interrupt was set by Processor A, then D0 will be in the logic '1' state. Table 3–32, left, summarizes the definitions of the Interrupt Identification Status Bits.

3.3.6 VME Command Registers

The text and tables beginning on the next page summarize the addresses and functions of the Command Registers associated with the VMEbus Slave interface on the Model 4269. In all cases, just write to the given address to cause the indicated function to execute.

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.6 VME Command Registers (continued)

3.3.6.1 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)

A VMEbus Master may interrupt either of the 'C40 processors on the 4269 by writing to the appropriate Set Interrupt Register. During normal operation, this interrupt is delivered to the 'C40's $\overline{\text{IIOF1}}$ input. Table 3–33, 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.

Table 3–33: Model 4269 – VME Set Interrupt Registers

Write to this address...	to Interrupt Processor...
A16_base+0x0010	A
A16_base+0x0014	B
A16_base+0x003C	A & B

Writing to A16_base+0x3C will interrupt both processors. This interrupt is reset when both processors write to their own Clear Interrupt Registers, at 0x0034 0004.

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

After receiving an interrupt from either of the processors on the 4269, 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 3–34, left, tells which address the VME Master must write, in order to clear the interrupt, depending upon which processor set it.

Table 3–34: Model 4269 – VME Clear Interrupt Registers

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

3.3.6.3 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 4269 issues an interrupt to VME. Table 3–35, below, shows the correspondence between the bits in this register and the VME Data Bus lines that are their destinations.

Table 3–35: Model 4269 – 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

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

3.3 VMEbus Slave Interface Memory Map (continued)

3.3.7 Test Bus Controller Registers

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

The Texas Instruments SN74ACT8990 is provided on the Model 4269 for use as a JTAG Test Bus Controller. That device interfaces a host computer to a 4–wire serial test bus on the 4269, 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 4269

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 4269's Processors

When power is initially applied to the Model 4269, the $\overline{\text{RESET}}$ inputs on the TMS320C40s are briefly held low by an RC network. This activation of the $\overline{\text{RESET}}$ 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 $\overline{\text{Reset}}$ 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 both processors in the 'C40 Reset Control Register (see Table 3–18 in Section 3.3.2 of this manual). The power–up reset 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 to the $\overline{\text{IIOF}}$ pins of Processor A. Another transfers the settings of jumper block JB7 to the $\overline{\text{IIOF}}$ pins on Processor B. 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 both processors to 0x0030 0000, which is the beginning address of each processor's EPROM.

Table 4–1: Model 4269 – Boot Address Jumper Blocks

JB2	JB7
Processor A	Processor B

Table 4-2: Model 4269 - '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 4269'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 4269 that are external to the 'C40s, as described in Table 4–4, on the next page.

Table 4–3: Model 4269 – SRAM Locations of Pointers to EPROM Functions

Host Slave Offset (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) VME Addressing: y = 0 for Processor A, y = 2 for Processor B
(2) 'C40 Addressing: zz = 00 for Processor A, zz = 08 for Processor B

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

4.2 Resetting and Booting the Model 4269's Processors (continued)

Table 4−4: Model 4269 − 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).
VME 'C40 Reset Control Reg	**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 Reg	**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 Reg	**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 or 0x8008 0000 for Processor B). 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 4269'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 both LEDs only blink around once per second, then that (or both) 'C40(s) has had trouble accessing its Global SRAM. If the factory boot code was executed and either (or both) LED(s) does not blink at all, then a more serious problem exists.

Grounding the Reset pin on the Model 4269's front panel, or asserting one of the Processor Reset Bits in the VME 'C40 Reset Control Register will initiate a reset cycle identical to the power−up reset described above for the processor(s) in question.

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

4.2 Resetting and Booting the Model 4269's Processors (continued)

4.2.1 Boot Loading User Code

During the power–up (or "cold") reset on the Model 4269, the boot load must always be from either the external EPROM or the Comm Port. However, processors on the 4269 can also be reset in the "warm" state, by using either the Reset pins on the front panel or the Reset Bits in the 'C40 Reset Control Register (see Section 3.3.2 for details about the Reset Control Register). 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 or 0x8008 0000 for Processor B). Next, write the address at which your boot code starts into the next Global SRAM location (i. e., 0x8000 0001 for Processor A or 0x8008 0001 for Processor B). 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 4269

'C40 programs in Common Object File Format (COFF) can be transferred into the 4269'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.

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.

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

4.3 Downloading Programs to the Model 4269 (continued)

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 4269 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 4269 via the JTAG connector on the front panel. Note that if the Emulator is to be connected to the Model 4269, jumpers must NOT be installed on jumper blocks JB22 and JB23, 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 4269'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 4269 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**.

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.4 (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.

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

4.4 Working with Interrupts on the Model 4269

The Model 4269 can issue interrupts to, and accept interrupts from, the VMEbus. Additionally, either 'C40 processor on the 4269 can interrupt the other 'C40. The sub–sections below will provide details on how the 4269's processors interrupt other devices and each other, and how interrupts are handled by the processors on the 4269.

4.4.1 The 4269 as a VMEbus Interrupter

The Model 4269 can participate in the VMEbus IACK cycle as an interrupter only. Enabling the 4269 as a VME interrupter is a two–step process. First, it must be decided which devices on the 4269 can issue interrupts to VME. Any one processor, both 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 4269 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 4269 – 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	–	–	–
Procs A & B	4 – 3	4 – 6	*15 – 17	15 – 13	16 – 18	16 – 14	4 – 2
* – Factory Default Setting							
NOTE: Procs A & B = either processor may interrupt VME – TBC = Test Bus Controller							

Next, one of the seven VME interrupt levels must be enabled for the 4269, 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, at the top of the next page, describes the settings of this jumper block.

The 'C40 processors on the Model 4269 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.

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

4.4 Working with Interrupts on the Model 4269 (continued)

Table 4−6: Model 4269 − 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, D1 is Processor B's, and D0 is Processor A's interrupt status bit. Table 4−7, at the bottom left of this 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 4269, the interrupt handler must write any data to 4269 address A16_base+0x20. A VME IRQ from Processor B is cleared by writing to A16_base+0x24. The VME Clear Interrupt Registers are summarized in Table 4−8, below. For information about clearing VME interrupts from the TBC, please see the data sheet for the 74ACT8990.

Table 4−7: Model 4269 − VME Interrupt ID Bits	
Status Bit	**Interrupter**
D4	TBC
D1	Processor B
D0	Processor A

Table 4−8: Model 4269 − VME Clear Interrupt Registers	
Write to this Address...	**to clear Int. from Proc...**
A16_base+0x0020	A
A16_base+0x0024	B

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

4.4 Working with Interrupts on the Model 4269 (continued)

4.4.2 Interrupting 'C40s on the 4269 from VME

By the strict VMEbus definition, the Model 4269 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 4269.

A VMEbus Master can interrupt Processor A on the Model 4269 by writing any data to address A16_base+0x10. Any write to A16_base+0x14 on the 4269 will issue a VMEbus interrupt to Processor B. Both 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 4269 – VME Set Interrupt Registers

Write to this Address...	to Interrupt Processor...
A16_base+0x0010	A
A16_base+0x0014	B
A16_base+0x003C	A & B

VMEbus Master interrupts drive the $\overline{\text{IIOFT}}$ inputs on the 'C40 processors. 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'.

4.4.3 Inter–Processor Interrupts on the 4269

Either 'C40 processor on the Model 4269 can interrupt the other processor. In each processor's Local bus Memory Map is a write–only address (0x0034 0005) for interrupting the other processor. The data written to this register is unused, and therefore trivial.

When a 'C40 on the 4269 interrupts the other 'C40, a bit (D13) is set in its own Status Register (as well as in the Status Register of the processor that it interrupts).

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

4.4 Working with Interrupts on the Model 4269 (continued)

4.4.3 Inter–Processor Interrupts on the 4269 (continued)

Table 4-10: Model 4269 - Status Register - Proc. Interrupt Bits

Bit	Interrupted Processor	
	A	B
D0	B	A
D4	V M E	

Interrupts from other on–board processor drive the IIOF0 input of the target 'C40. The 'C40 that receives an interrupt on IIOF0 reads his status register to verify that the other processor issued the interrupt. The LSB (D0) of each 'C40's Status Register is the incoming inter–processor interrupt status bit. Table 4–10, left, tells which of these bits indicates an interrupt from which other device. Another Local bus address (0x0034 0000, and again, the data written here is unused) is provided to clear inter–rupts from the other processor.

4.5 Using the Model 4269's Global Bus Lock Feature

The Model 4269'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 uncon–tested access to the Global data bus, and to the Global SRAM bank it is using, until it unlocks the bus.

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), and bus locks issued by one of the 'C40 processors on the 4269 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 4269. 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 are left to the discretion of the user.

4.5.1 Global Bus Locked by VME

When a VMEbus Master invokes the Global bus lock, it obtains uncontested use of the 4269'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 4269's shared memory, that neither of the 'C40 processors on the 4269 can access the other processor's Global SRAM, and that the locked Global SRAM bank is inaccessible to the processor that "owns" it (the other processor, however, can still access its own Global SRAM). 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 the Global SRAM bank that is not locked.

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

4.5 Using the Model 4269's Global Bus Lock Feature (continued)

4.5.1 Global Bus Locked by VME (continued)

To invoke the Global bus lock, the VMEbus Master 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. Once this bit has been set, all subsequent accesses by any VMEbus Master to the 4269 are locked to the Global SRAM in question, and none of the 4269's other resources are available to this master.

When the VMEbus Master 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 4269 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.

It is worth noting at this point that while invoking the Global bus lock does guarantee that no other device can use the 4269'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 a VMEbus Master that locks the 4269's Global bus **ALSO** keeps possession of the VMEbus for the duration of the block transfer.

4.5.2 Global Bus Locked by 'C40s on the 4269

The 'C40 processors on the 4269 can access their own Global SRAM at full processor speed (i. e., zero wait state), and the other Global SRAM 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 4269 can lock any ONE Global SRAM bank. The Global bus lock CANNOT be used to move data between two different Global SRAM banks. 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

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

4.5 Using the Model 4269's Global Bus Lock Feature (continued)

4.5.2 Global Bus Locked by 'C40s on the 4269 (continued)

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.

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 4269's arbiter may not react quickly enough to recognize an attempt to access the other 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.

NOTES: (continued)

(1) When a 'C40 is performing interlocked accesses to a Global SRAM bank in a tight loop, the 4269 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 4269'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.3 (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.

4.6 Using the Global Memory

The Global SRAM on the 4269 is accessible by any VMEbus Master and by both '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 Com−mon Object File Format (COFF) can be deposited here by the VME System Executive, and can either be executed directly from Global memory or moved by the 'C40s into their Local SRAM prior to execution.

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

4.6 Using the Model 4269's Global Bus Lock Feature (continued)

The 4269's Global SRAM is seen as two separate memory blocks of 256k longwords each from the 'C40 side, or 1 MByte each from the VMEbus side. The starting address of these memory blocks, from the 'C40's point of view, is 0x8000 0000 for Processor A's Global SRAM, or 0x8008 0000 for Processor B's Global SRAM. Any processor may access the other 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).

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 3.3.3.1 for further details). Processor B's bank of Global SRAM starts at address VME_base+0x0020 0000.

When multiple devices attempt to gain initial access the 4269's Global bus simultaneously, access will be granted in the following prioritized order: Processor A, Processor B 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 B, and B no longer needs the bus, priority would pass to VME, then to Processor A. Use of the Global SRAM by the 'C40s is facilitated by the Global bus locking feature of the Model 4269. This feature is described in detail in Section 4.5 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 4−11 through 4−22, 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 4269 to any another available resource, or to another region within that resource's address range.

The data in Tables 4−11 and 4−12 represent typical performance for a Model 4269 with 50 MHz 'C40s, using parallel instructions and 'C40 DMA transfer mode, respectively. Tables 4−13 and 4−14 represent the performance that can typically be expected under the same conditions, from a 50 MHz 'C40 on a 4269 with the Global bus lock used. Tables 4−15 and 4−16 represent the typical performance of a Model 4269 equipped with 60 MHz 'C40s, in parallel instruction and 'C40 DMA modes, respectively. Tables 4−17 and 4−18 show what can be expected when a 60 MHz 'C40 locks the Global bus. Tables 4−19 through 4−22 provide the same information for units that contain 40 MHz 'C40s.

'C40 Processor A on the 4269 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 4−11, 4−15 and 4−19 for 50 MHz, 60 MHz and 40 MHz devices, respectively. Tables 4−13, 4−17 and 4−21 describe such transfers with the Global bus lock applied.

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

4.6 Using the Model 4269's Global Bus Lock Feature (continued)

Table 4-11: Model 4269 - 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
'C40 Int. RAM	40 nsec	80 nsec	80 nsec	120 nsec
Local SRAM	40 nsec	160 nsec	80 nsec	120 nsec
Own Global SRAM	40 nsec	200 nsec	160 nsec	880 nsec
Other Global SRAM	120 nsec	120 nsec	880 nsec	1040 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4–12: Model 4269 – 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
'C40 Int. RAM	80 nsec	80 nsec	80 nsec	120 nsec
Local SRAM	80 nsec	160 nsec	80 nsec	120 nsec
Own Global SRAM	360 nsec	360 nsec	160 nsec	880 nsec
Other Global SRAM	520 nsec	520 nsec	880 nsec	1040 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4-13: Model 4269 - 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
'C40 Int. RAM	N/A	N/A	80 nsec	120 nsec
Local SRAM	N/A	N/A	80 nsec	120 nsec
Own Global SRAM	40 nsec	80 nsec	160 nsec	N/A
Other Global SRAM	120 nsec	120 nsec	N/A	240 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4-14: Model 4269 - 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
'C40 Int. RAM	N/A	N/A	80 nsec	120 nsec
Local SRAM	N/A	N/A	80 nsec	120 nsec
Own Global SRAM	80 nsec	80 nsec	160 nsec	N/A
Other Global SRAM	160 nsec	160 nsec	N/A	240 nsec
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

4.6 Using the Model 4269's Global Bus Lock Feature (continued)

Table 4-15: Model 4269 - 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
'C40 Int. RAM	33.3 nsec	66.7 nsec	66.7 nsec	100 nsec
Local SRAM	33.3 nsec	133.3 nsec	66.7 nsec	100 nsec
Own Global SRAM	33.3 nsec	166.7 nsec	133.3 nsec	733.3 nsec
Other Global SRAM	100 nsec	100 nsec	733.3 nsec	866.7 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4–16: Model 4269 – 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
'C40 Int. RAM	66.7 nsec	66.7 nsec	66.7 nsec	100 nsec
Local SRAM	66.7 nsec	133.3 nsec	66.7 nsec	100 nsec
Own Global SRAM	300 nsec	300 nsec	133.3 nsec	733.3 nsec
Other Global SRAM	433.3 nsec	433.3 nsec	733.3 nsec	866.7 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4-17: Model 4269 - 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
'C40 Int. RAM	N/A	N/A	66.7 nsec	100 nsec
Local SRAM	N/A	N/A	66.7 nsec	100 nsec
Own Global SRAM	33.3 nsec	66.7 nsec	133.3 nsec	N/A
Other Global SRAM	100 nsec	100 nsec	N/A	200 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4-18: Model 4269 - 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
'C40 Int. RAM	N/A	N/A	66.7 nsec	100 nsec
Local SRAM	N/A	N/A	66.7 nsec	100 nsec
Own Global SRAM	66.7 nsec	66.7 nsec	133.3 nsec	N/A
Other Global SRAM	133.3 nsec	133.3 nsec	N/A	200 nsec
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

4.6 Using the Model 4269's Global Bus Lock Feature (continued)

Table 4-19: Model 4269 - 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
'C40 Int. RAM	50 nsec	100 nsec	100 nsec	150 nsec
Local SRAM	50 nsec	200 nsec	100 nsec	150 nsec
Own Global SRAM	50 nsec	250 nsec	200 nsec	1100 nsec
Other Global SRAM	150 nsec	150 nsec	1100 nsec	1300 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4–20: Model 4269 – 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
'C40 Int. RAM	100 nsec	100 nsec	100 nsec	150 nsec
Local SRAM	100 nsec	200 nsec	100 nsec	150 nsec
Own Global SRAM	450 nsec	450 nsec	200 nsec	1100 nsec
Other Global SRAM	650 nsec	650 nsec	1100 nsec	1300 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4-21: Model 4269 - 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
'C40 Int. RAM	N/A	N/A	100 nsec	150 nsec
Local SRAM	N/A	N/A	100 nsec	150 nsec
Own Global SRAM	50 nsec	100 nsec	200 nsec	N/A
Other Global SRAM	150 nsec	150 nsec	N/A	300 nsec
All timing results are given as average for 1 longword transfer based on 1000 longword transfers				

Table 4-22: Model 4269 - 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
'C40 Int. RAM	N/A	N/A	100 nsec	150 nsec
Local SRAM	N/A	N/A	100 nsec	150 nsec
Own Global SRAM	100 nsec	100 nsec	200 nsec	N/A
Other Global SRAM	200 nsec	200 nsec	N/A	300 nsec
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

4.6 Using the Model 4269's Global Bus Lock Feature (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 4–12 and 4–14 document the performance of a 4269 with 50 MHz processors, in 'C40 DMA transfer mode, with the Global bus lock off and on, respectively. Tables 4–16 and 4–18 cover the same situations for 60 MHz processors, and Tables 4–20 and 4–22 give the same information for 40 MHz 'C40s.

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 4269. 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 4–11 through 4–22. Also note that the overheads listed above do not (and cannot) take possible arbitration delays into account. Table 4–23, below, summarizes the overhead information given above.

Table 4–23: Model 4269 – Global Bus Access Overheads – Maximum Initial Access Delays (Excluding Arbitration)

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
440 nsec	366.7 nsec	550 nsec	280 nsec	233.3 nsec	350 nsec

No VMEbus transfer data is included in the tables above, because the 'C40s on the 4269 cannot master the VMEbus. However, in a 4269 with 50 MHz 'C40s, the delay from the assertion of the DS signal from the master to the assertion of DTACK signal by the 4269 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 DS to a 50 MHz 4269's de–assertion of DTACK is 70 nsec, regardless of the state of the bus lock. For a 60 MHz 4269, the assertion delay is 240 nsec if the Global bus is not locked or 140 nsec if it is locked. The de–assertion delay on a 60 MHz board is 66.7 nsec in either case. For a 40 MHz 4269, 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 4–24, below.

Table 4–24: Model 4269: 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

Appendix A: Programming Examples

A.1 Slo_Blnk.C

The sample "C" source code below drives the two LEDs on the Model 4269's front panel with sinusoidally pulse-width modulated waveforms. Each processor drives one of the LEDs. The signals from processor A are 180° out of phase with the signals generated by processor B.

```
/*****************************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 PI           3.1415927
/*********************************************************************/
int angleStep = 200;
int fullDuty = 10000;
/*********************************************************************/

main()
{
   if (((*PROC_ID & 0x300000) == 0)||
       ((*PROC_ID & 0x300000) == 0x300000))
   {
      procA();                              /* processor A routine */
   }
   else
   {
      procB();                              /* processor B routine */
   }
}
/*********************************************************************/
```


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

A.1 **Slo_Blnk.C** (continued)

```
/*********************************************************************/
procA()
{
   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;
         wait(onDelay);

         *TIMER0_CTRL = LED_OFF;                /* turn proc A LED off */
         while((*COM0_CTRL & 0x1e0) != 0x0);
                                                /* wait for COM0 output */
         *COM0_OUTPUT = LED_ON;
         wait(offDelay);
      }
   }
}
/*********************************************************************/
wait(count)

int count;
{
   int i;
   for(i = 0; i < count; i++);
}
/*********************************************************************/
procB()
{
   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.2 Timer Routines

The two 'C' routines presented below can be used to start and stop Timer 1 on either of the Model 4269'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.3 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 4269's Global bus. This is beneficial when VME Masters, and the 'C40s are 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

/* 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

/* 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");

   /* Initialize TCK1 as a General Purpose IO Pin */
   tck1_on();
```


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

A.3 Using Timer Interrupts to Avoid Bus Timeouts (continued)

```
/* 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.3 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.4 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 4269. 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 4269 contains, and where they reside along the JTAG scan path.

```
; File name:  4269_70.cmd
; 4269 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_B"         TI320C4x        ;the first 'C40
"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 4269. 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 4269, as evidenced by the fact that all the addresses (from the 'C40 side) begin with 0x8000. For Processor B, that prefix would be 0x8008.

```
.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